

## **Aula 00 - Prof. Evandro Dalla Vecchia**

*TCE-GO (Analista de Controle Externo -  
Tecnologia da Informação) Sistemas  
Operacionais - 2024 (Pós-Edital)*

Autor:

**Equipe Direito Administrativo,  
Evandro Dalla Vecchia Pereira ,  
Herbert Almeida**

27 de Fevereiro de 2024

# Índice

1) Apresentação do Curso - Evandro .....	4
2) Linux - Conceitos Básicos - Teoria .....	7
3) Linux - Configuração de Serviços ou Funcionalidades - Teoria .....	32
4) Linux - Shell Script - Teoria .....	41
5) Linux - Configuração de Serviços ou Funcionalidades - Questões Comentadas - Multibancas .....	47
6) Linux - Shell - Questões Comentadas - Multibancas .....	72
7) Linux - Configuração de Serviços ou Funcionalidades - Lista de Questões - Multibancas .....	78
8) Linux - Shell - Lista de Questões - Multibancas .....	93
9) Linux - Serviços de Rede - DNS e DHCP - Teoria .....	97
10) Linux - Serviços de Rede - FTP e E-mail - Teoria .....	110
11) Linux - Gerenciamento de Processos - Teoria .....	117
12) Linux - Montagem de Volumes - Teoria .....	122
13) Linux - Superusuário - Teoria .....	125
14) Linux - Usuários - Teoria .....	128
15) Linux - Comandos Básicos - Teoria .....	133
16) Linux - Serviços de Rede - DNS e DHCP - Questões Comentadas - Multibancas .....	140
17) Linux - Serviços de Rede - FTP e E-mail - Questões Comentadas - Multibancas .....	157
18) Linux - Gerenciamento de Processos - Questões Comentadas - Multibancas .....	161
19) Linux - Montagem de Volumes - Questões Comentadas - Multibancas .....	165
20) Linux - Superusuário - Questões Comentadas - Multibancas .....	168
21) Linux - Usuários - Questões Comentadas - Multibancas .....	171
22) Linux - Comandos Básicos - Questões Comentadas - Multibancas .....	176
23) Linux - Serviços de Rede - DNS e DHCP - Lista de Questões - Multibancas .....	187
24) Linux - Serviços de Rede - FTP e E-mail - Lista de Questões - Multibancas .....	198
25) Linux - Gerenciamento de Processos - Lista de Questões - Multibancas .....	201
26) Linux - Montagem de Volumes - Lista de Questões - Multibancas .....	204
27) Linux - Superusuário - Lista de Questões - Multibancas .....	206
28) Linux - Usuários - Lista de Questões - Multibancas .....	208



# Índice

29) Linux - Comandos Básicos - Lista de Questões - Multibancas .....	211
--	-----



## APRESENTAÇÃO DO CURSO

Iniciamos nosso **Curso Regular de Sistemas Operacionais e Arquitetura de Computadores** em teoria e questões, voltado para provas **objetivas e discursivas** de concurso público. Tais assuntos são cobrados em diversos concursos em que há vagas específicas para a área de TI.

As aulas em PDF possuem por característica essencial a **didática**. Ao contrário do que encontramos em alguns livros, o curso todo se desenvolverá com uma leitura de fácil compreensão e assimilação.

Além disso, teremos videoaulas! Essas aulas destinam-se a complementar a preparação. Quando estiver cansado do estudo ativo (leitura e resolução de questões) ou até mesmo para a revisão, abordaremos alguns pontos da matéria por intermédio dos vídeos. Com outra didática, você disporá de um conteúdo complementar para a sua preparação. Ao contrário do PDF, evidentemente, **AS VIDEOAULAS NÃO ATENDEM A TODOS OS PONTOS QUE VAMOS ANALISAR NOS PDFS, NOSSOS MANUAIS ELETRÔNICOS**. Por vezes, haverá aulas com vários vídeos; outras que terão videoaulas apenas em parte do conteúdo. Nosso **foco** é sempre o **estudo ativo!**



## APRESENTAÇÃO PESSOAL

Meu nome é Evandro Dalla Vecchia Pereira, sou autor do livro "Perícia Digital - Da investigação à análise forense", Mestre em Ciência da Computação (UFRGS), Bacharel em Ciência da Computação (PUCRS), Técnico em Redes de Computadores (Etcom/UFRGS) e em Processamento de Dados (Urcamp). Perito Criminal na área de Perícia Digital desde 2004 no Instituto-Geral de Perícias/RS. Professor de pós-graduação em diversas instituições, nas áreas de Perícia Digital, Perícia Criminal e Auditoria de Sistemas. Lecionei em cursos de graduação de 2006 a 2017, nas instituições PUCRS, Unisinos, entre outras e sou professor em cursos de formação e aperfeiçoamento de Peritos Criminais, Delegados, Inspetores, Escrivães e Policiais Militares.

No Estratégia Concursos leciono desde o começo de 2018, inicialmente na área de Computação Forense e, na sequência, também assumi as áreas de Arquitetura de Computadores e Sistemas Operacionais, tanto na elaboração de materiais escritos como na gravação das videoaulas.

Deixarei abaixo meus contatos para quaisquer dúvidas ou sugestões. Terei o prazer em orientá-los da melhor forma possível nessa caminhada que estamos iniciando.

**Instagram:** @profevandroallavecchia

**Facebook:** <https://www.facebook.com/profevandroallavecchia>



## PARE TUDO! E PRESTE ATENÇÃO!!

Hoje eu faço parte de uma equipe **SENSACIONAL** de professores! Depois de muita luta conseguimos reunir **um time** de profissionais extremamente **QUALIFICADO** e sobretudo **COMPROMISSADO** em fazer o melhor pelos alunos. Para tal criamos um conjunto de ações para nos aproximarmos dos alunos, entendermos suas necessidades e evoluirmos nosso material para um patamar ainda mais diferenciado. São 3 as novidades que gostaria de convidá-lo a conhecer:

//estratégia tech



Nosso podcast alternativo ... livre, descontraído e com dicas rápidas que todo CANETA PRETA raiz deve ouvir. Já temos alguns episódios disponíveis e vários outros serão gravados nas próximas semanas ... acompanhe em:

<http://anchor.fm/estrategia-tech>



Telegram

a new era of messaging

Nosso grupo do Telegram é um local onde ouvimos os alunos e trocamos ideias com eles. Está crescendo a cada dia. A regra do grupo é: só vale falar sobre concursos. Lá divulgamos nossas aulas ao vivo e falamos sobre os concursos abertos, expectativas de novos concursos, revisões de véspera, e por aí vai...

[http://t.me/estrategia\\_ti](http://t.me/estrategia_ti)

Instagram



Criamos um perfil no Instagram ... e qual o objetivo? Fazer com que os alunos percam tempo nas redes sociais? Claro que não!! Estamos consolidando diversos posts dos professores! São dicas especiais, um patrimônio que deve ser explorado por todos os concurseiros de TI!

<http://instagram.com/estrategiaconcursosti>



## LINUX - CONCEITOS BÁSICOS

Começamos dizendo que o **Linux é um sistema operacional interativo**, ou seja, diferentemente dos sistemas em lote ou os sistemas de tempo real, ele permite uma interação do usuário diretamente com o computador, mesmo durante a execução de um programa. Essa interação ocorre através de dispositivos de entrada (teclado, mouse, entre outros) e saída (monitor de vídeo, impressora, entre outros).

O usuário "administrador", que tem "poder total" sobre o sistema é o **root** e seu diretório "home" é o **"/root"**. Os usuários "comuns" têm como seu diretório "home" o **"/home/nome\_usr"**, ex.: **"/home/evandro"**.

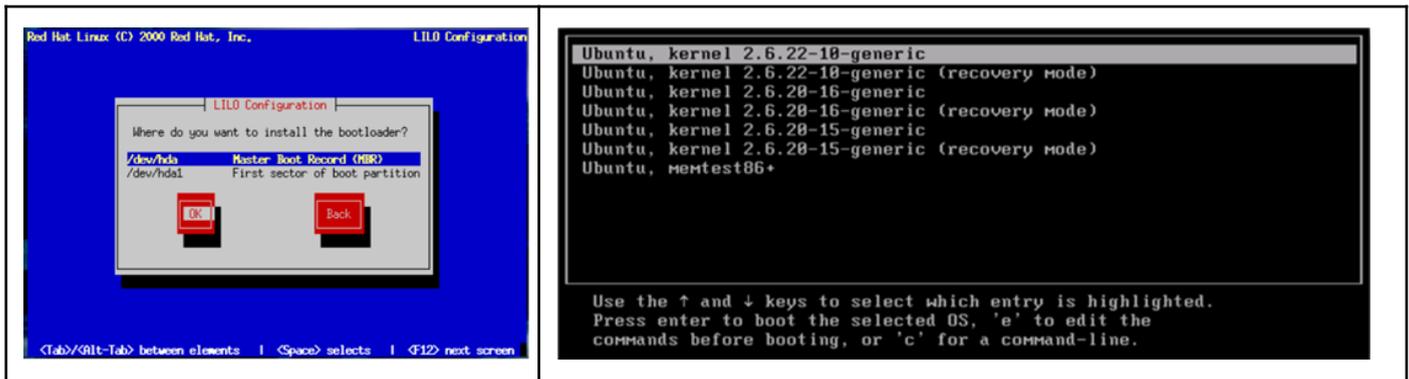
A instalação do Linux pode ocorrer de algumas formas, sendo as mais comuns através de uma mídia (CD, DVD, pen drive) ou através da rede. No caso da mídia, obviamente ela precisa ser **"bootável"** e a BIOS deve estar configurada para ler primeiro o CD/DVD ou a USB (se for pen drive). Abaixo uma tela de instalação a partir de um CD.



A instalação pela rede exige mais conhecimento e geralmente é feita por técnicos de TI nas empresas. É necessário configurar um servidor TFTP e um DHCP que irá fornecer a mídia de instalação para as máquinas da sua rede local. Se a BIOS da máquina cliente suportar, é possível então inicializar o sistema de instalação a partir da rede (usando PXE e TFTP) e prosseguir com a instalação a partir da rede.

Muitas pessoas utilizam o Linux e o Windows no mesmo computador, cada um em uma partição (com seus respectivos sistemas de arquivos). E como é possível escolher qual deles deve ser inicializado quando a máquina é ligada? Existem **gerenciadores de boot** que auxiliam nisso, sendo os mais conhecidos o **LILLO** (*Linux LOader*) e o **GRUB** (*GRand Unifield Bootloader*). Abaixo podemos ver os dois (à esquerda o LILLO e à direita o GRUB).





Ao escolher a partição, o *kernel* (veremos em breve o que é e para que serve) é carregado e inicializado. Na sequência há a execução dos scripts de inicialização do sistema.

Em relação à **portabilidade**, o Linux se sai muito bem! Ele não foi pensado para ser um sistema portátil, mas acabou indo nessa direção, tendo um dos núcleos de sistemas operacionais mais portáteis, sendo executado em sistemas desde o iPaq (um computador portátil) até o IBM S/390 (um mainframe). Na atualidade o Linux hoje funciona em dezenas de plataformas, desde mainframes até um relógio de pulso, passando por diversas arquiteturas: x86 (Intel, AMD), x86-64 (Intel EM64T, AMD64), ARM, PowerPC, Alpha, SPARC, entre outras, além de sistemas embarcados (*handhelds*, consoles de videogames, PVR – *Personal Video Recorder*, telefones celulares, TVs etc.).

## Sistema Multitarefa e Multiusuário

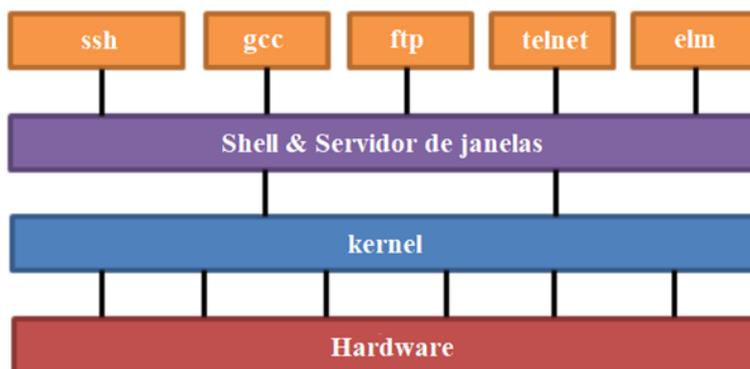
O Linux permite que vários processos sejam executados ao mesmo tempo (**multitarefa**), mesmo que haja apenas um processador na máquina. O sistema operacional faz o escalonamento dos processos no processador, dando a ilusão de um paralelismo (ocorre um pseudoparalelismo). O comando "ps", por exemplo, pode ser executado e você verá diversos processos em execução. Quando há a criação de um processo novo, ele passa a ser filho de outro e a chamada de sistema utilizada é a fork().

No Linux é possível que vários usuários utilizem ao mesmo tempo (**multiusuário**), então alguém pode deixar algumas aplicações em execução em sua sessão, bloquear o acesso e outra pessoa abrir uma outra sessão, executando novas aplicações. São permitidos vários usuários simultâneos, a não ser que seja definido um limite de sessões nos arquivos de configuração. O comando "who" pode ser executado para ver quem está logado no computador.

## Kernel, Shell e Servidor de Janelas

O **kernel** (núcleo) da maioria dos sistemas baseados em Unix (FreeBSD e derivados, Oracle Solaris, IBM AIX, HP-UX e todos os Linux e derivados) é **monolítico**. Mas o que é isso? Todo o conjunto de instruções de controle do hardware é executado no espaço de núcleo no modo de supervisão, ou seja, é um **único executável que possui todos os códigos de suporte necessários agregados**. Em teoria, um kernel monolítico pode travar o sistema inteiro se 1 módulo parar de funcionar adequadamente.





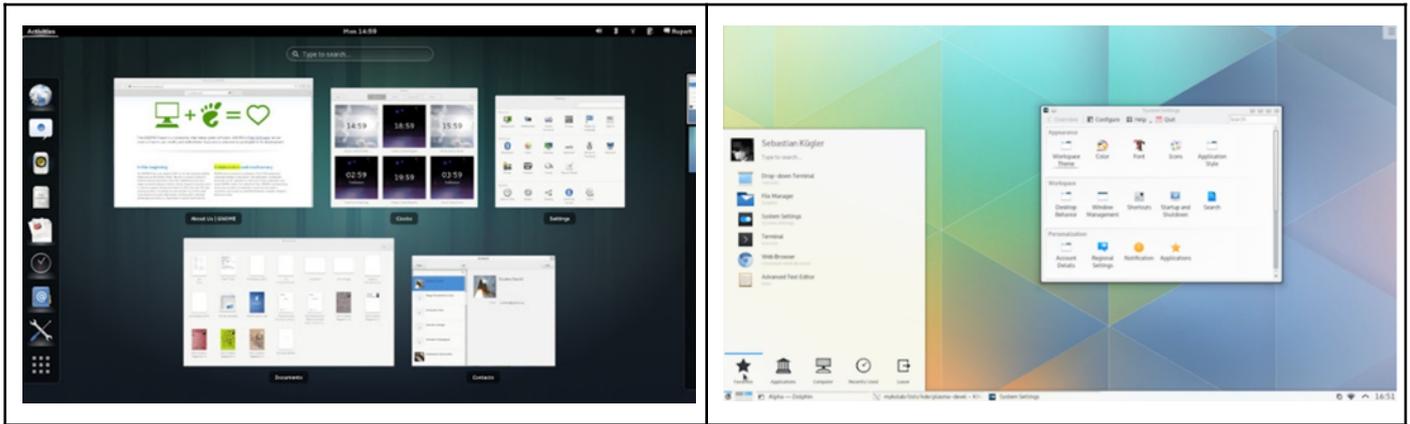
Na figura acima podemos ver cinco aplicativos que utilizam alguma interface (shell ou servidor de janelas), que por sua vez faz a "ligação" com o kernel. E somente o kernel faz a comunicação com o hardware (disco, teclado, monitor de vídeo etc.).

O shell é um interpretador de comandos (aquele terminal preto, geralmente) e sua função é ler a linha de comando, interpretar seu significado, executar o comando e devolver o resultado pelas saídas (monitor de vídeo, por exemplo). Na verdade, a interface shell é um arquivo executável, encarregado de interpretar comandos, enviá-los ao kernel e devolver resultados. Existem vários tipos de shell, sendo os mais comuns o sh (Bourne shell), o bash (Bourne again shell), o csh (C shell), o Tcsh (Tenex C shell), o ksh (Korn shell) e o zsh (Zero shell). Seus nomes normalmente correspondem ao nome do executável.

```
localhost:~# ls -las
total 40
 4 drwxr-xr-x  5 root   root    237 Jan  9  2021 .
 4 drwxrwxrwx 21 root   root    461 Jul 20 21:13 ..
 4 -rw-----  1 root   root     23 Jul 20 21:13 .ash_history
 4 drwx-----  3 root   root     61 Jul  5  2020 .cache
 4 drwx-----  5 root   root    124 Jul  5  2020 .mozilla
 4 drwxr-xr-x  4 root   root    202 Jul  6  2020 .wine
 4 -rw-r--r--  1 root   root    114 Jul  5  2020 bench.py
 4 -rw-r--r--  1 root   root     76 Jul  3  2020 hello.c
 4 -rw-r--r--  1 root   root     22 Jun 26  2020 hello.js
 4 -rw-r--r--  1 root   root    151 Jul  5  2020 readme.txt
localhost:~# █
```

Alguns dos servidores de janelas (ambientes gráficos) mais populares do mundo Linux são Gnome, KDE, Cinnamon, MATE, XFCE, Pantheon, Deepin, entre outros. Abaixo podemos ver uma tela do Gnome (à esquerda) e uma tela do KDE (à direita).





Uma característica interessante, por ser um sistema operacional de código aberto, é a possibilidade de **compilar o kernel**. Mas qual o motivo para isso? Para facilitar vamos a um exemplo: suponha que você tenha uma placa-mãe X e sua placa de rede integrada é uma da fabricante Y. No Kernel que você está usando (uma versão mais antiga) não há suporte para essa placa Y, ou seja, você não pode navegar na rede. Lendo a documentação do Kernel de uma versão mais recente, você constata que o suporte para Y foi adicionado. Então, se o kernel for atualizado será possível navegar na rede com tal placa com esse novo módulo.

Os **módulos são partes do kernel** que são carregadas somente quando solicitadas por algum aplicativo ou dispositivo e descarregadas da memória quando não são mais utilizadas. Dessa forma, evita-se a construção de um kernel grande (estático) que ocupe grande parte da memória com todos os drivers compilados e permite que partes do kernel ocupem a memória somente quando forem necessários.

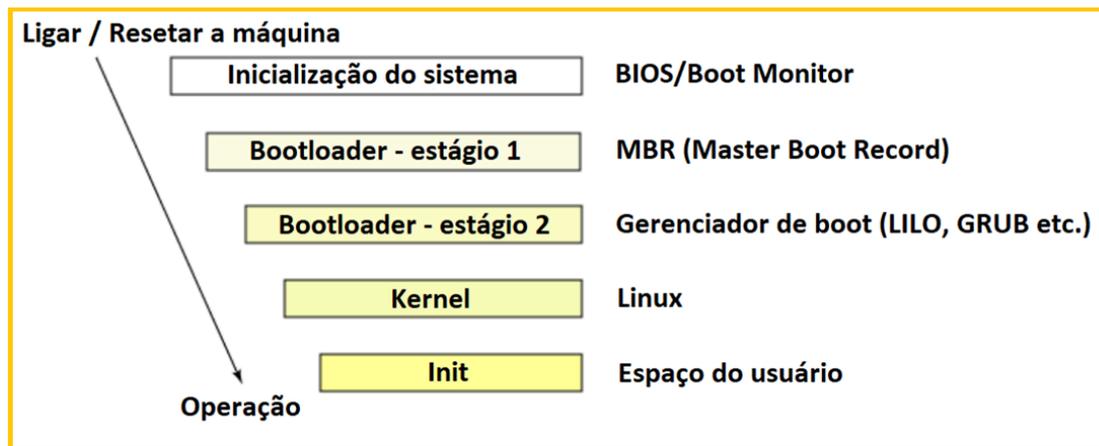
A localização padrão dos módulos do kernel é a seguinte: `"/lib/modules/versão_do_kernel/"` e um comando que permite obter informações acerca do servidor é o `"uname"`. Com o argumento `"-a"` é possível obter todas as informações, incluindo a versão do kernel. Abaixo um exemplo mostrando a versão do kernel (4.12):

```
[root@localhost lib]# uname -a  
Linux localhost 4.12.0-rc6-g48ec1f0-dirty #21 Fri Aug 4 21:02:28 CEST 2017 i586  
GNU/Linux  
[root@localhost lib]#
```

## Bootloader (Carregador de boot)

Ao inicializar o Linux, o gerenciador de inicialização é responsável por carregar a imagem do kernel e o disco RAM inicial ou sistema de arquivos (possui alguns arquivos críticos e drivers de dispositivo necessários para inicializar o sistema) na memória. Abaixo uma figura que mostra o passo a passo do boot.





Podemos observar que o boot possui dois estágios, descritos a seguir.

### Primeiro Estágio:

Em sistemas usando o método BIOS/MBR, o carregador de boot reside no primeiro setor do disco rígido, o Master Boot Record (MBR). Nesse estágio, o carregador de boot examina a tabela de partições e encontra uma partição inicializável. Depois, procura o carregador de inicialização do segundo estágio (ex.: GRUB) e o carrega na memória RAM.

Para sistemas que usam o método EFI/UEFI, o firmware UEFI lê seus dados do gerenciador de boot para determinar qual aplicativo UEFI deve ser lançado e de onde, ou seja, de qual disco e partição a partição EFI pode ser localizada. O firmware então inicia o aplicativo UEFI (ex.: GRUB), conforme definido na entrada de inicialização no gerenciador de inicialização do firmware.

### Segundo Estágio:

O carregador de inicialização do segundo estágio (ex.: GRUB) reside em /boot. Uma tela inicial é exibida, o que permite escolher qual sistema operacional será inicializado. Depois de escolhido o sistema operacional, o carregador de boot carrega o kernel do sistema operacional selecionado na memória RAM e passa o controle para ele.

### Funcionamento do boot no Linux

Os kernels geralmente são compactados, então seu primeiro trabalho é ser descompactado. Depois ele verificará e analisará o hardware do sistema e inicializará todos os drivers de dispositivo de hardware embutidos no kernel.

### Init (Systemd)

O systemd é um sistema de inicialização (init system) composto por um conjunto de programas que é executado em *background*<sup>1</sup> (*daemon*). Na atualidade a maioria das distribuições Linux utiliza o systemd para a execução do boot.

O systemd assume o controle assim que o kernel é ativado pelo gerenciador de bootloader (ex.: GRUB). A partir desse ponto são carregados todos os dispositivos e processos que são

<sup>1</sup> Veremos na sequência o conceito de background em detalhes.

inicializados com o sistema. Uma grande vantagem do systemd é a sua arquitetura e o modo de funcionamento. Nele são usadas unidades de socket (arquivos de configuração que codificam informações relacionadas à comunicação entre processos). Com isso, permite que todos os daemons requisitados no boot sejam carregados simultaneamente, bem como possibilita a transmissão coordenada entre dois sockets, o que resulta em uma rápida inicialização do sistema operacional.

## Arquitetura do systemd

Basicamente, a estrutura do systemd é composta por unidades (units). Abaixo, algumas classes de units do systemd:

- service: tais unidades são os serviços presentes no sistema operacional acessíveis ao usuário;
- timer: temporizadores usados para determinar ações para um serviço usando como base o tempo (não é o cron!);
- mount: arquivo de configuração que codifica informações sobre um diretório controlado e supervisionado pelo systemd;
- target: grupos de unidades que reúnem todas as units necessárias para iniciar um determinado serviço;
- snapshot: mecanismo usado para criar snapshots dinâmicos do estado atual do systemd manager, útil para retomar o estado após problemas;
- path: unidades especialmente utilizadas para monitorar arquivos e diretórios para eventos e, também, executar serviços;
- socket: arquivo de configuração que armazena informações acerca de um IPC ou soquete de rede, ou arquivo FIFO;
- swap: guarda informações relativas a dispositivos usados para swapping, bem como serviços que utilizam memória Swap.

Cada serviço é alocado pelo systemd em um grupo de controle dedicado (control group - cgroup). No cgroup são organizadas informações voltadas aos processos que fazem parte do grupo, como: limite, supervisão e contabilização de recursos computacionais que eles consomem.

O controle desses grupos é feito a partir de utilitários que acompanham o systemd, como por exemplo: journalctl, cgl, cgtop e systemctl (este último será descrito na sequência).

## Comandos básicos para controlar serviços no systemd

Para gerenciar o sistema e os serviços existe o aplicativo systemctl, que auxilia no controle do próprio systemd, além de atuar como gerenciador de serviços. Ou seja, ele permite monitorar, encerrar, iniciar, analisar, recarregar, checar o status dos processos, entre outras ações. Alguns exemplos de comandos são mostrados na sequência.

Listar todos os serviços disponíveis:

```
# systemctl list-unit-files --type=service
```

Ativar um serviço e habilitá-lo (ou desabilitá-lo) no boot:



```
# systemctl is-active httpd.service  
# systemctl enable httpd.service  
# systemctl disable httpd.service
```

Realizar ações básicas (iniciar, reiniciar, parar etc.) para determinado serviço:

```
# systemctl start httpd.service  
# systemctl restart httpd.service  
# systemctl stop httpd.service  
# systemctl reload httpd.service  
# systemctl status httpd.service
```

## Execução em Foreground ou Background

Os processos podem ser executados de duas formas: em primeiro plano (*foreground*) ou em segundo plano (*background*). Os processos que executam em *foreground* são os que necessitam de interação direta com o usuário, incluindo a troca de informações. Os processos em *background* não necessitam dessa interação com o usuário.

Há situações em que é necessário passar um processo que está sendo executado em *foreground* para *background* e vice-versa. Por exemplo, em uma sessão de transferência de arquivos entre computadores remotos, a baixa velocidade da linha de transmissão pode fazer com que o tempo de transferência leve muito tempo. Assim, seria interessante passar o processo para segundo plano, liberando o *shell* para outras atividades do usuário.

A passagem de um processo de *foreground* para *background* é realizada primeiro suspendendo o processo (CTRL + Z), seguido do comando "bg", que envia o processo para segundo plano (*background*). Observe que suspender a execução de um processo não significa finalizá-lo, apenas torná-lo temporariamente inativo!

A lista dos processos executados em *background* pode ser visualizada com o comando "jobs", que mostra o número da tarefa (*job*) associada. Se o usuário quiser interagir novamente com o processo, deve utilizar o comando "fg" seguido de % número\_job. Exemplo:

```
estrategia:~$ jobs  
[1] - Running                script.sh  
[2] + Suspended (tty output) teste.sh  
estrategia:~$ fg %2
```

O mais utilizado para executar em background é utilizar "&" logo após o nome do binário ou script. Por exemplo, se você deseja executar um script "aprovacao.sh", que vai realizar vários cálculos e no fim escrever em um arquivo, não é necessário que você fique com o *shell*



"travado", aguardando a conclusão da tarefa. Então é só digitar (note que o *shell* não fica esperando a conclusão do *script*, já aguarda o próximo comando):

```
estrategia:~$ aprovacao.sh &  
estrategia:~$
```

Processos que tipicamente só são executados em *background*, pois não possuem interação com o usuário, são os *daemons*. Geralmente possuem um nome que termina com a letra "d", como por exemplo, *syslogd* (*daemon* que gerencia o *log* do sistema). Em um ambiente Unix, o "processo pai" de um *daemon* é normalmente o processo *init* (PID=1). De forma geral, os sistemas operacionais iniciam *daemons* durante o processo de *boot*.

## Agendamento de Tarefas

Para o agendamento de tarefas a aplicação mais conhecida é o **Cron**, que já existe há muito tempo e passou por muitos estágios de evolução. Porém, a maioria das suas implementações atuais (*Vixie Cron*, *ISC Cron*, *BCron* etc.) ainda se baseiam no pressuposto de que o sistema está em funcionamento de forma ininterrupta. Isso é um problema quando se utiliza a virtualização, quando os sistemas que operam sob demanda, sendo ligados e desligados conforme sua necessidade.

Além disso, os usuários de *notebooks* suspendem ou desligam seus equipamentos para transportá-los, e até mesmo *desktops* e estações de trabalho são desligados para poupar energia. O resultado disso é que os *cronjobs* falham regularmente, pois o sistema se encontra desligado quando a tarefa deveria ser executada!

Para contornar essa situação, algumas distribuições contam com o *Anacron* como alternativa ao *Cron*. O *Anacron* permite a criação de listas de tarefas que serão realizadas em intervalos pré-definidos e, quando o *Anacron* é inicializado, ele verifica essas listas e executa as tarefas ainda não realizadas. No entanto, o *Anacron* possui algumas limitações. A primeira é que ele não é um *daemon*, então precisa ser executado sempre que for necessário.

Outra questão é que o *Anacron* não está preparado para lidar com períodos de tempo menores do que dias. A combinação desses problemas pode levar a situações nas quais o *Cron* e o *Anacron* funcionam ao mesmo tempo, e algumas tarefas podem ser executadas duas vezes ou nenhuma!

Uma solução é o *Fcron*, que faz o que o *Vixie Cron* e o *Anacron* fazem e ainda mais. É possível usar o *Fcron* para agendar *cronjobs* com data e hora fixas, com intervalos de tempo ou até mesmo de acordo com a disponibilidade do sistema.

Independente da implementação utilizada, a configuração é muito parecida, e vale destacar também que o *Cron* (ou uma de suas variantes) é um programa de nível multiusuário, ou seja, cada usuário pode agendar suas tarefas individualmente. Isso ocorre graças ao próprio arquivo onde são armazenados os *scripts*: o **Crontab**.



Como o próprio nome sugere, a estrutura do Crontab consiste em tabelas e nessas tabelas são preenchidas todas as informações referentes à tarefa: **minutos; horas; dias do mês; mês; dias da semana; comando**. Abaixo um exemplo do comando ("crontab -l" lista a crontab):

```
crontab -l  
55 * * * 5 /home/maria/backup.sh
```

Agora vamos analisar alguns exemplos comentados:

# executa cinco minutos depois da meia-noite, todos os dias (\* = tudo)

```
5 0 * * * $HOME/bin/script.sh
```

# executa às 14h15min no dia 1º de cada mês

```
15 14 1 * * $HOME/bin/script.sh
```

# executa às 22h nos dias de semana (1 a 5 = segunda-feira a sexta-feira)

```
0 22 * * 1-5 $HOME/bin/script.sh
```

## Sistemas de Arquivos

Quando falamos em armazenamento de informações a longo prazo temos três requisitos:

1. Deve ser possível armazenar um grande volume de informações;
2. As informações devem "sobreviver" ao término do processo (programa em execução);
3. Vários processos devem ser capazes de acessar as informações ao mesmo tempo (ex.: vários programas acessando o mesmo banco de dados).

A solução mais comum é armazenar as informações em mídias de armazenamento (HDs, SSDs, entre outras), em unidades chamadas **arquivos**. Tais informações armazenadas em arquivos devem ser persistentes, ou seja, não devem ser afetadas pela criação ou término de um processo. O arquivo geralmente só pode "sumir" se for excluído por seu criador (ou pelo administrador do sistema).

O sistema operacional faz o gerenciamento dos arquivos (estrutura, nome, como acessar, usar etc.) através do **sistema de arquivos**. Enquanto o usuário final "enxerga" os arquivos através de seus nomes, localizações, tamanhos, tipos, proprietários etc., os projetistas dos sistemas de arquivos devem se preocupar também com aspectos mais técnicos, como por exemplo se vão ser utilizadas listas encadeadas ou mapa de bits para monitorar o espaço de armazenamento livre.

Mesmo lidando com sistemas de arquivos típicos do Windows (FAT32, NTFS etc.), os sistemas de arquivos nativos do Linux são, entre outros:

**EXT** (EXTended File System): durante algum tempo o **EXT2** foi o sistema de arquivos padrão do Linux, herdando características de outros sistemas de arquivos e sendo baseado no UFS (usado



no FreeBSD e Solaris), além de possuir semelhanças com os sistemas de arquivos HFS e HFS+ (Apple).

A menor unidade de alocação do Ext é o bloco (o equivalente ao *cluster* em sistemas de arquivos FAT e NTFS), cujos tamanhos possíveis são 1024, 2048 e 4096 (escolhido no processo de formatação).

**DESPENCA NA PROVA!**



Quando falamos especificamente da família de sistemas de arquivos EXT, temos que ter em mente que a versão 2 não possuía *journaling*. Somente na versão 3 surgiu! Em seguida vamos ver mais detalhes.

A principal característica do EXT3 é o uso do recurso de *journaling*, onde o sistema de arquivos mantém um *journal* (diário) das alterações realizadas. Esse "diário" armazena uma lista das alterações realizadas, permitindo que o sistema de arquivos seja reparado de forma muito rápida após o desligamento incorreto. O Ext3 possui três modos de operação:

- *ordered* (default): o *journal* é atualizado no final de cada operação. Isso faz com que exista uma pequena perda de desempenho, já que a cabeça de leitura do HD precisa realizar duas operações de gravação, uma no arquivo que foi alterado e outra no *journal* (que é um arquivo especialmente formatado);
- ● *writeback*: o *journal* armazena apenas informações referentes à estrutura do sistema de arquivos (metadados) e não em relação aos arquivos propriamente ditos, é gravado de forma mais ocasional, aproveitando os momentos de inatividade;
- *journal*: é o mais seguro, todavia mais lento. Nesse modo, o *journal* armazena não apenas informações sobre as alterações, mas também uma cópia de segurança de todos os arquivos modificados, que ainda não foram gravados no disco. Por ser o mais lento, é o modo menos usado.

Para finalizar a família EXT, existe o EXT4, com melhorias que incluem, por exemplo, a ampliação do endereçamento para 48 bits, o que permite endereçar um volume virtualmente ilimitado de blocos e partições de até 1 exabyte (1EB = 1024 petabytes)! O limite de 2 TB para os arquivos também foi removido, abrindo espaço para o armazenamento de arquivos gigantescos (no máximo 1 EB, o mesmo tamanho do volume!).

Além do EXT2, EXT3, EXT4, existem outros sistemas de arquivos conhecidos (e cobrados em provas de concurso), a saber:

- **Reiser FS**: utilizado geralmente no Linux, foi o primeiro com suporte a *journaling* (incluído no núcleo Linux 2.4). Seu futuro é incerto depois que seu criador, Hans Reiser, foi condenado pelo assassinato de sua esposa. Há voluntários que continuam com o projeto;
- **XFS**: inicialmente desenvolvido pela Graphics para o seu sistema operacional IRIX, posteriormente teve seu código fonte liberado e foi adaptado para funcionar no



Linux. É um sistema de arquivos desenvolvido em 64 bits, compatível com sistemas de 32 bits. Em plataformas de 64 bits, possui um limite de tamanho de 8 EB para um volume e para cada arquivo. É um sistema de arquivos com journaling;

- **Btrfs** (B-tree file system) é um sistema de arquivos baseado no princípio cópia em gravação (copy-on-write - COW), inicialmente desenvolvido pela Oracle para ser usado no Linux. Foi projetado para solucionar problemas como a falta de agrupamento de discos ou volumes, snapshots, checksums e uso de múltiplos volumes simultaneamente nos sistemas de arquivos do Linux. Possui a limitação de tamanho de volume igual à limitação de tamanho de arquivo (16 EB);
- Red Hat **GFS**: é um sistema de arquivos de cluster que permite que um cluster acesse simultaneamente um disparador de obstáculo do qual é dividido entre os nós. Trata-se de um sistema de arquivos nativo que se conecta diretamente por meio da interface com a camada VFS da interface de sistema de arquivo Kernel Linux. O GFS emprega os metadados distribuídos e diários múltiplos para uma operação mais eficiente em um cluster;
- **NFS** (Network File System): protocolo de sistema de arquivos distribuído, originalmente desenvolvido pela Sun Microsystems, que permite que um usuário em um computador cliente acesse arquivos através de uma rede como se estivesse acessando na máquina local. Utiliza o sistema Open Network Computing Remote Procedure Call (ONC RPC);
- **Swap**: memória virtual do Linux que tem uma partição específica (diferente do Windows que utiliza um arquivo de paginação). Possui uma organização própria, sem utilizar um sistema de arquivos (ou pode ser entendido como tendo um sistema de arquivos próprio para isso).

## LVM (Logical Volume Manager)

O gerenciador de volumes lógicos (LVM) é um recurso que visa facilitar a vida dos administradores de sistema. Embora os usuários domésticos geralmente prefiram criar apenas uma ou duas partições ("/" ou "/" + "/home") durante a instalação do sistema, nos servidores são utilizadas muito mais partições por questões de segurança e desempenho.

O problema é que ao dividir o HD em várias partições diferentes surge o problema do aproveitamento do espaço, já que sempre algumas partições ficam cheias embora ainda exista muito espaço livre nas outras. Geralmente é possível redimensionar as partições, mas esse é sempre um processo demorado e que traz risco de perda de dados.

O LVM resolve o problema permitindo agrupar vários HDs em uma única unidade que pode ser dividida em vários volumes lógicos, vistos pelo sistema operacional como se fossem partições de disco. A grande vantagem é que novos volumes podem ser criados, excluídos ou redimensionados rapidamente e sem a necessidade de reiniciar o servidor.

Caso sejam usados HDs SCSI com suporte a hot-swap, é possível adicionar, remover ou ainda substituir HDs, fazendo as alterações necessárias nos volumes lógicos, tudo sem interrupções! O fato do sistema "visualizar" todos os HDs instalados como uma única unidade também facilita os backups.



Para utilizar o LVM é preciso compilar o Kernel ativando as opções "Multiple devices driver support (RAID and LVM)" e "Logical volume manager (LVM) Support". Em geral as distribuições já trazem esses dois componentes compilados como módulos, bastando ativá-los usando o comando "modprobe".

## Estrutura dos Diretórios

Uma característica forte do Linux é a sua estrutura de diretórios, que pode possuir pequenas modificações entre as distribuições, porém são semelhantes. Segundo o conceito FHS (*File Hierarchy Standard*), que é um padrão para que softwares e usuários possam prever a localização de arquivos e diretórios instalados, abaixo é mostrada a estrutura e o que é armazenado em cada diretório/subdiretório:

- /etc: dados de configuração (scripts de inicialização, tabela de sistemas de arquivo, configurações de login, configuração da fila de impressão, entre outros). A gente pensa que "etc" significa "etcetera", mas "etc" quer dizer "Environment Tables and Controls";
- /usr: maior parte dos aplicativos e bibliotecas do sistema;
- /usr/bin: executáveis dos programas;
- /usr/lib: bibliotecas e arquivos compartilhados;
- /usr/src: código-fonte de programas;
- /usr/doc: documentação em geral;
- /home: arquivos dos usuários (arquivos de trabalho, músicas, filmes, downloads etc.);
- /root: diretório home do root. Obs.: Esse diretório não é a mesma coisa que a raiz do sistema /;
- /var: sites hospedados (nem sempre), bases de dados do MySQL etc.;
- /var/log: logs de diversos serviços;
- /var/www: sites hospedados (nem sempre);
- /boot: imagem do Kernel e o initrd (initial ram disk), carregados no início do boot;
- /bin: comandos básicos, como "cd", "ls" e "cat";
- /lib: módulos do Kernel e bibliotecas básicas do sistema;
- /opt: arquivos de aplicativos adicionais (opcionais, ou "optionals"), que não são essenciais para o sistema. Cada aplicativo tem uma subpasta com seu respectivo nome;
- /proc: informações de depuração do Kernel, configurações que habilitam e desabilitam o suporte à algum elemento no Kernel;
- /tmp: arquivos temporários.

## Diretório /etc

Como já descrito no tópico anterior, o diretório /etc armazena dados de configuração. Vejamos alguns dos arquivos de configuração com suas descrições:

- /etc/group: arquivo de sistema que define os grupos aos quais os usuários pertencem em sistemas UNIX, como o Linux e o BSD;



- /etc/passwd: determina quem pode acessar legitimamente o sistema e o que eles podem fazer uma vez dentro;
- /etc/shadow: armazena as senhas dos usuários criptografadas, além de informações sobre datas de expiração e validade das contas;
- /etc/syslog.conf: usado para controlar a saída dos arquivos de log do daemon syslogd;
- /etc/profile: contém comandos que são executados para todos os usuários do sistema no momento do login. Somente o usuário root pode ter permissão para modificar esse arquivo;
- /etc/network/interfaces: usado pelos programas ifup e ifdown, respectivamente para ativar e desativar as interfaces de rede;
- /etc/resolv.conf: usado para configurar a resolução do host via DNS;
- /etc/exports: usado para especificar quais diretórios o servidor NFS (Network File System) exporta;
- /etc/ldap.conf: usado para a configuração padrão a ser aplicada na execução de clientes LDAP (Lightweight Directory Access Protocol);
- /etc/ssh/sshd\_config: configuração do SSH.

## Memória Virtual

O conceito de memória virtual está presente nos sistemas operacionais modernos (incluindo o Linux, claro). Sabemos que um programa necessita passar pela memória principal para ser executado, mas e se não há memória RAM suficiente para executar todo o programa? Aí surgiu a ideia de executar o programa em partes e apenas as partes que necessitam ser executadas no momento ficam na memória RAM.

A **memória virtual** é uma área de troca de dados que serve como uma extensão da memória RAM. Por exemplo, se há 20 MB de memória RAM livre e um programa de 30 MB é executado, uma parte é carregada e o restante fica na memória virtual, para depois serem copiados para a memória principal a fim de serem executados.

Isso ocorre à medida que a execução do programa for acontecendo e o programa em questão (ou outros programas) liberarem a memória que ocupavam para o uso de outro programa. Podemos verificar que há um fluxo contínuo de dados entre o disco rígido (onde fica a memória virtual) e a memória RAM. Por isso concluímos que pouca memória RAM implica em usar muita memória virtual, que por sua vez, implica em usar muita leitura/escrita no disco rígido, deixando o funcionamento geral do sistema mais lento!

Geralmente ao instalar qualquer distribuição GNU/Linux, na opção automático da partição do disco onde será instalado o sistema, também encontramos a criação automática da **partição Swap**. Se optar por uma instalação personalizada, deve-se atentar para a criação de uma partição e marcá-la como Swap. Abaixo é mostrada uma partição Swap com cerca de 16 GB através do Gparted (utilitário que lista e permite alterar partições, bastante utilizado).

Quando não há partição livre ou não há espaço para a criação da memória virtual e o sistema operacional necessita de memória de troca para o melhor desempenho dos processos é possível



criar o **arquivo de memória Swap**. Nada impede que ambas as soluções sejam utilizadas simultaneamente!

## Gerenciadores de Pacotes

Segundo Morimoto, os **pacotes são as "peças" que formam todas as distribuições Linux** e podem conter programas, bibliotecas de sistema ou mesmo algumas outras "coisas" como papéis de parede e ícones. Alguns programas grandes (KDE, por exemplo) são divididos em vários pacotes para que o usuário possa instalar apenas as partes que interessam, ficando com um sistema mais enxuto.

Alguns pacotes dependem de outros (um programa pode precisar de uma biblioteca que faz parte de outro pacote, por exemplo), as chamadas **dependências**. Para evitar que algo não funcione ou com pacotes desnecessários, o instalador automaticamente verifica as dependências de cada pacote, adicionando ou removendo pacotes relacionados a ele. É por isso que às vezes ao marcar um determinado pacote alguns outros são marcados junto.

Instalar ou remover pacotes no Linux hoje em dia é muito fácil, pois existem vários gerenciadores de pacotes para facilitar essa tarefa. Vamos ver apenas quatro deles, pois são os geralmente cobrados em provas de concurso.

**APT**: resolve dependências para sistemas baseados em Debian (incluindo o Ubuntu). Possui uma sintaxe muito simples. Vamos a alguns exemplos para:

- atualizar os repositórios de software: `sudo apt-get update` ou `sudo apt update` (a lista com os repositórios devem estar no arquivo `"/etc/apt/sources.list"`);
- atualizar seus pacotes já instalados (softwares): `sudo apt-get upgrade` ou `sudo apt upgrade`;
- uma atualização mais completa (tenta atualizar pacotes para a versão mais recente e remover dependências mais antigas ou não utilizadas): `sudo apt-get dist-upgrade` ou `sudo apt full-upgrade`;
- atualização dos repositórios e atualizar também os pacotes já instalados: `sudo apt-get update && sudo apt-get upgrade` ou `sudo apt update && sudo apt upgrade`;
- instalar um software: `sudo apt-get install nome_do_pacote` ou `sudo apt install nome_do_pacote`;
- remover um pacote: `sudo apt-get remove nome_do_pacote` ou `sudo apt remove nome_do_pacote`.

Ao remover um software do sistema usando o comando `apt-get remove`, o APT realiza todo o trabalho de remoção de dependências não utilizadas. Algumas dependências podem permanecer no sistema, então é só utilizar a opção `autoremove`: `sudo apt-get autoremove` ou `sudo apt autoremove`.



Para procurar um pacote a ser instalado, basta usar: `sudo apt-cache search termo_pesquisa` ou `sudo apt search termo_pesquisa`.

O APT atualmente não oferece a possibilidade de instalar um pacote de um *site*, ou seja, o usuário deve encontrar e baixar o pacote a ser instalado por conta própria, utilizando nesse caso o DPKG.

**DPKG:** criado para instalar pacotes ".deb" na distribuição Debian. Também é utilizado por outras distribuições, como Ubuntu, KNOPPIX etc. Os principais comandos utilizados pelo gerenciador de pacotes DPKG são `dpkg` e `dpkg-reconfigure`. A sintaxe é: `dpkg [opções] nome_pacote`

Opções:

- i : instalação simples.
- r : desinstala o pacote - exceto arquivos de configuração do pacote.
- P : desinstala o pacote - todos os arquivos do pacote.
- l : exibe os pacotes que estão instalados.
- p : exibe informações sobre o pacote instalado.
- s : exibe o status do pacote.
- l : exibe informações sobre pacotes não instalados.
- S : exibe o pacote do qual o arquivo faz parte.
- L : exibe os arquivos que fazem parte de um pacote instalado.
- c : exibe os arquivos que fazem parte de um pacote não instalado.
- help : exibe uma mensagem de ajuda.

Alguns exemplos:

- Instalação de um pacote: `# dpkg -i pacote.deb`
- Desinstalação de um pacote, mantendo os seus arquivos de configuração: `# dpkg -r pacote`
- Desinstalação de um pacote, inclusive os seus arquivos de configuração: `# dpkg -P pacote`
- Exibição de informações do pacote não instalado: `# dpkg -l pacote.deb`

Em relação ao comando `dpkg-reconfigure`: reconfigura pacotes ".deb" após terem sido instalados utilizando o `debconf` (sistema de configuração de pacotes ".deb"). Esse comando fará perguntas para reconfigurar o pacote.

Sintaxe: `dpkg-reconfigure opções nome_pacote`

Opções:



-a ou --all :: reconfigura todos os pacotes

-h ou --help :: exibe ajuda

Exemplo: Reconfigura o pacote ssh à # dpkg-reconfigure ssh

**RPM:** criado pela Red Hat para instalar pacotes ".rpm", sendo utilizado por várias distribuições, como Red Hat Enterprise/Fedora Core, Mandriva, SuSe, entre outras. O conjunto completo de modos básicos de operação do RPM é a criação, instalação, desinstalação, atualização, consulta e verificação de pacotes. O principal comando do gerenciador de pacotes RPM, é o rpm.

Sintaxe: rpm [opções] nome\_pacote

Algumas opções:

-i: instalação simples.

-v: exibe detalhes da instalação.

-h: mostra o caractere "#", enquanto o programa é instalado.

-U: atualização do programa de uma versão anterior para uma mais recente.

--nodeps: não procura dependências.

--force: força a instalação.

--root diretório: utiliza o sistema com raiz em diretório para todas as operações.

-e: desinstala o pacote.

--import: importa a chave pública GPG do distribuidor de um pacote.

--help: exibe uma mensagem de ajuda.

-q: consulta (*query*) se um pacote já está instalado.

Exemplos:

#rpm -q httpd à Consulta se o pacote está instalado.

# rpm -ivh tree.i386.rpm à Instala o pacote, exibindo detalhes da instalação e mostra o caractere "#", enquanto o programa é instalado.

# rpm -e tree à Desinstala o pacote tree.

**YUM:** resolve dependências de pacotes para distribuições que utilizam o RPM. É um gerenciador de pacotes padrão incluído em algumas distribuições baseados no REDHAT, incluindo o Fedora e CentOS. Muito parecido com o APT, vejamos alguns comandos:

- Atualização dos repositórios do YUM: sudo yum update;
- Instalação de um pacote: sudo yum install nome\_do\_pacote;



- Remoção de um pacote: `sudo yum remove nome_do_pacote`;
- Busca de um pacote que será instalado: `sudo yum search nome_do_pacote`.

O YUM não inclui um comando `autoremove` para encontrar e remover dependências não utilizadas, mas incluí um recurso para a instalação de um pacote a partir de uma URL (função que não existe no APT):

```
sudo yum install $url
```

## Configuração de Serviços ou Funcionalidades

Diversos serviços ou funcionalidades no Linux são configurados através de arquivos texto e alguns deles costumam ser cobrados em provas de concurso. Vamos a eles...

**Segurança da senha do usuário:** os arquivos que armazenam os dados dos usuários do Linux são o `/etc/passwd` e o `/etc/shadow`. Cuidado! O `passwd` não armazena a senha, embora o nome sugira isso! Vamos ver o exemplo de uma linha de cada arquivo, relativo ao usuário "estrategia". Primeiro o arquivo `passwd`:

```
estrategia:x:1010:1010:Estrategia Concursos:/home/estrategia:/bin/bash
```

Separado por dois pontos, podemos identificar os seguintes campos:

- nome do usuário = "estrategia";
- "senha" do usuário (o "x" indica que a senha está armazenada no arquivo `/etc/shadow`);
- identificação do usuário (uid = 1010);
- identificação do grupo (gid = 1010);
- comentário sobre o usuário (geralmente o nome completo) = "Estrategia Concursos";
- diretório principal (home) do usuário = `/home/estrategia`;
- *shell* utilizado, ao logar, pelo usuário = `/bin/bash`.

Agora vamos ao arquivo `shadow`:

```
estrategia:$6$AkC...CdbF:17355:0:99999:7:::
```

Separado por dois pontos, podemos identificar os seguintes campos:

- nome do usuário = "estrategia";
- *hash* da senha do usuário = "\$6\$AkC...CdbF" (obs.: inserir os três pontos no meio porque esse campo é muito grande);
- última vez (em dias) que a senha foi modificada, desde 1 de janeiro de 1970 = 17355;
- depois de alterada a senha o usuário só poderá alterar novamente depois de x dias = 0 (0 significa que está desabilitado);
- a partir do momento que o usuário altera a senha, é obrigado a alterá-la novamente depois de x dias = 99999;
- quantos dias antes do fim do prazo para alterar a senha, deve ser mostrado um alerta = 7.



Os outros dois campos estão vazios, mas significam: inativo (quanto tempo sua conta ficou inativa/desabilitada), expirada (quantos dias depois de 01/01/1970 que sua conta está desabilitada ou expirada).

**Configuração de interfaces de rede:** os programas ifup (habilita interfaces de rede) e ifdown (desabilita) se baseiam na configuração do arquivo "/etc/network/interfaces", o qual vamos ver um exemplo abaixo, com comentários.

```
auto lo                # configuração automática para lo
iface lo inet loopback # lo é uma interface de loopback
auto eth0              # configuração automática para eth0
allow-hotplug eth0    # ativa a interface eth0 (com o ifup)
iface eth0 inet static address 192.168.100.1 netmask 255.255.255.0 network 192.168.100.0
broadcast 192.168.100.255
# a última linha faz com que ifup ative a eth0 com o endereço fixo 192.168.100.1/24
```

Para os próximos dois serviços, é muito difícil cobrar algo muito específico, basicamente as bancas cobram os nomes de arquivos, então vamos direto ao ponto.

**DNS (Domain Name System):** o serviço de resolução de nomes tem um arquivo de configuração, o "/etc/resolv.conf". Lembre que "resolv" está ligado a "resolver um nome", fica mais fácil de lembrar.

**NFS (Network File System):** o sistema de arquivos em rede utilizado pelo Linux utiliza um arquivo que controla quais diretórios o servidor NFS exporta, o "/etc/exports".

## Interoperabilidade (Samba)

Quando falamos em sistemas de arquivos FAT, NTFS, EXT, XFS, entre outros, estamos falando em sistemas de arquivos "locais", os que permitem o acesso a uma mídia interna ou conectada a um computador, por exemplo. Existem também os sistemas de arquivos em rede, que dão a ilusão de um acesso local, porém o acesso é remoto. Os mais conhecidos são o NFS e o SMB/CIFS, os quais veremos a seguir.

O **NFS (Network File System)** é um protocolo de sistema de arquivos distribuído, **originado no Unix**, que permite que um usuário em um computador cliente acesse arquivos através de uma rede como se estivesse acessando na máquina local. O cliente NFS tem a finalidade de tornar o acesso remoto transparente para o usuário do computador, e esta interface cliente e servidor, executada pelo NFS através dos protocolos Cliente-Servidor, fica bem definida quando o usuário, ao chamar um arquivo/diretório no servidor, lhe parece estar acessando localmente, sendo que está trabalhando com arquivos remotamente.

Uma máquina Linux pode ser um servidor NFS e um cliente NFS, o que significa que pode exportar sistemas de arquivo para outros sistemas e montar sistemas de arquivo exportados de outras máquinas. Por exemplo, um grupo de usuários pode ter acesso aos arquivos de um determinado projeto usando um diretório compartilhado do sistema de arquivo NFS montado no diretório "/projeto". Para acessar os arquivos compartilhados, o usuário vai até o diretório



"/projeto", sem senhas ou comandos especiais para lembrar. Dessa forma, os usuários trabalham como se o diretório estivesse em suas máquinas locais.

Algumas **características do NFS** são:

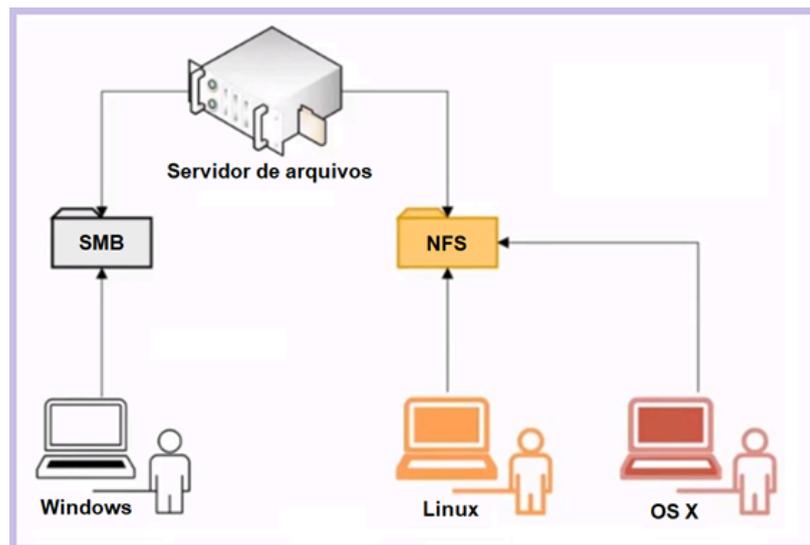
- Arquitetura cliente-servidor: o servidor recebe as requisições vindas do cliente, verifica a validade, executa e retorna ao cliente;
- *Stateless* (sem estado): considera cada requisição como uma transação independente que não está relacionada a qualquer requisição anterior, de forma que a comunicação consista em pares requisição/resposta independentes. Diante de uma falha no sistema, quando o servidor NFS volta a funcionar corretamente, o estado anterior é restaurado e o funcionamento continua como se não houvesse ocorrido nenhum problema;
- *Caching*: permite que as informações utilizadas mais recentemente sejam alocadas para posterior uso, ou pode carregar os dados em antecipação a operações futuras;
- *File Locking* (bloqueio de arquivo): permite que um processo tenha acesso exclusivo a um arquivo ou parte deste, e força outros processos que estão solicitando acesso ao mesmo arquivo, aguardarem a liberação;
- Utiliza o protocolo RPC (*Remote Procedure Call*): define um modo independente do sistema para processos se comunicarem através de uma rede de computadores, sendo que o cliente faz a chamada do procedimento remoto no servidor;
- Protocolo de transporte padrão é o UDP, no entanto, é possível utilizar o TCP;
- Compartilhar arquivos de um servidor NFS é conhecido como exportar os diretórios.

Para que os clientes possam acessar o servidor NFS é necessário que os seguintes *daemons* estejam em execução:

- *nfsd*: *daemon* NFS, atende requisições dos clientes NFS;
- *mountd*: *daemon* de montagem NFS, executa as solicitações passadas pelo *nfsd*;
- *portmap*: *daemon portmapper*, permite que clientes NFS descubram qual porta o servidor NFS está utilizando.

Na figura abaixo podemos ver o acesso a um servidor de arquivos através da rede. Note que máquinas Unix-like (ex.: Linux e OS X) utilizam o NFS como protocolo e as máquinas Windows utilizam o SMB/CIFS, que veremos o funcionamento na sequência.





Antes de começar a falar do protocolo em si, deixo bem claro que as questões de concurso já cobraram alguns detalhes do NFS, mas em relação ao SMB/CIFS a cobrança é superficial. Vamos lá...

O **SMB/CIFS** (*Server Message Block/Common Internet File System*) é um protocolo de redes cujo uso mais comum como é o compartilhamento de arquivos em uma rede local. Ele permite que o cliente manipule arquivos como se estes estivessem em sua máquina local (mesma coisa que o NFS, heim?). Algumas das operações suportadas são: leitura, escrita, criação, exclusão e renomeação, sendo que os arquivos/diretórios manipulados estão em um servidor remoto.

O protocolo SMB/CIFS funciona através do envio de pacotes do cliente para o servidor. O servidor recebe o pacote, verifica se a requisição é válida (se o cliente possui as permissões apropriadas), executa a requisição e retorna um pacote de resposta ao cliente. O cliente verifica o pacote de resposta para determinar se a requisição inicial obteve êxito. O protocolo mais utilizado para transporte confiável é o NetBIOS sobre TCP (NBT). Outros protocolos foram utilizados na camada de transporte, mas o NBT se tornou o mais utilizado.

O SMB/CIFS é muito utilizado pelos sistemas operacionais Windows, podendo funcionar como um servidor ou cliente. Isso não quer dizer que sistemas Unix-like não possuam também esse protocolo! Na verdade, a maioria dos sistemas Unix-Like (**Linux**, por exemplo) possuem uma implementação de cliente/servidor do SMB/CIFS via **Samba**, que é um "software servidor" para Linux que permite o gerenciamento e compartilhamento de recursos em redes formadas por computadores com o Windows. Dessa forma é possível utilizar o Linux como servidor de arquivos, servidor de impressão, entre outros, como se a rede utilizasse servidores Windows (Server 200x, 201x, entre outros).

Resumindo: o Samba é um componente que realiza a comunicação entre servidores Linux e Windows, permitindo que eles compartilhem recursos de disco e de impressão. O *daemon* `smbd` possui dois modos de compartilhamento. O primeiro modo (*sharemode*) é mais simples, gerando uma senha para cada recurso que é compartilhado. O segundo modo (*user*) permite o compartilhamento de vários recursos com um único *login* e senha. Os privilégios de acesso aos recursos devem ser sempre mediados pelos administradores do sistema.

Um outro componente interessante do pacote Samba é o *nmbd*, um servidor de nomes NetBios (mais conhecido por **WINS - Windows Internet Name Server**) responsável por entender e responder solicitações de resolução de nomes NetBios sobre IP. É ele o responsável pelo "aparecimento" do ícone do servidor Samba no ambiente de rede do Windows! Obs.: Se você não sabe o que é essa resolução de nomes, pense que é algo como o DNS, mas não de forma hierárquica, é uma resolução feita pelo Windows (sabendo isso já está bom para a prova 😊).

## Serviço de Diretório

Um serviço de diretório é um sistema de software que armazena, organiza e fornece acesso a informações em um diretório do sistema operacional. Diretórios podem ser muito limitados em escopo, suportando apenas um pequeno conjunto de tipos de nós e tipos de dados, ou eles podem ser muito amplos, suportando um conjunto de tipos arbitrário ou extensível. No DNS os nós são nomes de domínio e os itens de dados são endereços IP (e apelidos, nomes de servidor de e-mail etc.).

Em um diretório utilizado por um sistema operacional de rede, os nós representam **recursos** que são gerenciados pelo sistema operacional, incluindo usuários, computadores, impressoras, entre outros recursos compartilhados. Muitos serviços de diretório têm sido utilizados desde o aparecimento da Internet, mas o mais utilizado é um descendente do serviço de diretório X.500.

Antes de vermos uma implementação de serviço de diretório para o Linux (OpenLDAP), é importante entendermos o protocolo que fornece mecanismos de acesso aos objetos, o **LDAP (Lightweight Directory Access Protocol)**. Entidades internacionais (ITU, ISO, IETF, entre outras) trabalham na definição de padrões diversos, incluindo a padronização que dá suporte a serviços de diretórios. Um padrão de uso genérico é o **X.500** (da ISO) que possui uma grande abrangência, mas é muito complexo e não foi adotado em sua íntegra como um padrão de mercado. Um padrão mais "light" que de fato se tornou um padrão de mercado foi o LDAP.

O padrão LDAP define um sistema de nomeação hierárquico, no qual é possível referenciar qualquer objeto que esteja no AD. Um nome LDAP é composto pelo caminho completo do objeto (ex.: uma impressora, um computador etc.), partindo do domínio raiz até chegar ao objeto em si. Algumas abreviaturas (**atributos**) são utilizadas nessa nomenclatura hierárquica:

- **cn**: *common name* (nome da conta de um usuário, grupo etc.);
- **sn**: sobrenome (*surname*);
- **ou**: faz referência a uma unidade organizacional;
- **dc**: componente de domínio (normalmente o nome do domínio);
- **o**: nome da organização (geralmente o domínio raiz);
- **c**: *country* - país (normalmente não utilizado).

Vamos a um exemplo de um nome LDAP:

CN=evandrodv, OU=professores, DC=ti, DC=estrategiaconcursos.com.br à esse nome representa o usuário "evandrodv", cuja conta está contida na unidade organizacional "professores", no domínio "ti.estrategiaconcursos.com.br". Obs.: os dois componentes de domínio foram concatenados.



Por padrão um servidor LDAP "escuta" na porta 389 (TCP) e as principais características do protocolo são:

- Baseado em padrão aberto: qualquer desenvolvedor pode acessar sua especificação e realizar a implementação;
- Possui APIs bem definidas: facilita a vida dos programadores;
- Maior velocidade de consulta que um BD relacional;
- Replicável e distribuível;
- Facilita a localização de informações e recursos: pesquisa feita nome.

Algumas **operações (comandos)** que podem ser utilizados através do LDAP são:

- Bind: autentica e especifica a versão do protocolo LDAP;
- Search: procura/recupera entradas dos diretórios;
- Compare: compara uma entrada com determinado valor;
- Add: adiciona uma nova entrada;
- Delete: exclui uma entrada;
- Modify: modifica uma entrada;
- Modify DN: move ou renomeia uma entrada;
- Abandon: aborta uma requisição prévia;
- Unbind: fecha a conexão;
- Extended Operation: operação genérica para definir outras operações;
- StartTLS: protege a conexão com o TLS - implementada a partir da versão 3 do LDAP.

A representação dos dados é realizada através de uma estrutura hierárquica na forma de árvore (*Directory Information Tree - DIT*), a qual consiste em entradas de DNs (*Distinguished Names*). O LDAP utiliza a DIT como estrutura de dados fundamental:

Como podemos ver, a árvore de diretório possui uma forma hierárquica:

- Primeiro o diretório raiz;
- Na sequência: a rede corporativa, os departamentos e por fim os computadores dos usuários e os recursos de rede.

Alguns conceitos que também já foram cobrados em concursos são mostrados na sequência.

**Schema:** conjunto de objetos e atributos para o armazenamento. É modelado de acordo com o padrão X.500 da ISO.



Cada entrada (objeto) possui um identificador único (*dn - distinguished name*), o qual consiste de seu Relative Distinguished Name (RDN), construído de algum(ns) atributo(s) na entrada, seguido pelo DN da entrada pai.

**Escalabilidade:** é possível replicar servidores LDAP e incluir novos servidores à medida que aumenta a estrutura da organização. Ou seja, não é uma estrutura "engessada".

Como já mencionado anteriormente, uma solução bastante conhecida para Linux é o **OpenLDAP** (que também possui versões para Windows, Solaris, entre outros sistemas operacionais). Vamos ver como instalar o OpenLDAP:

```
sudo apt install slapd ldap-utils
```

O principal arquivo de configuração (pelo menos para concursos) é o `"/etc/ldap.conf"`, que é usado para configurar os padrões a serem aplicados quando da execução dos clientes LDAP.

## Distribuições Linux

As bancas raramente pedem detalhes de determinadas distribuições, mas é importante ter uma noção daquelas mais conhecidas/cobradas em concursos, conforme veremos a seguir.

### Red Hat Enterprise Linux (RHEL)

O Red Hat Enterprise Linux (RHEL) é uma distribuição Linux de classe empresarial desenvolvida pela Red Hat. Alguns aspectos importantes sobre o RHEL são:

- **Estabilidade e Segurança:** O RHEL é conhecido por sua estabilidade e segurança, tendo sido projetado para oferecer um ambiente operacional confiável para servidores e sistemas críticos de negócios (ambiente empresarial);
- **Compatibilidade Binária:** Oferece compatibilidade binária com outras distribuições baseadas em Red Hat, como o CentOS ou o Oracle Linux. Ou seja, os aplicativos desenvolvidos e testados no RHEL normalmente podem ser executados nessas distribuições sem modificações significativas;
- **Modelo de Assinatura:** O RHEL é distribuído através de um modelo de assinatura (software pago). As assinaturas incluem acesso ao software, suporte técnico, atualizações de segurança, entre outros itens;
- **Suporte de Longo Prazo:** Existe o suporte de longo prazo para cada versão principal do RHEL. As empresas podem implantar uma versão do RHEL e receber atualizações de segurança e manutenção por um período estendido, algo em torno de 10 anos;
- **Grande Conjunto de Software:** O RHEL é acompanhado por um amplo conjunto de software, incluindo ferramentas de gerenciamento de sistemas, servidores Web, bancos de dados, entre outros;



- Uso em Ambientes Empresariais: O RHEL foi projetado e é amplamente utilizado em ambientes corporativos, incluindo data centers, servidores, nuvens e ambientes de missão crítica. Isso ocorre por causa de sua estabilidade, segurança e suporte de longo prazo, visto que as organizações buscam uma plataforma confiável para suas operações de TI.

Resumindo, podemos dizer que o Red Hat Enterprise Linux é uma distribuição robusta e confiável, projetada para atender às necessidades das empresas em termos de estabilidade, segurança e suporte. É uma escolha popular para ambientes de produção críticos e é amplamente utilizado em todo o mundo.

## CentOS

CentOS é uma distribuição Linux de código aberto baseada no Red Hat Enterprise Linux (RHEL). É conhecida por sua estabilidade e é amplamente utilizada em servidores e ambientes de computação em nuvem. Trata-se de uma distribuição gratuita e é mantida pela comunidade de desenvolvedores. Oferece suporte de longo prazo e é uma escolha popular para empresas que desejam uma alternativa de baixo custo em relação ao RHEL. Alguns detalhes sobre o CentOS são:

- Origem/história: Foi lançado em 2004 como um clone do Red Hat Enterprise Linux (RHEL). É desenvolvido pela comunidade e mantido pelo CentOS Project, independente da Red Hat, embora seja altamente compatível com o RHEL;
- Objetivo: Fornecer uma distribuição estável, de código aberto e de uso gratuito que seja compatível com o RHEL. Geralmente é usado em servidores, infraestrutura de rede e ambientes de computação em nuvem;
- Compatibilidade: Existe a compatibilidade binária com o RHEL, ou seja, os aplicativos e ferramentas desenvolvidos para o RHEL geralmente podem ser executados no CentOS sem a necessidade de modificações significativas;
- Ciclo de Lançamento e Suporte: Oferece versões de lançamento regular, com suporte a longo prazo (LTS). As versões principais recebem atualizações de segurança e manutenção por um período estendido de tempo, algo em torno de 10 anos;
- Comunidade e Documentação: Existe uma grande comunidade de usuários e desenvolvedores que oferecem suporte através de fóruns, listas de discussão e outros meios online. Também existe vasta documentação disponível, incluindo guias de instalação, tutoriais e referências de configuração;
- Recursos e Pacotes: O CentOS inclui uma ampla gama de pacotes de software, incluindo ferramentas de sistema, servidores Web, bancos de dados, linguagens de programação, etc. Os administradores podem usar o gerenciador de pacotes YUM (Yellowdog Updater, Modified) para instalar, atualizar e remover softwares no sistema.

Resumindo, o CentOS é uma distribuição Linux popular e confiável, adequada para diversas situações, desde servidores corporativos até infraestrutura de nuvem. Essa distribuição oferece estabilidade, compatibilidade e suporte de longo prazo, tornando-se uma escolha popular para muitas organizações e profissionais de TI.



## Oracle Linux

Oracle Linux é uma distribuição baseada no Red Hat Enterprise Linux (RHEL) e desenvolvida pela Oracle Corporation. Alguns pontos importantes sobre o Oracle Linux são:

- Baseado no RHEL: O Oracle Linux é construído com base no código fonte do RHEL, fazendo com que compartilhe muitas características e funcionalidades com essa distribuição. Oferece compatibilidade binária com o RHEL, permitindo que aplicativos desenvolvidos e testados no RHEL sejam executados no Oracle Linux sem modificações significativas;
- Kernel Unbreakable Enterprise (KUE): Versão personalizada do kernel Linux desenvolvida pela Oracle, incluindo melhorias de desempenho, segurança e suporte a tecnologias específicas da Oracle;
- Modelo de Assinatura: Assim como o RHEL, o Oracle Linux é distribuído usando um modelo de assinatura (paga). As assinaturas incluem acesso ao software, suporte técnico, atualizações de segurança etc.;
- Recursos Adicionais: Além do sistema operacional em si, o Oracle Linux é acompanhado por uma diversos recursos adicionais, como o Oracle Linux Support, que oferece suporte técnico 24/7, e o Oracle Linux Premier Support, que inclui ferramentas de gerenciamento e monitoramento avançadas;
- Uso em Ambientes Oracle: O Oracle Linux é frequentemente utilizado em ambientes onde as soluções da Oracle são utilizadas, como bancos de dados Oracle Database, Oracle Middleware, entre outros. Ele é otimizado para integração e desempenho em ambientes Oracle;
- Open Source e Comunidade: Embora seja desenvolvido e mantido pela Oracle Corporation, por incrível que pareça, o Oracle Linux é uma distribuição de código aberto. A Oracle contribui para a comunidade de software livre e permite que os usuários contribuam e modifiquem o código fonte do Oracle Linux.

Resumindo, o Oracle Linux é uma distribuição empresarial robusta e confiável, baseada no RHEL e otimizada para integração com as soluções Oracle. Oferece compatibilidade, desempenho e suporte para empresas que dependem das tecnologias da Oracle em seus ambientes de TI.



## CONFIGURAÇÃO DE SERVIÇOS OU FUNCIONALIDADES

Diversos serviços ou funcionalidades no Linux são configurados através de arquivos texto e alguns deles costumam ser cobrados em provas de concurso. Vamos a eles...

**Segurança da senha do usuário:** os arquivos que armazenam os dados dos usuários do Linux são o `/etc/passwd` e o `/etc/shadow`. Cuidado! O `passwd` não armazena a senha, embora o nome sugira isso! Vamos ver o exemplo de uma linha de cada arquivo, relativo ao usuário "estrategia". Primeiro o arquivo `passwd`:

```
estrategia:x:1010:1010:Estrategia Concursos:/home/estrategia:/bin/bash
```

Separado por dois pontos, podemos identificar os seguintes campos:

- nome do usuário = "estrategia";
- "senha" do usuário (o "x" indica que a senha está armazenada no arquivo `/etc/shadow`);
- identificação do usuário (`uid = 1010`);
- identificação do grupo (`gid = 1010`);
- comentário sobre o usuário (geralmente o nome completo) = "Estrategia Concursos";
- diretório principal (home) do usuário = `/home/estrategia`;
- *shell* utilizado, ao logar, pelo usuário = `/bin/bash`.

Agora vamos ao arquivo `shadow`:

```
estrategia:$6$Akc...CdbF:17355:0:99999:7:::
```

Separado por dois pontos, podemos identificar os seguintes campos:

- nome do usuário = "estrategia";
- *hash* da senha do usuário = "\$6\$Akc...CdbF" (obs.: inserir os três pontos no meio porque esse campo é muito grande);
- última vez (em dias) que a senha foi modificada, desde 1 de janeiro de 1970 = 17355;
- depois de alterada a senha o usuário só poderá alterar novamente depois de x dias = 0 (0 significa que está desabilitado);
- a partir do momento que o usuário altera a senha, é obrigado a alterá-la novamente depois de x dias = 99999;
- quantos dias antes do fim do prazo para alterar a senha, deve ser mostrado um alerta = 7.

Os outros dois campos estão vazios, mas significam: inativo (quanto tempo sua conta ficou inativa/desabilitada), expirada (quantos dias depois de 01/01/1970 que sua conta está desabilitada ou expirada).



**Configuração de interfaces de rede:** os programas ifup (habilita interfaces de rede) e ifdown (desabilita) se baseiam na configuração do arquivo "/etc/network/interfaces", o qual vamos ver um exemplo abaixo, com comentários.

```
auto lo                # configuração automática para lo
iface lo inet loopback # lo é uma interface de loopback
auto eth0              # configuração automática para eth0
allow-hotplug eth0    # ativa a interface eth0 (com o ifup)
iface eth0 inet static address 192.168.100.1 netmask 255.255.255.0 network 192.168.100.0
broadcast 192.168.100.255
# a última linha faz com que ifup ative a eth0 com o endereço fixo 192.168.100.1/24
```

Para os próximos dois serviços, é muito difícil cobrar algo muito específico, basicamente as bancas cobram os nomes de arquivos, então vamos direto ao ponto.

**DNS (Domain Name System):** o serviço de resolução de nomes tem um arquivo de configuração, o "/etc/resolv.conf". Lembre que "resolv" está ligado a "resolver um nome", fica mais fácil de lembrar.

**NFS (Network File System):** o sistema de arquivos em rede utilizado pelo Linux utiliza um arquivo que controla quais diretórios o servidor NFS exporta, o "/etc/exports".

## Interoperabilidade (Samba)

Quando falamos em sistemas de arquivos FAT, NTFS, EXT, XFS, entre outros, estamos falando em sistemas de arquivos "locais", os que permitem o acesso a uma mídia interna ou conectada a um computador, por exemplo. Existem também os sistemas de arquivos em rede, que dão a ilusão de um acesso local, porém o acesso é remoto. Os mais conhecidos são o NFS e o SMB/CIFS, os quais veremos a seguir.

O **NFS (Network File System)** é um protocolo de sistema de arquivos distribuído, **originado no Unix**, que permite que um usuário em um computador cliente acesse arquivos através de uma rede como se estivesse acessando na máquina local. O cliente NFS tem a finalidade de tornar o acesso remoto transparente para o usuário do computador, e esta interface cliente e servidor, executada pelo NFS através dos protocolos Cliente-Servidor, fica bem definida quando o usuário, ao chamar um arquivo/diretório no servidor, lhe parece estar acessando localmente, sendo que está trabalhando com arquivos remotamente.

Uma máquina Linux pode ser um servidor NFS e um cliente NFS, o que significa que pode exportar sistemas de arquivo para outros sistemas e montar sistemas de arquivo exportados de outras máquinas. Por exemplo, um grupo de usuários pode ter acesso aos arquivos de um determinado projeto usando um diretório compartilhado do sistema de arquivo NFS montado no diretório "/projeto". Para acessar os arquivos compartilhados, o usuário vai até o diretório "/projeto", sem



senhas ou comandos especiais para lembrar. Dessa forma, os usuários trabalham como se o diretório estivesse em suas máquinas locais.

Algumas **características do NFS** são:

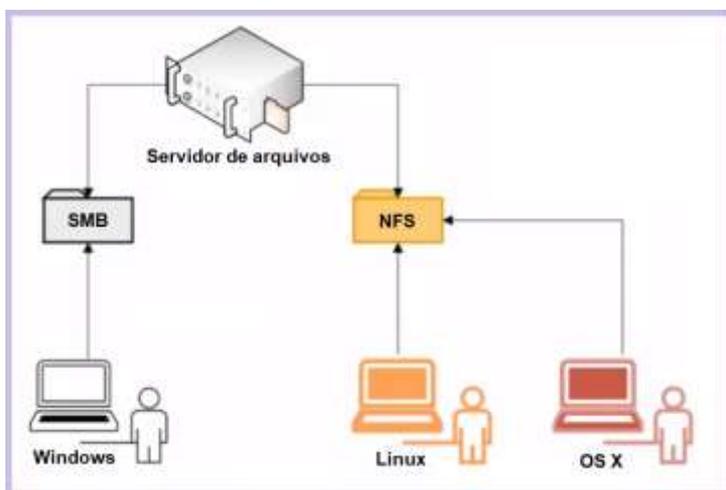
- **Arquitetura cliente-servidor:** o servidor recebe as requisições vindas do cliente, verifica a validade, executa e retorna ao cliente;
- **Stateless (sem estado):** considera cada requisição como uma transação independente que não está relacionada a qualquer requisição anterior, de forma que a comunicação consista em pares requisição/resposta independentes. Diante de uma falha no sistema, quando o servidor NFS volta a funcionar corretamente, o estado anterior é restaurado e o funcionamento continua como se não houvesse ocorrido nenhum problema;
- **Caching:** permite que as informações utilizadas mais recentemente sejam alocadas para posterior uso, ou pode carregar os dados em antecipação a operações futuras;
- **File Locking (bloqueio de arquivo):** permite que um processo tenha acesso exclusivo a um arquivo ou parte deste, e força outros processos que estão solicitando acesso ao mesmo arquivo, aguardarem a liberação;
- **Utiliza o protocolo RPC (Remote Procedure Call):** define um modo independente do sistema para processos se comunicarem através de uma rede de computadores, sendo que o cliente faz a chamada do procedimento remoto no servidor;
- **Protocolo de transporte** padrão é o UDP, no entanto, é possível utilizar o TCP;
- **Compartilhar arquivos** de um servidor NFS é conhecido como exportar os diretórios.

Para que os clientes possam acessar o servidor NFS é necessário que os seguintes *daemons* estejam em execução:

- *nfsd*: *daemon* NFS, atende requisições dos clientes NFS;
- *mountd*: *daemon* de montagem NFS, executa as solicitações passadas pelo *nfsd*;
- *portmap*: *daemon portmapper*, permite que clientes NFS descubram qual porta o servidor NFS está utilizando.

Na figura abaixo podemos ver o acesso a um servidor de arquivos através da rede. Note que máquinas Unix-like (ex.: Linux e OS X) utilizam o NFS como protocolo e as máquinas Windows utilizam o SMB/CIFS, que veremos o funcionamento na sequência.





Antes de começar a falar do protocolo em si, deixo bem claro que as questões de concurso já cobraram alguns detalhes do NFS, mas em relação ao SMB/CIFS a cobrança é superficial. Vamos lá...

O **SMB/CIFS** (*Server Message Block/ Common Internet File System*) é um protocolo de redes cujo uso mais comum como é o compartilhamento de arquivos em uma rede local. Ele permite que o cliente manipule arquivos como se estes estivessem em sua máquina local (mesma coisa que o NFS, heim?). Algumas das operações suportadas são: leitura, escrita, criação, exclusão e renomeação, sendo que os arquivos/diretórios manipulados estão em um servidor remoto.

O protocolo SMB/CIFS funciona através do envio de pacotes do cliente para o servidor. O servidor recebe o pacote, verifica se a requisição é válida (se o cliente possui as permissões apropriadas), executa a requisição e retorna um pacote de resposta ao cliente. O cliente verifica o pacote de resposta para determinar se a requisição inicial obteve êxito. O protocolo mais utilizado para transporte confiável é o NetBIOS sobre TCP (NBT). Outros protocolos foram utilizados na camada de transporte, mas o NBT se tornou o mais utilizado.

O SMB/CIFS é muito utilizado pelos sistemas operacionais Windows, podendo funcionar como um servidor ou cliente. Isso não quer dizer que sistemas Unix-like não possuam também esse protocolo! Na verdade, a maioria dos sistemas Unix-Like (**Linux**, por exemplo) possuem uma implementação de cliente/servidor do SMB/CIFS via **Samba**, que é um "software servidor" para Linux que permite o gerenciamento e compartilhamento de recursos em redes formadas por computadores com o Windows. Dessa forma é possível utilizar o Linux como servidor de arquivos, servidor de impressão, entre outros, como se a rede utilizasse servidores Windows (Server 200x, 201x, entre outros).





Resumindo: o Samba é um componente que realiza a comunicação entre servidores Linux e Windows, permitindo que eles compartilhem recursos de disco e de impressão. O *daemon* `smbd` possui dois modos de compartilhamento. O primeiro modo (*sharemode*) é mais simples, gerando uma senha para cada recurso que é compartilhado. O segundo modo (*user*) permite o compartilhamento de vários recursos com um único *login* e senha. Os privilégios de acesso aos recursos devem ser sempre mediados pelos administradores do sistema.

Um outro componente interessante do pacote Samba é o *nmbd*, um servidor de nomes NetBios (mais conhecido por **WINS - Windows Internet Name Server**) responsável por entender e responder solicitações de resoluções de nomes NetBios sobre IP. É ele o responsável pelo "aparecimento" do ícone do servidor Samba no ambiente de rede do Windows! Obs.: Se você não sabe o que é essa resolução de nomes, pense que é algo como o DNS, mas não de forma hierárquica, é uma resolução feita pelo Windows (sabendo isso já está bom para a prova 😊).

## Serviço de Diretório

Um serviço de diretório é um sistema de software que armazena, organiza e fornece acesso a informações em um diretório do sistema operacional. Diretórios podem ser muito limitados em escopo, suportando apenas um pequeno conjunto de tipos de nós e tipos de dados, ou eles podem ser muito amplos, suportando um conjunto de tipos arbitrário ou extensível. No DNS os nós são nomes de domínio e os itens de dados são endereços IP (e apelidos, nomes de servidor de e-mail etc.).

Em um diretório utilizado por um sistema operacional de rede, os nós representam **recursos** que são gerenciados pelo sistema operacional, incluindo usuários, computadores, impressoras, entre outros recursos compartilhados. Muitos serviços de diretório têm sido utilizados desde o aparecimento da Internet, mas o mais utilizado é um descendente do serviço de diretório X.500.

Antes de vermos uma implementação de serviço de diretório para o Linux (OpenLDAP), é importante entendermos o protocolo que fornece mecanismos de acesso aos objetos, o **LDAP (Lightweight Directory Access Protocol)**. Entidades internacionais (ITU, ISO, IETF, entre outras) trabalham na definição de padrões diversos, incluindo a padronização que dá suporte a serviços



de diretórios. Um padrão de uso genérico é o **X.500** (da ISO) que possui uma grande abrangência, mas é muito complexo e não foi adotado em sua íntegra como um padrão de mercado. Um padrão mais "light" que de fato se tornou um padrão de mercado foi o LDAP.

O padrão LDAP define um sistema de nomeação hierárquico, no qual é possível referenciar qualquer objeto que esteja no AD. Um nome LDAP é composto pelo caminho completo do objeto (ex.: uma impressora, um computador etc.), partindo do domínio raiz até chegar ao objeto em si. Algumas abreviaturas (**atributos**) são utilizadas nessa nomenclatura hierárquica:

- **cn**: *common name* (nome da conta de um usuário, grupo etc.);
- **sn**: sobrenome (*surname*);
- **ou**: faz referência a uma unidade organizacional;
- **dc**: componente de domínio (normalmente o nome do domínio);
- **o**: nome da organização (geralmente o domínio raiz);
- **c**: *country* - país (normalmente não utilizado).

Vamos a um exemplo de um nome LDAP:

CN=evandrodv, OU=professores, DC=ti, DC=estrategiaconcursos.com.br → esse nome representa o usuário "evandrodv", cuja conta está contida na unidade organizacional "professores", no domínio "ti.estrategiaconcursos.com.br". Obs.: os dois componentes de domínio foram concatenados.

Por padrão um servidor LDAP "**escuta**" na porta **389 (TCP)** e as principais características do protocolo são:

- Baseado em padrão aberto: qualquer desenvolvedor pode acessar sua especificação e realizar a implementação;
- Possui APIs bem definidas: facilita a vida dos programadores;
- Maior velocidade de consulta que um BD relacional;
- Replicável e distribuível;
- Facilita a localização de informações e recursos: pesquisa feita nome.

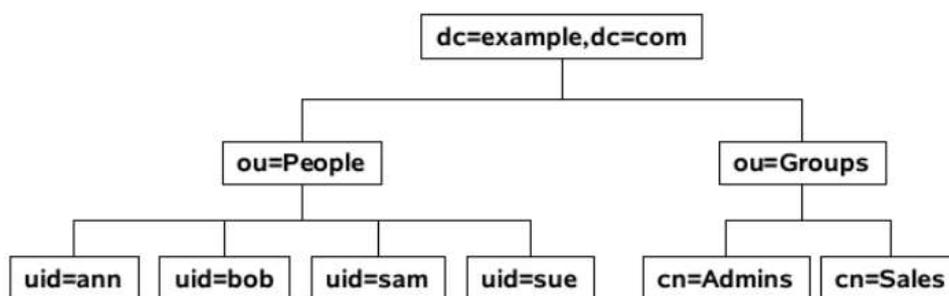
Algumas **operações (comandos)** que podem ser utilizados através do LDAP são:

- **Bind**: autentica e especifica a versão do protocolo LDAP;
- **Search**: procura/recupera entradas dos diretórios;
- **Compare**: compara uma entrada com determinado valor;
- **Add**: adiciona uma nova entrada;
- **Delete**: exclui uma entrada;
- **Modify**: modifica uma entrada;



- **Modify DN**: move ou renomeia uma entrada;
- **Abandon**: aborta uma requisição prévia;
- **Unbind**: fecha a conexão;
- **Extended Operation**: operação genérica para definir outras operações;
- **StartTLS**: protege a conexão com o TLS - implementada a partir da versão 3 do LDAP.

A representação dos dados é realizada através de uma estrutura hierárquica na forma de árvore (*Directory Information Tree - DIT*), a qual consiste em entradas de DN's (*Distinguished Names*). O LDAP utiliza a DIT como estrutura de dados fundamental:



Como podemos ver, a árvore de diretório possui uma forma hierárquica:

- Primeiro o diretório raiz;
- Após a rede corporativa, os departamentos e por fim os computadores dos usuários e os recursos de rede.

Alguns conceitos que também já foram cobrados em concursos são mostrados na sequência.

**Schema**: conjunto de objetos e atributos para o armazenamento. É modelado de acordo com o padrão X.500 da ISO.

Cada entrada (objeto) possui um identificador único (**dn - distinguished name**), o qual consiste de seu Relative Distinguished Name (RDN), construído de algum(ns) atributo(s) na entrada, seguido pelo DN da entrada pai.

**Escalabilidade**: é possível replicar servidores LDAP e incluir novos servidores à medida que aumenta a estrutura da organização. Ou seja, não é uma estrutura "engessada".



## Schema

- conjunto de objetos e atributos para o armazenamento

## Escalabilidade

- é possível replicar servidores LDAP e incluir novos servidores à medida que aumenta a estrutura da organização

Como já mencionado anteriormente, uma solução bastante conhecida para Linux é o **OpenLDAP** (que também possui versões para Windows, Solaris, entre outros sistemas operacionais). Vamos ver como instalar o OpenLDAP:

```
sudo apt install slapd ldap-utils
```

O principal arquivo de configuração (pelo menos para concursos) é o **"/etc/ldap.conf"**, que é usado para configurar os padrões a serem aplicados quando da execução dos clientes LDAP.



1. (CESPE/FUB - 2018) O Linux é um sistema operacional em que cada usuário consegue ter apenas um processo ativo por vez, processo esse que é iniciado automaticamente quando o sistema é carregado.

### Comentários:

O Linux permite que vários processos sejam executados ao mesmo tempo (multitarefa), mesmo que haja apenas um processador na máquina. O sistema operacional faz o escalonamento dos processos no processador, dando a ilusão de um paralelismo (ocorre um pseudoparalelismo). O comando "ps", por exemplo, pode ser executado e você verá diversos processos em execução. Quando há a criação de um processo novo, ele passa a ser filho de outro e a chamada de sistema utilizada é a fork(). Portanto, a questão está **errada**.

Gabarito: Errada

2. (FGV/AL-RO - 2018) Wallace é administrador de um servidor com sistema operacional CentOS e deseja compartilhar um diretório com os integrantes da empresa por meio do protocolo NFS. Para que a configuração desse compartilhamento possa ser efetuada, Wallace deve editar o arquivo /etc/



- A) services
- B) modules
- C) hosts
- D) export
- E) fstab

**Comentários:**

O Linux geralmente é intuitivo, então digamos que você tenha esquecido da aula, mas lembra que o NFS é um sistema de arquivos de rede e os diretórios a serem "exportados" devem estar listados no arquivo "export". Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D



## SHELL SCRIPT

Linguagens de script são linguagens de programação executadas do interior de programas, a partir de outras linguagens de programação, entre outras situações. Elas servem para estender a funcionalidade de um programa e/ou controlá-lo, acessando sua API e, são frequentemente usadas como ferramentas de **configuração e instalação** em sistemas operacionais, como por exemplo, em alguns sistemas operacionais da família Linux.

O Unix Shell é ao mesmo tempo um interpretador de comandos e uma linguagem de programação. Como interpretador de comandos, ele dá acesso ao rico conjunto de utilidades do GNU e como linguagem de programação ele permite que tais utilidades sejam combinadas. Arquivos contendo comandos podem ser criados e se tornar comandos (assim como os arquivos em lote no Windows - .BAT). Esses novos comandos têm o mesmo status de comandos de sistema como os do diretório /bin. Importante destacar que Shell é uma linguagem interpretada, ou seja, os comandos são interpretados na sequência, um a um, sem haver uma compilação.

### Shell

- é uma linguagem interpretada, ou seja, os comandos são interpretados na sequência, um a um, sem haver uma compilação

Na sequência vamos ver um passo a passo de como criar um shell script.

#### Criando um arquivo:

Há dois modos de realizar essa ação: via modo gráfico ou via terminal. No segundo caso, pode ser utilizado o comando vi, conforme mostrado a seguir (note a "extensão" .sh).

vi script.sh → Será criado e aberto um arquivo de leitura e escrita.

Outra opção é digitar o comando touch:

touch script.sh → cria um arquivo sem abri-lo.

#### Permissão ao arquivo:

Para começar a editar o arquivo, é necessário conceder a permissão de escrita a ele. Uma opção é utilizar o comando:

chmod 777 script.sh → o comando chmod é utilizado para conceder permissões em diretórios e arquivos, enquanto o valor 777 permite que o usuário tenha total liberdade



para editar o arquivo (na verdade dá liberdade total a qualquer usuário do sistema! Mas nosso foco agora não é estudar o chmod).

Pensando mais em segurança, pode-se, por exemplo, liberar apenas a execução (x):

`chmod +x script.sh` → como não foi especificado para quem foi liberada a execução (usuário, grupo ou todos), todos devem ter a permissão de execução.

### Editar e executar o arquivo:

Depois da permissão do arquivo, vamos para a edição do arquivo com o comando `vi`:

`vi script.sh` → é necessário digitar "i" para colocar o vi no modo "inserção".

Agora é necessário **definir** qual **Shell** será utilizado na **primeira linha** do arquivo (vamos escolher o bash, mas poderia ser outro interpretador de comandos, como sh, ksh ou csh):

```
#!/bin/bash
```

### Inserir comentários:

Basta utilizar o caractere "#" para inserir o comentário, como mostrado abaixo:

```
# A linha abaixo serve para bla bla bla
```

### Uso das variáveis:

Para declarar a variável, a sintaxe é a seguinte:

`nome_variavel = valor` → o valor pode ser string ou número. Para utilizar o valor, devemos colocar o símbolo "\$" na frente do nome que será inserido.

### Entrada de dados do usuário:

Se o script pedir ao usuário o fornecimento de dados para a entrada do processamento, o programador precisará digitar um comando ler esses dados:

```
read nome_variavel
```

### Comando de seleção:

Um comando simples que permite a execução de uma tarefa baseada na ação de um usuário que esteja usando o seu sistema é o condicional. A sintaxe é a seguinte:



```
if [ CONDIÇÃO ] ;  
  
then  
  
AÇÕES  
  
fi
```

É importante lembrar que para cada "if" aberto, deve haver um "fi" que feche a sequência. A mesma regra serve para os colchetes.

### Funções:

Para organizar, separar e estruturar um algoritmo, é necessário o uso de funções:

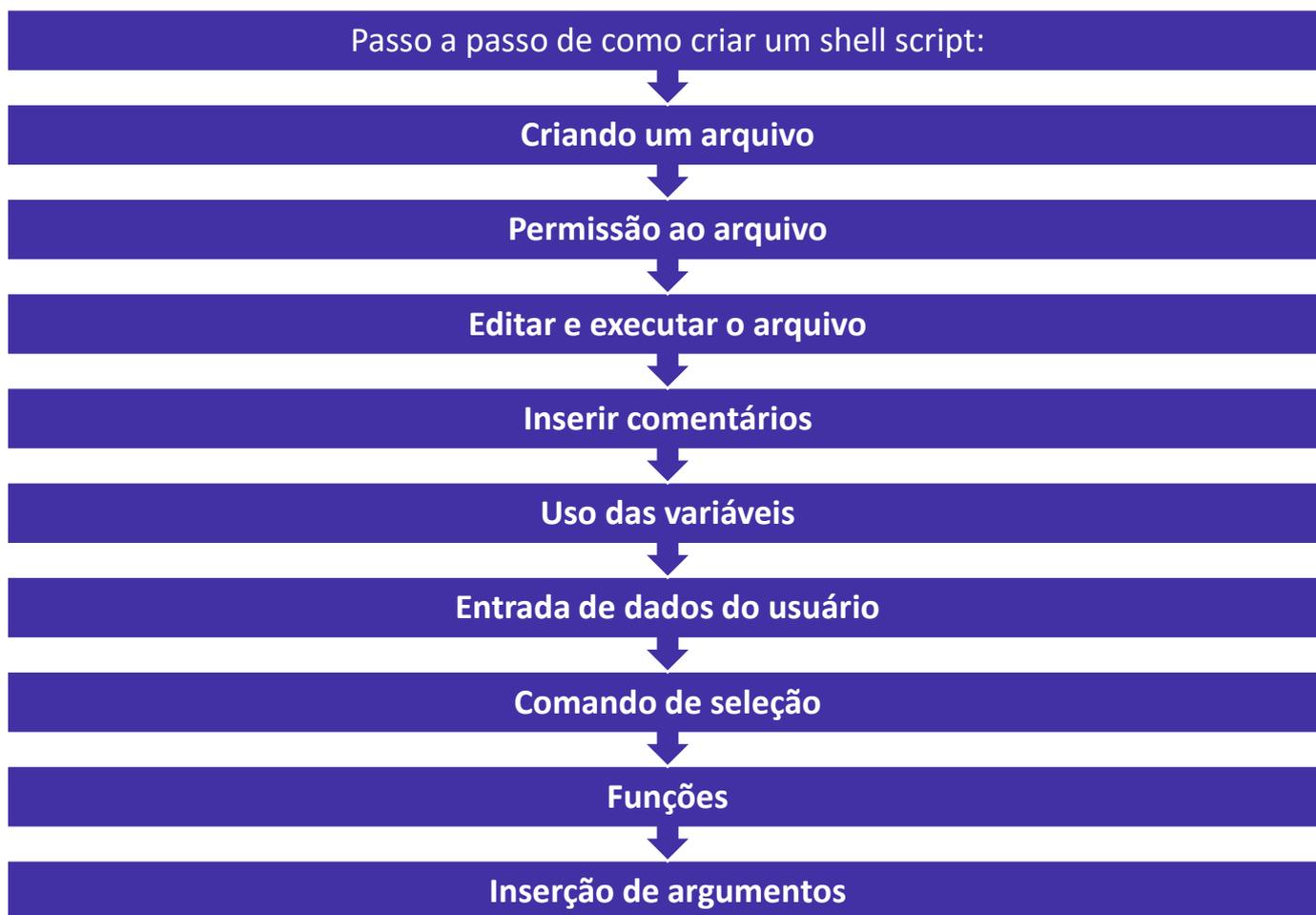
```
nome_funcao()  
  
{  
  
AÇÕES  
  
}
```

### Inserção de argumentos:

O shell script recebe dados fornecidos por outro programa ou por um usuário com a finalidade de produzir saídas, são os **argumentos**. Alguns nomes que qualificam os argumentos são:

- \$# - total de argumentos que foram passados;
- \$\* - retorno aos argumentos;
- \$0 - nome do script executado.





Um **exemplo** simples é mostrado abaixo (script.sh), sendo o script à esquerda e a saída de sua execução à direita. Note que na execução do script são passados 3 parâmetros (a, b, c) e na chamada da função "imprime()" são passados 4 parâmetros (um, dois, tres, quatro).

<pre>#!/bin/sh imprime () { echo "Sou o programa \$0" echo "Recebi \$# parametros" echo "Param 1: \$1" echo "Param 2: \$2" echo "Lista de parâmetros: \$*" } imprime um dois tres quatro echo "Sou o programa \$0" echo "Recebi \$# parametros" echo \$1 \$2 \$3</pre>	<pre>\$ ./script.sh a b c Sou o programa teste.sh Recebi 4 parametros Param 1: um Param 2: dois Lista de parâmetros: um dois tres quatro Sou o programa script.sh Recebi 3 parametros a b c</pre>
--	---



Vale lembrar que para um bom entendimento de um shell script, é importante conhecer o funcionamento dos comandos, e não são poucos! Mas, por enquanto, as questões sobre esse assunto não pegaram pesado.



1. (INSTITUTO AOCP/EBSERH - 2016) Em sistemas operacionais Linux é possível criar scripts para automatizar tarefas rotineiras. A extensão de arquivo utilizada para Shell Script é

- A) .bin
- B) .sl
- C) .bash
- D) .sh
- E) .gz

**Comentários:**

Vimos em aula o "script.sh", então o padrão utilizado como "extensão" é .sh. Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

2. (CESGRANRIO/Petrobras - 2012) No ambiente UNIX, existem vários interpretadores de linha de comando conhecidos como shell. É importante, para cada script, informar em que shell ele deve ser executado. Para isso, o usuário pode especificar o shell desejado

- A) na primeira linha do script.
- B) na última linha do script.
- C) em qualquer linha do script.
- D) em um arquivo à parte.



E) na linha de comando, após o nome do arquivo que contém o script.

**Comentários:**

É necessário definir qual Shell será utilizado na primeira linha do arquivo (vamos escolher o bash, mas poderia ser outro interpretador de comandos, como sh, ksh ou csh):

```
#!/bin/bash
```

Portanto, a **alternativa A** está correta e é o gabarito da questão.

**Gabarito:** Letra A



## QUESTÕES COMENTADAS - CONFIGURAÇÃO DE SERVIÇOS OU FUNCIONALIDADES - MULTIBANCAS

1. (AOCP/EBSERH - 2016) O NFS (Network File System) permite compartilhar sistemas de arquivos entre computadores conectados em rede e pode ser parte fundamental da infraestrutura da tecnologia da informação. Sobre NFS, analise as assertivas a seguir e assinale a alternativa que aponta a(s) correta(s).
- I. O NFS é considerado sem estado (stateless) e, portanto, quando um servidor NFS volta a funcionar, o estado anterior é restaurado e a informação não é perdida.
  - II. O NFS pode ser implementado do lado servidor e do lado cliente.
  - III. O NFS roda sobre o protocolo RPC (Remote Procedure Call), que define um modo independente do sistema para processos se comunicarem através de uma rede de computadores.
  - IV. O NFS suporta apenas UDP como protocolo de transporte, pois ele apresenta desempenho significativamente melhor que o TCP em redes locais.
- A) Apenas I e II.  
B) Apenas II e IV.  
C) Apenas III.  
D) Apenas III e I.  
E) Apenas I, II e III.

### Comentários:

(I) Exato! O NFS é stateless, cada requisição é uma transação independente que não está relacionada a qualquer requisição anterior. Então, quando um servidor NFS volta a funcionar, o estado anterior é restaurado e a informação não é perdida. (II) Existe quem faz requisições de acesso e manipulação de arquivos e diretórios (cliente) e quem recebe as requisições, avalia as permissões, executa e retorna (servidor). (III) O NFS utiliza o RPC (Remote Procedure Call), o cliente faz a chamada do servidor remoto (o servidor até pode estar na mesma máquina, mas o cenário mais comum é remoto). (IV) O protocolo padrão de transporte é o UDP, mas é possível optar pelo TCP. Portanto, a **alternativa E** está correta e é o gabarito da questão.

**Gabarito:** Letra E



2. (FGV/IBGE - 2017) Em ambiente Linux, a chamada ao sistema (system call) que implementa a criação de um novo processo é:

- A) create\_process;
- B) new\_process;
- C) fork;
- D) spawn;
- E) duplicate.

### Comentários:

No Linux a chamada de sistema fork() faz uma "bifurcação" de um processo, sendo que o novo vira o filho do outro. Ex.: em um shell você executa algum comando, então o fork() será chamado, o shell será o processo pai do processo novo (o comando digitado). Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

3. (FCC/TRF5 - 2017) Considere a figura abaixo que mostra a arquitetura do sistema operacional Linux



### A caixa

A) I representa a camada responsável pela interface entre o hardware e as aplicações. Dentre suas funções encontram-se gerenciamento de I/O, manutenção do sistema de arquivos, gerenciamento de memória e swapping, controle da fila de processos etc.

B) II representa a camada que permite o acesso a recursos através da execução de chamadas feitas por processos. Tais chamadas são geradas por funções padrão suportadas pelo kernel. Dentre suas funções estão habilitar funções padrão como open, read, write e close e manter a comunicação entre as aplicações e o kernel.



C) I é um processo que executa funções de leitura de comandos de entrada de um terminal, interpreta-os e gera novos processos, sempre que requisitados. É conhecido também como interpretador de comandos.

D) II é um processo que realiza modificações no shell, permitindo que funcionalidades do Linux sejam habilitadas ou desabilitadas, conforme a necessidade. Tal processo gera ganho de performance, pois à medida que customiza o shell, o usuário torna o Linux enxuto e adaptável.

E) I é um processo que realiza modificações no kernel, permitindo que funcionalidades do Linux sejam habilitadas ou desabilitadas, conforme a necessidade. Tal processo gera ganho de performance, pois à medida que customiza o kernel, o usuário torna o Linux enxuto e adaptável.

#### Comentários:

A caixa I pode ser um shell que recebe comandos do teclado, interpreta e executa, gerando novos processos (filhos) através da chamada de sistema fork(). Ex.: execução do comando "ls" no shell. Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

4. (CS-UFG/DEMAE - 2017) Na maior parte das distribuições Linux, a memória virtual é uma partição denominada

A) EXT

B) SWAP

C) SETUP

D) BIOS

#### Comentários:

Vimos que pode ser partição ou arquivo (ou até ambos), mas geralmente se utiliza uma partição de SWAP. Portanto, a **alternativa B** está correta e é o gabarito da questão.

**Gabarito:** Letra B

5. (Quadrix/COFECI - 2017) Não é possível realizar uma instalação automática, por meio de uma rede, no sistema operacional Linux.

#### Comentários:



Pode-se realizar uma instalação através de uma mídia (CD, DVD, pen drive) ou através da rede, desde que a placa de rede permita e a configuração esteja adequada para um boot pela rede. Portanto, a questão está **errada**.

Gabarito: Errada

6. (Colégio Pedro II/Colégio Pedro II - 2017) Sistema operacional é um conjunto de programas básicos e utilitários que fazem seu computador funcionar. O Debian é um sistema operacional, em cujo núcleo está o Kernel, o programa mais fundamental no computador e que faz todas as operações mais básicas, permitindo que você execute os outros programas. O Debian atualmente usa o Kernel Linux. Mas um sistema operacional não funciona somente com o Kernel, são necessários utilitários e aplicativos. O Debian utiliza estas ferramentas do projeto GNU. Por esse motivo, muitos utilizadores defendem que devemos chamar o sistema de "Debian GNU/Linux". Reconheça (1) o gerenciador de pacotes (programas) usado no Debian e em distribuições derivadas do Debian (Ubuntu, Knoppix, Big Linux etc.) que utiliza uma lista de dependências para instalar tudo o mais automaticamente possível, e (2) o arquivo de configuração utilizado por este gerenciador de pacote.

A) Gerenciador (1) - apt-get

Arquivo (2) - source.list

B) Gerenciador (1) - apt-get

Arquivo (2) - package.cfg

C) Gerenciador (1) - make install

Arquivo (2) - package.cfg

D) Gerenciador (1) - dpkg

Arquivo (2) - source.list

### Comentários:

Das alternativas mostradas, apenas "apt-get" é um gerenciador que utiliza uma lista de dependências e o nome dela é "sources.list". Nas alternativas aparece com um "s" a menos, mas tudo bem, relaxe...

Portanto, a **alternativa A** está correta e é o gabarito da questão.

Gabarito: Letra A



7. (CESPE/EMAP - 2018) O kernel do sistema operacional Linux tem a função de interpretar os comandos executados em um terminal.

**Comentários:**

Quem tem essa função é o shell! O kernel é o núcleo do sistema operacional! Portanto, a questão está **errada**.

**Gabarito:** Errada

8. (FGV/MPE-AL - 2018) Instalar, atualizar e remover pacotes no sistema operacional CentOS 7 é uma tarefa frequente para desenvolvedores de sistemas. Por isso, eventualmente podem ocorrer dúvidas sobre se determinado pacote está instalado ou qual é a versão que está sendo utilizada. Para dirimir essas dúvidas, sobre o pacote httpd devemos utilizar o comando

A) rpm -q httpd

B) yum check httpd

C) apt-get find httpd

D) find -iname httpd

E) crontab -l httpd

**Comentários:**

Para quem não lembra da parte da aula sobre o gerenciador utilizado no CentOS, pelo menos deve lembrar que a resposta só pode ser uma das 3 primeiras, certo? Senão, a coisa está feia.

Se também lembrar que "-q" é em relação a "query" (pesquisa), aí fica mais tranquilo para marcar a **alternativa A**, que é o gabarito da questão.

**Gabarito:** Letra A

9. (FGV/COMPESA - 2018) Com relação às características e tarefas de administração do sistema operacional Linux, analise as afirmativas a seguir.

I. O esquema de partição GPT (Guid Partition Table) oferece a possibilidade de até 128 partições primárias, com tamanhos maiores do que 2TB cada.

II. Ext4, XFS e exFAT são exemplos de sistemas de arquivo com jornal, utilizados pelo Linux.

III. A área de swap no Linux pode ser uma partição dedicada de swap, um arquivo de swap ou uma combinação de arquivos e partições de swap.



Está correto o que se afirma em

- A) I, somente.
- B) II, somente.
- C) III, somente.
- D) I e III, somente.
- E) I, II e III.

#### Comentários:

(I) Ao contrário da MBR, que oferece 4, a GPT realmente permite até 128, está correta! (II) exFAT é um sistema de arquivos do Windows, não do Linux! (III) Embora muitos pensem que tem que ser uma partição, pode ser um arquivo de swap ou ainda uma combinação. Portanto, a **alternativa D** está correta e é o gabarito da questão.

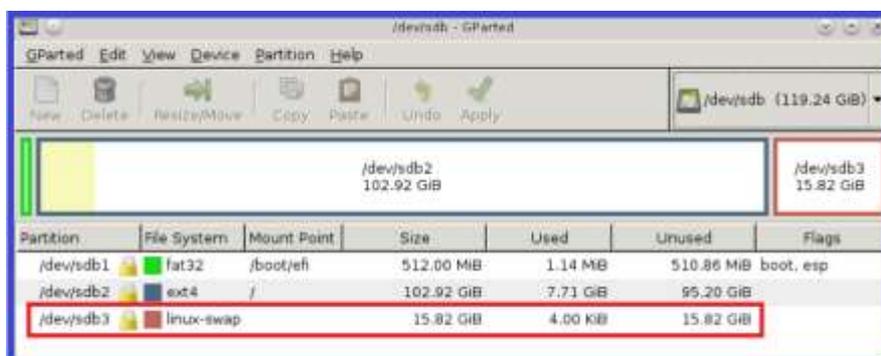
Gabarito: Letra D

10.(CS-UFG/SANEAGO-GO - 2018) Um utilitário GNU, presente em diversos sistemas operacionais linux, que lista as partições dos discos rígidos, é:

- A) fdisk.
- B) mcdisk.
- C) parted.
- D) gedit.

#### Comentários:

Na verdade, é GParted, mas tudo bem...releve uma letrinha 😊.



Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**11.(IF-RS/IF-RS - 2018) São tipos de sistemas de arquivos utilizados no Linux:**

- A) FAT 16, FAT 32, NTFS
- B) FAT 32, HFS, ExFAT
- C) Ext2, Ext3, Btrfs
- D) Ext2, Ext3, HFS
- E) Ext2, Ext3, NFS, SMB

**Comentários:**

Família FAT e NTFS= Windows.

HFS = Apple.

NFS e SMB = Sistemas de arquivos de rede.

Só sobrou a letra C! Lembrando: família EXT, Btrfs, XFS, JFS, ReiserFS, entre outros = Linux!

Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**12.(COPESE-UFT/UFT - 2018) Embora seja possível realizar boot de um sistema Linux a partir de um pendrive, a maioria das instalações do Linux o realiza a partir do disco rígido do computador. Esse processo consiste em duas fases básicas:**

1. Executar o carregador de boot a partir do dispositivo de boot;
2. Iniciar o kernel do Linux e iniciar os processos.

**Assinale a alternativa que contém um gerenciador de boot para sistemas Linux:**

- A) CISC (Complex Instruction Set Computer).
- B) GRUB (Grand Unified Bootloader).
- C) ORM (Object Relational Mapping).



D) CMS (Content Management System).

#### Comentários:

O LILO já foi bastante utilizado (mas nem aparece como opção) e o GRUB é mais utilizado ultimamente. Portanto, a **alternativa B** está correta e é o gabarito da questão.

**Gabarito:** Letra B

**13.(COPESE-UFT/UFT - 2018) Sistemas operacionais Linux permitem logins simultâneos de diferentes usuários. Assinale a alternativa que contém o comando a ser utilizado para visualizar os usuários logados em um determinado instante.**

A) listusers

B) ldconfig

C) who

D) uname

#### Comentários:

O Linux é um sistema operacional multiusuário, ou seja, pode ter N usuários logados ao mesmo tempo. Para saber QUEM está logado pode ser utilizado o comando "who". Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**14.(IF-RS/IF-RS - 2018) Considerando que se está logado como "root" no terminal de uma estação de trabalho com um sistema operacional Linux debian ou derivado, qual comando configura o DNS para 8.8.8.8?**

A) dns 8.8.8.8

B) echo 8.8.8.8 > /etc/hosts.conf

C) echo nameserver 8.8.8.8 > /etc/resolv.conf

D) nslookup 8.8.8.8

E) nameserver 8.8.8.8

#### Comentários:



Lembrando...o DNS resolve nomes, então o arquivo de configuração é o "resolv.conf" e sabemos que os arquivos de configuração por padrão ficam em "/etc". Só tem uma opção...a letra C.

Mas, continuando...o comando echo "escreve", então o comando completo "escreveu" "nameserver 8.8.8.8" no arquivo "/etc/resolv.conf", ou seja, configurou o serviço de DNS.

Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**15.(COPEVE-UFAL/UFAL - 2018) O assistente de tecnologia da informação precisa atualizar o sistema operacional GNU/Linux do servidor WEB de uma empresa. Qual comando, usando o modo root, deve ser utilizado?**

- A) upgrade distro-all.
- B) upgrade apt-distro.
- C) apt-get dist-upgrade.
- D) apt-get purge "nome da distro".
- E) apt-cache showpkg "nome da distro".

#### **Comentários:**

Sabemos que o gerenciador de pacotes é o APT, já descartamos as duas primeiras. Mesmo sem lembrar da aula, dá para ver que pela lógica a letra C indica um upgrade da distribuição Linux. Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**16.(COPEVE-UFAL/UFAL - 2018) Dadas as afirmativas quanto aos conceitos de Kernel no sistema operacional GNU/Linux,**

- I. Compilar o Kernel permite ao usuário remover drivers inúteis diminuindo o tempo de arranque do sistema operacional.
- II. Os módulos do Kernel do sistema estão armazenados no diretório /boot/.
- III. O comando uname -a exibe as informações do kernel do sistema.
- IV. Módulos são as partes do kernel que são carregadas somente quando são requisitadas por um aplicativo ou dispositivo.

Verifica-se que está(ão) correta(s)



- A) II, apenas.
- B) I e II, apenas.
- C) III e IV, apenas.
- D) I, III e IV, apenas.
- E) I, II, III e IV.

### Comentários:

(I) Compilar o kernel permite eliminar algo inútil ou acrescentar algo útil (ex.: um driver de uma placa de rede que você possui e não tinha antes). (II) Não! Ficam em "/lib/modules/versão\_do\_kernel". (III) Exato! Conforme tela abaixo, com destaque na versão do kernel. (IV) Perfeito! Assim não carrega o que não precisa para a memória!

```
[root@localhost lib]# uname -a  
Linux localhost 4.12.0-rc6-g48ec1f0-dirty #21 Fri Aug 4 21:02:28 CEST 2017 i586  
GNU/Linux  
[root@localhost lib]#
```

Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

**17.(FGV/AL-RO - 2018)** Wallace é administrador de um servidor com sistema operacional CentOS e deseja compartilhar um diretório com os integrantes da empresa por meio do protocolo NFS. Para que a configuração desse compartilhamento possa ser efetuada, Wallace deve editar o arquivo /etc/

- A) services
- B) modules
- C) hosts
- D) export
- E) fstab

### Comentários:



O Linux geralmente é intuitivo, então digamos que você tenha esquecido da aula, mas lembra que o NFS é um sistema de arquivos de rede e os diretórios a serem "exportados" devem estar listados no arquivo "export". Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

**18.(COPEVE-UFAL/UFAL - 2018) Analise a configuração de rede no GNU/Linux.**

```
auto lo
```

```
iface lo inet loopback
```

```
auto eth0
```

```
allow-hotplug eth0
```

```
iface eth0 inet static address 192.168.100.1 netmask 255.255.255.0 network 192.168.100.0  
broadcast 192.168.100.255
```

Dadas as afirmativas quanto à configuração de rede,

- I. O "iface eth0 inet static" informa ao sistema que a placa de rede terá o endereço IP estático.
- II. O "iface lo" informa ao sistema que a placa de rede receberá um endereço IP via servidor DHCP.
- III. Uma segunda placa de rede com fio instalada nesse computador irá receber a denominação eth1.
- IV. O "allow-hotplug" permite conectar dispositivos hotplug no computador.

Verifica-se que está(ão) correta(s)

- A) II, apenas.
- B) I e III, apenas.
- C) II e IV, apenas.
- D) I, III e IV, apenas.
- E) I, II, III e IV.

**Comentários:**

(I) O "iface eth0 inet static" informa ao sistema que a placa de rede terá o endereço IP estático, ou seja, não receberá um endereço através do protocolo DHCP. (II) O "iface lo" indica a interface de loopback (aquela para testes, geralmente o endereço IP é 127.0.0.1). (III) Uma segunda placa de rede com fio instalada nesse computador irá receber a denominação eth1, pois eth indica



"ethernet" e já existe a eth0. (IV) O "allow-hotplug" permite conectar dispositivos hotplug no computador ("allow" = permite, então ficou tranquila essa). Portanto, a **alternativa D** está correta e é o gabarito da questão.

Gabarito: Letra D

**19.(FGV/AL-RO - 2018)** Roberto trabalha como administrador de redes em uma empresa de cosméticos e sua principal função é a configuração de estações de trabalho dos seus colaboradores. Rotineiramente, Roberto instala e remove pacotes nas estações de trabalho às quais ele dá suporte.

Assinale a opção que indica o comando utilizado por Roberto para desinstalar pacotes RPM nas estações de trabalho com sistema operacional Linux CentOS.

- A) rpm -Uvh pacote
- B) rpm -e pacote
- C) rpm -ivh pacote
- D) rpm -qi pacote
- E) rpm -qa | grep pacote

#### Comentários:

Algumas opções do rpm:

-i: instalação simples.

-v: exibe detalhes da instalação.

-h: mostra o caractere "#", enquanto o programa é instalado.

-U: atualização do programa de uma versão anterior para uma mais recente.

--nodeps: não procura dependências.

--force: força a instalação.

--root diretório: utiliza o sistema com raiz em diretório para todas as operações.

-e: desinstala o pacote.

--import: importa a chave pública GPG do distribuidor de um pacote.



--help: exibe uma mensagem de ajuda.

-q: consulta (query) se um pacote já está instalado.

Portanto, a **alternativa B** está correta e é o gabarito da questão.

**Gabarito:** Letra B

**20.(Quadrix/CFBio - 2018) Carregamento e inicialização do kernel e execução dos scripts de inicialização do sistema são algumas das etapas do processo de inicialização de um Linux típico.**

**Comentários:**

Questão redonda, pois o núcleo do S.O. (Linux) deve ser carregado, depois os scripts de inicialização definidos pelo usuário (ou om a configuração padrão mesmo). Portanto, a questão está **correta**.

**Gabarito:** Correta

**21.(Quadrix/CRM-PR - 2018) Um computador pessoal com sistema operacional Linux, versão desktop, pode ter no máximo cinco usuários simultâneos.**

**Comentários:**

Só teria o limite de 5 se fosse configurado previamente, mas por padrão não tem limite - até estourar a memória! 😊. Portanto, a questão está **errada**.

**Gabarito:** Errada

**22.(FADESP/IF-PA - 2018) Sobre os pacotes do Sistema Operacional (SO) Debian 9.5, é correto afirmar que**

- A) o apt é um programa que pode substituir o dpkg para o gerenciamento de pacotes do SO.
- B) os pacotes devem ser distribuídos no formato .tar.gz.
- C) o dpkg-source compila e executa código fonte de um pacote.
- D) um pacote marcado como Essential não pode ser desinstalado do Sistema Operacional.
- E) cada pacote deve especificar as dependências sobre outros pacotes para seu funcionamento.

**Comentários:**



É comum que para uma aplicação funcione ela precise de alguns pacotes, porque alguns dependem de outros. Os gerenciadores de pacotes ajudam a verificar essas dependências. Portanto, a **alternativa E** está correta e é o gabarito da questão.

**Gabarito:** Letra E

**23.(CONSULPAM/Câmara de Juiz de Fora-MG - 2018) Em relação ao Sistema Operacional Linux, marque o item INCORRETO:**

- A) Sua arquitetura é composta por um núcleo monolítico cujas funções são: gerenciar a memória, operar as entradas e saídas e o acesso aos arquivos.
- B) Outra característica do Linux é com relação aos drivers de dispositivos e suporte a rede, os quais podem ser compactadas e utilizadas como se fossem módulos ou bibliotecas (LKM em inglês Loadable Kernel Modules), separados pela parte principal, cujo carregamento pode ser ativado após a execução do núcleo.
- C) No quesito portabilidade, o Linux funciona com eficiência em plataformas como x64 da Intel (EM64T e AMD64) PowerPC, Alpha, SPARC, porém é de difícil instalação nos sistemas embarcados como PVR, celulares, Tv's e Handhelds.
- D) A partir da década de 90, ao passo que a distribuição do Linux se popularizou, foi também limitada, pois se torna uma alternativa no uso de software livre, contra os sistemas operacionais da Apple (Mac OS) e Microsoft (Windows).

#### Comentários:

Em relação à portabilidade, o Linux se sai muito bem! Ele não foi pensado para ser um sistema portátil, mas acabou indo nessa direção, tendo um dos núcleos de sistemas operacionais mais portáteis, sendo executado em sistemas desde o iPaq (um computador portátil) até o IBM S/390 (um mainframe). Na atualidade o Linux hoje funciona em dezenas de plataformas, desde mainframes até um relógio de pulso, passando por diversas arquiteturas: x86 (Intel, AMD), x86-64 (Intel EM64T, AMD64), ARM, PowerPC, Alpha, SPARC, entre outras, além de sistemas embarcados (handhelds, consoles de videogames, PVR – Personal Video Recorder, telefones celulares, TVs etc.). Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**24.(CESPE/FUB - 2018) O Linux é um sistema operacional em que cada usuário consegue ter apenas um processo ativo por vez, processo esse que é iniciado automaticamente quando o sistema é carregado.**

#### Comentários:



O Linux permite que vários processos sejam executados ao mesmo tempo (multitarefa), mesmo que haja apenas um processador na máquina. O sistema operacional faz o escalonamento dos processos no processador, dando a ilusão de um paralelismo (ocorre um pseudoparalelismo). O comando "ps", por exemplo, pode ser executado e você verá diversos processos em execução. Quando há a criação de um processo novo, ele passa a ser filho de outro e a chamada de sistema utilizada é a fork(). Portanto, a questão está **errada**.

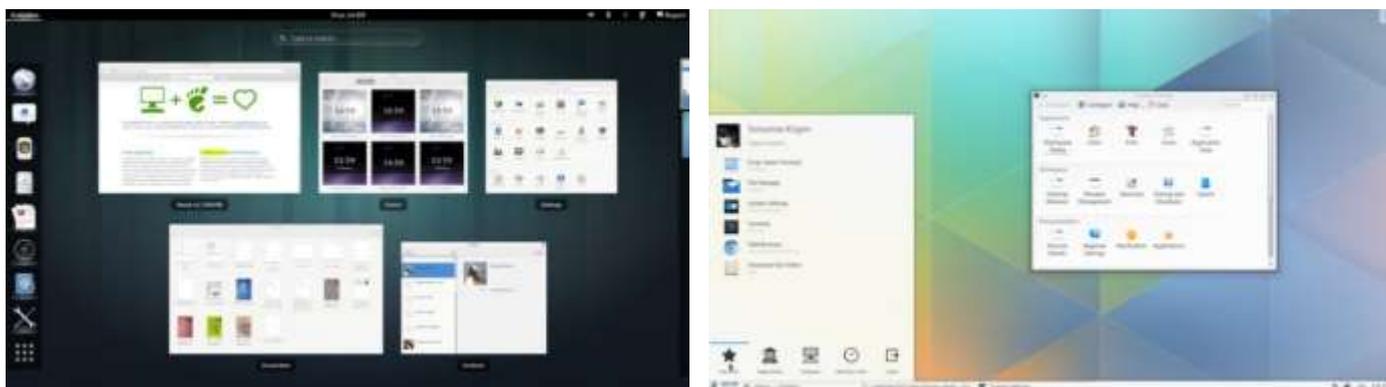
**Gabarito:** Errada

**25.(COPESE/UFT - 2018) O sistema operacional Linux é reconhecido por permitir diversos níveis de personalização, inclusive de suportar o uso de vários ambientes gráficos. Assinale a alternativa que NÃO constituiu uma interface gráfica usada no Linux**

- A) Pantheon.
- B) XFCM.
- C) MATE Desktop.
- D) Cinnamon Desktop.

#### **Comentários:**

O servidor de janelas é um ambiente gráfico (que se parece bastante com o sistema operacional Windows), tornando possível que tarefas sejam executadas sem a digitação de comandos (mas o usuário pode abrir um shell dentro do ambiente gráfico, se quiser). Abaixo podemos ver dois exemplos, o Gnome (à esquerda) e o KDE (à direita).



Além desses dois, outros exemplos são Pantheon, MATE, Cinnamon, XFCE, Deepin, entre outros. Sacanagem foi colocar "XFCM" como alternativa, pois o correto seria "XFCE"! Portanto, a **alternativa B** está correta e é o gabarito da questão.

**Gabarito:** Letra B



26.(CCV-UFC/UFC - 2018) De acordo com a linha do arquivo '/ etc / passwd' exibida abaixo, qual das seguintes afirmações é verdadeira?

```
jferreira:x:502:1000:Joao Ferreira:/home/jferreira:/bin/bash
```

- A) O usuário Joao Ferreira possui a senha 'x'.
- B) Shadow passwords (senhas ocultas) são utilizados no sistema.
- C) O usuário Joao Ferreira pertence ao grupo com groupID 502.
- D) Membros do groupID 502 podem ler o diretório /home/jferreira.
- E) O nome de usuário (username) 'jferreira' pertence ao grupo 'jferreira'.

#### Comentários:

O segundo campo seria a senha (antigamente até era), mas há um bom tempo o hash da senha é colocado no arquivo /etc/shadow e no lugar dela é colocado um "x". Ou seja, esse "x" indica que shadow passwords (senhas ocultas) são utilizados no sistema. Portanto, a **alternativa B** está correta e é o gabarito da questão.

**Gabarito:** Letra B

27.(Quadrix/CRM-PR - 2018) É possível instalar, em um mesmo computador desktop, os sistemas operacionais Windows 7 e Linux, de forma independente.

#### Comentários:

É possível, sim! Cada um em uma partição e um gerenciador de partição é utilizado para criar um menu e mostrá-lo quando você liga a máquina. Um exemplo clássico é o GRUB. Portanto, a questão está **correta**.

**Gabarito:** Correta

28.(Quadrix/CRM-PR - 2018) O usuário root do sistema Linux tem autonomia para acessar arquivos de outros usuários, com exceção dos usuários administradores.

#### Comentários:

Root pode tudo! Até mesmo acessar os arquivos de outros usuários administradores! Portanto, a questão está **errada**.

**Gabarito:** Errada



29.(IDECAN/CRF-SP - 2018) "O arquivo \_\_\_\_\_ só é legível pelo superusuário e serve para manter senhas criptografadas protegidas contra o acesso não autorizado. Ele também fornece informações sobre contas que não são disponíveis em \_\_\_\_\_." Assinale a alternativa que completa correta e sequencialmente a afirmativa sobre o Sistema Operacional Linux.

- A) /etc/passwd; /etc/group
- B) /etc/group; /etc/passwd
- C) /etc/passwd; etc/shadow
- D) /etc/shadow; /etc/passwd

#### Comentários:

Lembre-se que no /etc/passwd é possível verificar alguns dados do usuário (menos a senha, tem um "x" em seu lugar). No /etc/shadow tem o hash da senha, além de outras informações (expiração da senha, entre outras). Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

30.(UERR/IPERON - 2018) Um administrador de uma rede baseada em Linux deseja guardar no servidor os scripts especiais para iniciar ou interromper módulos e programas diversos. Para isso, ele precisa guardar essas informações dentro do diretório:

- A) /srv
- B) /opt
- C) /etc
- D) /lib
- E) home

#### Comentários:

A gente pensa que "etc" significa "etcetera", mas o diretório quer dizer "environment tables and controls", então os arquivos de configuração geralmente se encontram em /etc. Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C



31.(UERR/IPERON - 2018) Em um computador com Linux, deseja-se instalar o sistema de arquivos ext3, mas com o nível de Journaling no qual se grava mudanças em arquivos de metadados, que força que a escrita do conteúdo dos arquivos seja feita após a marcação de seus metadados. Esse nível de Journaling é denominado:

- A) ordered
- B) cshell.
- C) iso9660
- D) journal.
- E) fsreiser.

#### Comentários:

O Ext3 possui três modos de operação:

- *ordered (default)*: o journal é atualizado no final de cada operação. Isso faz com que exista uma pequena perda de desempenho, já que a cabeça de leitura do HD precisa realizar duas operações de gravação, uma no arquivo que foi alterado e outra no journal (que é um arquivo especialmente formatado);
- *writeback*: o journal armazena apenas informações referentes à estrutura do sistema de arquivos (metadados) e não em relação aos arquivos propriamente ditos, é gravado de forma mais ocasional, aproveitando os momentos de inatividade.
- *journal*: é o mais seguro, todavia mais lento. Nesse modo, o journal armazena não apenas informações sobre as alterações, mas também uma cópia de segurança de todos os arquivos modificados, que ainda não foram gravados no disco. Por ser o mais lento, é o modo menos usado.

Portanto, a **alternativa A** está correta e é o gabarito da questão.

**Gabarito:** Letra A

32.(CS-UFG/IF Goiano - 2019) Os sistemas Linux buscam padronizar diretórios para a localização de arquivos. Comandos essenciais do sistema operacional, tais como cat, tar, su, rm e pwd, em geral, estão localizados nos seguintes diretórios:

- A) /lib e /var



- B) /usr/lib e /lib
- C) /bin e /usr/bin
- D) /usr/bin e /var

### Comentários:

Comandos são executáveis (binários), então, como sabemos que a estrutura de diretórios no Linux é organizada pelo tipo de informação, só nos resta a alternativa que possui "bin" duas vezes. Vejamos:

- /usr/bin: executáveis dos programas;
- /bin: comandos básicos, como cd, ls e cat.

Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**33.(FCC/SEFAZ-BA - 2019) Um Auditor Fiscal da área de Tecnologia da Informação deseja desinstalar um pacote chamado java-1.6.0-openjdk.x86\_64 em linha de comando, como usuário root, no Red Hat Enterprise Linux 6. Para isso, terá que utilizar o comando**

- A) apt-get uninstall java-1.6.0-openjdk.x86\_64
- B) apt-get remove -rf java-1.6.0-openjdk.x86\_64
- C) rm -rf java-1.6.0-openjdk.x86\_64
- D) yum remove java-1.6.0-openjdk.x86\_64
- E) apt-get -rf java-1.6.0-openjdk.x86\_64

### Comentários:

YUM: resolve dependências de pacotes para distribuições que utilizam o RPM. É um gerenciador de pacotes padrão incluído em algumas distribuições baseados no REDHAT, incluindo o Fedora e CentOS. Muito parecido com o APT, vejamos alguns comandos:

- Atualização dos repositórios do YUM: sudo yum update;
- Instalação de um pacote: sudo yum install nome\_do\_pacote;
- Remoção de um pacote: sudo yum remove nome\_do\_pacote;
- Busca de um pacote que será instalado: sudo yum search nome\_do\_pacote.



Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

**34.(IF-MS/IF-MS - 2019)** O sistema operacional Linux possui várias partições (áreas) em sua estrutura, cada uma com uma função definida. Assinale a alternativa que apresenta a partição que abriga a pasta raiz do sistema que contém arquivos essenciais ao seu pleno funcionamento:

- A) /boot
- B) /bin
- C) /
- D) /etc
- E) /root

**Comentários:**

No Linux o diretório raiz é representado por "/", enquanto no Windows é representado por "\". Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**35.(VUNESP/Câmara de Sertãozinho-SP - 2019)** Considere que o usuário de um computador com sistema operacional Linux deseja executar um comando em segundo plano (background) em um terminal Shell. Para efetivar essa ação, o usuário deve executar o comando

- A) e teclar a combinação Ctrl+c
- B) e teclar a combinação Ctrl+x
- C) seguido de bg
- D) seguido de &
- E) seguido de !

**Comentários:**

O mais utilizado para executar em background é utilizar "&" logo após o nome do binário ou script. Por exemplo, se você deseja executar um script "aprovacao.sh", que vai realizar vários



cálculos e no fim escrever em um arquivo, não é necessário que você fique com o shell "travado", aguardando a conclusão da tarefa. Então é só digitar (note que o shell não fica esperando a conclusão do script, já aguarda o próximo comando):

```
estrategia:~$ aprovacao.sh &  
estrategia:~$
```

Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

**36.(COSEAC/UFF - 2019) No Sistema Operacional Linux o diretório local padrão do superusuário é:**

- A) /home.
- B) /usr.
- C) /root.
- D) /lib.
- E) /dev.

**Comentários:**

Os "usuários comuns" possuem como diretório local padrão "/home/nome\_usr" (ex.: /home/evandro), enquanto o root possui como diretório local padrão o "/root". Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**37.(CESPE/SLU-DF - 2019) Windows e Linux são classificados como sistemas operacionais de tempo real crítico, porque fornecem garantias absolutas de que todas as suas ações ocorrerão dentro de intervalos de tempo determinados.**

**Comentários:**

Windows e Linux são sistemas operacionais interativos, pois recebem instruções por teclado, mouse, caneta ótica etc., realiza as tarefas, mostra os resultados via impressora, monitor de vídeo etc. Logo, há muita interação, o que é diferente de sistemas de tempo real ou ainda os sistemas de lote (são as 3 classificações relacionadas a esse assunto). Portanto, a questão está **errada**.

**Gabarito:** Errada



38.(INAZ do Pará/CORE-SP - 2019) "Todos os arquivos e diretórios do sistema Linux instalado no computador partem de uma única origem: o diretório raiz. Mesmo que estejam armazenados em outros dispositivos físicos." - Disponível em: <https://canaltech.com.br/linux/entendendo-a-estrutura-de-diretorios-dolinux/>. Acesso em: 13.12.2018.

Qual pasta no sistema operacional LINUX tem a função de armazenar arquivos de dispositivos do sistema?

A) /dev.

B) /etc.

C) /lib.

D) /var.

E) /proc.

#### Comentários:

Dispositivos em inglês são devices. Como Linux geralmente é intuitivo...a resposta é "/dev". Portanto, a **alternativa A** está correta e é o gabarito da questão.

Gabarito: Letra A

39.(INAZ do Pará/CORE-SP - 2019) "O agendamento de tarefas é um recurso muito interessante para a administração de sistemas operacionais. É possível programar a execução de scripts de manutenção do sistema, disparar envio de newsletters, gerar relatórios de análises de logs, entre outros" - Disponível em: [https://www.vivaolinux.com.br/dica/Agendamento-de-tarefas-no-Linux-\(cron-e-at\)](https://www.vivaolinux.com.br/dica/Agendamento-de-tarefas-no-Linux-(cron-e-at)). Acesso em: 13.12.2018.

Qual programa do sistema operacional LINUX executa tarefas de modo automático, em horários pré-determinados, e executa tarefas que não foram executadas, enquanto o sistema esteve desligado?

A) Fcron.

B) Cron.

C) Anacron.

D) Schedule.

E) Crontab.



## Comentários:

Para o agendamento de tarefas a aplicação mais conhecida é o Cron, que já existe há muito tempo e passou por muitos estágios de evolução. Porém, a maioria das suas implementações atuais (Vixie Cron, ISC Cron, BCron etc.) ainda se baseiam no pressuposto de que o sistema está em funcionamento de forma ininterrupta. Isso é um problema quando se utiliza a virtualização, quando os sistemas que operam sob demanda, sendo ligados e desligados conforme sua necessidade.

Para contornar essa situação, algumas distribuições contam com o Anacron como alternativa ao Cron. O Anacron permite a criação de listas de tarefas que serão realizadas em intervalos pré-definidos e, quando o Anacron é inicializado, ele verifica essas listas e executa as tarefas ainda não realizadas. No entanto, o Anacron possui algumas limitações. A primeira é que ele não é um daemon, então precisa ser executado sempre que for necessário.

Outra questão é que o Anacron não está preparado para lidar com períodos de tempo menores do que dias. A combinação desses problemas pode levar a situações nas quais o Cron e o Anacron funcionem ao mesmo tempo, e algumas tarefas podem ser executadas duas vezes ou nenhuma!

Uma solução é o Fcron, que faz o que o Vixie Cron e o Anacron fazem e ainda mais. É possível usar o Fcron para agendar cronjobs com data e hora fixas, com intervalos de tempo ou até mesmo de acordo com a disponibilidade do sistema.

Portanto, a **alternativa A** está correta e é o gabarito da questão.

**Gabarito:** Letra A

**40. (IDECAN/IF-PB - 2019)** O Debian, uma das distribuições do sistema operacional Linux, possui um conjunto de pastas com nome pré-estabelecidos. Essas pastas possuem significado específico para o sistema operacional, visando atender necessidades do mesmo. Assinale a alternativa que indica corretamente o nome da pasta que armazena arquivos referentes às instalações de programas não oficiais da distribuição do sistema operacional.

A) /bin

B) /lib

C) /opt

D) /var

E) /proc

## Comentários:



Programas não oficiais poderiam ser chamados de "opcionais" (optionals), então vamos usar a lógica de nomes do Linux... "/opt". Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**41.(UFRR/UFRR - 2019) A respeito de diretórios no sistema operacional Linux, a estrutura é conhecida como árvore invertida, isto é, a raiz da árvore de diretórios é o topo. Qual é a representação do diretório raiz, e nome do usuário que possui privilégio para escrever neste diretório.**

- A) (raiz) e usuário root
- B) \\ (raiz) e usuário administrador
- C) / (raiz) e usuário root
- D) [ (raiz) e usuário administrador
- E) ]/ (raiz) e usuário root

#### **Comentários:**

Quem está acostumado com Windows sabe que o diretório raiz é "\", mas no Linux é invertida: "/". No Windows o usuário que "pode tudo" é o "Administrador", enquanto no Linux é o "root". Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**42.(IF-ES/IF-ES - 2019) O Network File System (NFS) – Sistemas de Arquivos em Rede – tem como um dos principais propósitos dar suporte a um sistema heterogêneo, no qual clientes e servidores estejam possivelmente executando sistemas operacionais e hardwares diferentes. Sobre o NFS, é CORRETO afirmar:**

- A) O NFS utiliza dois protocolos cliente-servidor, em que o primeiro é responsável pela montagem e o segundo é para acesso de diretório e arquivos.
- B) O servidor tem total gerência sobre o ponto de montagem nos clientes.
- C) Os serviços NFS são implementados apenas nos servidores Linux.
- D) Como critério de segurança, os clientes não podem ter acesso aos atributos dos arquivos.



E) O NFS faz uso de máquinas distintas para servidores e clientes, impossibilitando que a mesma máquina seja tanto cliente quanto servidor.

### Comentários:

Para que os clientes possam acessar o servidor NFS é necessário que os seguintes daemons estejam em execução:

- nfsd: daemon NFS, atende requisições dos clientes NFS;
- mountd: daemon de montagem NFS, executa as solicitações passadas pelo nfsd;
- portmap: daemon portmapper, permite que clientes NFS descubram qual porta o servidor NFS está utilizando.

Para essa questão, esqueça o portmap, pois só serve para descobrir a porta do servidor. Os outros dois daemons fazem o papel dos protocolos citados na alternativa A: um atende requisições dos clientes (acessos a diretórios e arquivos) e o outro faz a montagem.

Portanto, a **alternativa A** está correta e é o gabarito da questão.

**Gabarito:** Letra A

**43.(FCC/Prefeitura de Manaus-AM - 2019) Em computador com sistema operacional Linux e o OpenLDAP instalado, o programador editou o arquivo ldap.conf para configurar**

- A) a autenticação do servidor.
- B) a execução do serviço do LDAP.
- C) o banco de dados a ser utilizado.
- D) a geração de índices do banco de dados.
- E) o acesso dos clientes ao diretório.

### Comentários:

O principal arquivo de configuração (pelo menos para concursos) é o "/etc/ldap.conf", que é usado para configurar os padrões a serem aplicados quando da execução dos clientes LDAP. Portanto, a **alternativa E** está correta e é o gabarito da questão.

**Gabarito:** Letra E



## QUESTÕES COMENTADAS - SHELL SCRIPT - MULTIBANCAS

1. (CESGRANRIO/Petrobras - 2011) Em um sistema Unix, um arquivo de script chamado teste.sh foi copiado para o diretório /tmp. No shell do sistema, o usuário submeteu dois comandos: cd /tmp e teste.sh. Após a execução do segundo comando, o shell informou uma mensagem de erro, indicando comando não encontrado. O que deve ser feito para corrigir o problema que gerou essa mensagem?
- A) Certificar que o usuário não entrou com letras maiúsculas.
  - B) Certificar que o arquivo tem permissão para ser executado.
  - C) Omitir a extensão .sh ao entrar com o nome do script.
  - D) Incluir ./ antes do nome do script.
  - E) Mover o arquivo para o diretório home e executá-lo.

### Comentários:

No Linux, quando um executável (binário ou um script com permissão de execução) não estiver no path (lista de diretórios que permitem a execução), é possível executar através de "./" antes do nome do executável. Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

2. (CESGRANRIO/Petrobras - 2012) No ambiente UNIX, existem vários interpretadores de linha de comando conhecidos como shell. É importante, para cada script, informar em que shell ele deve ser executado.

Para isso, o usuário pode especificar o shell desejado

- A) na primeira linha do script.
- B) na última linha do script.
- C) em qualquer linha do script.
- D) em um arquivo à parte.
- E) na linha de comando, após o nome do arquivo que contém o script.

### Comentários:



É necessário definir qual Shell será utilizado na primeira linha do arquivo (vamos escolher o bash, mas poderia ser outro interpretador de comandos, como sh, ksh ou csh):

```
#!/bin/bash
```

Portanto, a **alternativa A** está correta e é o gabarito da questão.

**Gabarito:** Letra A

3. (CEPERJ/Rio previdência - 2014) Shell script é uma linguagem de script para Linux, nada mais do que comandos do próprio Linux que são executados em uma determinada sequência para uma determinada finalidade. Nesse contexto, duas situações são listadas a seguir.

I- No terminal ou modo gráfico, deseja-se criar um arquivo que possa ser editado para que se torne o primeiro shell script a ser criado, sendo necessário utilizar um comando CMD1.

II- Para que seja possível executar o shell script criado, é preciso atribuir a este o direito de execução; para isso é necessário usar um comando CMD2.

Exemplos de CMD1 e de CMD2 são, respectivamente:

- A) touch shell1.sh e exec +x shell1.sh
- B) touch shell1.sh e chmod +x shell1.sh
- C) create shell1.sh e chmod +x shell1.sh
- D) new shell1.sh e chmod +x shell1.sh
- E) new shell1.sh e exec +x shell1.sh

#### Comentários:

Criando um arquivo:

Há dois modos de realizar essa ação: via modo gráfico ou via terminal. No segundo caso, pode ser utilizado o comando vi, conforme mostrado a seguir (note a "extensão" .sh).

vi script.sh: Será criado e aberto um arquivo de leitura e escrita.

Outra opção é digitar o comando touch:

touch script.sh: cria um arquivo sem abri-lo.

Permissão ao arquivo:



Para começar a editar o arquivo, é necessário conceder a permissão de escrita a ele. Uma opção é utilizar o comando:

`chmod 777 script.sh`: o comando `chmod` é utilizado para conceder permissões em diretórios e arquivos, enquanto o valor `777` permite que o usuário tenha total liberdade para editar o arquivo (na verdade dá liberdade total a qualquer usuário do sistema! Mas nosso foco agora não é estudar o `chmod`).

Pensando mais em segurança, pode-se, por exemplo, liberar apenas a execução (x):

`chmod +x script.sh`: como não foi especificado para quem foi liberada a execução (usuário, grupo ou todos), todos devem ter a permissão de execução.

Portanto, a **alternativa B** está correta e é o gabarito da questão.

**Gabarito:** Letra B

4. (FUMARC/AL-MG - 2014) No Linux, considere a existência de um arquivo "arquivo.txt" no diretório corrente, com o seguinte conteúdo:

linha 1 valor 10 linha 2 valor 20 linha 3 valor 30

Deseja-se produzir, a partir do arquivo, a seguinte saída:

1 - 10 2 - 20 3 - 30

O script shell usado com comandos AWK que produz a saída desejada é:

- A) `awk arquivo.txt '{ print $2" - "$4 }'`
- B) `cat arquivo.txt | awk '{ print $2" - "$4 }'`
- C) `cat arquivo.txt | awk '{ print $1" - "$3"\n" }'`
- D) `cat arquivo.txt | awk '{ printf($2" - "$4) }'`

#### Comentários:

O comando `cat` é usado para unir, criar e exibir arquivos. No caso dessa questão, serve para exibir.

O `awk` é um utilitário especializado em manipulação de texto.

O pipe ("`|`") serve para uma comunicação entre os processos, ou seja, a saída do que está na esquerda dele é passada para o que está à direita.

Então:



cat arquivo.txt: exibe o conteúdo do arquivo e esse conteúdo é passado para:

awk '{ print \$2" - "\$4 }': imprime o 2º argumento, um traço ("-"), o 4º argumento e continua processando o texto até o fim da linha. Logo, imprime:

linha 1 valor 10 linha 2 valor 20 linha 3 valor 30

Lembrando os traços entre o 2º e o 4º argumentos, fica assim:

1 - 10 2 - 20 3 - 30

Portanto, a **alternativa B** está correta e é o gabarito da questão.

**Gabarito:** Letra B

5. (VUNESP/CRO-SP - 2015) No sistema operacional UNIX, uma aplicação pode ser desenvolvida por meio de um script que é um conjunto de comandos que podem ser executados pelo interpretador de comandos (shell). Considerando a existência do interpretador /bin/sh, a primeira linha de um arquivo de script deve conter:

- A) \*/bin/sh
- B) \$/bin/sh
- C) &/bin/sh
- D) #!/bin/sh
- E) #\*/bin/sh

#### Comentários:

Para definir qual Shell será utilizado, deve-se especificar na primeira linha do arquivo (vamos escolher o sh, conforme a questão, mas poderia ser outro interpretador de comandos, como bash, ksh ou csh):

```
#!/bin/sh
```

Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

6. (INSTITUTO AOCP/EBSERH - 2016) Em sistemas operacionais Linux é possível criar scripts para automatizar tarefas rotineiras. A extensão de arquivo utilizada para Shell Script é



- A) .bin
- B) .sl
- C) .bash
- D) .sh
- E) .gz

#### Comentários:

Vimos em aula o "script.sh", então o padrão utilizado como "extensão" é .sh. Portanto, a alternativa D está correta e é o gabarito da questão.

Gabarito: Letra D

7. (FCC/ELETOBRAS-ELETROSUL - 2016) Um profissional de TI está usando um computador com sistema operacional Linux que utiliza no shell o interpretador de comandos bash. Ele está logado como usuário teste e criou o seguinte arquivo shell script:

```
1 #!/bin/bash
2 echo 'Eletrosul- Centrais Elétricas S.A.'
3 $ variavel= 'Eu estou logado como usuário $user'
4 $ echo $variavel
```

Considerando que 1, 2, 3 e 4 indicam as linhas do arquivo e que este tenha sido salvo com o nome exemplo, é correto afirmar:

- A) Para o arquivo ser executável, é necessário acionar o comando \$ chmod +x exemplo. Depois disto o arquivo poderá ser executado com ./exemplo.
- B) A linha 1 indica que todas as outras linhas abaixo deverão ser executadas pelo compilador sh, que se localiza em /bin/bash.
- C) Após ser executado, o arquivo imprimirá na tela apenas frase "Eletrosul – Centrais Elétricas S. A. " utilizando o comando echo.
- D) Ao acionar o comando file arquivo é possível ver que a definição dele é Bourne-Again Shell Script, que se refere ao bash script.
- E) As linhas 3 e 4 farão com que seja impresso na tela Eu estou logado como usuário \$teste.



## Comentários:

Vamos analisar o script:

`#!/bin/bash`: especifica o shell bash

`echo 'Eletrosul- Centrais Elétricas S.A.'`: imprime na tela 'Eletrosul- Centrais Elétricas S.A.'

`$ variavel= 'Eu estou logado como usuário $user'`: A variável recebe a string 'Eu estou logado com o usuário teste'

`$ echo $variavel`: desconsidere \$ inicial, então imprime 'Eu estou logado com o usuário teste'

Analisando as alternativas, a única correta é a que menciona que para o arquivo (script) ser executável, é necessário acionar o comando "chmod +x exemplo" (dá a permissão de execução a todos, pois não especificou a quem seria). Depois disto o arquivo poderá ser executado com "./exemplo".

Portanto, a **alternativa A** está correta e é o gabarito da questão.

**Gabarito:** Letra A



## LISTA DE QUESTÕES - CONFIGURAÇÃO DE SERVIÇOS OU FUNCIONALIDADES - MULTIBANCAS

1. (AOCP/EBSERH - 2016) O NFS (Network File System) permite compartilhar sistemas de arquivos entre computadores conectados em rede e pode ser parte fundamental da infraestrutura da tecnologia da informação. Sobre NFS, analise as assertivas a seguir e assinale a alternativa que aponta a(s) correta(s).
    - I. O NFS é considerado sem estado (stateless) e, portanto, quando um servidor NFS volta a funcionar, o estado anterior é restaurado e a informação não é perdida.
    - II. O NFS pode ser implementado do lado servidor e do lado cliente.
    - III. O NFS roda sobre o protocolo RPC (Remote Procedure Call), que define um modo independente do sistema para processos se comunicarem através de uma rede de computadores.
    - IV. O NFS suporta apenas UDP como protocolo de transporte, pois ele apresenta desempenho significativamente melhor que o TCP em redes locais.
  
  - A) Apenas I e II.
  - B) Apenas II e IV.
  - C) Apenas III.
  - D) Apenas III e I.
  - E) Apenas I, II e III.
2. (FGV/IBGE - 2017) Em ambiente Linux, a chamada ao sistema (system call) que implementa a criação de um novo processo é:
    - A) create\_process;
    - B) new\_process;
    - C) fork;
    - D) spawn;
    - E) duplicate.



3. (FCC/TRF5 - 2017) Considere a figura abaixo que mostra a arquitetura do sistema operacional Linux



**A caixa**

A) I representa a camada responsável pela interface entre o hardware e as aplicações. Dentre suas funções encontram-se gerenciamento de I/O, manutenção do sistema de arquivos, gerenciamento de memória e swapping, controle da fila de processos etc.

B) II representa a camada que permite o acesso a recursos através da execução de chamadas feitas por processos. Tais chamadas são geradas por funções padrão suportadas pelo kernel. Dentre suas funções estão habilitar funções padrão como open, read, write e close e manter a comunicação entre as aplicações e o kernel.

C) I é um processo que executa funções de leitura de comandos de entrada de um terminal, interpreta-os e gera novos processos, sempre que requisitados. É conhecido também como interpretador de comandos.

D) II é um processo que realiza modificações no shell, permitindo que funcionalidades do Linux sejam habilitadas ou desabilitadas, conforme a necessidade. Tal processo gera ganho de performance, pois à medida que customiza o shell, o usuário torna o Linux enxuto e adaptável.

E) I é um processo que realiza modificações no kernel, permitindo que funcionalidades do Linux sejam habilitadas ou desabilitadas, conforme a necessidade. Tal processo gera ganho de performance, pois à medida que customiza o kernel, o usuário torna o Linux enxuto e adaptável.

4. (CS-UFG/DEMAE - 2017) Na maior parte das distribuições Linux, a memória virtual é uma partição denominada

- A) EXT
- B) SWAP
- C) SETUP
- D) BIOS



5. (Quadrix/COFECI - 2017) Não é possível realizar uma instalação automática, por meio de uma rede, no sistema operacional Linux.
6. (Colégio Pedro II/Colégio Pedro II - 2017) Sistema operacional é um conjunto de programas básicos e utilitários que fazem seu computador funcionar. O Debian é um sistema operacional, em cujo núcleo está o Kernel, o programa mais fundamental no computador e que faz todas as operações mais básicas, permitindo que você execute os outros programas. O Debian atualmente usa o Kernel Linux. Mas um sistema operacional não funciona somente com o Kernel, são necessários utilitários e aplicativos. O Debian utiliza estas ferramentas do projeto GNU. Por esse motivo, muitos utilizadores defendem que devemos chamar o sistema de "Debian GNU/Linux". Reconheça (1) o gerenciador de pacotes (programas) usado no Debian e em distribuições derivadas do Debian (Ubuntu, Knoppix, Big Linux etc.) que utiliza uma lista de dependências para instalar tudo o mais automaticamente possível, e (2) o arquivo de configuração utilizado por este gerenciador de pacote.

A) Gerenciador (1) - apt-get

Arquivo (2) - source.list

B) Gerenciador (1) - apt-get

Arquivo (2) - package.cfg

C) Gerenciador (1) - make install

Arquivo (2) - package.cfg

D) Gerenciador (1) - dpkg

Arquivo (2) - source.list

7. (CESPE/EMAP - 2018) O kernel do sistema operacional Linux tem a função de interpretar os comandos executados em um terminal.

8. (FGV/MPE-AL - 2018) Instalar, atualizar e remover pacotes no sistema operacional CentOS 7 é uma tarefa frequente para desenvolvedores de sistemas. Por isso, eventualmente podem ocorrer dúvidas sobre se determinado pacote está instalado ou qual é a versão que está sendo utilizada. Para dirimir essas dúvidas, sobre o pacote httpd devemos utilizar o comando

A) rpm -q httpd



- B) yum check httpd
- C) apt-get find httpd
- D) find -iname httpd
- E) crontab -l httpd

9. (FGV/COMPESA - 2018) Com relação às características e tarefas de administração do sistema operacional Linux, analise as afirmativas a seguir.

I. O esquema de partição GPT (Guid Partition Table) oferece a possibilidade de até 128 partições primárias, com tamanhos maiores do que 2TB cada.

II. Ext4, XFS e exFAT são exemplos de sistemas de arquivo com jornal, utilizados pelo Linux.

III. A área de swap no Linux pode ser uma partição dedicada de swap, um arquivo de swap ou uma combinação de arquivos e partições de swap.

Está correto o que se afirma em

- A) I, somente.
- B) II, somente.
- C) III, somente.
- D) I e III, somente.
- E) I, II e III.

10.(CS-UFG/SANEAGO-GO - 2018) Um utilitário GNU, presente em diversos sistemas operacionais linux, que lista as partições dos discos rígidos, é:

- A) fsdisk.
- B) mcdisk.
- C) parted.
- D) gedit.

11.(IF-RS/IF-RS - 2018) São tipos de sistemas de arquivos utilizados no Linux:

- A) FAT 16, FAT 32, NTFS



- B) FAT 32, HFS, ExFAT
- C) Ext2, Ext3, Btrfs
- D) Ext2, Ext3, HFS
- E) Ext2, Ext3, NFS, SMB

**12.(COPESE-UFT/UFT - 2018)** Embora seja possível realizar boot de um sistema Linux a partir de um pendrive, a maioria das instalações do Linux o realiza a partir do disco rígido do computador. Esse processo consiste em duas fases básicas:

1. Executar o carregador de boot a partir do dispositivo de boot;
2. Iniciar o kernel do Linux e iniciar os processos.

Assinale a alternativa que contém um gerenciador de boot para sistemas Linux:

- A) CISC (Complex Instruction Set Computer).
- B) GRUB (Grand Unified Bootloader).
- C) ORM (Object Relational Mapping).
- D) CMS (Content Management System).

**13.(COPESE-UFT/UFT - 2018)** Sistemas operacionais Linux permitem logins simultâneos de diferentes usuários. Assinale a alternativa que contém o comando a ser utilizado para visualizar os usuários logados em um determinado instante.

- A) listusers
- B) ldconfig
- C) who
- D) uname

**14.(IF-RS/IF-RS - 2018)** Considerando que se está logado como "root" no terminal de uma estação de trabalho com um sistema operacional Linux debian ou derivado, qual comando configura o DNS para 8.8.8.8?

- A) dns 8.8.8.8
- B) echo 8.8.8.8 > /etc/hosts.conf



C) echo nameserver 8.8.8.8 > /etc/resolv.conf

D) nslookup 8.8.8.8

E) nameserver 8.8.8.8

**15.(COPEVE-UFAL/UFAL - 2018)** O assistente de tecnologia da informação precisa atualizar o sistema operacional GNU/Linux do servidor WEB de uma empresa. Qual comando, usando o modo root, deve ser utilizado?

A) upgrade distro-all.

B) upgrade apt-distro.

C) apt-get dist-upgrade.

D) apt-get purge "nome da distro".

E) apt-cache showpkg "nome da distro".

**16.(COPEVE-UFAL/UFAL - 2018)** Dadas as afirmativas quanto aos conceitos de Kernel no sistema operacional GNU/Linux,

I. Compilar o Kernel permite ao usuário remover drivers inúteis diminuindo o tempo de arranque do sistema operacional.

II. Os módulos do Kernel do sistema estão armazenados no diretório /boot/.

III. O comando uname -a exibe as informações do kernel do sistema.

IV. Módulos são as partes do kernel que são carregadas somente quando são requisitadas por um aplicativo ou dispositivo.

Verifica-se que está(ão) correta(s)

A) II, apenas.

B) I e II, apenas.

C) III e IV, apenas.

D) I, III e IV, apenas.

E) I, II, III e IV.



17.(FGV/AL-RO - 2018) Wallace é administrador de um servidor com sistema operacional CentOS e deseja compartilhar um diretório com os integrantes da empresa por meio do protocolo NFS. Para que a configuração desse compartilhamento possa ser efetuada, Wallace deve editar o arquivo /etc/

- A) services
- B) modules
- C) hosts
- D) export
- E) fstab

18.(COPEVE-UFAL/UFAL - 2018) Analise a configuração de rede no GNU/Linux.

```
auto lo
iface lo inet loopback
auto eth0
allow-hotplug eth0
iface eth0 inet static address 192.168.100.1 netmask 255.255.255.0 network 192.168.100.0
broadcast 192.168.100.255
```

Dadas as afirmativas quanto à configuração de rede,

- I. O "iface eth0 inet static" informa ao sistema que a placa de rede terá o endereço IP estático.
- II. O "iface lo" informa ao sistema que a placa de rede receberá um endereço IP via servidor DHCP.
- III. Uma segunda placa de rede com fio instalada nesse computador irá receber a denominação eth1.
- IV. O "allow-hotplug" permite conectar dispositivos hotplug no computador.

Verifica-se que está(ão) correta(s)

- A) II, apenas.
- B) I e III, apenas.
- C) II e IV, apenas.



D) I, III e IV, apenas.

E) I, II, III e IV.

**19.(FGV/AL-RO - 2018)** Roberto trabalha como administrador de redes em uma empresa de cosméticos e sua principal função é a configuração de estações de trabalho dos seus colaboradores. Rotineiramente, Roberto instala e remove pacotes nas estações de trabalho às quais ele dá suporte.

Assinale a opção que indica o comando utilizado por Roberto para desinstalar pacotes RPM nas estações de trabalho com sistema operacional Linux CentOS.

A) rpm -Uvh pacote

B) rpm -e pacote

C) rpm -ivh pacote

D) rpm -qi pacote

E) rpm -qa | grep pacote

**20.(Quadrix/CFBio - 2018)** Carregamento e inicialização do kernel e execução dos scripts de inicialização do sistema são algumas das etapas do processo de inicialização de um Linux típico.

**21.(Quadrix/CRM-PR - 2018)** Um computador pessoal com sistema operacional Linux, versão desktop, pode ter no máximo cinco usuários simultâneos.

**22.(FADESP/IF-PA - 2018)** Sobre os pacotes do Sistema Operacional (SO) Debian 9.5, é correto afirmar que

A) o apt é um programa que pode substituir o dpkg para o gerenciamento de pacotes do SO.

B) os pacotes devem ser distribuídos no formato .tar.gz.

C) o dpkg-source compila e executa código fonte de um pacote.

D) um pacote marcado como Essential não pode ser desinstalado do Sistema Operacional.

E) cada pacote deve especificar as dependências sobre outros pacotes para seu funcionamento.

**23.(CONSULPAM/Câmara de Juiz de Fora-MG - 2018)** Em relação ao Sistema Operacional Linux, marque o item INCORRETO:



- A) Sua arquitetura é composta por um núcleo monolítico cujas funções são: gerenciar a memória, operar as entradas e saídas e o acesso aos arquivos.
- B) Outra característica do Linux é com relação aos drivers de dispositivos e suporte a rede, os quais podem ser compactadas e utilizadas como se fossem módulos ou bibliotecas (LKM em inglês Loadable Kernel Modules), separados pela parte principal, cujo carregamento pode ser ativado após a execução do núcleo.
- C) No quesito portabilidade, o Linux funciona com eficiência em plataformas como x64 da Intel (EM64T e AMD64) PowerPC, Alpha, SPARC, porém é de difícil instalação nos sistemas embarcados como PVR, celulares, Tv's e Handhelds.
- D) A partir da década de 90, ao passo que a distribuição do Linux se popularizou, foi também limitada, pois se torna uma alternativa no uso de software livre, contra os sistemas operacionais da Apple (Mac OS) e Microsoft (Windows).

**24.(CESPE/FUB - 2018) O Linux é um sistema operacional em que cada usuário consegue ter apenas um processo ativo por vez, processo esse que é iniciado automaticamente quando o sistema é carregado.**

**25.(COPESE/UFT - 2018) O sistema operacional Linux é reconhecido por permitir diversos níveis de personalização, inclusive de suportar o uso de vários ambientes gráficos. Assinale a alternativa que NÃO constituiu uma interface gráfica usada no Linux**

- A) Pantheon.
- B) XFCM.
- C) MATE Desktop.
- D) Cinnamon Desktop.

**26.(CCV-UFC/UFC - 2018) De acordo com a linha do arquivo '/ etc / passwd' exibida abaixo, qual das seguintes afirmações é verdadeira?**

```
jferreira:x:502:1000:Joao Ferreira:/home/jferreira:/bin/bash
```

- A) O usuário Joao Ferreira possui a senha 'x'.
- B) Shadow passwords (senhas ocultas) são utilizados no sistema.
- C) O usuário Joao Ferreira pertence ao grupo com groupID 502.
- D) Membros do groupID 502 podem ler o diretório /home/jferreira.



E) O nome de usuário (username) 'jferreira' pertence ao grupo 'jferreira'.

**27.**(Quadrix/CRM-PR - 2018) É possível instalar, em um mesmo computador desktop, os sistemas operacionais Windows 7 e Linux, de forma independente.

**28.**(Quadrix/CRM-PR - 2018) O usuário root do sistema Linux tem autonomia para acessar arquivos de outros usuários, com exceção dos usuários administradores.

**29.**(IDECAN/CRF-SP - 2018) "O arquivo \_\_\_\_\_ só é legível pelo superusuário e serve para manter senhas criptografadas protegidas contra o acesso não autorizado. Ele também fornece informações sobre contas que não são disponíveis em \_\_\_\_\_." Assinale a alternativa que completa correta e sequencialmente a afirmativa sobre o Sistema Operacional Linux.

A) /etc/passwd; /etc/group

B) /etc/group; /etc/passwd

C) /etc/passwd; etc/shadow

D) /etc/shadow; /etc/passwd

**30.**(UERR/IPERON - 2018) Um administrador de uma rede baseada em Linux deseja guardar no servidor os scripts especiais para iniciar ou interromper módulos e programas diversos. Para isso, ele precisa guardar essas informações dentro do diretório:

A) /srv

B) /opt

C) /etc

D) /lib

E) home

**31.**(UERR/IPERON - 2018) Em um computador com Linux, deseja-se instalar o sistema de arquivos ext3, mas com o nível de Journaling no qual se grava mudanças em arquivos de metadados, que força que a escrita do conteúdo dos arquivos seja feita após a marcação de seus metadados. Esse nível de Journaling é denominado:

A) ordered

B) cshell.



C) iso9660

D) journal.

E) fsreiser.

**32.(CS-UFG/IF Goiano - 2019)** Os sistemas Linux buscam padronizar diretórios para a localização de arquivos. Comandos essenciais do sistema operacional, tais como cat, tar, su, rm e pwd, em geral, estão localizados nos seguintes diretórios:

A) /lib e /var

B) /usr/lib e /lib

C) /bin e /usr/bin

D) /usr/bin e /var

**33.(FCC/SEFAZ-BA - 2019)** Um Auditor Fiscal da área de Tecnologia da Informação deseja desinstalar um pacote chamado java-1.6.0- openjdk.x86\_64 em linha de comando, como usuário root, no Red Hat Enterprise Linux 6. Para isso, terá que utilizar o comando

A) apt-get uninstall java-1.6.0-openjdk.x86\_64

B) apt-get remove -rf java-1.6.0-openjdk.x86\_64

C) rm -rf java-1.6.0-openjdk.x86\_64

D) yum remove java-1.6.0-openjdk.x86\_64

E) apt-get -rf java-1.6.0-openjdk.x86\_64

**34.(IF-MS/IF-MS - 2019)** O sistema operacional Linux possui várias partições (áreas) em sua estrutura, cada uma com uma função definida. Assinale a alternativa que apresenta a partição que abriga a pasta raiz do sistema que contém arquivos essenciais ao seu pleno funcionamento:

A) /boot

B) /bin

C) /

D) /etc



E) /root

**35.(VUNESP/Câmara de Sertãozinho-SP - 2019)** Considere que o usuário de um computador com sistema operacional Linux deseja executar um comando em segundo plano (background) em um terminal Shell. Para efetivar essa ação, o usuário deve executar o comando

A) e teclar a combinação Ctrl+c

B) e teclar a combinação Ctrl+x

C) seguido de bg

D) seguido de &

E) seguido de !

**36.(COSEAC/UFF - 2019)** No Sistema Operacional Linux o diretório local padrão do superusuário é:

A) /home.

B) /usr.

C) /root.

D) /lib.

E) /dev.

**37.(CESPE/SLU-DF - 2019)** Windows e Linux são classificados como sistemas operacionais de tempo real crítico, porque fornecem garantias absolutas de que todas as suas ações ocorrerão dentro de intervalos de tempo determinados.

**38.(INAZ do Pará/CORE-SP - 2019)** "Todos os arquivos e diretórios do sistema Linux instalado no computador partem de uma única origem: o diretório raiz. Mesmo que estejam armazenados em outros dispositivos físicos." - Disponível em: <https://canaltech.com.br/linux/entendendo-a-estrutura-de-diretorios-dolinux/>. Acesso em: 13.12.2018.

Qual pasta no sistema operacional LINUX tem a função de armazenar arquivos de dispositivos do sistema?

A) /dev.

B) /etc.



- C) /lib.
- D) /var.
- E) /proc.

**39.(INAZ do Pará/CORE-SP - 2019)** "O agendamento de tarefas é um recurso muito interessante para a administração de sistemas operacionais. É possível programar a execução de scripts de manutenção do sistema, disparar envio de newsletters, gerar relatórios de análises de logs, entre outros" - Disponível em: [https://www.vivaolinux.com.br/dica/Agendamento-de-tarefas-no-Linux-\(cron-e-at\)](https://www.vivaolinux.com.br/dica/Agendamento-de-tarefas-no-Linux-(cron-e-at)). Acesso em: 13.12.2018.

Qual programa do sistema operacional LINUX executa tarefas de modo automático, em horários pré-determinados, e executa tarefas que não foram executadas, enquanto o sistema esteve desligado?

- A) Fcron.
- B) Cron.
- C) Anacron.
- D) Schedule.
- E) Crontab.

**40.(IDECAN/IF-PB - 2019)** O Debian, uma das distribuições do sistema operacional Linux, possui um conjunto de pastas com nome pré-estabelecidos. Essas pastas possuem significado específico para o sistema operacional, visando atender necessidades do mesmo. Assinale a alternativa que indica corretamente o nome da pasta que armazena arquivos referentes às instalações de programas não oficiais da distribuição do sistema operacional.

- A) /bin
- B) /lib
- C) /opt
- D) /var
- E) /proc

**41.(UFRR/UFRR - 2019)** A respeito de diretórios no sistema operacional Linux, a estrutura é conhecida como árvore invertida, isto é, a raiz da árvore de diretórios é o topo. Qual é a



representação do diretório raiz, e nome do usuário que possui privilégio para escrever neste diretório.

- A) (raiz) e usuário root
- B) \\ (raiz) e usuário administrador
- C) / (raiz) e usuário root
- D) [ (raiz) e usuário administrador
- E) ]/ (raiz) e usuário root

**42.(IF-ES/IF-ES - 2019) O Network File System (NFS) – Sistemas de Arquivos em Rede – tem como um dos principais propósitos dar suporte a um sistema heterogêneo, no qual clientes e servidores estejam possivelmente executando sistemas operacionais e hardwares diferentes. Sobre o NFS, é CORRETO afirmar:**

- A) O NFS utiliza dois protocolos cliente-servidor, em que o primeiro é responsável pela montagem e o segundo é para acesso de diretório e arquivos.
- B) O servidor tem total gerência sobre o ponto de montagem nos clientes.
- C) Os serviços NFS são implementados apenas nos servidores Linux.
- D) Como critério de segurança, os clientes não podem ter acesso aos atributos dos arquivos.
- E) O NFS faz uso de máquinas distintas para servidores e clientes, impossibilitando que a mesma máquina seja tanto cliente quanto servidor.

**43.(FCC/Prefeitura de Manaus-AM - 2019) Em computador com sistema operacional Linux e o OpenLDAP instalado, o programador editou o arquivo ldap.conf para configurar**

- A) a autenticação do servidor.
- B) a execução do serviço do LDAP.
- C) o banco de dados a ser utilizado.
- D) a geração de índices do banco de dados.
- E) o acesso dos clientes ao diretório.



## GABARITO



## GABARITO

- |           |             |            |
|-----------|-------------|------------|
| 1- E      | 16- D       | 31- A      |
| 2- C      | 17- D       | 32- C      |
| 3- C      | 18- D       | 33- D      |
| 4- B      | 19- B       | 34- C      |
| 5- Errada | 20- Correta | 35- D      |
| 6- A      | 21- Errada  | 36- C      |
| 7- Errada | 22- E       | 37- Errada |
| 8- A      | 23- C       | 38- A      |
| 9- D      | 24- Errada  | 39- A      |
| 10- C     | 25- B       | 40- C      |
| 11- C     | 26- B       | 41- C      |
| 12- B     | 27- Correta | 42- A      |
| 13- C     | 28- Errada  | 43- E      |
| 14- C     | 29- D       |            |
| 15- C     | 30- C       |            |



## LISTA DE QUESTÕES - SHELL SCRIPT - MULTIBANCAS

1. (CESGRANRIO/Petrobras - 2011) Em um sistema Unix, um arquivo de script chamado teste.sh foi copiado para o diretório /tmp. No shell do sistema, o usuário submeteu dois comandos: cd /tmp e teste.sh. Após a execução do segundo comando, o shell informou uma mensagem de erro, indicando comando não encontrado. O que deve ser feito para corrigir o problema que gerou essa mensagem?

- A) Certificar que o usuário não entrou com letras maiúsculas.
- B) Certificar que o arquivo tem permissão para ser executado.
- C) Omitir a extensão .sh ao entrar com o nome do script.
- D) Incluir ./ antes do nome do script.
- E) Mover o arquivo para o diretório home e executá-lo.

2. (CESGRANRIO/Petrobras - 2012) No ambiente UNIX, existem vários interpretadores de linha de comando conhecidos como shell. É importante, para cada script, informar em que shell ele deve ser executado.

Para isso, o usuário pode especificar o shell desejado

- A) na primeira linha do script.
- B) na última linha do script.
- C) em qualquer linha do script.
- D) em um arquivo à parte.
- E) na linha de comando, após o nome do arquivo que contém o script.

3. (CEPERJ/Rio previdência - 2014) Shell script é uma linguagem de script para Linux, nada mais do que comandos do próprio Linux que são executados em uma determinada sequência para uma determinada finalidade. Nesse contexto, duas situações são listadas a seguir.

I- No terminal ou modo gráfico, deseja-se criar um arquivo que possa ser editado para que se torne o primeiro shell script a ser criado, sendo necessário utilizar um comando CMD1.

II- Para que seja possível executar o shell script criado, é preciso atribuir a este o direito de execução; para isso é necessário usar um comando CMD2.



Exemplos de CMD1 e de CMD2 são, respectivamente:

- A) touch shell1.sh e exec +x shell1.sh
- B) touch shell1.sh e chmod +x shell1.sh
- C) create shell1.sh e chmod +x shell1.sh
- D) new shell1.sh e chmod +x shell1.sh
- E) new shell1.sh e exec +x shell1.sh

4. (FUMARC/AL-MG - 2014) No Linux, considere a existência de um arquivo "arquivo.txt" no diretório corrente, com o seguinte conteúdo:

linha 1 valor 10 linha 2 valor 20 linha 3 valor 30

Deseja-se produzir, a partir do arquivo, a seguinte saída:

1 - 10 2 - 20 3 - 30

O script shell usado com comandos AWK que produz a saída desejada é:

- A) awk arquivo.txt '{ print \$2" - "\$4 }'
- B) cat arquivo.txt | awk '{ print \$2" - "\$4 }'
- C) cat arquivo.txt | awk '{ print \$1" - "\$3"\n" }'
- D) cat arquivo.txt | awk '{ printf(\$2" - "\$4) }'

5. (VUNESP/CRO-SP - 2015) No sistema operacional UNIX, uma aplicação pode ser desenvolvida por meio de um script que é um conjunto de comandos que podem ser executados pelo interpretador de comandos (shell). Considerando a existência do interpretador /bin/sh, a primeira linha de um arquivo de script deve conter:

- A) \*/bin/sh
- B) \$/bin/sh
- C) &/bin/sh
- D) #!/bin/sh
- E) #\*/bin/sh



6. (INSTITUTO AOCP/EBSERH - 2016) Em sistemas operacionais Linux é possível criar scripts para automatizar tarefas rotineiras. A extensão de arquivo utilizada para Shell Script é

- A) .bin
- B) .sl
- C) .bash
- D) .sh
- E) .gz

7. (FCC/ELETROBRAS-ELETROSUL - 2016) Um profissional de TI está usando um computador com sistema operacional Linux que utiliza no shell o interpretador de comandos bash. Ele está logado como usuário teste e criou o seguinte arquivo shell script:

```
1 #!/bin/bash
2 echo 'Eletrosul- Centrais Elétricas S.A.'
3 $ variavel= 'Eu estou logado como usuário $user'
4 $ echo $variavel
```

Considerando que 1, 2, 3 e 4 indicam as linhas do arquivo e que este tenha sido salvo com o nome exemplo, é correto afirmar:

- A) Para o arquivo ser executável, é necessário acionar o comando `$ chmod +x exemplo`. Depois disto o arquivo poderá ser executado com `./exemplo`.
- B) A linha 1 indica que todas as outras linhas abaixo deverão ser executadas pelo compilador sh, que se localiza em `/bin/bash`.
- C) Após ser executado, o arquivo imprimirá na tela apenas frase "Eletrosul – Centrais Elétricas S. A. " utilizando o comando echo.
- D) Ao acionar o comando `file` arquivo é possível ver que a definição dele é Bourne-Again Shell Script, que se refere ao bash script.
- E) As linhas 3 e 4 farão com que seja impresso na tela Eu estou logado como usuário \$teste.



## GABARITO



GABARITO

1- D

2- A

3- B

4- B

5- D

6- D

7- A



## SERVIÇOS DE REDE – DNS E DHCP

Nesta seção vamos focar em dois serviços de rede bastante conhecidos, o DNS e o DHCP. Iniciamos com os conceitos e na sequência veremos como são implementados no Linux.

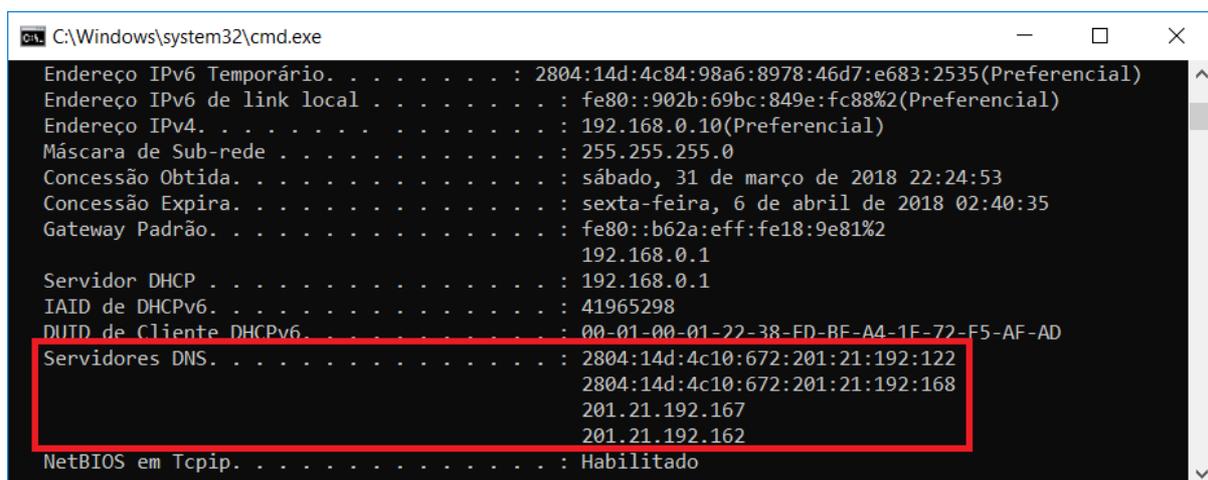
### DNS (Domain Name System)

Os equipamentos conectados à Internet necessitam de um endereço IP e através dele conseguimos identificar "quem" acessou determinado conteúdo ou realizou determinada ação, por exemplo. Mas, quem "gosta" de números é a máquina! Nós, meros mortais, temos dificuldade em memorizar muitos números (endereços IP, por exemplo). E os servidores (Web, de e-mail, de arquivos etc.) são acessados através do endereço IP que foi alocado a eles.

Uma solução para isso foi a criação de um serviço que "faz o meio de campo", ou seja, traduz nomes (bem mais fáceis de memorizar) para os endereços IP equivalentes. É o famoso *Domain Name System* (DNS). Assim, é possível acessar uma página Web através de um nome, sendo que de forma transparente ao usuário, esse nome é traduzido ao endereço IP onde se encontra o servidor Web que contém a página e a requisição é realizada a esse servidor.

Um conceito mais formal (Tanenbaum) é o seguinte: o DNS é definido como um esquema hierárquico de atribuição de nomes baseado no domínio e de um sistema de banco de dados distribuído. O DNS atua na camada de aplicação e utiliza como protocolo de transporte o UDP para as consultas/respostas e o TCP para transferências de zonas (entre servidores DNS). Tanto com o UDP como com o TCP, a porta utilizada é a 53.

Abaixo é mostrada uma tela com a resposta para o comando `ipconfig /all` (no Linux o comando equivalente é o `ifconfig`) mostrando, entre outras informações, os servidores DNS locais.



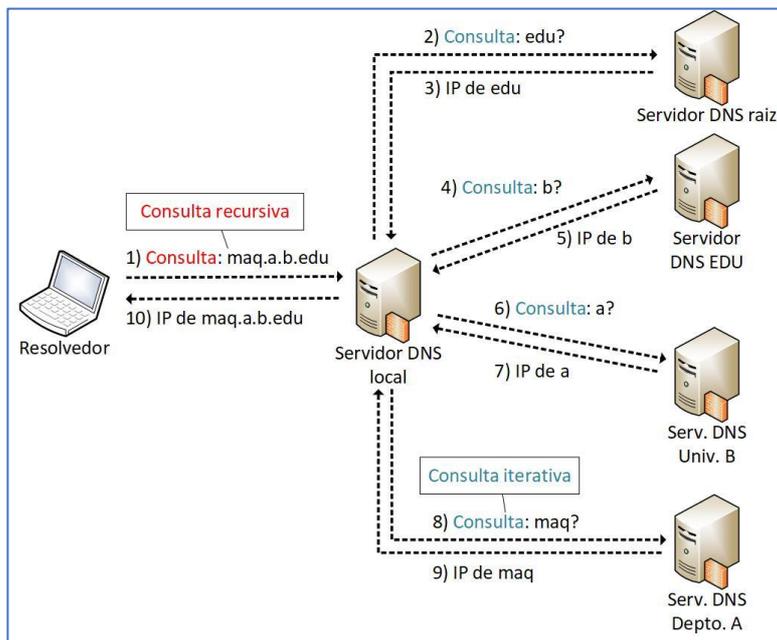
```
C:\Windows\system32\cmd.exe
Endereço IPv6 Temporário. . . . . : 2804:14d:4c84:98a6:8978:46d7:e683:2535(Preferencial)
Endereço IPv6 de link local . . . . . : fe80::902b:69bc:849e:fc88%2(Preferencial)
Endereço IPv4. . . . . : 192.168.0.10(Preferencial)
Máscara de Sub-rede . . . . . : 255.255.255.0
Concessão Obtida. . . . . : sábado, 31 de março de 2018 22:24:53
Concessão Expira. . . . . : sexta-feira, 6 de abril de 2018 02:40:35
Gateway Padrão. . . . . : fe80::b62a:eff:fe18:9e81%2
                          192.168.0.1
Servidor DHCP . . . . . : 192.168.0.1
IAID de DHCPv6. . . . . : 41965298
DUID de Cliente DHCPv6. . . . . : 00-01-00-01-22-38-FD-BE-A4-1E-72-F5-AF-AD
Servidores DNS. . . . . : 2804:14d:4c10:672:201:21:192:122
                          2804:14d:4c10:672:201:21:192:168
                          201.21.192.167
                          201.21.192.162
NetBIOS em Tcpi. . . . . : Habilitado
```

Um passo a passo de uma solicitação de um cliente DNS (seu computador, por exemplo) a um servidor DNS local é mostrado a seguir.



- 1) O aplicativo (ex.: navegador) chama o **resolvedor**, passando o nome que se deseja a tradução para endereço IP;
- 2) O resolvedor realiza uma consulta ao servidor DNS local;
- 3) O servidor DNS local responde ao resolvedor;
- 4) O resolvedor informa o endereço IP ao aplicativo.

Ok, mas e se for um nome que o servidor DNS local não conhece? Seja porque nunca foi solicitado, ou por que tal informação já não se encontra mais em sua *cache*? Bom, aí é melhor olhar a figura abaixo.



A consulta realizada ao servidor DNS local é chamada **consulta recursiva**, pois o resolvedor envia a consulta e recebe a resposta final, sem precisar enviar uma consulta a cada servidor DNS de nível superior. Já em **consultas iterativas**, a resposta à requisição DNS pode ser parcial, obrigando o solicitante a encaminhar novas requisições DNS a outros servidores até obter a resposta final desejada.

A delegação de domínios de mais alto nível (*top-level domain* - TLD), tais como .com, .edu, .br, .mx, entre outros, é de responsabilidade da ICANN (*Internet Corporation for Assigned Names and Numbers*). Para o Brasil (TLD .br), o responsável é o CGI.br<sup>1</sup>, conforme podemos ver abaixo.

<sup>1</sup> Base de dados de domínios TLD disponível em <<http://www.iana.org/domains/root/db>>.



.booking	generic	Booking.com B.V.
.boots	generic	THE BOOTS COMPANY PLC
.bosch	generic	Robert Bosch GMBH
.bostik	generic	Bostik SA
.boston	generic	Boston TLD Management, LLC
.bot	generic	Amazon Registry Services, Inc.
.boutique	generic	Binky Moon, LLC
.box	generic	NS1 Limited
.bq	country-code	Not assigned
.br	country-code	Comite Gestor da Internet no Brasil
.bradesco	generic	Banco Bradesco S.A.
.bridgestone	generic	Bridgestone Corporation
.broadway	generic	Celebrate Broadway, Inc.
.broker	generic	DOTBROKER REGISTRY LTD

CURIOSIDADE



Note que existe um TLD ".bradesco", relacionado ao Banco Bradesco. Faça um teste em seu navegador: digite "bradesco.com.br" e "bradesco.bradesco". Qual o resultado? No momento em que testei, ambos direcionam para uma nova URL: "https://banco.bradesco/html/classic/index.shtm", pertencente ao domínio ".bradesco".

Então, se alguém quiser registrar um domínio com o sufixo .br, pode verificar se há disponibilidade desse domínio, através da URL <http://registro.br>. Se houver, pode realizar a solicitação, efetuar o pagamento e informar as configurações solicitadas pelo CGI.br sobre o provedor onde a página será hospedada (servidores DNS).

Na medida em que novos domínios são cadastrados, eles são propagados pela Internet e em poucas horas todos os servidores DNS do mundo são capazes de traduzir o domínio para o endereço IP equivalente onde está hospedado o serviço. A figura abaixo mostra a estrutura DNS, desde a raiz, os TLDs, domínios de segundo e terceiro níveis e o computador lá na ponta.



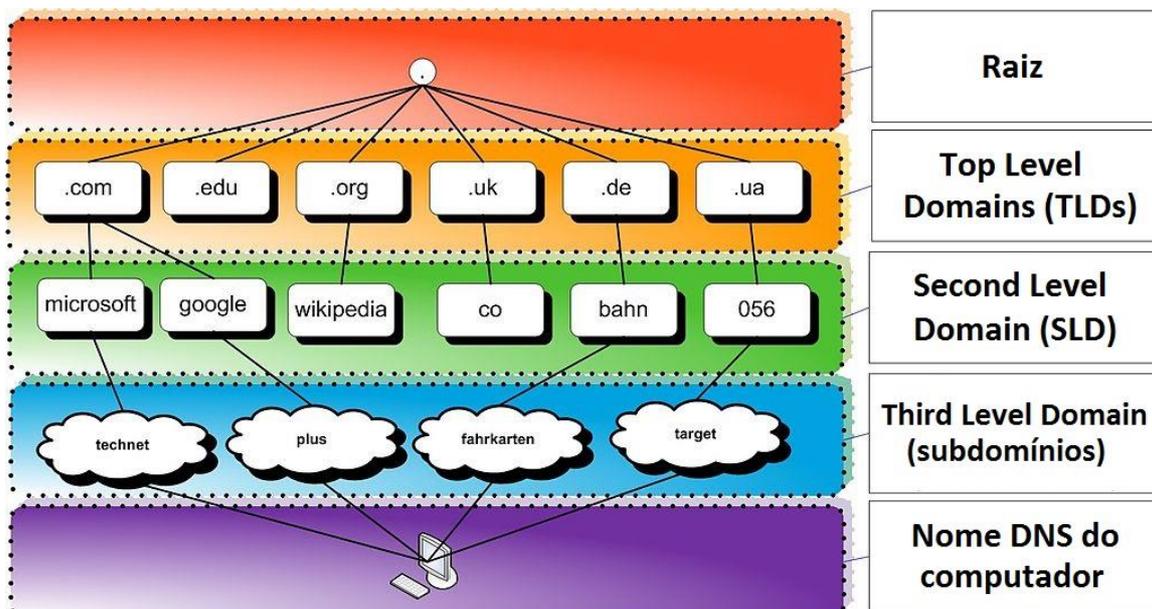
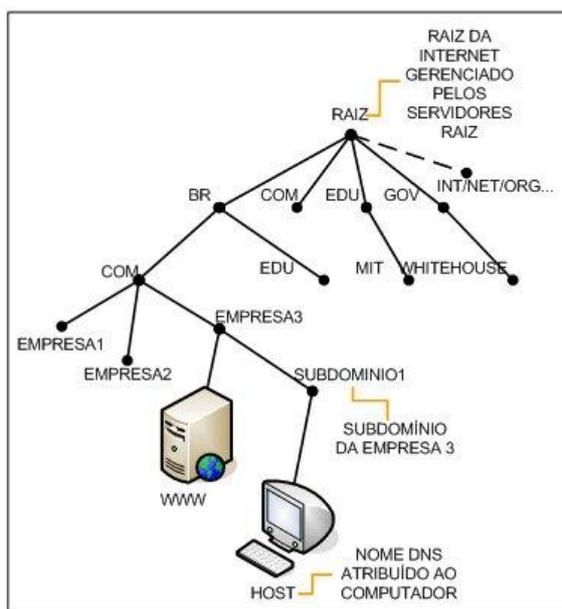


Figura adaptada de <<https://hugoemiliano.info/2017/07/05/servicos-e-protocolos-dns/>>.

Por exemplo, a URL <www.microsoft.com> pode ser compreendida da seguinte forma:

- .com: Top Level Domain (TLD);
- microsoft: Second Level Domain (SLD);
- não há terceiro nível (subdomínio) para essa URL;
- www: Nome do computador (www é um nome padrão para servidores Web).

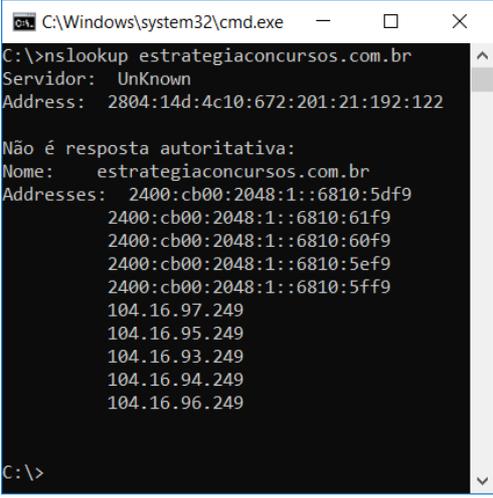
Abaixo uma outra figura, mostrando um exemplo com subdomínio. Nesse caso a URL completa para acessar o HOST seria <HOST.SUBDOMINIO1.EMPRESA3.COM.BR>.



Fonte: <[http://www.abusar.org/dns\\_como.html](http://www.abusar.org/dns_como.html)>.

Para não haver consultas constantes a servidores DNS de mais alto nível (mais próximos da raiz, ou a própria raiz), os servidores DNS possuem uma memória cache<sup>2</sup>, permitindo a resposta imediata ao solicitante (quando tiver a informação). Quando não tiver a informação, deve-se buscar nos níveis superiores.

É possível também, em sistemas operacionais como Windows e Linux, configurar em traduções fixas, de domínio para endereço IP (arquivo hosts, como já vimos). Uma ferramenta comum ao Windows e Linux para obter informações sobre registros de DNS de um determinado domínio, host ou IP é o *nslookup* (vale a pena utilizá-la, pois há questões que cobram o seu conhecimento):



```
C:\Windows\system32\cmd.exe - □ ×
C:\>nslookup estrategiaconcursos.com.br
Servidor: UnKnown
Address: 2804:14d:4c10:672:201:21:192:122

Não é resposta autoritativa:
Nome: estrategiaconcursos.com.br
Addresses: 2400:cb00:2048:1::6810:5df9
           2400:cb00:2048:1::6810:61f9
           2400:cb00:2048:1::6810:60f9
           2400:cb00:2048:1::6810:5ef9
           2400:cb00:2048:1::6810:5ff9
           104.16.97.249
           104.16.95.249
           104.16.93.249
           104.16.94.249
           104.16.96.249

C:\>
```

Em relação ao padrão Unix, o servidor de nomes mais conhecido é o BIND, que é conjunto de softwares DNS, que contém um *daemon* servidor de nomes (*named*), uma biblioteca *resolver* (nosso "resolvedor") e outros programas. O Bind é mantido pela ISC (*Internet Software Consortium* - <<https://www.isc.org/downloads/bind/>>).

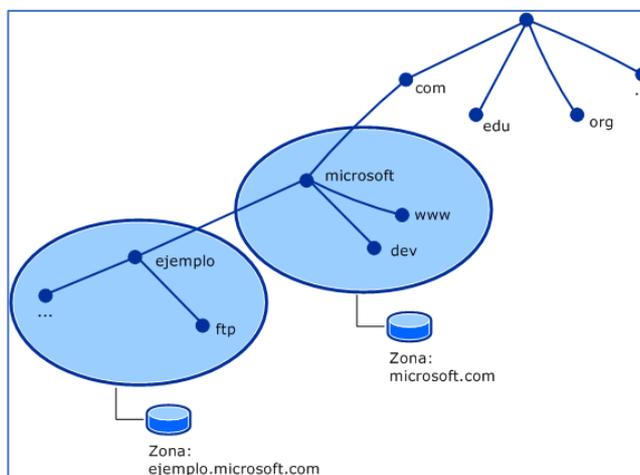
O arquivo de configuração para o resolvedor é o **resolv.conf** e fica localizado em /etc. Trata-se de um arquivo em texto plano usualmente criado pelo administrador ou por aplicações que gerenciam tarefas de configuração.

O espaço de nomes do DNS é dividido em zonas não superpostas. Cada zona está associada a um ou mais servidores de nomes, que mantêm o banco de dados para a zona. A figura abaixo mostra tal conceito:

---

<sup>2</sup> Responsável por armazenar consultas recentes, respondendo ao solicitante diretamente.





Fonte: <<http://un-newbie.blogspot.com.br/2014/03/introduccion-transferencia-de-zona-y.html>>.

Os **registros de recursos (RRs)** são o banco de dados do DNS. São compostos por tuplas de cinco campos: <nome\_domínio, tempo\_vida, classe, tipo, valor>, descritos abaixo:

- Nome: chave de pesquisa primária para atender as consultas;
- Tempo\_vida (TTL): tempo que deve permanecer em *cache* (em segundos);
- Classe: geralmente IN (Internet);
- Tipo: SOA, A, AAAA etc. (tabela a seguir);
- Valor: número, nome de domínio ou *string* ASCII.

Tipo	Significado	Valor
SOA	Início de autoridade ( <i>Start of Authority</i> ).	Parâmetros para essa zona.
A	Endereço IPv4.	Inteiro de 32 bits.
AAAA	Endereço IPv6.	Inteiro de 128 bits.
MX	Troca de mensagens de e-mail.	Prioridade, domínio disposto a aceitar e-mails.
NS	Servidor de nomes.	Nome de um servidor para este domínio.
CNAME	Nome canônico ( <i>alias</i> = apelido).	Nome de domínio.
PTR	Ponteiro (usado para o DNS reverso <sup>3</sup> )	Nome alternativo de um end. IP.
SPF	Estrutura de política do transmissor.	Codificação de texto da política de envio de mensagens de e-mail.
SRV	Identifica computadores que hospedam serviços específicos.	Host que o oferece.
TXT	Informações sobre um servidor, rede, <i>datacenter</i> etc.	Texto ASCII com descrições.

<sup>3</sup> Envia um endereço IP como consulta e recebe o nome como resposta.



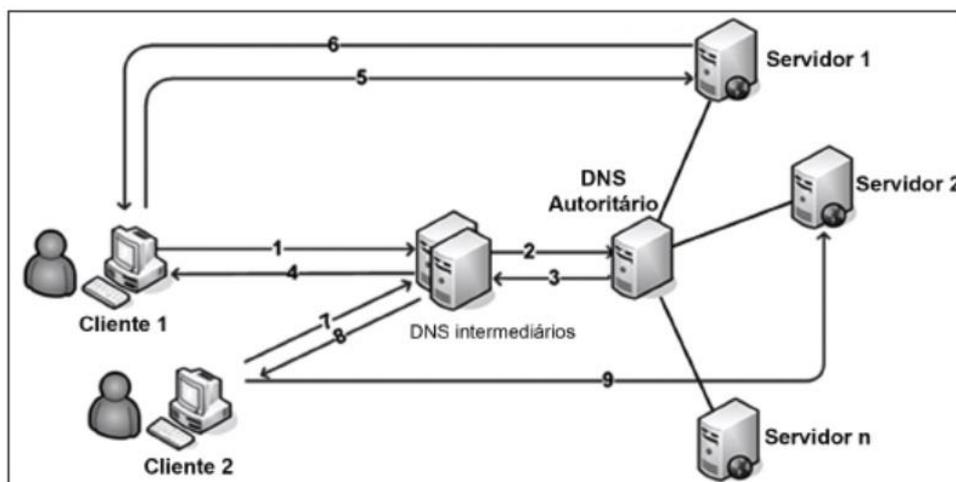
Outro conceito cobrado em provas de concurso é a resposta autoritativa ou não-autoritativa. Vejamos suas definições:

Um servidor DNS autoritativo possui autoridade sobre um nome de domínio. O DNS autoritativo dita qual será o apontamento da tabela de DNS do seu sítio. Uma **resposta autoritativa**<sup>4</sup> de um servidor é a garantia de estar atualizada, enquanto uma **resposta não-autoritativa** pode estar desatualizada (*cache* com informações antigas, por exemplo). Existe um percentual elevado de respostas não autoritativas que estão perfeitamente corretas, casos em que mudanças de endereçamento são raros.

Servidores primários e secundários são autoritativos para os seus domínios, porém não o são sobre informações a respeito de outros domínios mantidas em *cache*. **Servidores caching-only** nunca são autoritativos, mas possuem a vantagem de reduzir a quantidade de tráfego DNS na rede.

Uma política que pode ser adotada para equilibrar as vantagens de cada técnica é colocar um servidor secundário ou *caching-only* em cada segmento de rede ou subrede. É admissível uma máquina ser servidora primária para um domínio e servidora secundária para outros domínios.

Para não sobrecarregar servidores, existe uma abordagem popular e simples: o **balanceamento** de carga por **DNS**. Considere a figura abaixo e o passo a passo mostrado na sequência.



Fonte: Prova FCC/ELETROSUL - 2016.

1. Cliente 1 tenta acessar o *site*, é realizada uma pesquisa no DNS local para determinar qual é o endereço IP correspondente;
2. O pedido de endereço chega ao servidor de DNS autoritário do domínio;
3. A primeira vez que esta consulta é feita, o servidor DNS remoto pode retornar todos os registros de endereços que ele tem para o site;

4. O servidor DNS local, em seguida, determina o endereço de registro para retornar ao cliente;
5. Se todos os registros são retornados, o cliente utilizará o primeiro que lhe é atribuído;
6. O servidor responde ao cliente e atende ao pedido;
7. A cada pedido, o algoritmo Round Robin roda os endereços e retorna pela ordem em que eles estão;
8. Cada consulta DNS irá resultar em um cliente usando um endereço diferente;
9. Esta rotação de endereços irá distribuir pedidos para os servidores.

No Linux o servidor DNS mais utilizado e cobrado em concursos é o BIND (*Berkeley Internet Name Domain*), que se encontra na versão 9 (por isso o nome do pacote é bind9). Para instalar o bind9 e ferramentas relacionadas pode-se utilizar o gerenciador de pacotes APT (ou o apt-get), da seguinte forma:

```
$ sudo apt-get install bind9 bind9utils bind9-doc dnsutils
```

É importante saber quais são e para que servem os arquivos de configuração. Um dos mais cobrados em concursos é o **/etc/resolv.conf**, que identifica os locais dos computadores de servidor DNS. Nesse arquivo deve ser indicado o nome de domínio DNS da rede, e qual o servidor DNS irá resolver as consultas de nomes. Um exemplo é mostrado a seguir.

```
domain ti.teste.com.br  
search ti.teste.com.br  
teste.com.br  
nameserver 192.168.1.1  
nameserver 192.168.2.1
```

Cada parâmetro é colocado em uma linha. As três palavras-chave normalmente usadas são:

- *domain*: especifica o nome do domínio local;
- *search*: lista de pesquisa para a procura do nome de servidor, normalmente determinado pelo domínio do servidor local;
- *nameserver*: Endereço IP do servidor de nomes que o resolvedor deve pesquisar, utilizados na ordem em que estão listados;

O arquivo resolv.conf também serve para especificar o tipo (nível) de serviço a ser executado pelo software, a saber:

- *resolver-only system*: não requer que o sistema local tenha um servidor DNS em execução, requer apenas o resolver;
- *caching-only server*: servidor que não possui uma cópia da tabela de zonas, ou seja é não-autoritativo, possuindo uma grande quantidade de registros em cache;



- *primary server (master server)*: onde todas atualizações manuais devem ser realizadas, é autoritativo;
- *secondary server*: busca informações de zona em um *master server*, é autoritativo.

O principal arquivo de configuração do Bind é o **/etc/bind/named.conf** (em versões antigas: /etc/named.conf). Por padrão o Bind já vem configurado para trabalhar como um servidor DNS de *cache*, que pode ser usado tanto localmente quanto por outros dispositivos da rede local. Dentro do arquivo de configuração é possível encontrar entradas tais como:

<pre>zone "." {     type hint;     file "/etc/bind/db.root"; };  zone "localhost" {     type master;     file "/etc/bind/db.local"; }; zone "127.in-addr.arpa" {     type master;     file "/etc/bind/db.127"; };</pre>	<pre>zone "0.in-addr.arpa" {     type master;     file "/etc/bind/db.0"; };  zone "255.in-addr.arpa" {     type master;     file "/etc/bind/db.255"; };</pre>
---	---

Pode-se ver que cada uma das seções indica a localização de um arquivo, onde vai a configuração referente a ela. Por exemplo, na primeira seção ("zone ".") é indicado o arquivo "/etc/bind/db.root", que contém os endereços dos 14 root servers, que o Bind contactará na hora de resolver os domínios.

Esta configuração vem incluída por padrão e não deve ser alterada, a menos que seja um usuário experiente. Ao configurar o servidor DNS algumas novas zonas são incluídas (novas seções de configuração), contendo os domínios que se deseja configurar.

O serviço referente ao Bind pode se chamar "bind" ou "named", de acordo com a distribuição. Nos derivados do Debian o controle do serviço é realizado através do comando **/etc/init.d/bind9** (ou /etc/init.d/bind para a versão 8), enquanto nas distribuições derivadas do Red Hat utiliza-se o comando **service named**:

```
# /etc/init.d/bind9
restart
ou
# service named restartc
```



Servidores DNS públicos (*forwarders*), como por exemplo o Google Public DNS, podem ser configurados no arquivo **named.conf.options**. Nesse arquivo também pode ser definida a lista de controle de acessos (ACL). Um exemplo é mostrado abaixo.

<pre>options {     directory "/var/cache/bind";     // Troca a porta entre servidores DNS     query-source address * port *;     forward only;     forwarders { 192.168.1.1; };     auth-nxdomain no;     interface-interval 0;     // Escuta apenas em interfaces locais     listen-on-v6 { none; };     listen-on { 127.0.0.1; 192.168.0.1; };</pre>	<pre>// Não transfere info de zona a DNS secundário     allow-transfer { none; };     allow-query { internals; };     allow-recursion { internals; };     // Não torna versão pública do BIND     version none; };</pre>
--	--

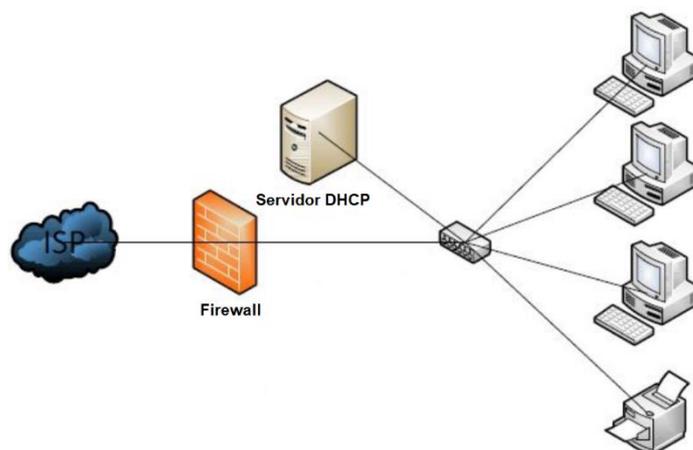
Para finalizar é importante destacar um arquivo útil para máquinas que são acessadas frequentemente: o **/etc/hosts**. A inclusão de um computador nesse arquivo dispensa a consulta a um servidor DNS para obter um endereço IP. Ou seja, a tradução ocorre automaticamente ao ler tal arquivo, sem ter que consultar o serviço de DNS. Um exemplo é mostrado abaixo.

- |                                  |                  |            |
|----------------------------------|------------------|------------|
| # Aqui são colocados comentários |                  |            |
| # End. IP                        | Nome do host     | Alias      |
| (opcional)                       |                  |            |
| 127.0.0.1                        | localhost        | localmente |
| 192.168.0.1                      | www.teste.com.br | teste      |

## DHCP (Dynamic Host Configuration Protocol)

O **DHCP** (*Dynamic Host Configuration Protocol*) é um protocolo que tem a função de configurar os endereços IP dos computadores de uma rede de forma dinâmica. Ou seja, deve haver pelo menos um servidor DHCP pré-configurado para receber solicitações de computadores que não possuem endereço IP (clientes), o servidor verificar qual endereço IP disponível e envia ao solicitante. Abaixo podemos ver um exemplo:





No cenário mostrado, os três PCs e a impressora estão configurados como clientes DHCP. Cada um deles deve enviar uma mensagem *broadcast* (mensagem a todos, pois não sabe quem é o servidor DHCP), o servidor DHCP recebe a mensagem, verifica qual endereço IP está disponível (ou até mesmo reservado para o solicitante), oferece esse endereço ao cliente e, por fim, o cliente aceita (comunicando o servidor que vai utilizar tal endereço IP).

Algumas informações que devem ser configuradas no servidor:

- Escopo: intervalo de endereços IP que estarão disponíveis para atribuição automática. Também pode se referir ao intervalo que não será distribuído. Ex.: endereços IP disponíveis = 192.168.1.100 a 192.168.1.150;
- Máscara de rede: usada para fazer a divisão da rede de computadores. Uma rede classe C possui a máscara 255.255.255.0;
- Gateway: dispositivo que serve para interligar a rede local com a Internet, ex. na figura: o elemento central (pode ser um modem/roteador), que liga os cinco dispositivos da rede local (servidor + 3 PCs + impressora) à Internet (passando por um firewall), ex.: 192.168.1.1;
- DNS: endereço do servidor DNS a ser consultado, ex.: 8.8.8.8 (esse é o servidor DNS do Google).

**Importante:** o servidor DHCP precisa de pelo menos uma interface de rede configurada com endereço IP fixo!

O `dhcpd` é o servidor DHCP no Linux. Ele é útil, por exemplo, em uma máquina agindo como um roteador em uma rede local.

Obs.: `dhcpd` (*daemon* do servidor DHCP) não é o mesmo que `dhcpcd` (*daemon* do cliente DHCP).

### Instalação:

Deve-se instalar o `dhcp`, disponível nos repositórios oficiais.

### Uso:



O dhcpd inclui dois arquivos: dhcpd4.service e dhcpd6.service, que podem ser usados para controlar o *daemon*. Eles iniciam o *daemon* em todas as interfaces de rede para IPv4 e IPv6, respectivamente.

### Configuração:

Deve-se atribuir um endereço IPv4 estático para a interface que se deseja usar (ex.: eth0).

O arquivo /etc/dhcpd.conf contém as configurações. Esse arquivo pode se parecer com:

```
option domain-name-servers 8.8.8.8,  
8.8.4.4;  
option subnet-mask 255.255.255.0;  
option routers 139.96.30.100;  
subnet 139.96.30.0 netmask  
255.255.255.0 {  
    range 139.96.30.150 139.96.30.250;  
}
```



1. (IBFC/Polícia Científica-PR - 2017) Servidores DNS (Domain Name Server) são responsáveis pela conversão do nome dos diversos servidores espalhados pela Internet para seu número IP e vice-versa. Servidores de DNS trabalham de forma colaborativa e hierárquica. Assinale a alternativa a que apresenta o nome dado aos servidores que se encontram no topo da hierarquia de DNS:

- A) Root Name Servers
- B) Main servers
- C) International Name Servers
- D) Controllers Servers
- E) Master Servers

### Comentários:



A hierarquia DNS é uma árvore invertida, ou seja, os servidores raiz (Root Name Servers) ficam no topo, logo abaixo ficam os TLDs (Top Level Domains), depois os de segundo nível e assim por diante. Portanto, a **alternativa A** está correta e é o gabarito da questão.

**Gabarito:** Letra A

2. (FCC/PRODATER - 2016) No Serviço de Nomes de Domínio – DNS existem diferentes tipos de servidores distribuídos hierarquicamente que armazenam informações também de forma hierárquica. Considerando o nome: [www.empresa.com](http://www.empresa.com), o domínio [.com](http://www.empresa.com) é gerenciado pelo servidor

- A) Global.
- B) PDR.
- C) Authoritative.
- D) TLD.
- E) Root.

#### Comentários:

Os domínios de mais alto nível são classificados em genéricos ("[.com](http://www.empresa.com)", "[.edu](http://www.empresa.edu)", "[.bradesco](http://www.empresa.bradesco)", etc.) ou códigos de países ("[.br](http://www.empresa.br)", "[.mx](http://www.empresa.mx)", "[.uy](http://www.empresa.uy)", "[.jp](http://www.empresa.jp)", etc.). Em inglês é conhecido como TLD (Top Level Domain) e os servidores DNS TLD gerenciam tais domínios. Abaixo podemos ver mais exemplos de TLDs genéricos e dois exemplos de códigos de países. Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D



## SERVIÇOS DE REDE – FTP E E-MAIL

Agora vamos focar no serviço de transferência de arquivos, utilizando o protocolo FTP, e no serviço de e-mail.

### FTP (File Transfer Protocol)

Como o próprio nome deixa claro, o FTP é um protocolo de transferência de arquivos. Trata-se de um protocolo padrão, independente de hardware. O FTP é baseado no protocolo de transporte TCP, o que garante a entrega dos pacotes. O servidor FTP utiliza as **portas 20** para os **dados** e a **21** para o **controle**. Vamos ver como ele funciona:

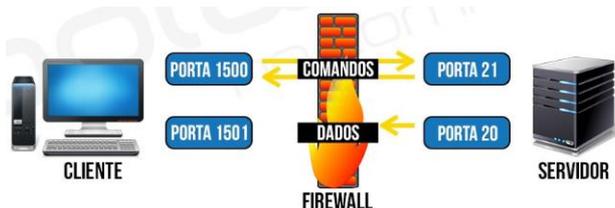
Um cliente realiza uma conexão TCP para a porta 21 do servidor. Essa conexão (conexão de controle) permanece aberta ao longo da sessão enquanto uma segunda conexão (conexão de dados), é estabelecida na porta 20 do servidor e em alguma porta do cliente (comunicada no diálogo entre cliente e servidor). A conexão de controle (porta 21) é utilizada para administração da sessão (comandos, identificação) entre cliente e servidor.

O servidor responde na conexão de controle com três dígitos de código de estado em ASCII com uma mensagem de texto opcional. Por exemplo, "200" ou "200 OK" significa que o último comando obteve êxito. Os números representam o número do código e o texto opcional representa as explicações ou parâmetros necessários. Uma transferência de arquivo em progresso, sobre uma conexão de dados, pode ser abortada utilizando uma mensagem de interrupção enviada sobre a conexão de controle.

O FTP pode ser executado em modo ativo ou passivo, os quais determinam como a conexão de dados é estabelecida. No **modo ativo**, o cliente envia para o servidor o endereço IP e o número da porta na qual ele irá ouvir e então o servidor inicia a conexão TCP. No exemplo abaixo vemos que o cliente informou ao servidor que a porta 1501 aguardará uma conexão vinda do servidor para receber os dados.



Mas e se houver um firewall que não tenha liberado a porta 1501? Olhe o que acontece:



Em situações onde o cliente está atrás de um firewall e inapto para aceitar entradas de conexões TCP, o **modo passivo** pode ser utilizado. O cliente envia um comando PASV para o servidor e recebe um endereço IP e um número de porta como resposta, os quais o cliente utiliza para abrir a conexão de dados com o servidor. Na figura abaixo podemos ver que o cliente solicitou uma conexão no modo passivo e o servidor "disse": "Vou abrir a porta 2345 e você pode se conectar nela para receber os dados". Então, nesse caso, a porta 20 não será utilizada.



A transferência de dados pode ser feita em qualquer um dos três modos a seguir:

- Modo fluxo: dado é enviado como um fluxo contínuo, liberando o FTP de fazer algum processamento. Todo processamento é deixado para o TCP. Nenhum indicador de fim de arquivo é necessário, a menos que o dado esteja dividido dentro de registros;
- Modo de bloco: o FTP quebra o dado dentro de vários blocos (bloco de cabeçalho, contagem de byte e campo de dado) e então passa-o para o TCP;
- Modo comprimido: dado é comprimido utilizando um algoritmo simples.

## Transferência de dados

Modo fluxo

Modo de bloco

Modo comprimido

O **acesso a servidores FTP** pode ocorrer de dois modos: através de uma interface ou através da linha de comando. Tanto usuários Linux como usuários Windows podem acessar através dos dois modos. O modo linha de comando está presente em qualquer distribuição Linux-like e Windows.

A partir de qualquer navegador credenciado (Internet Explorer, Chrome, Firefox, entre outros), também é possível acessar um servidor FTP digitando na barra de endereço:

ftp://[username]:[password]@[servidor]

ou

ftp://[username]:[password]@[servidor]:[porta]

Os **comandos** abaixo podem ser executados no FTP através da **linha de comando**. Os comandos do FTP podem ser abreviados, desde que não formem expressões ambíguas.



- !: executa o comando na máquina local;
- ?: semelhante a help;
- append: adiciona dados a um arquivo existente;
- ascii: configura o tipo de transferência de arquivos para ASCII;
- bell: emite um bip quando um comando é executado;
- binary: configura o tipo de transferência de arquivos para binário;
- bye: encerra a sessão FTP;
- cd: seguido de caminho/diretório muda para o diretório informado;
- delete: apaga um arquivo. Para mais de um arquivo utiliza-se mdelete;
- debug: estabelece a modalidade de depuração;
- dir: mostra o conteúdo do diretório servidor atual;
- disconnect: semelhante a bye;
- get: obtém um arquivo do servidor. Para mais de um arquivo utiliza-se mget;
- glob: seleciona a expansão para nomes de arquivo;
- hash: demonstra cada bloco do arquivo durante a transferência. Cada bloco compõe-se de 1024 bytes;
- help: lista sumariamente todos comandos disponíveis;
- literal: permite enviar comandos arbitrários;
- ls: mostra uma lista abreviada do conteúdo do diretório servidor. Para mais de uma pasta utiliza-se mls;
- mkdir: cria um diretório ou subdiretório no servidor;
- prompt: ativa/desativa o modo interativo;
- put: envia um arquivo ao servidor. Para enviar mais de um arquivo utiliza-se mput;
- pwd: mostra o diretório de trabalho;
- quit: finaliza a sessão FTP;
- quote: envia subcomandos do servidor FTP, como se encontram no servidor;
- recv: similar a get;
- remotehelp: solicita ajuda do servidor FTP remoto;
- rename: renomeia um arquivo;
- send: semelhante a put;
- status: obtém informações de estado do servidor;
- trace: demonstra o caminho percorrido pelo arquivo na transferência;
- type: especifica o tipo de representação;
- user: inicia a sessão no servidor;
- verbose: ativa/desativa a modalidade literal.

Seguindo a lógica adotada para servidores no Linux, o **servidor FTP** é executado através de um *daemon*, o **ftpd**.



## E-mail

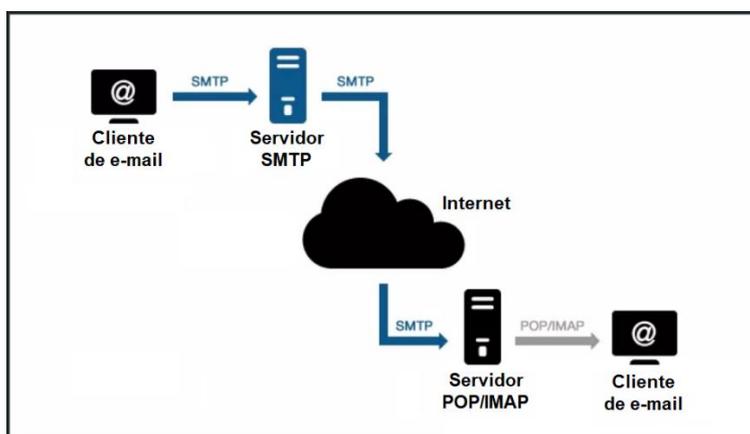
Antes de compreender como funciona o serviço (servidor) de e-mail no Linux, é importante entender a diferença dos 3 protocolos mais utilizados para esse serviço. Esqueça por um instante a utilização de HTTP ou HTTPS, no caso dos Webmails, vamos focar nos protocolos exclusivos para e-mail mesmo, vamos lá...

**SMTP** (*Simple Mail Transfer Protocol*): utilizado quando o e-mail é entregue a partir de um cliente de e-mail (Outlook, Thunderbird, entre outros) a um servidor de e-mail ou quando o e-mail é entregue a partir de um servidor de e-mail para outro servidor. Ou seja, se você enviar um e-mail a partir de um software cliente, ele utilizará o SMTP até o seu servidor e depois o seu servidor utilizará também o SMTP para entregar ao servidor destino, quando este ficará na caixa de e-mails do usuário destinatário. A **porta padrão** de um servidor SMTP é a **25**.

**IMAP** (*Internet Message Access Protocol*): também chamado de IMAP4, possui acesso a todas as pastas da conta de e-mail e deixa o *status* das mensagens igual tanto no servidor como no software e realiza a sincronia das mensagens, mantendo a conexão para que as alterações e mensagens novas recebidas sejam atualizadas quase que em tempo real. A **porta padrão** de um servidor IMAP é a **143**.

**POP** (Post Office Protocol): mais conhecido como POP3, esse protocolo baixa as mensagens do servidor e as armazena localmente no computador, não deixando uma cópia das mensagens no servidor (a menos que seja marcada a opção "deixar uma cópia no servidor") nas configurações do programa de e-mail. Esse protocolo tem acesso apenas à Caixa de Entrada, não conseguindo baixar nenhuma outra pasta da conta de e-mail. A **porta padrão** de um servidor POP3 é a **110**.

Abaixo podemos ver o relacionamento entre os 3 protocolos.



Podemos ver que a diferença principal é que o POP baixa as mensagens para o cliente local (permite que elas sejam armazenadas no servidor, mas não existe sincronia nesse processo).



O IMAP4 é a evolução do POP3, sincronizando a caixa de e-mails com o conteúdo do servidor, permitindo que um e-mail seja sincronizado entre vários locais sem perda de mensagens entre locais diferentes (salvo diretórios locais, todas as mensagens dentro do diretório do e-mail são sincronizadas).

No Linux, os servidores de e-mail mais comuns são o **sendmail** e o **postfix**. O postfix é considerado um MTA (Mail Transfer Agent) que se apresenta como alternativa ao sendmail, tendo como objetivo ser mais rápido, seguro e fácil de configurar que o sendmail, além de tentar manter a compatibilidade com ele. Vamos ver um pouco dos arquivos de configuração e funcionamento do postfix a seguir.

O postfix contém um arquivo denominado **main.cf**, no qual estão as configurações gerais do daemon, localizado no diretório **/etc/postfix**, pela instalação padrão. Abaixo serão demonstrados os principais parâmetros de configuração para ter-se um servidor de e-mail funcional. Nos exemplos abaixo será utilizado o nome do servidor como "mail.dominio.com.br" e o domínio "dominio.com.br", apenas para ilustração.

```
myhostname = mail.dominio.com.br
```

```
mydomain = dominio.com.br
```

O parâmetro "mydestination" deve receber os domínios que o servidor de e-mail é responsável. Por exemplo, mail.dominio.com.br, ftp.dominio.com.br, www.dominio.com.br, dominio2.com.br etc.:

```
mydestination = $myhostname, localhost.$mydomain, $mydomain, mail.$mydomain,  
www.$mydomain, dominio2.com.br
```

O parâmetro "mynetworks" serve para determinar as redes do provedor:

```
mynetworks = 192.168.0.0/16, 127.0.0.0/8
```

Também podem ser informados os domínios em um arquivo:

```
mynetworks = /etc/postfix/mynetworks
```

Pela configuração padrão do postfix, ele permite relay de e-mails das suas redes e de seus domínios. Se houver clientes que não fazem parte da rede e que necessitam utilizar o seu servidor de e-mail para o envio de mensagens, é possível criar um arquivo com a lista de clientes através do parâmetro "smtpd\_recipient\_restrictions" e habilitar o envio de mensagens, como mostrado abaixo:

```
smtpd_recipient_restrictions = permit_mynetworks
```



```
check_client_access hash:/etc/postfix/client_access
```

```
check_relay_domains
```

Exemplo do arquivo /etc/postfix/client\_access:

```
dial.amigos.com.br OK
```

```
10.0.0 OK
```

```
falcatrua.com.br REJECT
```

Esse parâmetro não consta na configuração padrão do postfix e pode ser incluído no final do arquivo de configuração main.cf.

A entrega de e-mails no **postfix** pode ser feita de diversas formas: enviando os e-mails para "/var/spool/mail/user" (padrão), no formato "Maildir/" (utilizado pelo qmail) entre outros.

**Iniciando o servidor:** # postfix start

Caso seja feita alguma alteração na configuração do servidor de e-mail, para atualizá-lo, basta digitar o comando: # postfix reload

Para desativar o servidor de e-mail: # postfix stop

Uma característica importante no Linux são os logs, que, na sua maioria, são armazenados em /var/log. Esse é o caso também dos logs relacionados aos servidores de e-mail, útil quando são necessárias informações sobre postfix, smtpd ou qualquer serviço relacionado ao e-mail que esteja sendo executado no servidor. Por padrão, esses logs ficam em **/var/log/maillog** ou **/var/log/mail.log**.



1. (FCC/Prefeitura de Manaus-AM - 2019) Um Assistente de TI foi incumbido de configurar o protocolo para ser utilizado na troca de mensagens entre dois servidores de e-mail. A escolha correta do protocolo para essa finalidade é:

A) IMAP.

B) POP3.



- C) SNMP.
- D) SMTP.
- E) POP4.

**Comentários:**

SMTP (Simple Mail Transfer Protocol): utilizado quando o e-mail é entregue a partir de um cliente de e-mail (Outlook, Thunderbird, entre outros) a um servidor de e-mail ou quando o e-mail é entregue a partir de um servidor de e-mail para outro servidor. Ou seja, se você enviar um e-mail a partir de um software cliente, ele utilizará o SMTP até o seu servidor e depois o seu servidor utilizará também o SMTP para entregar ao servidor destino, quando este ficará na caixa de e-mails do usuário destinatário. A porta padrão de um servidor SMTP é a 25. Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

**2. (FUNCAB/MDA - 2014) São exemplos de servidores de correio eletrônico do Linux:**

- A) postfix e sendmail.
- B) fetchmail e sendmail.
- C) pine e postfixe.
- D) mail e pine.
- E) samba e postfixe.

**Comentários:**

Alguns exemplos de servidores de e-mail no Linux: qmail, postfix e sendmail. Portanto, a **alternativa A** está correta e é o gabarito da questão.

**Gabarito:** Letra A



## GERENCIAMENTO DE PROCESSOS

As estruturas internas do kernel registram informações sobre cada processo, incluindo, entre outras:

- Mapa de espaço de endereços do processo;
- Estado atual (espera, parado, em execução etc.);
- Prioridade de execução;
- Informações sobre recursos;
- Informações sobre arquivos e portas de rede;
- Máscara de sinalização (quais sinais estão bloqueados);
- Proprietário do processo.



O **PID** (Process Id) é o número de identificação de um processo, atribuído na ordem em que são criados. O **PPID** é o PID do processo pai. **Não há chamada** de sistema que **crie** um **processo** no Linux, o que acontece é que um processo existente realiza seu clone - chamada de sistema **fork()** - e o processo clone pode trocar o programa por um diferente. O original é o pai e a cópia gerada é o filho.

O **UID** (User Id) é a identificação do usuário que criou e o **GID** (Group Id) é a identificação do grupo. Para definir a prioridade de agendamento (quanto tempo de CPU o processo recebe), existe um valor de **niceness** ("gentileza"). Esses valores vão de -20 a 19 (zero é o padrão), sendo -20 a maior prioridade e 19 a menor (Cuidado! É "invertido", assim mesmo!). No momento do clone do processo, o valor é herdado do processo pai.

O comando **nice** define a prioridade de um processo (padrão = 10) antes de executar o programa, enquanto o comando **renice** altera a prioridade de processo já em execução. O proprietário pode aumentar o valor (diminuir a prioridade), mas não pode o contrário! O root pode tudo, claro! Vamos a alguns exemplos:



\$ nice -n 5 ~/bin/teste → Reduz prioridade (eleva nice) por um fator de 5:  $10 + 5 = 15$ ;

\$ sudo renice -6 7943 → Configura o valor de nice como -6;

\$ sudo renice 8 -u evandro → Configura o valor de nice como 8 para processos de evandro.

Como vimos anteriormente, a chamada de sistema **fork()** clona um processo, criando um filho:

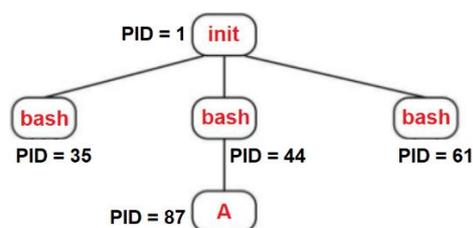
- Retorna 0 para o filho;
- O pai recebe o PID do filho.

Geralmente em seguida o filho executa um novo programa, ou seja, muda o texto do programa e os segmentos de dados e pilha são reiniciados, recebe um PID distinto e suas próprias informações contábeis. Dessa forma cada um sabe o seu papel.

Vários processos são criados na inicialização, sendo o **init** é o mais notável, sendo o responsável por:

- Executar os scripts de inicialização;
- Ser o "paizão" dos processos, pois todos são descendentes dele, exceto os criados pelo kernel;
- Chamar `exit_` quando um processo se completa (`exit` notifica o kernel de que ele está pronto para expirar, fornecendo um código - motivo, 0 = término normal);
- Receber os processos órfãos.

Em relação à **hierarquia de processos** existe apenas 1 pai e 0 ou mais filhos. Por exemplo, no Linux o processo `init` (PID = 1) é o primeiro a ser executado, logo após o carregamento do Kernel. A função dele é controlar todos os outros processos que são executados no computador. Digamos que a partir dele sejam abertos 3 *shells* (`bash`) e a partir de um deles seja executado um programa "A". Abstraindo a existência de outros processos, a hierarquia descrita ficaria assim (PIDs inventados, com exceção do `init`):



**Sinais** podem ser enviados aos processos (interrupções em nível de processo) de algumas formas:

- Comunicação entre processos;
- Extinguir, interromper ou suspender processos (ex. no terminal: CTRL+C ou CTRL+Z);
- Enviados com o comando **kill** (não apenas "mata", pode enviar outros sinais);
- Enviados pelo *kernel* (alguma infração, ex.: divisão por 0);



- Enviados pelo *kernel* (situação "interessante", ex.: morte de processo filho).

Vamos enfatizar o seguinte, o comando **kill** envia qualquer sinal (não é porque o nome do comando é "matar" que ele serve apenas para isso, porém geralmente é o uso mais frequente). Sintaxe:

```
kill [-SINAL] PID
```

O comando **killall** tem o mesmo propósito de **kill**, mas é especificado o nome do processo (se houver mais de um processo com o mesmo nome, o sinal é enviado a todos):

```
killall [-SINAL] NOME_PROC
```

Alguns dos sinais são (o padrão é TERM):

#	Nome	Descrição
1	HUP	Suspender
2	INT	Interromper
3	QUIT	Abandonar
9	KILL	Matar
15	TERM	Término do software

O sinal pode ser especificado pelo nome ou pelo seu código equivalente. Por exemplo, para enviar o sinal KILL para um processo com PID 4321:

```
kill -SIGKILL 4321
```

```
kill -9 4321
```

Para enviar o sinal TERM para um processo com nome abcd:

```
killall -SIGTERM abcd
```

Os **estados dos processos** em sistemas Linux podem ser:

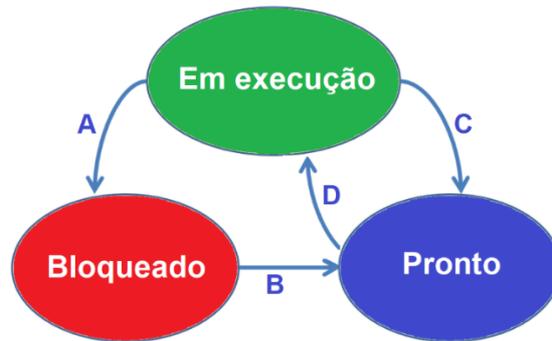
Estado	Significado
Executável	Pode ser executado (pronto).
Dormente	Aguardando algum recurso (ex.: E/S).
Zumbi	Tentando se destruir.



Parado

Suspensão (sem permissão para ser executado).

*Daemons* e *shells* interativos passam a maior parte do tempo “dormindo” (E/S ou conexões de rede). Vamos ver um desenho bastante utilizado em aulas de sistemas operacionais (não especificamente o Linux):



O estado executável (pronto) é quando o processo está aguardando para utilizar a CPU.

O estado dormente (bloqueado) é quando estava em execução e fez uma chamada de sistema que não pôde ser completada imediatamente (ex.: ler um arquivo).

O estado zumbi é quando o processo terminou a execução, mas ainda não teve o seu status coletado.

O estado parado é quando está proibido de executar (sinais STOP ou TSTP, sinal CONT para reiniciar).

## Comandos para Monitoramento de Processos

Para o **monitoramento de processos** existem basicamente três comandos, mas o mais conhecido é o ps. Vejamos os comandos e alguns argumentos já cobrados em provas de concurso:

- **ps**: pode exibir PID, UID, prioridade, terminal de controle de processos, memória consumida, estado atual
    - a: mostra todos os processos existentes;
    - u: exibe o nome do usuário que iniciou determinado processo e a hora que isso ocorreu;
    - x: exibe os processos que não estão associados a terminais;
    - l: exibe mais campos no resultado;
    - e: exibe as variáveis de ambiente relacionadas aos processos;
    - f: exibe a árvore de execução dos processos.
- ➔ Visão geral sobre o sistema, bastante utilizado na "vida real" e cobrado em provas:  
ps aux



- **top**: mostra um resumo dos processos ativos e uso de recursos, atualizando regularmente (processos mais ativos aparecem no topo, padrão de atualização = 10 segundos);
- **jobs**: mostra os processos que estão parados ou rodando em segundo plano.

Processos em segundo plano são iniciados usando o símbolo “&” no final da linha de comando ou através do comando bg:

```
$ ./teste &
```

```
$ bg ./teste
```

Quando um comando é executado em segundo plano, o terminal fica liberado para que outros comandos sejam digitados.



1. (CESPE/EBSERH - 2018) No sistema operacional Linux, é possível alterar a prioridade de um processo já iniciado com o uso do comando nice.

**Comentários:**

O nice é utilizado para executar um programa e o renice é utilizado para alterar a prioridade do processo já em execução. Portanto, a questão está **errada**.

**Gabarito:** Errada

2. (CESPE/TJ-SE - 2014) Alguns programas podem apresentar problemas que resultem no travamento do sistema operacional, o que pode ser resolvido, no Linux, por meio do comando Kill, que finaliza o processo, funcionalidade que pode ser acessada por meio de outro terminal.

**Comentários:**

O comando kill serve para enviar um sinal para um processo através de seu PID. Esse sinal pode ser para finalizar um processo, entre outros. A questão não fala que o kill serve apenas para finalizar, diz que é possível finalizar um processo com ele. Portanto, a questão está **correta**.

**Gabarito:** Correta



## MONTAGEM DE VOLUMES

Todos arquivos acessíveis em um sistema UNIX são arranjados em uma grande árvore, com a raiz "/". Tais arquivos podem estar espalhados em diversos dispositivos (HDs, SSDs, pen drives, DVDs etc.). O comando **mount** serve para anexar o sistema de arquivos encontrado no dispositivo (ou uma partição) à grande árvore de arquivos. Algumas distribuições montam automaticamente e o ponto de montagem padrão é o diretório **/mnt**.

O arquivo **/etc/fstab** possui informações sobre os sistemas de arquivos que o sistema pode montar, e o arquivo **/etc/mtab** possui uma lista dos dispositivos montados (através do comando **mount**).

### Comandos para Montagem e Desmontagem

O comando **mount**, sem parâmetros, mostra quais sistemas de arquivos estão montados:

```
[root@localhost ~]# mount
root on / type 9p (rw,relatime,dirsync,access=client,trans=virtio)
devtmpfs on /dev type devtmpfs (rw,relatime,size=125948k,nr_inodes=31487,mode=755)
none on /proc type proc (rw,relatime)
none on /sys type sysfs (rw,relatime)
devpts on /dev/pts type devpts (rw,relatime,mode=600,ptmxmode=000)
[root@localhost ~]#
```

A forma básica do comando **mount** possui dois parâmetros:

- Dispositivo contendo o sistema de arquivos a ser montado (ex.: **/dev/sda3**);
- Ponto de montagem (ex.: **/teste**).

### comando mount

Dispositivo contendo  
o sistema de arquivos  
a ser montado

Ponto de montagem



Assim, temos, por exemplo:

```
[root@estrategia ~]# mount /dev/sda3 /teste
```

Obs.: /teste deve existir antes de aplicar o comando!

Geralmente o tipo de sistema de arquivos é detectado automaticamente, mas pode-se especificar o tipo com o parâmetro **-t** (ex.: sistema de arquivos EXT4):

```
[root@estrategia ~]# mount -t ext4 /dev/sda3 /teste
```

Alguns exemplos de sistemas de arquivos suportados são cifs, ext2, ext3, ext4, hfs, iso9660, nfs, reiserfs, vfat, xfs, entre outros.

O parâmetro **-o** possibilita definir opções, separadas por vírgulas, como por exemplo:

- ro: somente leitura;
- rw: leitura e escrita;
- exec: permite a execução de binários;
- noexec: não permite a execução de binários;
- remount: tenta remontar um sistema de arquivos já montado.

Exemplo:

```
mount -o remount,rw,noexec /dev/sda3 /teste
```

O parâmetro **-a** monta todos os sistemas de arquivos listados no arquivo fstab (exceto as linhas com a palavra-chave noauto). Importante ressaltar que os programas leem do fstab, porém não escrevem nele. O parâmetro -a pode ser utilizado para montar "tudo" sem realizar um boot.

O comando **umount** retira o sistema de arquivos da árvore hierárquica de arquivos. Não há a desmontagem caso o sistema de arquivos esteja ocupado (com arquivos abertos, swap em uso etc.):

```
umount /teste
```



1. (CESPE/TCE-PA - 2016) A opção `mount -t vfat` permite fazer o becape dentro de uma partição de disco DOS/Windows, caso em que a hierarquia de diretório aparece como parte do sistema de arquivos Linux.

**Comentários:**

Vimos que `-t` é o parâmetro para especificar o tipo de sistema de arquivos e que "vfat" especifica a família FAT, a qual é utilizada em sistemas DOS/Windows. Portanto, a questão está **correta**.

**Gabarito:** Correta

2. (CESPE/TCE-RN - 2015) O Linux tem a capacidade de montar sistemas de arquivos do dispositivo `/dev/sdc3`, que tem o sistema de arquivos do tipo ext4 e, no ponto de montagem `/teste`, deve ser utilizado o comando `mount -t ext4 /dev/sdc3 /teste`.

**Comentários:**

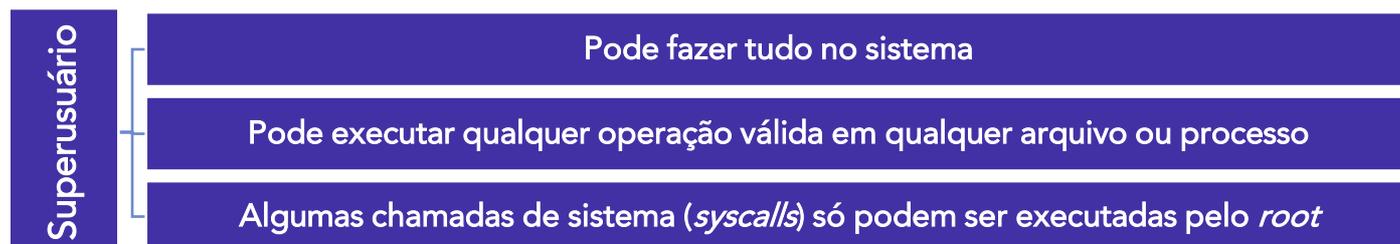
Semelhante à questão anterior. Portanto, a questão está **correta**.

**Gabarito:** Correta



## SUPERUSUÁRIO

O **superusuário** é aquele que pode fazer tudo no sistema! Pode executar qualquer operação válida em qualquer arquivo ou processo. Algumas chamadas de sistema (*syscalls*) só podem ser executadas pelo *root* (nome padrão para o superusuário, mas pode ter outros).



O **root** é uma conta com **UID=0**, sendo possível modificar o nome de usuário dessa conta ou criar contas com UID=0 (ações não recomendadas, porém possíveis).

Alguns exemplos de operações restritas, que somente o *root* pode realizar:

- Modificar o diretório raiz;
- Criar arquivos de dispositivo;
- Configurar o relógio;
- Aumentar limites de uso de recursos e as prioridades de processos;
- Definir o nome do host;
- Configurar interfaces de rede;
- Abrir portas de rede privilegiadas (< 1024);
- Desligar o sistema.

Por padrão, o diretório home do *root* é **/root** e **/sbin** é o diretório que armazena os binários de sistema importantes que são utilizados usados pelo *root*.

Quando um usuário "**comum**" utiliza o shell, aparece **\$** no prompt, enquanto para o **root** aparece **#**. Isso é importante para verificar em uma questão na prova, para saber se quem está executando é um usuário comum ou o *root*! Abaixo podemos ver que o *root* está utilizando o shell. Claro que está configurado para aparecer o nome do usuário à esquerda, mas mesmo que não estivesse, poderíamos ver o caractere "#", que indica ser o *root*.

```
root@evandro-HP: ~/DHAV2
Arquivo Editar Ver Pesquisar Terminal Ajuda
Gerando arquivo: /media/evandro/CASOS-1TB/saida-imagem012/canal4/dir0/arq20180505175052_000000011263.h264 de tamanho: 525338
Gerando arquivo: /media/evandro/CASOS-1TB/saida-imagem012/canal7/dir0/arq20180505175044_000000011264.h264 de tamanho: 393507
Gerando arquivo: /media/evandro/CASOS-1TB/saida-imagem012/canal7/dir0/arq20180505175046_000000011265.h264 de tamanho: 92548
Processou 11266 arquivos
root@evandro-HP:~/DHAV2#
```



## Comandos

Para um usuário comum se tornar root, pode-se utilizar o comando **su** (*substitute user*). Note que o comando su serve para "substituir" um usuário e não precisa ser necessariamente o root! Mas na prática, geralmente se utiliza para se tornar o root mesmo.

O comando **su** sem argumentos solicita a senha de *root* e inicia um shell de *root*. Não registra nenhum comando executado como root, mas cria log de quem e quando se tornou root, o que é útil para auditoria de alguma coisa errada realizada no período. Para substituir por outras identidades, a sintaxe é: **su nomeUsuario** → pede a senha do nomeUsuario. É útil para testar em nome do nomeUsuario.

Se duas ou mais pessoas souberem a senha do root, pode acontecer uma situação em que alguém faz o login como root, "faz coisas que não deveria" e ninguém saberá quem foi. Por questões de segurança há a possibilidade de desabilitar o *login* de *root*.

O comando **sudo** permite a usuários comuns obter privilégios de outro usuário, em geral o root, para executar tarefas específicas dentro do sistema de maneira segura e controlável pelo administrador. Ao ser **executado**, há uma consulta ao arquivo **/etc/sudoers**, que possui os usuários e comandos habilitados em cada host. É solicitada a senha do próprio usuário que executou o sudo e, na sequência, é executado o comando que se encontra após o sudo. Importante salientar que é mantido um log dos comandos executados, hosts, usuários, diretório, data/hora etc. O exemplo mais comum é sem especificar um usuário:

```
$ sudo apt-get install abc
```

No exemplo acima, como não é especificado um usuário, então o "apt-get install abc" será executado como root, após o usuário ter digitado a sua senha. Pode haver configuração de tempo sem ter que digitar a senha novamente (até 5 minutos). Vamos ver mais dois exemplos:

```
$ sudo -u evandro /home/evandro/prog → executa o "prog" em nome de "evandro".
```

```
$ sudo -g prof /home/teste/prog → seta o grupo primário para "prof" no momento da execução de "prog".
```



1. (CESPE/ABIN - 2018) O Linux não impede a alteração do nome do superusuário, nem a criação de contas com UID igual a 0, embora essas ações não sejam recomendadas.

**Comentários:**

Como vimos na aula, não é recomendado fazer isso, mas é possível alterar o nome do root para qualquer coisa, e, também é possível criar novas contas de superusuário (com UID = 0). Portanto, a questão está **correta**.

**Gabarito:** Correta

2. (IBFC/EBSERH - 2016) O comando sudo do sistema operacional Linux é muito poderoso, permitindo que usuários comuns obtenham privilégios de super usuário. Por questões de segurança, o administrador precisa definir no arquivo \_\_\_\_\_, quais usuários podem executar sudo, em quais computadores podem fazê-lo e quais comandos podem executar através dele. Assinale a alternativa que complete correta e respectivamente a lacuna:

- A) /etc/sudoers
- B) /root/sudouser
- C) /root/sudoers
- D) /usr/sudouser
- E) /etc/sudouser

**Comentários:**

Pense o seguinte: os arquivos de configuração geralmente ficam no diretório "/etc". E para executar o sudo existe um arquivo com as configurações de quem e do que pode ser executado com o sudo, que é o arquivo "sudoers". Em inglês, quem caminha ("walk") é um "walker", então quem faz um "sudo" é um "sudoer", assim fica mais fácil lembrar 😊. Portanto, a **alternativa A** está correta e é o gabarito da questão.

**Gabarito:** Letra A



## USUÁRIOS, GRUPOS, PERMISSÕES DE ACESSO

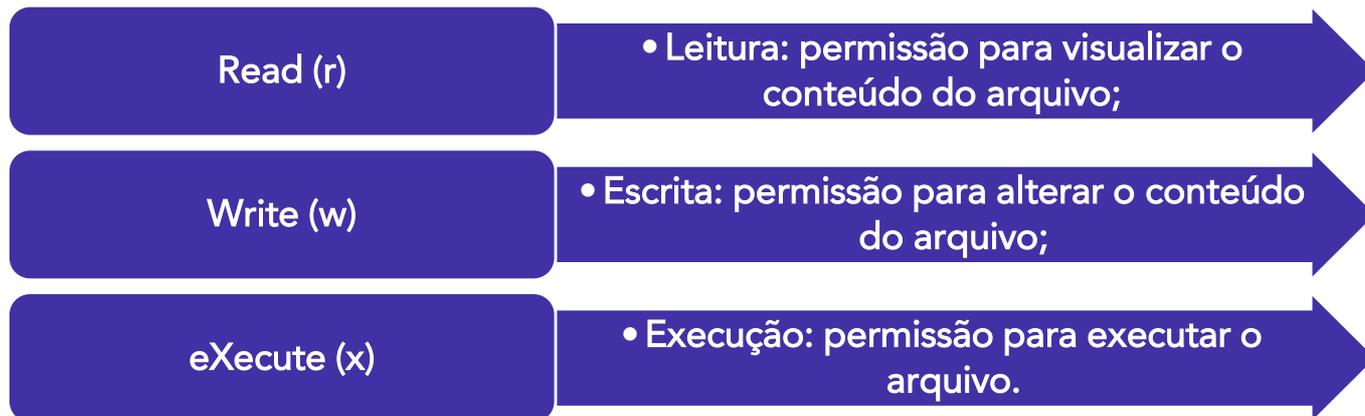
Um novo usuário pode ser adicionado utilizando-se ferramentas, como **adduser** ou **useradd**. A partir desse momento o usuário criado é adicionado nos arquivos `/etc/passwd` e `/etc/shadow`. Os arquivos recebem um número **UID** referente à identificação do **usuário (dono)**, **GID** referente ao seu **grupo**, entre outros dados.

Os números UID e GID variam de 0 a 65535 (dependendo do sistema, o valor limite pode ser maior). No caso do **usuário root**, esses valores são **sempre 0 (zero)**. Assim, para fazer com que um usuário tenha os mesmos privilégios que o root, é necessário que seu UID e GID sejam iguais a 0. Isso informa ao sistema que o usuário em questão é um superusuário.

Cada arquivo/diretório no Linux possui um conjunto de **permissões de acesso** e propriedades em **três níveis**: usuário dono do arquivo (**U**), grupo proprietário (**G**) e qualquer outro usuário que não se encaixe nos níveis anteriores (**O**).

Os **tipos de permissão** são:

- Read (**r**) - Leitura: permissão para visualizar o conteúdo do arquivo;
- Write (**w**) - Escrita: permissão para alterar o conteúdo do arquivo;
- eXecute (**x**) - Execução: permissão para executar o arquivo.



Dessa forma definimos a seguinte ordem:

U			G			O		
r	w	x	r	w	x	r	w	x

Existem também algumas **permissões especiais**:

- **setuid**: privilégio do dono e não de quem executou, ex.: `passwd` executado pelo evandro → privilégio de root, mas quem executou foi o usuário evandro!



- **setgid**: tem efeito tanto em arquivos como em diretórios. Para um arquivo, executa com privilégios do grupo do usuário dono. Para um diretório, o grupo dos arquivos criados dentro do diretório será o mesmo do diretório pai. Abaixo podemos notar um "s" no lugar do "x" do grupo. Isso demonstra que o setgid está com valor 1.

```
$ ls -ld teste  
drwxrwsr-x. 2 egdoc egdoc 4096 Nov 1 17:25 teste
```

- **sticky bit**: quando usado em um diretório, todos os arquivos dentro dele poderão ser modificados somente por seus donos. Exemplo típico: "/tmp", um diretório temporário onde todos podem colocar arquivos, mas cada um pode modificar o seu. Abaixo podemos notar um "t" no lugar do "x" dos outros. Isso demonstra que o sticky bit está com valor 1.

```
$ ls -ld /tmp  
drwxrwxrwt. 14 root root 300 Nov 1 16:48 /tmp
```

Vamos rever nossa figura do "UGO" com as permissões especiais:

			U			G			O		
<u>uid</u>	<u>gid</u>	<u>sticky</u>	r	w	x	r	w	x	r	w	x

Analisando apenas as permissões especiais, se o setuid estiver ativo temos 100 = 4 (em decimal). Se tivermos o setgid ativo, temos 010 = 2 e se o sticky bit estiver ativo, temos 001 = 1.

Para **visualizar as permissões** é comum utilizarmos o argumento "-l" no comando "ls", pois esse argumento lista arquivos com informações adicionais, incluindo colunas com as permissões, nome do usuário e grupo do dono:

```
[root@localhost etc]# ls -l teste  
-rw-r--r-- 1 root root 6 Jul 22 18:43 teste  
[root@localhost etc]#
```

Na figura acima podemos ver que se trata de um arquivo (o primeiro caractere é "-" = arquivo, se fosse diretório seria "d"). Depois vemos as permissões do Usuário (rw-), Grupo (r--) e Outros (r--). Não vemos permissões especiais.

## Modificando as Permissões

Ok, e se eu quiser **modificar as permissões** depois de ter criado um arquivo? Aí temos o comando **chmod**. Vamos ver algumas formas de utilizá-lo:



`chmod u=rwx, g=rx, o=x nome_arquivo` → modifica as permissões do “nome\_arquivo” para:

- usuário = completo (*read, write, execute*);
- grupo = leitura e execução (*read, execute*);
- outros = apenas execução (*execute*).

`chmod 0751 /home/evandro/arquivo.txt` → aqui temos um exemplo utilizando números em octal e temos a presença das permissões especiais (o primeiro número). Como o primeiro número é 0, fica fácil, pois em binário ficaria 000 (setuid = 0, setgid = 0, sticky bit = 0). Agora vamos ao UGO:

- usuário = 7 = 111 = completo (*read, write, execute*);
- grupo = 5 = 101 = leitura e execução (*read, execute*);
- outros = 1 = 001 = apenas execução (*execute*).

Acabamos de ver que os comandos acima são equivalentes em termos das permissões concedidas!

Um último exemplo:

`chmod +x teste` → dá a permissão de execução ao arquivo “teste”, mas para quem? U? G? O? Como não está explícito para quem é, todos recebem o “+x”. Veja abaixo antes e depois de aplicar o “`chmod +x teste`”. Tem um “`ls -l teste`” antes e depois para podermos verificar que realmente tanto o usuário, como o grupo e os outros receberam a permissão de execução (x).

```
[root@localhost etc]# ls -l teste
-rw-r--r-- 1 root root 6 Jul 22 18:43 teste
[root@localhost etc]# chmod +x teste
[root@localhost etc]# ls -l teste
-rwxr-xr-x 1 root root 6 Jul 22 18:43 teste
[root@localhost etc]#
```

## Modificando o Dono e/ou o Grupo

Agora vamos ver como modificar o dono ou o grupo.

Embora o comando **chown** signifique *change owner*, ele permite a alteração do dono ou do grupo do arquivo/diretório. A opção “-R” aplica a arquivos/diretórios recursivamente e “-c” mostra o resultado. Vamos ver uns exemplos a seguir (note que o prompt mudou para “#”, o que indica que está executando como *root*, afinal tem que ser *root* para mudar um arquivo de dono ou de grupo).

```
# chown -R evandro /teste
```

→ “evandro” virou dono da pasta /teste, de forma recursiva.



```
# chown evandro:inf /teste
```

→ “/teste” passou a ter o dono “evandro” e o grupo “inf” (mudou os dois de uma vez!).

Por fim, um comando que altera apenas o grupo de um arquivo/diretório, o **chgrp**:

```
#chgrp -R inf /teste
```

→ Recursivamente, mudou o dono de grupo do diretório “/teste” para “inf”.



1. (CCV-UFC/UFC - 2019) Qual dos itens abaixo contém o comando do sistema operacional Linux capaz de mudar o grupo de um arquivo ou diretório do sistema?

- A) df
- B) sed
- C) chown
- D) chmod
- E) passwd

#### Comentários:

Questão interessante porque não mostra o comando “chgrp”. Mas sabemos que o chown permite a alteração de usuário/grupo, ou seja, é possível trocar o grupo também! Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

2. (FGV/MPE-AL - 2018) Para definir o script `/usr/local/bin/meuscript` como executável no sistema operacional Linux, devemos usar o comando

- A) `chmod og /usr/local/bin/meuscript`
- B) `chmod 000 /usr/local/bin/meuscript`
- C) `chmod +x /usr/local/bin/meuscript`



D) `chmod ugo-x /usr/local/bin/meuscript`

E) `chmod 666 /usr/local/bin/meuscript`

### Comentários:

A única que adiciona a execução "x" é a letra C. Como não especifica para "quem", todo o UGO recebe a permissão para a execução.

Na letra E, se transformarmos 666 em binário, temos: 110 110 110 = rw- rw- rw- (não tem "x" para ninguém!).

Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C



## COMANDOS BÁSICOS

Durante a aula vimos comandos relacionados a determinados assuntos (gerenciamento de processos, superusuário, entre outros). Agora vamos ver alguns outros comandos que também costumam ser cobrados em provas de concurso. Primeiro vamos focar em comandos utilizados para **criar/excluir arquivos e diretórios, trocar o diretório corrente e verificar o diretório corrente**:

- **mkdir**: "make directory", cria um diretório, ex.: `mkdir /home/evandro/teste` cria o diretório "teste" dentro de /home/evandro (se não houver esse caminho, ocorre um erro);
- **cd**: "change directory", muda o diretório corrente, exemplos:
  - `cd ..` (muda para o diretório pai);
  - `cd /` (muda para o diretório raiz);
  - `cd ../teste` (sobe um nível e "entra" no diretório teste);
  - `cd teste` ("entra" no diretório teste, que deve estar abaixo do diretório atual);
- **pwd** (print working directory): imprime o caminho completo do diretório de trabalho corrente.
- **rmdir**: remove o diretório, ex.: `rmdir /root/teste`;
- **touch**: cria um arquivo vazio (entre outras funções), ex.: `touch nome_arquivo`;
- **rm**: remove um arquivo, ex.: `rm nome_arquivo`;
- **echo**: pode criar um arquivo de texto, se utilizado o redirecionamento de saída ">" ou ">>", exemplos:
  - `echo oi > arquivo.txt` → cria um arquivo com o conteúdo "oi";
  - `echo oi >> arquivo.txt` → adiciona no fim do arquivo.txt o conteúdo "oi" (se não existir, cria o arquivo);

Vamos ver uma sequência desses comandos a seguir, para melhorar o entendimento.



```
[root@localhost ~]# mkdir teste
[root@localhost ~]# cd teste
[root@localhost teste]# pwd
/root/teste
[root@localhost teste]# mkdir opa
[root@localhost teste]# ls
opa
[root@localhost teste]# rmdir opa
[root@localhost teste]# ls
[root@localhost teste]# touch arquivo_novo
[root@localhost teste]# ls -las
total 8
  4 drwxr-xr-x  2 root    root      66 Nov 19  2019 .
  4 drwx-----  4 root    root     176 Oct 24  2017 ..
  0 -rw-r--r--  1 root    root       0 Nov 19  2019 arquivo_novo
[root@localhost teste]# rm arquivo_novo
[root@localhost teste]# ls -las
total 8
  4 drwxr-xr-x  2 root    root      37 Nov 19  2019 .
  4 drwx-----  4 root    root     176 Oct 24  2017 ..
[root@localhost teste]# cd /
[root@localhost ]# pwd
/
[root@localhost ]#
```

Existe uma infinidade de comandos, ferramentas ou aplicativos para o **monitoramento/diagnóstico** em sistemas Linux. Vamos focar em alguns, de acordo com a cobrança em provas de concurso. Como existem diversos argumentos para cada comando, vamos focar naqueles que já foram alvo de questões.

**ls:** lista o conteúdo de um diretório. Os parâmetros mais cobrados são:

- -a (all): todos os arquivos, não ignora as entradas que começam com ".";
- -l (long): utiliza um formato de listagem longa;
- -s (size): imprime o tamanho de cada arquivo.

**cat:** concatena arquivos e mostra na saída padrão. Sintaxe: cat ARQ1 ARQ2. Mas se for utilizado o direcionador "simples" (>), coloca o conteúdo do primeiro arquivo sobre o conteúdo do segundo e não mostra na tela. Se for utilizado o direcionador "duplo" (>>), concatena o primeiro ao segundo e não mostra na tela. Quando usado com apenas um arquivo, mostra seu conteúdo. Abaixo um exemplo com o direcionador duplo e depois somente com um arquivo.

```
[root@localhost teste]# cat oi.txt >> tchau.txt
[root@localhost teste]# cat tchau.txt
tchau
oi
[root@localhost teste]#
```

Se o usuário quiser ver caracteres "não imprimíveis", pode utilizar alguns parâmetros:



- -v, --show-nonprinting: utiliza “^” e “M-” para mostrar caracteres não imprimíveis, exceto para Line Feed (quebra de linha) e TAB;
- -E, --show-ends: mostra “\$” no fim de cada linha;
- -T, --show-tabs: mostra TAB como “^I”
- -A, --show-all: equivalente aos três anteriores juntos (-vET).

**tail:** a tradução literal é “rabo”, então pense no rabo de um cachorro, é o que fica por último...é o propósito desse comando, mostrar a última parte de um arquivo. Então, se um arquivo é pequeno, tanto faz usar o “cat” ou o “tail”, mas se for grande, depende o que o usuário quer ver!

Abaixo podemos ver uma sequência desses comandos. Primeiro, o pwd mostra que o diretório corrente é o /root/teste. O comando ls com os parâmetros -las (os 3 parâmetros combinados: l, a, s) mostra três arquivos .txt, com tamanhos 4, 6 e 150 bytes. Os arquivos menores, com certeza é possível ver todo o conteúdo com o comando cat. O de 150 bytes seria possível se não houver muitas quebras de linha (mas eu criei um arquivo com os números 01 a 50 com quebra de linha, então o melhor seria usar o tail, caso se deseje ver apenas o final do arquivo). Então, podemos ver todo o conteúdo de tchau.txt e apenas o final do arquivo numeros.txt.

```
[root@localhost teste]# pwd
/root/teste
[root@localhost teste]# ls -las
total 20
 4 drwxr-xr-x  2 root   root   115 Jan  8 19:21 .
 4 drwx----- 4 root   root   157 Oct 24  2017 ..
 4 -rw-r--r--  1 root   root   150 Jan  8 19:29 numeros.txt
 4 -rw-r--r--  1 root   root    4 Jan  8 19:21 ola.txt
 4 -rw-r--r--  1 root   root    6 Jan  8 19:21 tchau.txt
[root@localhost teste]# cat tchau.txt
tchau
[root@localhost teste]# tail numeros.txt
41
42
43
44
45
46
47
48
49
50
[root@localhost teste]#
```

Uma opção para atualizar a exibição de dados que são inseridos no arquivo é a -f (follow).

**find:** procura por arquivos em uma hierarquia de diretórios. Vamos ver algumas maneiras de utilizar o find, com os devidos parâmetros, a seguir.

- Listar todos os arquivos localizados em um determinado diretório, incluindo os arquivos armazenados nos subdiretórios: **\$ find .**
- Definir o nível de “profundidade” na navegação dos subdiretórios (maxdepth), se quiser localizar apenas os arquivos em um diretório, sem os subdiretórios, utiliza-se profundidade 1, por exemplo:



`$ find ./teste -maxdepth 1 -name *.php` → traz como resultado os arquivos PHP dentro do diretório ./teste (obs.: “-name” é *case sensitive* e “-iname” é *case insensitive*)

- Tirar dos resultados alguma coisa, por exemplo, localizar arquivos denominados “comando\*”, de forma que o “comando.php” não apareça (utilizar a negação, com o símbolo `!` de exclamação):

`$ find ./teste -name 'comando*' !-name '*.php'`

- Utilizar o OR para estabelecer mais critérios (parâmetro `-o`), como por exemplo, arquivos JPG e TXT: `$ find -name '*.jpg' -o -name '*.txt'`
- Buscar somente arquivos (`-type f`): `$ find . -type f` → busca apenas os arquivos no diretório corrente (`.`), conforme mostrado abaixo:

```
[root@localhost teste]# ls
numeros.txt  oi.txt      opa          tchau.txt
[root@localhost teste]# find . -type f
./oi.txt
./tchau.txt
./numeros.txt
[root@localhost teste]#
```

- Buscar somente diretórios (`-type d`): `$ find ./teste -type d -name "exemplo*"`
- Buscar arquivos com determinadas permissões: `$ find . -type f -perm 0740` → busca os arquivos com permissão 7 (rwx), 4 (r--), 0 (---)
- Buscar arquivos a partir um determinado proprietário (owner) ou grupo: `$ find . -user evandro` ou `# find /var/www -group adm`
- Buscar arquivos e diretórios com base em períodos de modificação:
  - `# find / -mtime 5` → arquivos modificados há 5 dias
  - `# find / -atime 5` → arquivos acessados há 5 dias
  - `# find / -mtime +10 -mtime -60` → arquivos modificados entre 10 e 60 dias atrás
- Buscar arquivos baseado no tamanho (`-size`): `# find / -size +10M` → procura arquivos maiores que 10 MB.

**grep:** mostra as linhas de um arquivo que “batem” (match) com um determinado padrão. Um exemplo comum é utilizar o cat para “visualizar” o arquivo e a saída dele passar como entrada para o grep, através do pipe (`|`). Ao lado podemos ver um exemplo em que é aplicado o cat no arquivo numeros.txt e em cima dessa saída (visualização do conteúdo de numeros.txt) é aplicado o grep 3, ou seja, mostra todas as linhas do arquivo que possuem o número 3.



```
[root@localhost teste]# cat numeros.txt | grep 3
03
13
23
30
31
32
33
34
35
36
37
38
39
43
[root@localhost teste]#
```

**more:** faz com que um conteúdo de texto seja mostrado uma tela por vez. Se aplicarmos “more números.txt”, teremos as duas telas a seguir, sendo necessário pressionar o espaço para pular para a tela seguinte, ou o ENTER para “andar” de linha em linha:

```
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
--More-- (88% of 150 bytes) [root@localhost teste]#
```

**df:** mostra a quantidade de espaço em disco disponível no sistema de arquivos contendo cada argumento de nome de arquivo. Se nenhum nome de arquivo é fornecido, o espaço disponível em todos os sistemas de arquivos montados é mostrado.

**free:** mostra a quantidade de memória utilizada e livre no sistema.

**which** (“qual”): mostra o caminho completo de determinado comando. Abaixo vemos, por exemplo, o caminho completo dos comandos mount, ls e cat.

```
[root@localhost ~]# which mount
/bin/mount
[root@localhost ~]# which ls
/bin/ls
[root@localhost ~]# which cat
/bin/cat
[root@localhost ~]#
```





1. (IESES/BAHIAGÁS - 2016) Um administrador de sistema gostaria de acompanhar o arquivo de log '/var/log/apache.log' em tempo real. Este arquivo está em um servidor Linux Ubuntu 14.04. Supondo que o administrador possua acesso de superusuário ao sistema através do bash. Qual dos comandos a seguir, executado uma única vez, poderia auxiliar o administrador a acompanhar este log dinamicamente (sem interferência adicional)?

- A) tail -rt /var/log/apache.log
- B) more -rt /var/log/apache.log
- C) more -f /var/log/apache.log
- D) cat /var/log/apache.log
- E) tail -f /var/log/apache.log

#### Comentários:

Para ver os dados no fim do arquivo o comando é tail. Uma opção para atualizar a exibição de dados que são inseridos no arquivo é a "-f" (follow = seguir). Portanto, a **alternativa E** está correta e é o gabarito da questão.

Gabarito: Letra E

2. (INSTITUTO AOCP/EBSERH - 2016) Um administrador de sistema operacional Linux necessita procurar determinados textos, dada uma palavra, dentro de um arquivo texto. O comando que possibilita executar essa operação é o

- A) find.
- B) more.
- C) whereis.
- D) grep.



E) pipe.

### Comentários:

grep: mostra as linhas de um arquivo que "batem" (match) com um determinado padrão. Um exemplo comum é utilizar o cat para "visualizar" o arquivo e a saída dele passar como entrada para o grep, através do pipe (|). Ao lado podemos ver um exemplo em que é aplicado o cat no arquivo numeros.txt e em cima dessa saída (visualização do conteúdo de numeros.txt) é aplicado o grep 3, ou seja, mostra todas as linhas do arquivo que possuem o número 3. Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D



## QUESTÕES COMENTADAS - SERVIÇOS DE REDE – DNS E DHCP - MULTIBANCAS

1. (FGV/MEC - 2009) No UNIX, o DNS é implementado por meio do software Berkeley Internet Name Domain (BIND), que segue a filosofia cliente/servidor.

O cliente do BIND é denominado "resolver", responsável por gerar as "queries" enquanto que o servidor do BIND é chamado por "named", responsável por responder às "queries".

Configurações BIND são descritas pelo tipo de serviço a ser executado pelo software, existindo quatro níveis de serviço que podem ser definidos nessas configurações: "resolver-only systems", "caching-only servers", "primary servers" e "secondary servers".

Os parâmetros que definem a configuração do "resolver-only systems" são inseridas no seguinte arquivo:

- A) /etc/resolv.sys
- B) /etc/resolv.cfg
- C) /etc/resolv.bind
- D) /etc/resolv.conf
- E) /etc/resolv.parm

### Comentários:

O arquivo resolv.conf também serve para especificar o tipo (nível) de serviço a ser executado pelo software, a saber:

- resolver-only system: não requer que o sistema local tenha um servidor DNS em execução, requer apenas o resolver;
- caching-only server: servidor que não possui uma cópia da tabela de zonas, ou seja é não-autoritativo, possuindo uma grande quantidade de registros em cache;
- primary server (master server): onde todas atualizações manuais devem ser realizadas, é autoritativo;
- secondary server: busca informações de zona em um master server, é autoritativo.

Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D



2. (FCC/TCE-SP - 2010) Em Linux, o `/etc/resolv.conf` é um arquivo texto simples, com um parâmetro por linha e especificações de endereços de servidores DNS. Nesse arquivo existem três palavras chaves normalmente usadas, que são

- A) domain, search e resolv.
- B) search, resolv e order.
- C) domain, search e nameserver.
- D) search, nameserver e order.
- E) search nameserver e resolv.

#### Comentários:

Mais uma vez vamos ver um exemplo do `resolv.conf`, para memorizar 😊

```
domain ti.teste.com.br
search ti.teste.com.br teste.com.br
nameserver 192.168.1.1
nameserver 192.168.2.1
```

Portanto, a **alternativa C** está correta e é o gabarito da questão.

Gabarito: Letra C

3. (FCC/TCE-SP - 2010) No Linux, ele é muito útil para máquinas que são acessadas frequentemente, pois a inclusão de um computador neste arquivo dispensa a consulta a um servidor DNS para obter um endereço IP, sendo muito útil para máquinas que são acessadas frequentemente. Trata-se de:

- A) `/etc/nameserver`
- B) `/etc/localhost`
- C) `/etc/ipsec`
- D) `/etc/hosts`
- E) `/etc/dnshost`

#### Comentários:



Um exemplo do arquivo hosts (que também existe no Windows!):

```
# Aqui são colocados comentários
# End. IP      Nome do host      Alias (opcional)
127.0.0.1     localhost         localmente
192.168.0.1   www.teste.com.br teste
```

Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

4. (AOCP/UFGD - 2014) Em algumas distribuições Linux encontramos um arquivo que identifica os locais dos computadores de servidor de nome DNS, computadores estes que são usados pelo TCP/IP para traduzir os nomes de Internet. Assinale a alternativa que apresenta o nome deste arquivo.

- A) DNS.conf.
- B) resolver.conf.
- C) resolv.conf.
- D) translate.conf.
- E) change.conf.

**Comentários:**

Mesmo que a questão anterior! Note que o resolv.conf é bastante cobrado! Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

5. (IADES/TRE-PA - 2014) Um servidor Linux pode hospedar o serviço de resolução de nomes de uma rede de computadores. Conhecido por DNS, esse serviço é indispensável em uma rede que possua conexão com a internet. O nome de um pacote que implementa o DNS, muito utilizado em sistemas operacionais Linux, é

- A) Firefox.
- B) Apache.
- C) Squid.



D) Postfix.

E) BIND.

#### Comentários:

O pacote é conhecido como BIND (versão 8, ou bind9 - versão 9). Portanto, a **alternativa E** está correta e é o gabarito da questão.

**Gabarito:** Letra E

**6. (FCC/CNMP - 2015) O serviço de nome de domínios (DNS) possui uma arquitetura do tipo cliente/servidor na qual a base de dados é distribuída por toda internet. Nessa arquitetura, o acesso ao servidor DNS para buscar o relacionamento IP/Domínio é feito pelo cliente que é o**

A) Browser.

B) DNS Cache.

C) DNS Resolver.

D) DNS Searcher.

E) Gateway.

#### Comentários:

Conforme vimos, o browser (navegador) não resolve o nome! Supondo que uma URL seja solicitada através de um browser, o "resolvedor DNS" terá o papel de resolver o DNS (o próprio nome já diz isso). Pode ser através do arquivo hosts (máquina local) ou através de solicitação ao servidor DNS "mais próximo" (mais comum - geralmente o servidor DNS utilizado pelo provedor de acesso à Internet). Se o servidor DNS tiver a informação em cache, ele responde, senão faz uma busca nos servidores superiores na hierarquia DNS, atualiza sua cache e responde ao "resolvedor". Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**7. (CESPE/FUB - 2015) O protocolo DNS (domain name service), localizado no nível de aplicação da camada de transporte do TCP, é responsável pelo mapeamento de nomes e de endereços.**

#### Comentários:

O DNS está localizado na camada de aplicação. A questão misturou "nível de aplicação da camada de transporte do TCP". Portanto, a questão está **errada**.



Gabarito: Errada

8. (FGV/TCE-SE - 2015) Um programa precisa simular o comportamento de um cliente DNS. Para funcionar adequadamente, o programa precisa enviar as consultas para um servidor DNS, especificamente para a sua porta:

A) udp/23

B) icmp/34

C) tcp/22

D) ip/50

E) udp/53

#### Comentários:

Como já vimos, o protocolo de transporte utilizado para consultas DNS é o UDP e a porta é a 53. Essas informações devem estar enraizadas em seu cérebro! Para a transferência de zonas de DNS, é utilizado o TCP (na mesma porta: 53) como protocolo de transporte. Portanto, a **alternativa E** está correta e é o gabarito da questão.

Gabarito: Letra E

9. (CESPE/TRE-MT - 2015) Acerca do servidor DNS/BIND (Domain Name System/Berkeley Internet Domain), cuja funcionalidade é resolver nomes da rede, assinale a opção correta.

A) Cada domínio tem seus registros de recursos e o registro de domínio denominado NS (name server), o qual é utilizado para definir propriedades básicas do domínio e sua zona.

B) Um servidor DNS utiliza LDAP para fazer armazenamento das zonas de domínio para uma rápida resolução de um nome.

C) O BIND, que utiliza a porta 53, é um programa de código aberto utilizado pela maior parte dos servidores DNS.

D) Os domínios de um servidor DNS são organizados na Internet sobre uma estrutura de dados do tipo lista encadeada, sendo o primeiro elemento da lista um ponto.

E) O protocolo HTTP implementa, por padrão, um servidor de resolução de nomes amplamente utilizado na Internet conhecido como DNS.

#### Comentários:



Para quem não está acostumado com essa parte de servidores, memorize o seguinte: servidor DNS é quase sinônimo de BIND! Se você ainda não o conhece e não leu sobre ele, recomendo uma passada rápida na URL <https://www.isc.org/downloads/bind/>. Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**10.(IADES/CRC-MG - 2015)** Quanto ao programa que é normalmente utilizado para testar se um servidor DNS (Domain Name Server) está funcionando corretamente, ou seja, resolvendo nomes para os endereços IP, assinale a alternativa correta.

- A) ping
- B) tracert
- C) nslookup
- D) ipconfig
- E) net host

**Comentários:**

(A) Serve para ver se “está vivo”; (B) Serve para traçar uma rota (roteadores intermediários); (C) Exatamente! ns = name server, lookup = “dar uma olhada”, se você nunca usou, experimente agora no prompt: nslookup estrategiaconcursos.com.br; (D) verifica configurações das interfaces de rede; (E) o comando net não possui o parâmetro host! Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**11.(FGV/TCE-SE - 2015)** Em um equipamento rodando Linux, um administrador deseja alterar o endereço default do servidor DNS usado para consultas. Para isso, ele precisa alterar o arquivo:

- A) /etc/named.conf
- B) /etc/bind.conf
- C) /etc/network.conf
- D) /etc/dns.conf
- E) /etc/resolv.conf



### Comentários:

Um exemplo de arquivo resolv.conf:

```
domain ti.teste.com.br
search ti.teste.com.br teste.com.br
nameserver 192.168.1.1
nameserver 192.168.2.1
```

Portanto, a **alternativa E** está correta e é o gabarito da questão.

**Gabarito:** Letra E

12.(FGV/COMPESA - 2016) O sistema de Nomes de Domínio - DNS permite transformar nomes digitados em um navegador WEB em um endereço de rede. O nome do host e o nível do domínio para o domínio "system.master.com" são, respectivamente,

- A) system e segundo nível.
- B) system e terceiro nível.
- C) master e terceiro nível.
- D) master e segundo nível.
- E) .com e terceiro nível.

### Comentários:

Um servidor DNS raiz "aponta" para servidores de primeiro nível (TLD). No domínio mostrado, o TLD é o ".com", que aponta para um servidor DNS responsável por "master" (segundo nível), que por sua vez, aponta para um terceiro nível, o host "system", no caso da questão. Portanto, a **alternativa B** está correta e é o gabarito da questão.

**Gabarito:** Letra B

13.(FCC/PRODATER - 2016) No Serviço de Nomes de Domínio – DNS existem diferentes tipos de servidores distribuídos hierarquicamente que armazenam informações também de forma hierárquica. Considerando o nome: www.empresa.com, o domínio .com é gerenciado pelo servidor

- A) Global.
- B) PDR.



- C) Authoritative.
- D) TLD.
- E) Root.

#### Comentários:

Os domínios de mais alto nível são classificados em genéricos (".com", ".edu", ".bradesco", etc.) ou códigos de países (".br", ".mx", ".uy", ".jp", etc.). Em inglês é conhecido como TLD (Top Level Domain) e os servidores DNS TLD gerenciam tais domínios. Abaixo podemos ver mais exemplos de TLDs genéricos e dois exemplos de códigos de países. Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

**14. (IBFC/EBSERH-HUAP - 2016) O Domain Name System (DNS) é um sistema de gerenciamento de nomes hierárquicos e distribuídos. A funcionalidade do DNS Reverso vem a ser:**

- A) reverter a funcionalidade básica de um DNS padrão obtendo o endereço MAC.
- B) resolver um endereço IP, buscando o nome de domínio associado ao host.
- C) resolver o nome do domínio de um host qualquer para seu endereço IP correspondente.
- D) com base na região geográfica, obter automaticamente um endereço IP local.
- E) resolver o problema da reversão do IPv6 para o IPv4 de uma forma rápida e automática.

#### Comentários:

A questão começa com a definição de DNS, segundo o Tanenbaum. O DNS tem a função de traduzir um nome em um endereço, então o DNS reverso faz o contrário: traduz um endereço em um nome. É definido através do registro PTR:

PTR    Ponteiro (usado para o DNS reverso)    Nome alternativo de um end. IP.

Portanto, a **alternativa B** está correta e é o gabarito da questão.

**Gabarito:** Letra B

**15. (FUNRIO/CM Nova Iguaçu - 2016) Um administrador de rede está configurando o DNS de um servidor e vai trabalhar com o registro que define as características da zona a ser configurada, tais como, o nome da zona e o nome do servidor, que é a autoridade para a referida zona, ou**



seja, o servidor DNS no qual está a zona que foi criada originalmente. Esse registro é conhecido pela sigla

- A) HINFO.
- B) MX.
- C) SOA.
- D) CNAME.

#### Comentários:

Mais uma vez aquela tabela, agora copiei apenas a linha sobre o "SOA":

SOA Início de autoridade (Start of Authority). Parâmetros para essa zona.

Ou seja, se falar sobre características/parâmetros de uma zona, trata-se do registro Start of Authority (SOA). Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**16.(CESPE/TCE-PR - 2016) Uma consulta DNS inicial típica, originada de uma máquina de usuário e encaminhada ao servidor de nomes local para um nome de domínio externo não armazenado em cache DNS, será do tipo**

- A) raiz.
- B) domínio de alto nível.
- C) iterativa.
- D) recursiva.
- E) direta.

#### Comentários:

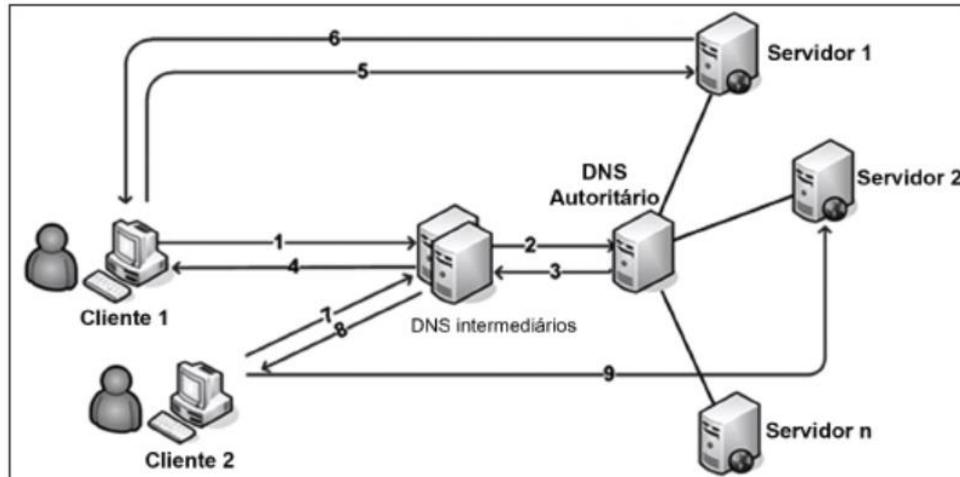
O foco da questão é a consulta realizada de um equipamento ao servidor DNS local, supondo que a informação não esteja na memória cache desse servidor local. Nesse caso, o servidor DNS local irá requisitar ao servidor DNS TLD, receberá como resposta o próximo servidor DNS a ser consultado e repetirá o processo até ter toda a informação desejada. Armazenará em sua cache e responderá ao cliente (equipamento que fez a solicitação inicial). Note que o cliente fez uma



requisição apenas e obteve uma resposta, mas o servidor DNS local realizou consultas recursivas. Portanto, a **alternativa D** está correta e é o gabarito da questão.

Gabarito: Letra D

17.(FCC/ELETROSUL - 2016) O balanceamento de carga por DNS é uma abordagem popular e simples para as solicitações de balanceamento de servidores. Considere a figura abaixo.



De acordo com a figura, geralmente são estes os passos que ocorrem assim que é feita uma consulta DNS:

1. Quando um cliente tenta acessar o site, é realizada uma pesquisa no DNS local para determinar qual é o endereço I..... correspondente;
2. O pedido de endereço chega ao servidor de DNS II..... do domínio;
3. A primeira vez que esta consulta é feita, o servidor DNS remoto pode retornar todos os registros de endereços que ele tem para o site;
4. O servidor DNS III....., em seguida, determina o endereço de registro para retornar ao cliente;
5. Se todos os registros são retornados, o cliente utilizará o primeiro que lhe é atribuído;
6. O servidor responde ao cliente e atende ao pedido;
7. A cada pedido, o algoritmo Round Robin roda os endereços e retorna pela ordem em que eles estão;
8. Cada consulta DNS irá resultar em um cliente usando um endereço IV.....;
9. Esta rotação de endereços irá distribuir pedidos para os servidores.

As lacunas de I a IV são, correta e respectivamente, preenchidas com:

A) TCP – autoritário – local – TCP igual ao obtido em 1



- B) IP – autoritário – local – IP diferente
- C) do servidor – local – autoritário – de um servidor
- D) do servidor – autoritário – local – do servidor igual ao obtido em 1
- E) IP – local – autoritário – IP igual ao obtido em 1

### Comentários:

Quando um cliente tenta acessar o site, é realizada uma pesquisa no DNS local para determinar qual é o endereço IP (DNS traduz nome em endereço IP) correspondente; O pedido de endereço chega ao servidor de DNS autoritário (após o servidor DNS local, chega ao servidor que possui os registros atualizados = autoritário) do domínio; O servidor DNS local (depois de retornar do servidor autoritário, o servidor local atualiza a cache e envia ao cliente), em seguida, determina o endereço de registro para retornar ao cliente; Cada consulta DNS irá resultar em um cliente usando um endereço IP diferente (essa é a ideia do algoritmo Round Robin, o balanceamento de carga). Portanto, a **alternativa B** está correta e é o gabarito da questão.

**Gabarito:** Letra B

**18.(CESPE/TCE-SC - 2016) Após o servidor local SMTP aceitar uma mensagem para subsequente envio, é necessário determinar o endereço do servidor de email do destinatário. Essa etapa é realizada mediante consulta DNS a um servidor de nomes capaz de prover a informação, no qual serão verificados os registros especiais MX (mail exchange).**

### Comentários:

Vamos relembrar a tabela, focando na linha sobre "MX":

Tipo	Significado	Valor
SOA	Início de autoridade ( <i>Start of Authority</i> ).	Parâmetros para essa zona.
A	Endereço IPv4.	Inteiro de 32 bits.
AAAA	Endereço IPv6.	Inteiro de 128 bits.
MX	Troca de mensagens de e-mail.	Prioridade, domínio disposto a aceitar e-mails.
NS	Servidor de nomes.	Nome de um servidor para este domínio.
CNAME	Nome canônico ( <i>alias</i> = apelido).	Nome de domínio.
PTR	Ponteiro (usado para o DNS reverso <sup>1</sup> )	Nome alternativo de um end. IP.

<sup>1</sup> Envia um endereço IP como consulta e recebe o nome como resposta.



SPF	Estrutura de política do transmissor.	Codificação de texto da política de envio de mensagens de e-mail.
SRV	Identifica computadores que hospedam serviços específicos.	Host que o oferece.
TXT	Informações sobre um servidor, rede, <i>datacenter</i> etc.	Texto ASCII com descrições.

Portanto, a questão está **correta**.

**Gabarito:** Correta

**19.(CESPE/Polícia Científica-PE - 2016) Na operação de um serviço DNS, se uma consulta a um servidor de alto nível retornar uma resposta positiva para o servidor DNS local, o mapeamento nome/endereço será**

- A) armazenado definitivamente no banco de dados do servidor DNS local.
- B) replicado para todos os servidores DNS alcançáveis para que estejam disponíveis a outras consultas.
- C) repassado pelo módulo tradutor a um servidor raiz para confirmar a resposta recebida.
- D) armazenado em cache local e pode ser mantido por um tempo definido por configuração.
- E) repassado ao módulo tradutor do provedor de serviços para realizar a entrega à aplicação.

**Comentários:**

Quando o servidor DNS local busca uma informação nova, ele guarda na cache, por tempo determinado. Esse tempo varia de segundos/minutos a um dia, por exemplo. Depende do tempo definido no campo TTL. Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

**20.(IBFC/Polícia Científica-PR - 2017) Servidores de DNS (Domain Name System) têm como função converter endereços IP em seu respectivo nome e vice-versa. Para sua configuração, são utilizados arquivos denominados mapas de domínio (zone). Esses arquivos são compostos por entradas chamadas RR (Resource Record). O tipo básico de RR que estabelece a correspondência entre um nome canônico e um endereço IP é indicado por:**

- A) PTR



- B) MX
- C) NS
- D) A
- E) SOA

#### Comentários:

Quando é realizada uma consulta pelo nome, esperando como resposta um endereço IP, os RRs possíveis são:

A	Endereço IPv4.	Inteiro de 32 bits.
AAAA	Endereço IPv6.	Inteiro de 128 bits.

Como a questão não mostra "AAAA" como alternativa, sobrou apenas o "A", que retorna um endereço IPv4.

Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

**21.(IBFC/Polícia Científica-PR - 2017) Servidores DNS (Domain Name Server) são responsáveis pela conversão do nome dos diversos servidores espalhados pela Internet para seu número IP e vice-versa. Servidores de DNS trabalham de forma colaborativa e hierárquica. Assinale a alternativa a que apresenta o nome dado aos servidores que se encontram no topo da hierarquia de DNS:**

- A) Root Name Servers
- B) Main servers
- C) International Name Servers
- D) Controllers Servers
- E) Master Servers

#### Comentários:



A hierarquia DNS é uma árvore invertida, ou seja, os servidores raiz (Root Name Servers) ficam no topo, logo abaixo ficam os TLDs (Top Level Domains), depois os de segundo nível e assim por diante. Portanto, a **alternativa A** está correta e é o gabarito da questão.

**Gabarito:** Letra A

**22. (CESPE/Polícia Federal - 2018) As atualizações entre servidores DNS utilizam o UDP, enquanto as consultas feitas a servidores DNS utilizam o TCP (ou, opcionalmente, o SCTP).**

**Comentários:**

Vamos tentar utilizar a lógica: o UDP é bem mais leve, então deve ser utilizado em consultas DNS, pois tais consultas ocorrem em grande número na Internet e se alguma mensagem for perdida não há grandes problemas, bastando realizar uma nova consulta. As atualizações entre servidores DNS são bem menos frequentes que as consultas e é importante que mensagens não sejam perdidas pela rede, logo, é interessante que se utilize TCP, correto? Bom, essa lógica é a utilizada no DNS! E, como podemos ver, a questão inverteu o TCP com o UDP. Portanto, a questão está **errada**.

**Gabarito:** Errada

**23. (IF-RS/IF-RS - 2018) Considerando que se está logado como "root" no terminal de uma estação de trabalho com um sistema operacional Linux debian ou derivado, qual comando configura o DNS para 8.8.8.8?**

- A) dns 8.8.8.8
- B) echo 8.8.8.8 > /etc/hosts.conf
- C) echo nameserver 8.8.8.8 > /etc/resolv.conf
- D) nslookup 8.8.8.8
- E) nameserver 8.8.8.8

**Comentários:**

nameserver: Endereço IP do servidor de nomes que o resolvedor deve pesquisar, utilizados na ordem em que estão listados. Tal diretiva deve estar no arquivo /etc/resolv.conf, então com o comando echo é possível escrever "nameserver 8.8.8.8" em "/etc/resolv.conf" sem ter que abrir o arquivo.

Portanto, a **alternativa C** está correta e é o gabarito da questão.



Gabarito: Letra C

24.(FAURGS/BANRISUL - 2018) Um cliente DNS, ao fazer a requisição DNS para resolver o nome [www.banrisul.com.br](http://www.banrisul.com.br), recebe como resposta uma mensagem do tipo non authoritative. Esse tipo de resposta é obtido por meio de um registro DNS (resource record - RR) armazenado

- A) em um cache de DNS.
- B) em um servidor interativo sem cache DNS.
- C) em um servidor recursivo sem cache DNS.
- D) no arquivo de zona do DNS primário.
- E) no arquivo de zona do DNS secundário.

#### Comentários:

Um servidor DNS autoritativo possui autoridade sobre um nome de domínio. O DNS autoritativo dita qual será o apontamento da tabela de DNS do seu sítio. Uma resposta autoritativa de um servidor é a garantia de estar atualizada, enquanto uma resposta não-autoritativa pode estar desatualizada (cache com informações antigas, por exemplo). Existe um percentual elevado de respostas não autoritativas que estão perfeitamente corretas, casos em que mudanças de endereçamento são raros.

Servidores primários e secundários são autoritativos para os seus domínios, porém não o são sobre informações a respeito de outros domínios mantidas em cache. Servidores caching-only nunca são autoritativos, mas possuem a vantagem de reduzir a quantidade de tráfego DNS na rede.

Portanto, a **alternativa A** está correta e é o gabarito da questão.

Gabarito: Letra A

25.(FAPEC/UFMS - 2018) Considere as afirmações a seguir sobre a instalação e configuração de um servidor de DNS (Bind9) no sistema operacional Debian 9.

I - Quando o servidor local de DNS não consegue traduzir o nome, ele deve solicitar a um servidor público (forwarders). O Google Public DNS é um exemplo de forwarders e deve ser configurado no arquivo `named.conf.options`.

II - A instalação do servidor pode ser realizada pelo usuário root com o comando: `apt install bind9 bind9-doc dnsutils`

III - A lista de controle de acessos ou ACL é definida no arquivo `named.conf.options`.



Está(ão) correta(s):

- A) Apenas I.
- B) Apenas II.
- C) Apenas III.
- D) Apenas II e III.
- E) I, II e III.

**Comentários:**

Todas as afirmativas foram abordadas no decorrer da aula. Portanto, a **alternativa E** está correta e é o gabarito da questão.

**Gabarito:** Letra E

**26.(CESPE/EMAP - 2018)** Em um servidor DNS que esteja utilizando Bind, o arquivo `/etc/named.conf` mantém as configurações de funcionamento do serviço DNS. Nesse servidor, é possível configurar a porta que deve escutar o serviço; assim, para que o serviço escute a porta 53, deve-se configurar a opção seguinte.

```
Listening 53 { any;};
```

**Comentários:**

No arquivo `named.conf` são configuradas informações de zonas. O próprio nome do arquivo já sugere isso (`named` = nomes de domínios). Não há a configuração da porta em que o serviço deve "escutar" nesse arquivo! Portanto, a questão está **errada**.

**Gabarito:** Errada

**27.(CESPE/TRT8 - 2013)** Para alterar o arquivo de configuração do serviço de DHCP no Linux, deve-se acessar o diretório

- A) `/dev`.
- B) `/mnt`.
- C) `/etc`.
- D) `/usr/bin`.



E) /boot.

### Comentários:

Por padrão, os arquivos de configuração ficam no diretório "/etc", como é o caso do "dhcpd.conf". Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**28.(SUGEP-UFRPE/UFRPE - 2018)** Para acesso à Internet, servidores Windows e Linux utilizam um serviço para atribuição dinâmica de IPs aos microcomputadores conectados à rede, em que é marcada a opção associada a "o IP será atribuído automaticamente pelo servidor", na configuração.

Esse serviço é conhecido pela sigla:

- A) DNS
- B) HTTP
- C) DHCP
- D) TCP
- E) IP

### Comentários:

O DHCP (Dynamic Host Configuration Protocol) é um protocolo que tem a função de configurar os endereços IP dos computadores de uma rede de forma dinâmica. Ou seja, deve haver pelo menos um servidor DHCP pré-configurado para receber solicitações de computadores que não possuem endereço IP (clientes), o servidor verificar qual endereço IP disponível e envia ao solicitante. Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C



## QUESTÕES COMENTADAS - SERVIÇOS DE REDE – FTP E E-MAIL - MULTIBANCAS

1. (CESPE/HEMOBRÁS - 2008) Tanto o Linux quanto o Windows possuem softwares clientes para FTP (file transfer protocol) e que podem ser utilizados para a troca de arquivos em uma rede local.

### Comentários:

É comum que sistemas operacionais conhecidos, como o Windows e o Linux, tenham software cliente FTP para se conectar a um servidor e enviar/baixar arquivos. Portanto, a questão está correta.

**Gabarito:** Correta

2. (INSTITUTO AOCP/UFPB - 2014) Um estagiário executou o comando FTP no servidor Linux. Esse comando possibilita

- A) permitir a abertura de portas para um acesso remoto a um servidor da rede.
- B) realizar a manutenção de disco e de memória de um computador remotamente.
- C) trocar informações de desempenho de redes utilizando um canal de comunicação protegido e exclusivo.
- D) realizar uma verificação de permissões de acesso na rede e no computador que o usuário está utilizando.
- E) trocar arquivos entre computadores interligados, independente do sistema operacional e da rede utilizada.

### Comentários:

O comando ftp, tanto no Linux como no Windows, chama o cliente FTP, que permite trocar arquivos entre computadores interligados, independente do sistema operacional e da rede utilizada. Por exemplo, um cliente FTP Linux pode se conectar a um servidor FTP Windows e tudo irá funcionar tranquilamente. Portanto, a **alternativa E** está correta e é o gabarito da questão.

**Gabarito:** Letra E

3. (VUNESP/SAAE de Barretos-SP - 2018) Os serviços de acesso a dados da internet são disponibilizados em uma arquitetura Cliente/Servidor. Por exemplo, considerando um servidor



com sistema operacional Linux, o serviço padrão que responde às requisições ftp é denominado

- A) ftpd.
- B) ftps.
- C) ftpsrv.
- D) samba.
- E) smb.

#### Comentários:

Seguindo a lógica adotada para servidores no Linux, o servidor FTP é executado através de um daemon, o ftpd (geralmente o daemon tem o nome do serviço, acrescido de um "d" no fim). Portanto, a **alternativa A** está correta e é o gabarito da questão.

**Gabarito:** Letra A

4. (FUNIVERSA/PC-DF - 2012) Em um servidor de e-mail com sistema operacional Linux, especialmente quando se utiliza o serviço Postfix, pode-se auditar o tráfego de entrada e saída de mensagens de correio eletrônico por meio dos registros conhecidos por logs. Em que arquivo desse sistema operacional encontram-se os logs das atividades de envio e recebimento de mensagens de correio eletrônico (e-mail)?

- A) No diretório "/etc/postfix".
- B) No arquivo "/var/log/maillog".
- C) Na pasta "/var/mail/<nome\_do\_usuario>".
- D) No arquivo "/usr/local/bin/postfix".
- E) No arquivo de registro "/dev/tty".

#### Comentários:

Uma característica importante no Linux são os logs, que, na sua maioria, são armazenados em /var/log. Esse é o caso também dos logs relacionados aos servidores de e-mail, útil quando são necessárias informações sobre postfix, smtpd ou qualquer serviço relacionado ao e-mail que esteja sendo executado no servidor. Por padrão, esses logs ficam em /var/log/maillog ou /var/log/mail.log. Portanto, a **alternativa B** está correta e é o gabarito da questão.



Gabarito: Letra B

5. (FUNCAB/MDA - 2014) São exemplos de servidores de correio eletrônico do Linux:

- A) postfix e sendmail.
- B) fetchmail e sendmail.
- C) pine e postfixe.
- D) mail e pine.
- E) samba e postfixe.

**Comentários:**

Alguns exemplos de servidores de e-mail no Linux: qmail, postfix e sendmail. Portanto, a **alternativa A** está correta e é o gabarito da questão.

Gabarito: Letra A

6. (FGV/TJ-GO - 2014) A empresa Y passou a adotar uma política de uso de software livre e resolveu mudar o seu correio eletrônico, que era baseado em Microsoft Exchange. O novo ambiente deveria rodar em ambiente Linux. Uma das possíveis escolhas do novo software de correio eletrônico recai no programa:

- A) nginx;
- B) puppet;
- C) apache;
- D) postfix;
- E) bind.

**Comentários:**

Alguns exemplos de servidores de e-mail no Linux: qmail, postfix e sendmail. Portanto, a **alternativa D** está correta e é o gabarito da questão.

Gabarito: Letra D



7. (FCC/Prefeitura de Manaus-AM - 2019) Um Assistente de TI foi incumbido de configurar o protocolo para ser utilizado na troca de mensagens entre dois servidores de e-mail. A escolha correta do protocolo para essa finalidade é:

- A) IMAP.
- B) POP3.
- C) SNMP.
- D) SMTP.
- E) POP4.

#### Comentários:

SMTP (Simple Mail Transfer Protocol): utilizado quando o e-mail é entregue a partir de um cliente de e-mail (Outlook, Thunderbird, entre outros) a um servidor de e-mail ou quando o e-mail é entregue a partir de um servidor de e-mail para outro servidor. Ou seja, se você enviar um e-mail a partir de um software cliente, ele utilizará o SMTP até o seu servidor e depois o seu servidor utilizará também o SMTP para entregar ao servidor destino, quando este ficará na caixa de e-mails do usuário destinatário. A porta padrão de um servidor SMTP é a 25. Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D



## QUESTÕES COMENTADAS - GERENCIAMENTO DE PROCESSOS - MULTIBANCAS

1. (CESPE/TJ-SE - 2014) Alguns programas podem apresentar problemas que resultem no travamento do sistema operacional, o que pode ser resolvido, no Linux, por meio do comando Kill, que finaliza o processo, funcionalidade que pode ser acessada por meio de outro terminal.

### Comentários:

O comando kill serve para enviar um sinal para um processo através de seu PID. Esse sinal pode ser para finalizar um processo, entre outros. A questão não fala que o kill serve apenas para finalizar, diz que é possível finalizar um processo com ele. Portanto, a questão está **correta**.

**Gabarito:** Correta

2. (FCC/CNMP - 2015) O comando ps do Linux permite parâmetros (ou opções) para mostrar:
- I. os processos criados por você e de outros usuários do sistema.
  - II. o nome de usuário que iniciou o processo e hora em que o processo foi iniciado.
  - III. as variáveis de ambiente no momento da inicialização do processo.
  - IV. a árvore de execução de comandos (comandos que são chamados por outros comandos).
- Para realizar o que descrevem os itens I, II, III e IV são utilizados, correta e respectivamente, os parâmetros

A) f / u / e / a

B) f / a / e / u

C) u / e / f / a

D) e / u / a / f

E) a / u / e / f

### Comentários:

A = All, U = user, E = environment (ambiente), F = tree (esse fugiu da lógica de nomes, mas os outros dá para "adivinhar", se não souber). Portanto, a **alternativa E** está correta e é o gabarito da questão.

**Gabarito:** Letra E



3. (FCC/TRT14 - 2016) Para listar todos os processos que estavam em execução em um computador com o sistema operacional Linux instalado, um usuário utilizou o comando ps. Para que esse comando exiba informações detalhadas de cada processo, como o nome do usuário que iniciou o processo, o número identificador do processo, a porcentagem de utilização da CPU e da memória pelo processo e a hora em que cada processo foi iniciado, este comando deve ser utilizado com o parâmetro

A) aux.

B) -top.

C) vim.

D) pstree.

E) -zcf.

#### Comentários:

A = All, U = user, X = não executado no terminal "atual" (foge um pouco da lógica dos nomes, mas os outros estão mais tranquilos). Portanto, a **alternativa A** está correta e é o gabarito da questão.

Gabarito: Letra A

4. (FGV/IBGE - 2017) Em ambiente Linux, a chamada ao sistema (system call) que implementa a criação de um novo processo é:

A) create\_process;

B) new\_process;

C) fork;

D) spawn;

E) duplicate.

#### Comentários:

Não há chamada de sistema que crie um processo no Linux, o que acontece é que um processo existente realiza seu clone - chamada de sistema fork() - e o processo clone pode trocar o programa por um diferente. O original é o pai e a cópia gerada é o filho. Portanto, a **alternativa C** está correta e é o gabarito da questão.



Gabarito: Letra C

5. (FGV/Prefeitura de Niterói-RJ - 2018) Sobre o gerenciamento de processos no sistema operacional Linux, assinale a afirmativa correta.

- A) O comando ps lista os processos em execução no sistema, trazendo informações sobre o quanto de processamento ou de memória cada um deles está consumindo.
- B) O ID do processo não pode ser reutilizado depois que o processo termina.
- C) Por meio do comando renice é possível alterar a prioridade de execução de um processo.
- D) Por padrão, os processos executados por um usuário iniciam com nível de prioridade -20, isto é, a prioridade mais baixa possível.
- E) Processos órfãos não possuem PPID.

**Comentários:**

(A) O ps pode exibir PID, UID, prioridade, terminal de controle de processos, memória consumida, estado atual. O quanto de processamento está consumindo, não! (B) Ex.: se o processo 4000 é terminado, depois de algum tempo outro processo pode utilizar o PID 4000, sem problemas. (C) Isso aí! O renice altera a prioridade de um processo já em execução, enquanto o nice apenas atribui um valor de prioridade a um programa a ser executado. (D) O padrão é iniciar com prioridade 0 e -20 é a maior prioridade possível! A menor é 19 (isso mesmo, invertido). (E) Possuem PPID de um processo que já não existe mais. Quando isso é constatado o processo init assume a "paternidade" desse processo órfão.

Portanto, a **alternativa C** está correta e é o gabarito da questão.

Gabarito: Letra C

6. (CESPE/EBSERH - 2018) No sistema operacional Linux, é possível alterar a prioridade de um processo já iniciado com o uso do comando nice.

**Comentários:**

O nice é utilizado para executar um programa e o renice é utilizado para alterar a prioridade do processo já em execução. Portanto, a questão está **errada**.

Gabarito: Errada



7. (CESPE/TJ-PA - 2020) No Linux, um processo, por si só, não é elegível para receber tempo de CPU. Essa ação depende, basicamente, do seu estado da execução. O processo está administrativamente proibido de executar no estado

- A) pronto.
- B) dormente.
- C) executável.
- D) parado.
- E) zumbi.

**Comentários:**

Os estados dos processos em sistemas Linux podem ser:

Estado	Significado
Executável	Pode ser executado (pronto).
Dormente	Aguardando algum recurso (ex.: E/S).
Zumbi	Tentando se destruir.
Parado	Suspenso (sem permissão para ser executado).

Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D



## QUESTÕES COMENTADAS - MONTAGEM DE VOLUMES - MULTIBANCAS

1. (FCC/TRT4 - 2015) A possibilidade de compartilhar arquivos entre diferentes sistemas operacionais é fundamental para aumentar a produtividade computacional. A montagem automática de uma partição com sistema de arquivos CIFS, durante o boot do servidor com sistema operacional Linux, deve ser configurada no arquivo

- A) /etc/fstab.
- B) /boot/mount.
- C) /etc/mount.
- D) /boot/inittab.
- E) /etc/initd.

### Comentários:

O arquivo /etc/fstab possui informações sobre os sistemas de arquivos que o sistema pode montar, e o arquivo /etc/mtab possui uma lista dos dispositivos montados (através do comando mount). Portanto, a **alternativa A** está correta e é o gabarito da questão.

**Gabarito:** Letra A

2. (UFRRJ/UFRRJ - 2015) Qual o comando para agregar uma mídia formatada em NTFS no sistema de arquivos de um sistema operacional UNIX-like?

- A) `umount -t ntfs /dev/diskid/DISK-JP1440HA196TKS /mnt.`
- B) `mount /mnt /dev/diskid/DISK-JP1440HA196TKS -t ntfs.`
- C) `umout /mnt /dev/diskid/DISK-JP1440HA196TKS -t ntfs.`
- D) `mount -t ntfs /dev/diskid/DISK-JP1440HA196TKS /mnt.`
- E) `umount /dev/diskid/DISK-JP1440HA196TKS -t ntfs /mnt.`

### Comentários:

A forma básica do comando mount possui dois parâmetros:



- Dispositivo contendo o sistema de arquivos a ser montado (ex.: /dev/sda3);
- Ponto de montagem (ex.: /teste).

Além disso, temos o parâmetro -t que serve para definir o tipo de sistema de arquivos.

Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

**3. (CESPE/TCE-RN - 2015)** O Linux tem a capacidade de montar sistemas de arquivos do dispositivo /dev/sdc3, que tem o sistema de arquivos do tipo ext4 e, no ponto de montagem /teste, deve ser utilizado o comando `mount -t ext4 /dev/sdc3 /teste`.

**Comentários:**

Semelhante à questão anterior. Portanto, a questão está **correta**.

**Gabarito:** Correta

**4. (FCC/TRT23 - 2016)** Uma partição NFS remota deve ser montada em um computador com sistema operacional Linux. Para especificar, no comando mount, que a partição é NFS, deve-se utilizar a opção:

- A) -n.
- B) -f.
- C) -i.
- D) -s.
- E) -t.

**Comentários:**

-n: monta s/ escrever em /etc/mtab;

-f: "fakes", útil pra debugar;

-i: não chama o /sbin/mount;

-s: tolera opções "sloopy", ao invés de falhar;

-t: especifica o tipo de sistema de arquivos.



Portanto, a **alternativa E** está correta e é o gabarito da questão.

5. (CESPE/TCE-PA - 2016) A opção `mount -t vfat` permite fazer o backup dentro de uma partição de disco DOS/Windows, caso em que a hierarquia de diretório aparece como parte do sistema de arquivos Linux.

#### Comentários:

Vimos que `-t` é o parâmetro para especificar o tipo de sistema de arquivos e que `"vfat"` especifica a família FAT, a qual é utilizada em sistemas DOS/Windows. Portanto, a questão está **correta**.

**Gabarito:** Correta



## QUESTÕES COMENTADAS – SUPERUSUÁRIO - MULTIBANCAS

1. (FCM/IF Sudeste-MG - 2016) O Linux permite ao superusuário (root) executar qualquer operação válida em qualquer arquivo ou processo, inclusive algumas chamadas de sistema podem ser executadas somente pelo superusuário. A operação que NÃO é exclusiva do superusuário é
- A) configurar interfaces de rede.
  - B) configurar o relógio do sistema.
  - C) alterar as permissões de um arquivo.
  - D) aumentar os limites de uso dos recursos.
  - E) definir o nome de host (hostname) do sistema.

### Comentários:

De todas as alternativas mostradas, sabemos que alterar as permissões de um arquivo qualquer um pode fazer, não é? Claro que qualquer um pode alterar as permissões (comando `chmod`) do seu arquivo e não de outros (a não ser que tenha permissão para isso). Portanto, a **alternativa C** está correta e é o gabarito da questão.

Gabarito: Letra C

2. (IBFC/EBSERH - 2016) O comando `sudo` do sistema operacional Linux é muito poderoso, permitindo que usuários comuns obtenham privilégios de super usuário. Por questões de segurança, o administrador precisa definir no arquivo \_\_\_\_\_, quais usuários podem executar `sudo`, em quais computadores podem fazê-lo e quais comandos podem executar através dele. Assinale a alternativa que complete correta e respectivamente a lacuna:
- A) `/etc/sudoers`
  - B) `/root/sudour`
  - C) `/root/sudoers`
  - D) `/usr/sudour`
  - E) `/etc/sudour`



### Comentários:

Pense o seguinte: os arquivos de configuração geralmente ficam no diretório "/etc". E para executar o sudo existe um arquivo com as configurações de quem e do que pode ser executado com o sudo, que é o arquivo "sudoers". Em inglês, quem caminha ("walk") é um "walker", então quem faz um "sudo" é um "sudoer", assim fica mais fácil lembrar 😊. Portanto, a **alternativa A** está correta e é o gabarito da questão.

Gabarito: Letra A

3. (INAZ do Pará/DPE-PR - 2017) Qual a alternativa que corresponde à linha de comando para que usuários comuns possam utilizar o comando administrativo apt-get dist-upgrade?

- A) # apt-get dist-upgrade
- B) \$ sudo apt-get gw dist-upgrade
- C) # sudo apt-get gw up dist-upgrade
- D) \$ sudo apt-get dist-upgrade
- E) # sudo apt-get dist-upgrade gw now

### Comentários:

Usuário comum tem o prompt "\$", então temos duas alternativas possíveis. Para executar o "apt-get dist-upgrade" com o "sudo" é só colocar o "sudo" na frente. Portanto, a **alternativa D** está correta e é o gabarito da questão.

Gabarito: Letra D

4. (CESPE/ABIN - 2018) O Linux não impede a alteração do nome do superusuário, nem a criação de contas com UID igual a 0, embora essas ações não sejam recomendadas.

### Comentários:

Como vimos na aula, não é recomendado fazer isso, mas é possível alterar o nome do root para qualquer coisa, e, também é possível criar novas contas de superusuário (com UID = 0). Portanto, a questão está **correta**.

Gabarito: Correta

5. (COSEAC/UFF - 2019) No Sistema Operacional Linux o diretório local padrão do superusuário é:



- A) /home.
- B) /usr.
- C) /root.
- D) /lib.
- E) /dev.

### Comentários:

/home – home dos usuários “comuns”.

/usr – onde a maioria dos programas ficam instalados, executáveis e bibliotecas de todos os principais programas.

/root – é o home do root!

/lib – bibliotecas necessárias para os binários essenciais para as pastas /bin e /sbin.

/dev – links para dispositivos de hardware.

Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C



## QUESTÕES COMENTADAS - USUÁRIOS, GRUPOS, PERMISSÕES DE ACESSO - MULTIBANCAS

1. (FUNCAB/MDA - 2014) No sistema operacional Linux, a criação de um grupo permite que um conjunto de usuários diferentes possua acesso a um mesmo arquivo. O comando do Linux que tem como objetivo alterar o grupo de um arquivo/diretório é o

- A) chown.
- B) chgroup.
- C) chset.
- D) chfile.
- E) chgrp.

### Comentários:

Sabemos que o chown permite a alteração de usuário/grupo, mas a questão fala somente em grupo. Tem uma pegadinha na letra B, mas seguindo o padrão de 5 letras (chown), a resposta é "chgrp". Portanto, a **alternativa E** está correta e é o gabarito da questão.

Gabarito: Letra E

2. (FGV/Prefeitura de Niterói-RJ - 2018) Pedro é o proprietário do arquivo header.txt em um sistema Linux e gostaria de assegurar que somente ele tivesse permissão de leitura, gravação e execução a este arquivo, enquanto que todos os demais usuários com acesso ao sistema tivessem somente a permissão de leitura.

Assinale a opção que indica o comando que pode ser usado para conseguir esse objetivo.

- A) chmod ug+r header.txt
- B) chmod 766 header.txt
- C) chmod 722 header.txt
- D) chmod +r header.txt
- E) chmod 744 header.txt

### Comentários:



Vamos fazer na sequência o UGO (usuário, grupo, outros):

$$U = rwx = 111 = 7$$

$$G = r-- = 100 = 4$$

$$O = r-- = 100 = 4$$

Portanto, a **alternativa E** está correta e é o gabarito da questão.

**Gabarito:** Letra E

**3. (FGV/MPE-AL - 2018) Para definir o script /usr/local/bin/meuscript como executável no sistema operacional Linux, devemos usar o comando**

- A) `chmod og /usr/local/bin/meuscript`
- B) `chmod 000 /usr/local/bin/meuscript`
- C) `chmod +x /usr/local/bin/meuscript`
- D) `chmod ugo-x /usr/local/bin/meuscript`
- E) `chmod 666 /usr/local/bin/meuscript`

#### **Comentários:**

A única que adiciona a execução "x" é a letra C. Como não especifica para "quem", todo o UGO recebe a permissão para a execução.

Na letra E, se transformarmos 666 em binário, temos: 110 110 110 = rw- rw- rw- (não tem "x" para ninguém!).

Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**4. (IF-ES/IF-ES - 2019) Os administradores de redes precisam entender com clareza sobre o gerenciamento de contas de usuários nos Sistemas Operacionais de Servidores Linux. Em relação ao sistema de contas Linux, analise as afirmativas abaixo:**

I – O arquivo /etc/shadow serve para manter senhas criptografadas seguras quanto a acesso não autorizado.

II – Um novo usuário pode ser adicionado utilizando-se ferramentas, como adduser ou useradd.



III – Os usuários dos sistemas Linux recebem um número UID referente a sua identificação, porém o usuário root, por questões de segurança, não dispõe de UID.

É CORRETO o que se afirma em:

- A) I, II e III.
- B) I e II, apenas
- C) I e III, apenas.
- D) II e III, apenas.
- E) Nenhuma das afirmativas está correta.

#### Comentários:

(I) O arquivo /etc/shadow serve para manter o hash das senhas, entre outras informações relacionadas. (II) Um novo usuário pode ser adicionado utilizando-se ferramentas, como adduser ou useradd e automaticamente os dados serão acrescentados nos arquivos /etc/passwd e /etc/shadow. (III) Os usuários dos sistemas Linux recebem um número UID referente a sua identificação. O root recebe o UID 0. Portanto, a **alternativa B** está correta e é o gabarito da questão.

**Gabarito:** Letra B

5. (CS-UFG/IF Goiano - 2019) Na configuração do Sistema Linux, o comando chmod modifica as permissões de arquivos com respeito à execução, leitura e escrita. A expressão chmod 751 arq2 é equivalente a:

- A) chmod u=rx,g=rwx,o=x arq2
- B) chmod u=rx,g=rwx,o=r arq2
- C) chmod u=rwx,g=rx,o=x arq2
- D) chmod u=rwx,g=rx,o=r arq2

#### Comentários:

U = 7 = 111 = rwx

G = 5 = 101 = r-x

O = 1 = 001 = --x



Tirando os "tracinhos":  $u=rwx, g=rx, o=x$

Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**6. (CCV-UFC/UFC - 2019) Qual dos itens abaixo contém o comando do sistema operacional Linux capaz de mudar o grupo de um arquivo ou diretório do sistema?**

- A) df
- B) sed
- C) chown
- D) chmod
- E) passwd

**Comentários:**

Questão interessante porque não mostra o comando "chgrp". Mas sabemos que o chown permite a alteração de usuário/grupo, ou seja, é possível trocar o grupo também! Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

**7. (FCC/TJ-MA - 2019) Um Analista digitou o comando  $chmod u=rwx, g=rx, o=r$  processo para definir as permissões de acesso ao arquivo processo. O comando equivalente usando a notação octal é:**

- A)  $chmod 713$  processo
- B)  $chmod 777$  processo
- C)  $chmod 134$  processo
- D)  $chmod 754$  processo
- E)  $chmod 671$  processo

**Comentários:**

$U = rwx = 111 = 7$



$$G = r-x = 101 = 5$$

$$O = 1-- = 100 = 4$$

Então, temos "chmod 754 processo".

Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D



## QUESTÕES COMENTADAS - COMANDOS BÁSICOS - MULTIBANCAS

1. (FGV/Câmara Municipal do Recife-PE - 2014) No Linux, o comando

```
cat something | grep 'something'
```

tem o efeito de:

- A) mostrar as linhas do arquivo something que incluem a palavra something;
- B) adicionar a palavra something ao conteúdo do arquivo something;
- C) criar uma cópia do arquivo something denominada grpe;
- D) mover o arquivo something para um novo diretório denominado something;
- E) restringir o acesso ao arquivo something ao usuário something.

### Comentários:

O `cat NOME_ARQUIVO` mostra o conteúdo desse arquivo como saída, mas como há um pipe (`|`) em seguida, essa saída vira a entrada para o comando seguinte. Então, o `grep` realiza a pesquisa do termo 'something' sobre o conteúdo do arquivo e mostra todas as linhas que possuem tal termo.

Portanto, a **alternativa A** está correta e é o gabarito da questão.

**Gabarito:** Letra A

2. (VUNESP/DESENVOLVESP - 2014) O superusuário do sistema operacional Linux deseja verificar a informação da última linha do arquivo `/var/log/dmesg`. Para efetivar essa ação por meio do uso de apenas um comando, ou seja, sem qualquer outra ação, deve-se executar, na linha de comando:

- A) `$ cat /var/log/dmesg`
- B) `$ cat /var/log/dmesg >`
- C) `$ more /var/log/dmesg >>`
- D) `$ tail /var/log/dmesg`
- E) `$ top /var/log/dmesg`



### Comentários:

tail: a tradução literal é “rabo”, então pense no rabo de um cachorro, é o que fica por último...é o propósito desse comando, mostrar a última parte de um arquivo. Então, se um arquivo é pequeno, tanto faz usar o “cat” ou o “tail”, mas se for grande, depende o que o usuário quer ver!

Como a questão fala em ver a última linha, o tail pode ser utilizado, mostrando não só a última, mas as últimas linhas do arquivo.

Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

**3. (VUNESP/SP-URBANISMO - 2014)** Um usuário padrão do sistema operacional Linux, “logado” no seu diretório home, deseja saber o diretório no qual está instalado o comando mount. Para obter tal informação, esse usuário poderá utilizar o seguinte comando:

- A) cat mount
- B) find mount
- C) grep mount
- D) where mount
- E) which mount

### Comentários:

which (“qual”): mostra o caminho completo de determinado comando. Abaixo vemos, por exemplo, o caminho completo dos comandos mount, ls e cat. Portanto, a **alternativa E** está correta e é o gabarito da questão.

**Gabarito:** Letra E

**4. (FCC/TRF4 - 2014)** Marcos, usuário de um computador com sistema operacional Linux Red Hat listou o conteúdo do seu diretório home e observou a presença do arquivo manual.txt com 31.251 bytes de tamanho, o que representa cerca de 20 páginas de texto se visualizado em um terminal Linux padrão. Para que Marcos possa visualizar diretamente o final do arquivo manual.txt, sem a necessidade de iniciar a visualização a partir do começo do arquivo, ele deve executar o comando:

- A) cat manual.txt | more



- B) more manual.txt
- C) list manual.txt | end
- D) tail manual.txt
- E) cat manual.txt | end

### Comentários:

tail: a tradução literal é "rabo", então pense no rabo de um cachorro, é o que fica por último...é o propósito desse comando, mostrar a última parte de um arquivo. Então, se um arquivo é pequeno, tanto faz usar o "cat" ou o "tail", mas se for grande, depende o que o usuário quer ver! Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

### 5. (FGV/TJ-BA - 2015) O seguinte comando, invocado em uma shell do Linux:

```
find . -type f
```

escreverá na saída padrão:

- A) o número de arquivos regulares descendentes do diretório corrente;
- B) os caminhos (pathnames) relativos para os arquivos regulares descendentes do diretório corrente;
- C) os caminhos (pathnames) absolutos para os arquivos regulares descendentes do diretório corrente;
- D) os caminhos (pathnames) relativos para os diretórios descendentes do diretório corrente;
- E) os caminhos (pathnames) absolutos para os arquivos regulares e diretórios descendentes do diretório corrente.

### Comentários:

O ponto (.) indica diretório corrente e o "-type f" indica apenas os arquivos (files), sem os diretórios:

```
[root@localhost teste]# ls
numeros.txt  oi.txt      opa        tchau.txt
[root@localhost teste]# find . -type f
./oi.txt
./tchau.txt
./numeros.txt
[root@localhost teste]# █
```



Portanto, a **alternativa B** está correta e é o gabarito da questão.

Gabarito: Letra B

6. (IF-RR/IF-RR - 2015) O comando find permite buscar arquivos filtrando por nome, tamanho, data, dono, grupo, permissões, tipo e outros. A busca é recursiva, por padrão, mas pode ser restringida a uma determinada profundidade. O comando GNU find faz parte das findutils GNU e está instalado em cada sistema Ubuntu. Um usuário possui uma distribuição Ubuntu14.04 LTS e precisa encontrar um arquivo de áudio com tamanho superior a 5MB, dentre as opções abaixo, qual alternativa melhor representa o comando para a busca pelo arquivo.

- A) find \$HOME -iname '\*.mp4' -size -5M
- B) find \$HOME -iname '\*.mp4' -size +5M
- C) find \$HOME -iname '\*.jpeg' -size +5M
- D) find \$HOME -iname '\*.ogg' +size -5M
- E) find \$HOME -iname '\*.ogg' -size +5M

#### Comentários:

\$HOME indica o caminho do diretório home do usuário, -iname indica o nome com case insensitive. A questão fala em tamanho maior que 5MB, então o parâmetro correto é "-size +5M", e das extensões mostradas, o referente a áudio é o ".ogg". Poderia haver confusão com o ".mp4", mas lembre-se que esse formato é de áudio e vídeo, enquanto o ".ogg" é apenas de áudio! Portanto, a **alternativa E** está correta e é o gabarito da questão.

Gabarito: Letra E

7. (FCM/IF Sudeste-MG - 2016) Alguns comandos em Linux podem combinar diferentes parâmetros, como o comando a seguir:

```
$ find /home/usuario/ -name '*.txt' -mtime 2;
```

Esta linha de comando procura

- A) todos os arquivos com extensão ".txt", dentro da pasta "/home/usuario".
- B) e remove todos os arquivos com extensão ".txt" dentro da pasta "/home/usuario".



- C) todos os arquivos com extensão “.txt”, modificados a mais de 2 dias, dentro da pasta “/home/usuario” e subpastas.
- D) todos os arquivos modificados a mais de 2 dias, dentro da pasta “/home/usuario”, e subpastas, e os move para a pasta “name”.
- E) e copia para o diretório corrente todos os arquivos com nome “\*.txt”, criados a menos de 2 dias, dentro da pasta “/home/usuario”.

#### Comentários:

Procura em “/home/usuario/” os arquivos com qualquer nome, desde que terminem com “.txt”, com a data de modificação (mtime) 2 dias (incluindo os “quebrados”, ou seja, 2 dias e pouco...). Portanto, a **alternativa C** está correta e é o gabarito da questão.

**Gabarito:** Letra C

8. (FCM/IFF-RS - 2016) Em um ambiente Linux, desejam-se encontrar todos os arquivos com a extensão “.sh” a partir do diretório corrente. O comando a ser utilizado é o

- A) find -all \*.sh /
- B) grep - all \*.sh
- C) find -user -i .sh
- D) ps -name \ \*.sh
- E) find . -name \\*.sh

#### Comentários:

Para encontrar os caminhos de determinados arquivos, utilizamos o find. O ponto (.) indica o diretório corrente e o parâmetro -name indica o nome dos arquivos que se procura, ou seja, “\*.sh”. Portanto, a **alternativa E** está correta e é o gabarito da questão.

**Gabarito:** Letra E

9. (IESES/BAHIAGÁS - 2016) O comando find está presente em vários sistemas operacionais Unix-like, sendo muito conhecido pela sua versatilidade. Considere o uso do bash em uma distribuição de Linux Ubuntu 14.04, na qual o comando find esteja disponível. Um usuário gostaria então de listar somente arquivos de extensões ‘.java’ e ‘.py’ em seu diretório ‘/home/usuario’. Assinale a alternativa que executa esta tarefa corretamente.



- A) `find /home/usuario -type f -iname '*.java' -iname '*.py'`
- B) `find /home/usuario -type d -a \( -iname '*.java' -o -iname '*.py' \)`
- C) `find /home/usuario -type f -o \( -iname '*.java' -o -iname '*.py' \)`
- D) `find /home/usuario -type f -a \( -iname '*.java' -o -iname '*.py' \)`
- E) `find /home/usuario -type f -a \( -iname '*.java' -a -iname '*.py' \)`

#### Comentários:

O foco são os arquivos, então utiliza-se o “-type f”. Os arquivos desejados são os com extensão .java e .py, então deve-se pesquisar os arquivos .java OU .py, ou seja, utiliza-se o parâmetro “-o” para indicar o OU. Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

**10.(CESPE/FUNPRESP-JUD - 2016) Os comandos find e which podem ser utilizados para a obtenção de informações acerca de programas instalados no Linux; o primeiro localiza o caminho (path) onde o comando está instalado, e o segundo, a versão do programa.**

#### Comentários:

find: procura por arquivos em uma hierarquia de diretórios.

which (“qual”): mostra o caminho completo de determinado comando.

O find procura qualquer tipo de arquivo/diretório, e o which mostra o caminho completo de um comando (não a sua versão). Portanto, a questão está **errada**.

**Gabarito:** Errada

**11.(INSTITUTO AOCP/EBSERH - 2016) Um administrador de sistema operacional Linux necessita procurar determinados textos, dada uma palavra, dentro de um arquivo texto. O comando que possibilita executar essa operação é o**

- A) find.
- B) more.
- C) whereis.
- D) grep.



E) pipe.

### Comentários:

grep: mostra as linhas de um arquivo que "batem" (match) com um determinado padrão. Um exemplo comum é utilizar o cat para "visualizar" o arquivo e a saída dele passar como entrada para o grep, através do pipe (|). Ao lado podemos ver um exemplo em que é aplicado o cat no arquivo numeros.txt e em cima dessa saída (visualização do conteúdo de numeros.txt) é aplicado o grep 3, ou seja, mostra todas as linhas do arquivo que possuem o número 3. Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

**12. (IESES/BAHIAGÁS - 2016) Qual dos comandos apresentados a seguir pode ser utilizado para juntar o conteúdo de um arquivo texto de log do Linux, denominado de log1, ao conteúdo de um arquivo texto de log, denominado log2?**

A) cat log2 << log1

B) cat log2 < log1

C) cat log1 || log2

D) cat log1 > log2

E) cat log1 >> log2

### Comentários:

cat: concatena arquivos e mostra na saída padrão. Sintaxe: cat ARQ1 ARQ2. Mas se for utilizado o direcionador "simples" (>), coloca o conteúdo do primeiro arquivo sobre o conteúdo do segundo e não mostra na tela. Se for utilizado o direcionador "duplo" (>>), concatena o primeiro ao segundo e não mostra na tela. Quando usado com apenas um arquivo, mostra seu conteúdo. Abaixo um exemplo com o direcionador duplo e depois somente com um arquivo.

```
[root@localhost teste]# cat oi.txt >> tchau.txt
[root@localhost teste]# cat tchau.txt
tchau
oi
[root@localhost teste]#
```

Portanto, a **alternativa E** está correta e é o gabarito da questão.

**Gabarito:** Letra E



13.(IESES/BAHIAGÁS - 2016) Um administrador de sistema gostaria de acompanhar o arquivo de log `'/var/log/apache.log'` em tempo real. Este arquivo está em um servidor Linux Ubuntu 14.04. Supondo que o administrador possua acesso de superusuário ao sistema através do bash. Qual dos comandos a seguir, executado uma única vez, poderia auxiliar o administrador a acompanhar este log dinamicamente (sem interferência adicional)?

- A) `tail -rt /var/log/apache.log`
- B) `more -rt /var/log/apache.log`
- C) `more -f /var/log/apache.log`
- D) `cat /var/log/apache.log`
- E) `tail -f /var/log/apache.log`

#### Comentários:

Para ver os dados no fim do arquivo o comando é `tail`. Uma opção para atualizar a exibição de dados que são inseridos no arquivo é a `"-f"` (follow = seguir). Portanto, a **alternativa E** está correta e é o gabarito da questão.

Gabarito: Letra E

14.(CONSULPLAN/Câmara de Belo Horizonte-MG - 2018) No Sistema Operacional GNU/Linux, o comando `cat` é utilizado para concatenar arquivos. Sua sintaxe básica é: `cat [opções] arquivo`. Uma das opções utilizadas é `-A`, que tem como função:

- A) Numerar todas as linhas na saída.
- B) Exibir todos os caracteres especiais.
- C) Numerar apenas as linhas não vazias.
- D) Exibir caracteres não passíveis de impressão (caracteres de controle).

#### Comentários:

Se o usuário quiser ver caracteres "não imprimíveis", pode utilizar alguns parâmetros:

- `-v, --show-nonprinting`: utiliza `"^"` e `"M-"` para mostrar caracteres não imprimíveis, exceto para Line Feed (quebra de linha) e TAB;
- `-E, --show-ends`: mostra `"$"` no fim de cada linha;
- `-T, --show-tabs`: mostra TAB como `"^I"`



- -A, --show-all (mostra tudo): equivalente aos três anteriores juntos (-vET).

Portanto, a **alternativa B** está correta e é o gabarito da questão.

**Gabarito:** Letra B

**15.(IDECAN/IF-PB - 2019) A respeito dos comandos que podem ser utilizados para verificar o uso de recursos em um sistema operacional Linux, assinale a alternativa que indica, respectivamente, o nome do comando responsável por exibir informações referentes ao uso da memória RAM e uso do espaço em disco.**

- A) cat e du
- B) locate e df
- C) cat e free
- D) free e df
- E) du e free

**Comentários:**

df: mostra a quantidade de espaço em disco disponível no sistema de arquivos contendo cada argumento de nome de arquivo. Se nenhum nome de arquivo é fornecido, o espaço disponível em todos os sistemas de arquivos montados é mostrado.

free: mostra a quantidade de memória utilizada e livre no sistema.

Portanto, a **alternativa D** está correta e é o gabarito da questão.

**Gabarito:** Letra D

**16.(COSEAC/UFF - 2019) No Linux, o comando para listar o conteúdo de um diretório e o comando que mostra o caminho por inteiro do diretório em que o usuário se encontra são, respectivamente:**

- A) find e more.
- B) cat e grep.
- C) lpr e cat.
- D) df e finger.



E) ls e pwd.

### Comentários:

ls: lista o conteúdo de um diretório. Os parâmetros mais cobrados são:

- -a (all): todos os arquivos, não ignora as entradas que começam com ".";
- -l (long): utiliza um formato de listagem longa;
- -s (size): imprime o tamanho de cada arquivo.

pwd (print working directory): imprime o caminho completo do diretório de trabalho corrente. Portanto, a **alternativa E** está correta e é o gabarito da questão.

**Gabarito:** Letra E

17.(VUNESP/Câmara de Tatuí-SP - 2019) Considere a execução da sequência de comandos a seguir, em um terminal shell do Linux:

```
# cd /root
```

```
# mkdir -p /root/foo/bar
```

```
# pwd
```

- O resultado impresso na tela após a execução do último comando será:

A) /root

B) /root/foo/bar

C) bar

D) /root/foo

E) /foo/bar

### Comentários:

O comando cd troca de diretório, "indo" para /root.

O comando mkdir cria o diretório /root/foo/bar. O argumento "-p" faz com que, se houver erro no comando, que tente o diretório pai ("parent"). Vamos considerar que não houve erro.

O comando pwd mostra o diretório corrente. Aí tem uma leve pegadinha, pois foi criado o diretório /root/foo/bar, mas o diretório corrente ainda é o /root, pois o comando cd fez o "deslocamento" até esse diretório e não saiu mais! Logo, pwd deve retornar /root.



Portanto, a **alternativa A** está correta e é o gabarito da questão.

**Gabarito:** Letra A

**18.(CS-UFG/IF Goiano - 2019) No sistema operacional GNU/Linux, usando a linha de comando, deseja-se executar o seguinte: a) criar um diretório chamado IFG; b) criar um arquivo de texto chamado Concurso.txt; c) apagar o arquivo de texto Concurso.txt; d) apagar o diretório IFG. Qual é a sequência de comandos a ser empregada? Obs.: o sinal “,” é um mero separador entre os comandos**

- A) mkdir IFG , touch Concurso.txt , rm Concurso.txt , rmdir IFG
- B) mkdir IFG , create Concurso.txt , kill Concurso.txt , killdir IFG
- C) create IFG , touch Concurso.txt , rm Concurso.txt , rmdir IFG
- D) create IFG , create Concurso.txt , kill Concurso.txt , killdir IFG

#### **Comentários:**

Questão bem objetiva. O mkdir serve para criar um diretório. Uma forma de criar um arquivo texto é através do comando touch. Para remover um arquivo utilizamos o comando rm. E para excluir um diretório, temos o comando rmdir. Portanto, a **alternativa A** está correta e é o gabarito da questão.

**Gabarito:** Letra A



## LISTA DE QUESTÕES - SERVIÇOS DE REDE – DNS E DHCP - MULTIBANCAS

1. (FGV/MEC - 2009) No UNIX, o DNS é implementado por meio do software Berkeley Internet Name Domain (BIND), que segue a filosofia cliente/servidor.

O cliente do BIND é denominado "resolver", responsável por gerar as "queries" enquanto que o servidor do BIND é chamado por "named", responsável por responder às "queries".

Configurações BIND são descritas pelo tipo de serviço a ser executado pelo software, existindo quatro níveis de serviço que podem ser definidos nessas configurações: "resolver-only systems", "caching-only servers", "primary servers" e "secondary servers".

Os parâmetros que definem a configuração do "resolver-only systems" são inseridas no seguinte arquivo:

A) /etc/resolv.sys

B) /etc/resolv.cfg

C) /etc/resolv.bind

D) /etc/resolv.conf

E) /etc/resolv.parm

2. (FCC/TCE-SP - 2010) Em Linux, o /etc/resolv.conf é um arquivo texto simples, com um parâmetro por linha e especificações de endereços de servidores DNS. Nesse arquivo existem três palavras chaves normalmente usadas, que são

A) domain, search e resolv.

B) search, resolv e order.

C) domain, search e nameserver.

D) search, nameserver e order.

E) search nameserver e resolv.

3. (FCC/TCE-SP - 2010) No Linux, ele é muito útil para máquinas que são acessadas frequentemente, pois a inclusão de um computador neste arquivo dispensa a consulta a um



servidor DNS para obter um endereço IP, sendo muito útil para máquinas que são acessadas frequentemente. Trata-se de:

A) /etc/nameserver

B) /etc/localhost

C) /etc/ipsec

D) /etc/hosts

E) /etc/dnshost

4. (AOCP/UFGD - 2014) Em algumas distribuições Linux encontramos um arquivo que identifica os locais dos computadores de servidor de nome DNS, computadores estes que são usados pelo TCP/IP para traduzir os nomes de Internet. Assinale a alternativa que apresenta o nome deste arquivo.

A) DNS.conf.

B) resolver.conf.

C) resolv.conf.

D) translate.conf.

E) change.conf.

5. (IADES/TRE-PA - 2014) Um servidor Linux pode hospedar o serviço de resolução de nomes de uma rede de computadores. Conhecido por DNS, esse serviço é indispensável em uma rede que possua conexão com a internet. O nome de um pacote que implementa o DNS, muito utilizado em sistemas operacionais Linux, é

A) Firefox.

B) Apache.

C) Squid.

D) Postfix.

E) BIND.



6. (FCC/CNMP - 2015) O serviço de nome de domínios (DNS) possui uma arquitetura do tipo cliente/servidor na qual a base de dados é distribuída por toda internet. Nessa arquitetura, o acesso ao servidor DNS para buscar o relacionamento IP/Domínio é feito pelo cliente que é o

- A) Browser.
- B) DNS Cache.
- C) DNS Resolver.
- D) DNS Searcher.
- E) Gateway.

7. (CESPE/FUB - 2015) O protocolo DNS (domain name service), localizado no nível de aplicação da camada de transporte do TCP, é responsável pelo mapeamento de nomes e de endereços.

8. (FGV/TCE-SE - 2015) Um programa precisa simular o comportamento de um cliente DNS. Para funcionar adequadamente, o programa precisa enviar as consultas para um servidor DNS, especificamente para a sua porta:

- A) udp/23
- B) icmp/34
- C) tcp/22
- D) ip/50
- E) udp/53

9. (CESPE/TRE-MT - 2015) Acerca do servidor DNS/BIND (Domain Name System/Berklay Internet Domain), cuja funcionalidade é resolver nomes da rede, assinale a opção correta.

- A) Cada domínio tem seus registros de recursos e o registro de domínio denominado NS (name server), o qual é utilizado para definir propriedades básicas do domínio e sua zona.
- B) Um servidor DNS utiliza LDAP para fazer armazenamento das zonas de domínio para uma rápida resolução de um nome.
- C) O BIND, que utiliza a porta 53, é um programa de código aberto utilizado pela maior parte dos servidores DNS.



D) Os domínios de um servidor DNS são organizados na Internet sobre uma estrutura de dados do tipo lista encadeada, sendo o primeiro elemento da lista um ponto.

E) O protocolo HTTP implementa, por padrão, um servidor de resolução de nomes amplamente utilizado na Internet conhecido como DNS.

**10.(IADES/CRC-MG - 2015) Quanto ao programa que é normalmente utilizado para testar se um servidor DNS (Domain Name Server) está funcionando corretamente, ou seja, resolvendo nomes para os endereços IP, assinale a alternativa correta.**

A) ping

B) tracert

C) nslookup

D) ipconfig

E) net host

**11.(FGV/TCE-SE - 2015) Em um equipamento rodando Linux, um administrador deseja alterar o endereço default do servidor DNS usado para consultas. Para isso, ele precisa alterar o arquivo:**

A) /etc/named.conf

B) /etc/bind.conf

C) /etc/network.conf

D) /etc/dns.conf

E) /etc/resolv.conf

**12.(FGV/COMPESA - 2016) O sistema de Nomes de Domínio - DNS permite transformar nomes digitados em um navegador WEB em um endereço de rede. O nome do host e o nível do domínio para o domínio "system.master.com" são, respectivamente,**

A) system e segundo nível.

B) system e terceiro nível.

C) master e terceiro nível.

D) master e segundo nível.



E) .com e terceiro nível.

**13.(FCC/PRODATER - 2016) No Serviço de Nomes de Domínio – DNS existem diferentes tipos de servidores distribuídos hierarquicamente que armazenam informações também de forma hierárquica. Considerando o nome: www.empresa.com, o domínio .com é gerenciado pelo servidor**

A) Global.

B) PDR.

C) Authoritative.

D) TLD.

E) Root.

**14.(IBFC/EBSERH-HUAP - 2016) O Domain Name System (DNS) é um sistema de gerenciamento de nomes hierárquicos e distribuídos. A funcionalidade do DNS Reverso vem a ser:**

A) reverter a funcionalidade básica de um DNS padrão obtendo o endereço MAC.

B) resolver um endereço IP, buscando o nome de domínio associado ao host.

C) resolver o nome do domínio de um host qualquer para seu endereço IP correspondente.

D) com base na região geográfica, obter automaticamente um endereço IP local.

E) resolver o problema da reversão do IPv6 para o IPv4 de uma forma rápida e automática.

**15.(FUNRIO/CM Nova Iguaçu - 2016) Um administrador de rede está configurando o DNS de um servidor e vai trabalhar com o registro que define as características da zona a ser configurada, tais como, o nome da zona e o nome do servidor, que é a autoridade para a referida zona, ou seja, o servidor DNS no qual está a zona que foi criada originalmente. Esse registro é conhecido pela sigla**

A) HINFO.

B) MX.

C) SOA.

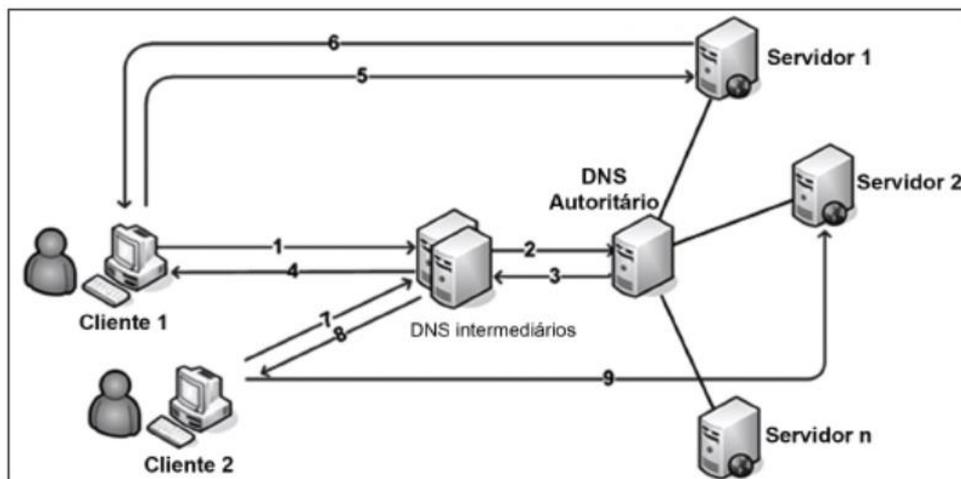
D) CNAME.



16.(CESPE/TCE-PR - 2016) Uma consulta DNS inicial típica, originada de uma máquina de usuário e encaminhada ao servidor de nomes local para um nome de domínio externo não armazenado em cache DNS, será do tipo

- A) raiz.
- B) domínio de alto nível.
- C) iterativa.
- D) recursiva.
- E) direta.

17.(FCC/ELETROSUL - 2016) O balanceamento de carga por DNS é uma abordagem popular e simples para as solicitações de balanceamento de servidores. Considere a figura abaixo.



De acordo com a figura, geralmente são estes os passos que ocorrem assim que é feita uma consulta DNS:

1. Quando um cliente tenta acessar o site, é realizada uma pesquisa no DNS local para determinar qual é o endereço I..... correspondente;
2. O pedido de endereço chega ao servidor de DNS II..... do domínio;
3. A primeira vez que esta consulta é feita, o servidor DNS remoto pode retornar todos os registros de endereços que ele tem para o site;
4. O servidor DNS III....., em seguida, determina o endereço de registro para retornar ao cliente;
5. Se todos os registros são retornados, o cliente utilizará o primeiro que lhe é atribuído;
6. O servidor responde ao cliente e atende ao pedido;



7. A cada pedido, o algoritmo Round Robin roda os endereços e retorna pela ordem em que eles estão;

8 . Cada consulta DNS irá resultar em um cliente usando um endereço IV.....;

9. Esta rotação de endereços irá distribuir pedidos para os servidores.

As lacunas de I a IV são, correta e respectivamente, preenchidas com:

- A) TCP – autoritário – local – TCP igual ao obtido em 1
- B) IP – autoritário – local – IP diferente
- C) do servidor – local – autoritário – de um servidor
- D) do servidor – autoritário – local – do servidor igual ao obtido em 1
- E) IP – local – autoritário – IP igual ao obtido em 1

**18.(CESPE/TCE-SC - 2016)** Após o servidor local SMTP aceitar uma mensagem para subsequente envio, é necessário determinar o endereço do servidor de email do destinatário. Essa etapa é realizada mediante consulta DNS a um servidor de nomes capaz de prover a informação, no qual serão verificados os registros especiais MX (mail exchange).

**19.(CESPE/Polícia Científica-PE - 2016)** Na operação de um serviço DNS, se uma consulta a um servidor de alto nível retornar uma resposta positiva para o servidor DNS local, o mapeamento nome/endereço será

- A) armazenado definitivamente no banco de dados do servidor DNS local.
- B) replicado para todos os servidores DNS alcançáveis para que estejam disponíveis a outras consultas.
- C) repassado pelo módulo tradutor a um servidor raiz para confirmar a resposta recebida.
- D) armazenado em cache local e pode ser mantido por um tempo definido por configuração.
- E) repassado ao módulo tradutor do provedor de serviços para realizar a entrega à aplicação.

**20.(IBFC/Polícia Científica-PR - 2017)** Servidores de DNS (Domain Name System) têm como função converter endereços IP em seu respectivo nome e vice-versa. Para sua configuração, são utilizados arquivos denominados mapas de domínio (zone). Esses arquivos são compostos por entradas chamadas RR (Resource Record). O tipo básico de RR que estabelece a correspondência entre um nome canônico e um endereço IP é indicado por:



- A) PTR
- B) MX
- C) NS
- D) A
- E) SOA

**21.**(IBFC/Polícia Científica-PR - 2017) Servidores DNS (Domain Name Server) são responsáveis pela conversão do nome dos diversos servidores espalhados pela Internet para seu número IP e vice-versa. Servidores de DNS trabalham de forma colaborativa e hierárquica. Assinale a alternativa a que apresenta o nome dado aos servidores que se encontram no topo da hierarquia de DNS:

- A) Root Name Servers
- B) Main servers
- C) International Name Servers
- D) Controllers Servers
- E) Master Servers

**22.**(CESPE/Polícia Federal - 2018) As atualizações entre servidores DNS utilizam o UDP, enquanto as consultas feitas a servidores DNS utilizam o TCP (ou, opcionalmente, o SCTP).

**23.**(IF-RS/IF-RS - 2018) Considerando que se está logado como "root" no terminal de uma estação de trabalho com um sistema operacional Linux debian ou derivado, qual comando configura o DNS para 8.8.8.8?

- A) dns 8.8.8.8
- B) echo 8.8.8.8 > /etc/hosts.conf
- C) echo nameserver 8.8.8.8 > /etc/resolv.conf
- D) nslookup 8.8.8.8
- E) nameserver 8.8.8.8



24.(FAURGS/BANRISUL - 2018) Um cliente DNS, ao fazer a requisição DNS para resolver o nome `www.banrisul.com.br`, recebe como resposta uma mensagem do tipo `non authoritative`. Esse tipo de resposta é obtido por meio de um registro DNS (`resource record - RR`) armazenado

- A) em um cache de DNS.
- B) em um servidor interativo sem cache DNS.
- C) em um servidor recursivo sem cache DNS.
- D) no arquivo de zona do DNS primário.
- E) no arquivo de zona do DNS secundário.

25.(FAPEC/UFMS - 2018) Considere as afirmações a seguir sobre a instalação e configuração de um servidor de DNS (Bind9) no sistema operacional Debian 9.

I - Quando o servidor local de DNS não consegue traduzir o nome, ele deve solicitar a um servidor público (`forwarders`). O Google Public DNS é um exemplo de `forwarders` e deve ser configurado no arquivo `named.conf.options`.

II - A instalação do servidor pode ser realizada pelo usuário `root` com o comando: `apt install bind9 bind9-doc dnsutils`

III - A lista de controle de acessos ou ACL é definida no arquivo `named.conf.options`.

Está(ão) correta(s):

- A) Apenas I.
- B) Apenas II.
- C) Apenas III.
- D) Apenas II e III.
- E) I, II e III.

26.(CESPE/EMAP - 2018) Em um servidor DNS que esteja utilizando Bind, o arquivo `/etc/named.conf` mantém as configurações de funcionamento do serviço DNS. Nesse servidor, é possível configurar a porta que deve escutar o serviço; assim, para que o serviço escute a porta 53, deve-se configurar a opção seguinte.

`Listening 53 { any;};`



27.(CESPE/TRT8 - 2013) Para alterar o arquivo de configuração do serviço de DHCP no Linux, deve-se acessar o diretório

- A) /dev.
- B) /mnt.
- C) /etc.
- D) /usr/bin.
- E) /boot.

28.(SUGEP-UFRPE/UFRPE - 2018) Para acesso à Internet, servidores Windows e Linux utilizam um serviço para atribuição dinâmica de IPs aos microcomputadores conectados à rede, em que é marcada a opção associada a "o IP será atribuído automaticamente pelo servidor", na configuração.

Esse serviço é conhecido pela sigla:

- A) DNS
- B) HTTP
- C) DHCP
- D) TCP
- E) IP



## GABARITO



## GABARITO

1- D

2- C

3- D

4- C

5- E

6- C

7- Errada

8- E

9- C

10- C

11- E

12- B

13- D

14- B

15- C

16- D

17- B

18- Correta

19- D

20- D

21- A

22- Errada

23- C

24- A

25- E

26- Errada

27- C

28- C



## LISTA DE QUESTÕES - SERVIÇOS DE REDE – FTP E E-MAIL - MULTIBANCAS

1. (CESPE/HEMOBRÁS - 2008) Tanto o Linux quanto o Windows possuem softwares clientes para FTP (file transfer protocol) e que podem ser utilizados para a troca de arquivos em uma rede local.
2. (INSTITUTO AOCP/UFPB - 2014) Um estagiário executou o comando FTP no servidor Linux. Esse comando possibilita
  - A) permitir a abertura de portas para um acesso remoto a um servidor da rede.
  - B) realizar a manutenção de disco e de memória de um computador remotamente.
  - C) trocar informações de desempenho de redes utilizando um canal de comunicação protegido e exclusivo.
  - D) realizar uma verificação de permissões de acesso na rede e no computador que o usuário está utilizando.
  - E) trocar arquivos entre computadores interligados, independente do sistema operacional e da rede utilizada.
3. (VUNESP/SAAE de Barretos-SP - 2018) Os serviços de acesso a dados da internet são disponibilizados em uma arquitetura Cliente/Servidor. Por exemplo, considerando um servidor com sistema operacional Linux, o serviço padrão que responde às requisições ftp é denominado
  - A) ftpd.
  - B) ftps.
  - C) ftpsrv.
  - D) samba.
  - E) smb.
4. (FUNIVERSA/PC-DF - 2012) Em um servidor de e-mail com sistema operacional Linux, especialmente quando se utiliza o serviço Postfix, pode-se auditar o tráfego de entrada e saída de mensagens de correio eletrônico por meio dos registros conhecidos por logs. Em que



arquivo desse sistema operacional encontram-se os logs das atividades de envio e recebimento de mensagens de correio eletrônico (e-mail)?

- A) No diretório `"/etc/postfix"`.
- B) No arquivo `"/var/log/maillog"`.
- C) Na pasta `"/var/mail/<nome_do_usuario>"`.
- D) No arquivo `"/usr/local/bin/postfix"`.
- E) No arquivo de registro `"/dev/tty"`.

5. (FUNCAB/MDA - 2014) São exemplos de servidores de correio eletrônico do Linux:

- A) postfix e sendmail.
- B) fetchmail e sendmail.
- C) pine e postfixe.
- D) mail e pine.
- E) samba e postfixe.

6. (FGV/TJ-GO - 2014) A empresa Y passou a adotar uma política de uso de software livre e resolveu mudar o seu correio eletrônico, que era baseado em Microsoft Exchange. O novo ambiente deveria rodar em ambiente Linux. Uma das possíveis escolhas do novo software de correio eletrônico recai no programa:

- A) nginx;
- B) puppet;
- C) apache;
- D) postfix;
- E) bind.

7. (FCC/Prefeitura de Manaus-AM - 2019) Um Assistente de TI foi incumbido de configurar o protocolo para ser utilizado na troca de mensagens entre dois servidores de e-mail. A escolha correta do protocolo para essa finalidade é:



- A) IMAP.
- B) POP3.
- C) SNMP.
- D) SMTP.
- E) POP4.

## GABARITO



## GABARITO

- 1- Correta
- 2- E
- 3- A

- 4- B
- 5- A
- 6- D

- 7- D



## LISTA DE QUESTÕES - GERENCIAMENTO DE PROCESSOS - MULTIBANCAS

1. (CESPE/TJ-SE - 2014) Alguns programas podem apresentar problemas que resultem no travamento do sistema operacional, o que pode ser resolvido, no Linux, por meio do comando Kill, que finaliza o processo, funcionalidade que pode ser acessada por meio de outro terminal.
2. (FCC/CNMP - 2015) O comando ps do Linux permite parâmetros (ou opções) para mostrar:
  - I. os processos criados por você e de outros usuários do sistema.
  - II. o nome de usuário que iniciou o processo e hora em que o processo foi iniciado.
  - III. as variáveis de ambiente no momento da inicialização do processo.
  - IV. a árvore de execução de comandos (comandos que são chamados por outros comandos).Para realizar o que descrevem os itens I, II, III e IV são utilizados, correta e respectivamente, os parâmetros

A) f / u / e / a

B) f / a / e / u

C) u / e / f / a

D) e / u / a / f

E) a / u / e / f

3. (FCC/TRT14 - 2016) Para listar todos os processos que estavam em execução em um computador com o sistema operacional Linux instalado, um usuário utilizou o comando ps. Para que esse comando exiba informações detalhadas de cada processo, como o nome do usuário que iniciou o processo, o número identificador do processo, a porcentagem de utilização da CPU e da memória pelo processo e a hora em que cada processo foi iniciado, este comando deve ser utilizado com o parâmetro

A) aux.

B) -top.

C) vim.

D) pstree.



E) -zcf.

**4. (FGV/IBGE - 2017) Em ambiente Linux, a chamada ao sistema (system call) que implementa a criação de um novo processo é:**

A) create\_process;

B) new\_process;

C) fork;

D) spawn;

E) duplicate.

**5. (FGV/Prefeitura de Niterói-RJ - 2018) Sobre o gerenciamento de processos no sistema operacional Linux, assinale a afirmativa correta.**

A) O comando ps lista os processos em execução no sistema, trazendo informações sobre o quanto de processamento ou de memória cada um deles está consumindo.

B) O ID do processo não pode ser reutilizado depois que o processo termina.

C) Por meio do comando renice é possível alterar a prioridade de execução de um processo.

D) Por padrão, os processos executados por um usuário iniciam com nível de prioridade -20, isto é, a prioridade mais baixa possível.

E) Processos órfãos não possuem PPID.

**6. (CESPE/EBSERH - 2018) No sistema operacional Linux, é possível alterar a prioridade de um processo já iniciado com o uso do comando nice.**

**7. (CESPE/TJ-PA - 2020) No Linux, um processo, por si só, não é elegível para receber tempo de CPU. Essa ação depende, basicamente, do seu estado da execução. O processo está administrativamente proibido de executar no estado**

A) pronto.

B) dormente.

C) executável.

D) parado.



E) zumbi.

## GABARITO



## GABARITO

1- Correta

2- E

3- A

4- C

5- C

6- Errada

7- D



## LISTA DE QUESTÕES - MONTAGEM DE VOLUMES - MULTIBANCAS

1. (FCC/TRT4 - 2015) A possibilidade de compartilhar arquivos entre diferentes sistemas operacionais é fundamental para aumentar a produtividade computacional. A montagem automática de uma partição com sistema de arquivos CIFS, durante o boot do servidor com sistema operacional Linux, deve ser configurada no arquivo  
  - A) /etc/fstab.
  - B) /boot/mount.
  - C) /etc/mount.
  - D) /boot/inittab.
  - E) /etc/initd.
2. (UFRRJ/UFRRJ - 2015) Qual o comando para agregar uma mídia formatada em NTFS no sistema de arquivos de um sistema operacional UNIX-like?  
  - A) `umount -t ntfs /dev/diskid/DISK-JP1440HA196TKS /mnt.`
  - B) `mount /mnt /dev/diskid/DISK-JP1440HA196TKS -t ntfs.`
  - C) `umout /mnt /dev/diskid/DISK-JP1440HA196TKS -t ntfs.`
  - D) `mount -t ntfs /dev/diskid/DISK-JP1440HA196TKS /mnt.`
  - E) `umount /dev/diskid/DISK-JP1440HA196TKS -t ntfs /mnt.`
3. (CESPE/TCE-RN - 2015) O Linux tem a capacidade de montar sistemas de arquivos do dispositivo /dev/sdc3, que tem o sistema de arquivos do tipo ext4 e, no ponto de montagem /teste, deve ser utilizado o comando `mount -t ext4 /dev/sdc3 /teste.`
4. (FCC/TRT23 - 2016) Uma partição NFS remota deve ser montada em um computador com sistema operacional Linux. Para especificar, no comando `mount`, que a partição é NFS, deve-se utilizar a opção:  
  - A) `-n.`
  - B) `-f.`



C) -i.

D) -s.

E) -t.

5. (CESPE/TCE-PA - 2016) A opção `mount -t vfat` permite fazer o becape dentro de uma partição de disco DOS/Windows, caso em que a hierarquia de diretório aparece como parte do sistema de arquivos Linux.

## GABARITO



## GABARITO

1- A

2- D

3- Correta

4- E

5- Correta



## LISTA DE QUESTÕES – SUPERUSUÁRIO - MULTIBANCAS

1. (FCM/IF Sudeste-MG - 2016) O Linux permite ao superusuário (root) executar qualquer operação válida em qualquer arquivo ou processo, inclusive algumas chamadas de sistema podem ser executadas somente pelo superusuário. A operação que NÃO é exclusiva do superusuário é

  - A) configurar interfaces de rede.
  - B) configurar o relógio do sistema.
  - C) alterar as permissões de um arquivo.
  - D) aumentar os limites de uso dos recursos.
  - E) definir o nome de host (hostname) do sistema.
2. (IBFC/EBSERH - 2016) O comando sudo do sistema operacional Linux é muito poderoso, permitindo que usuários comuns obtenham privilégios de super usuário. Por questões de segurança, o administrador precisa definir no arquivo \_\_\_\_\_, quais usuários podem executar sudo, em quais computadores podem fazê-lo e quais comandos podem executar através dele. Assinale a alternativa que complete correta e respectivamente a lacuna:

  - A) /etc/sudoers
  - B) /root/sudour
  - C) /root/sudoers
  - D) /usr/sudour
  - E) /etc/sudour
3. (INAZ do Pará/DPE-PR - 2017) Qual a alternativa que corresponde à linha de comando para que usuários comuns possam utilizar o comando administrativo apt-get dist-upgrade?

  - A) # apt-get dist-upgrade
  - B) \$ sudo apt-get gw dist-upgrade
  - C) # sudo apt-get gw up dist-upgrade
  - D) \$ sudo apt-get dist-upgrade



E) # sudo apt-get dist-upgrade gw now

4. (CESPE/ABIN - 2018) O Linux não impede a alteração do nome do superusuário, nem a criação de contas com UID igual a 0, embora essas ações não sejam recomendadas.

5. (COSEAC/UFF - 2019) No Sistema Operacional Linux o diretório local padrão do superusuário é:

A) /home.

B) /usr.

C) /root.

D) /lib.

E) /dev.

## GABARITO



# GABARITO

1- C

2- A

3- D

4- Correta

5- C



## LISTA DE QUESTÕES - USUÁRIOS, GRUPOS, PERMISSÕES DE ACESSO - MULTIBANCAS

1. (FUNCAB/MDA - 2014) No sistema operacional Linux, a criação de um grupo permite que um conjunto de usuários diferentes possua acesso a um mesmo arquivo. O comando do Linux que tem como objetivo alterar o grupo de um arquivo/diretório é o

- A) chown.
- B) chgroup.
- C) chset.
- D) chfile.
- E) chgrp.

2. (FGV/Prefeitura de Niterói-RJ - 2018) Pedro é o proprietário do arquivo header.txt em um sistema Linux e gostaria de assegurar que somente ele tivesse permissão de leitura, gravação e execução a este arquivo, enquanto que todos os demais usuários com acesso ao sistema tivessem somente a permissão de leitura.

Assinale a opção que indica o comando que pode ser usado para conseguir esse objetivo.

- A) chmod ug+r header.txt
- B) chmod 766 header.txt
- C) chmod 722 header.txt
- D) chmod +r header.txt
- E) chmod 744 header.txt

3. (FGV/MPE-AL - 2018) Para definir o script `/usr/local/bin/meuscript` como executável no sistema operacional Linux, devemos usar o comando

- A) chmod og /usr/local/bin/meuscript
- B) chmod 000 /usr/local/bin/meuscript
- C) chmod +x /usr/local/bin/meuscript



D) `chmod ugo-x /usr/local/bin/meuscript`

E) `chmod 666 /usr/local/bin/meuscript`

4. (IF-ES/IF-ES - 2019) Os administradores de redes precisam entender com clareza sobre o gerenciamento de contas de usuários nos Sistemas Operacionais de Servidores Linux. Em relação ao sistema de contas Linux, analise as afirmativas abaixo:

I – O arquivo `/etc/shadow` serve para manter senhas criptografadas seguras quanto a acesso não autorizado.

II – Um novo usuário pode ser adicionado utilizando-se ferramentas, como `adduser` ou `useradd`.

III – Os usuários dos sistemas Linux recebem um número UID referente a sua identificação, porém o usuário `root`, por questões de segurança, não dispõe de UID.

É CORRETO o que se afirma em:

A) I, II e III.

B) I e II, apenas

C) I e III, apenas.

D) II e III, apenas.

E) Nenhuma das afirmativas está correta.

5. (CS-UFG/IF Goiano - 2019) Na configuração do Sistema Linux, o comando `chmod` modifica as permissões de arquivos com respeito à execução, leitura e escrita. A expressão `chmod 751 arq2` é equivalente a:

A) `chmod u=rx,g=rwx,o=x arq2`

B) `chmod u=rx,g=rwx,o=r arq2`

C) `chmod u=rwx,g=rx,o=x arq2`

D) `chmod u=rwx,g=rx,o=r arq2`

6. (CCV-UFC/UFC - 2019) Qual dos itens abaixo contém o comando do sistema operacional Linux capaz de mudar o grupo de um arquivo ou diretório do sistema?

A) `df`

B) `sed`



- C) chown
- D) chmod
- E) passwd

7. (FCC/TJ-MA - 2019) Um Analista digitou o comando `chmod u=rwx,g=rx,o=r processo` para definir as permissões de acesso ao arquivo processo. O comando equivalente usando a notação octal é:

- A) `chmod 713 processo`
- B) `chmod 777 processo`
- C) `chmod 134 processo`
- D) `chmod 754 processo`
- E) `chmod 671 processo`

## GABARITO



## GABARITO

- 1- E
- 2- E
- 3- C

- 4- B
- 5- C
- 6- C

- 7- D



## LISTA DE QUESTÕES - COMANDOS BÁSICOS - MULTIBANCAS

1. (FGV/Câmara Municipal do Recife-PE - 2014) No Linux, o comando

`cat something | grep 'something'`

tem o efeito de:

- A) mostrar as linhas do arquivo something que incluem a palavra something;
- B) adicionar a palavra something ao conteúdo do arquivo something;
- C) criar uma cópia do arquivo something denominada grpe;
- D) mover o arquivo something para um novo diretório denominado something;
- E) restringir o acesso ao arquivo something ao usuário something.

2. (VUNESP/DESENVOLVESP - 2014) O superusuário do sistema operacional Linux deseja verificar a informação da última linha do arquivo `/var/log/dmesg`. Para efetivar essa ação por meio do uso de apenas um comando, ou seja, sem qualquer outra ação, deve-se executar, na linha de comando:

- A) `$ cat /var/log/dmesg`
- B) `$ cat /var/log/dmesg >`
- C) `$ more /var/log/dmesg >>`
- D) `$ tail /var/log/dmesg`
- E) `$ top /var/log/dmesg`

3. (VUNESP/SP-URBANISMO - 2014) Um usuário padrão do sistema operacional Linux, "logado" no seu diretório home, deseja saber o diretório no qual está instalado o comando mount. Para obter tal informação, esse usuário poderá utilizar o seguinte comando:

- A) `cat mount`
- B) `find mount`
- C) `grep mount`



D) where mount

E) which mount

4. (FCC/TRF4 - 2014) Marcos, usuário de um computador com sistema operacional Linux Red Hat listou o conteúdo do seu diretório home e observou a presença do arquivo manual.txt com 31.251 bytes de tamanho, o que representa cerca de 20 páginas de texto se visualizado em um terminal Linux padrão. Para que Marcos possa visualizar diretamente o final do arquivo manual.txt, sem a necessidade de iniciar a visualização a partir do começo do arquivo, ele deve executar o comando:

A) cat manual.txt | more

B) more manual.txt

C) list manual.txt | end

D) tail manual.txt

E) cat manual.txt | end

5. (FGV/TJ-BA - 2015) O seguinte comando, invocado em uma shell do Linux:

```
find . -type f
```

escreverá na saída padrão:

A) o número de arquivos regulares descendentes do diretório corrente;

B) os caminhos (pathnames) relativos para os arquivos regulares descendentes do diretório corrente;

C) os caminhos (pathnames) absolutos para os arquivos regulares descendentes do diretório corrente;

D) os caminhos (pathnames) relativos para os diretórios descendentes do diretório corrente;

E) os caminhos (pathnames) absolutos para os arquivos regulares e diretórios descendentes do diretório corrente.

6. (IF-RR/IF-RR - 2015) O comando find permite buscar arquivos filtrando por nome, tamanho, data, dono, grupo, permissões, tipo e outros. A busca é recursiva, por padrão, mas pode ser restringida a uma determinada profundidade. O comando GNU find faz parte das findutils GNU e está instalado em cada sistema Ubuntu. Um usuário possui uma distribuição



Ubuntu14.04 LTS e precisa encontrar um arquivo de áudio com tamanho superior a 5MB, dentre as opções abaixo, qual alternativa melhor representa o comando para a busca pelo arquivo.

- A) find \$HOME -iname '\*.mp4' -size -5M
- B) find \$HOME -iname '\*.mp4' -size +5M
- C) find \$HOME -iname '\*.jpeg' -size +5M
- D) find \$HOME -iname '\*.ogg' +size -5M
- E) find \$HOME -iname '\*.ogg' -size +5M

7. (FCM/IF Sudeste-MG - 2016) Alguns comandos em Linux podem combinar diferentes parâmetros, como o comando a seguir:

```
$ find /home/usuario/ -name '*.txt' -mtime 2;
```

Esta linha de comando procura

- A) todos os arquivos com extensão ".txt", dentro da pasta "/home/usuario".
- B) e remove todos os arquivos com extensão ".txt" dentro da pasta "/home/usuario".
- C) todos os arquivos com extensão ".txt", modificados a mais de 2 dias, dentro da pasta "/home/usuario" e subpastas.
- D) todos os arquivos modificados a mais de 2 dias, dentro da pasta "/home/usuario", e subpastas, e os move para a pasta "name".
- E) e copia para o diretório corrente todos os arquivos com nome "\*.txt", criados a menos de 2 dias, dentro da pasta "/home/usuario".

8. (FCM/IFF-RS - 2016) Em um ambiente Linux, desejam-se encontrar todos os arquivos com a extensão ".sh" a partir do diretório corrente. O comando a ser utilizado é o

- A) find -all \*.sh /
- B) grep - all \*.sh
- C) find -user -i .sh
- D) ps -name \ \*.sh



E) `find . -name \*.sh`

9. (IESES/BAHIAGÁS - 2016) O comando `find` está presente em vários sistemas operacionais Unix-like, sendo muito conhecido pela sua versatilidade. Considere o uso do `bash` em uma distribuição de Linux Ubuntu 14.04, na qual o comando `find` esteja disponível. Um usuário gostaria então de listar somente arquivos de extensões `'.java'` e `'.py'` em seu diretório `'/home/usuario'`. Assinale a alternativa que executa esta tarefa corretamente.

A) `find /home/usuario -type f -iname '*.java' -iname '*.py'`

B) `find /home/usuario -type d -a \( -iname '*.java' -o -iname '*.py' \)`

C) `find /home/usuario -type f -o \( -iname '*.java' -o -iname '*.py' \)`

D) `find /home/usuario -type f -a \( -iname '*.java' -o -iname '*.py' \)`

E) `find /home/usuario -type f -a \( -iname '*.java' -a -iname '*.py' \)`

10. (CESPE/FUNPRESP-JUD - 2016) Os comandos `find` e `which` podem ser utilizados para a obtenção de informações acerca de programas instalados no Linux; o primeiro localiza o caminho (path) onde o comando está instalado, e o segundo, a versão do programa.

11. (INSTITUTO AOCP/EBSERH - 2016) Um administrador de sistema operacional Linux necessita procurar determinados textos, dada uma palavra, dentro de um arquivo texto. O comando que possibilita executar essa operação é o

A) `find`.

B) `more`.

C) `whereis`.

D) `grep`.

E) `pipe`.

12. (IESES/BAHIAGÁS - 2016) Qual dos comandos apresentados a seguir pode ser utilizado para juntar o conteúdo de um arquivo texto de log do Linux, denominado de `log1`, ao conteúdo de um arquivo texto de log, denominado `log2`?

A) `cat log2 << log1`

B) `cat log2 < log1`



- C) `cat log1 || log2`
- D) `cat log1 > log2`
- E) `cat log1 >> log2`

**13.(IESES/BAHIAGÁS - 2016)** Um administrador de sistema gostaria de acompanhar o arquivo de log `'/var/log/apache.log'` em tempo real. Este arquivo está em um servidor Linux Ubuntu 14.04. Supondo que o administrador possua acesso de superusuário ao sistema através do bash. Qual dos comandos a seguir, executado uma única vez, poderia auxiliar o administrador a acompanhar este log dinamicamente (sem interferência adicional)?

- A) `tail -rt /var/log/apache.log`
- B) `more -rt /var/log/apache.log`
- C) `more -f /var/log/apache.log`
- D) `cat /var/log/apache.log`
- E) `tail -f /var/log/apache.log`

**14.(CONSULPLAN/Câmara de Belo Horizonte-MG - 2018)** No Sistema Operacional GNU/Linux, o comando `cat` é utilizado para concatenar arquivos. Sua sintaxe básica é: `cat [opções] arquivo`. Uma das opções utilizadas é `-A`, que tem como função:

- A) Numerar todas as linhas na saída.
- B) Exibir todos os caracteres especiais.
- C) Numerar apenas as linhas não vazias.
- D) Exibir caracteres não passíveis de impressão (caracteres de controle).

**15.(IDECAN/IF-PB - 2019)** A respeito dos comandos que podem ser utilizados para verificar o uso de recursos em um sistema operacional Linux, assinale a alternativa que indica, respectivamente, o nome do comando responsável por exibir informações referentes ao uso da memória RAM e uso do espaço em disco.

- A) `cat` e `du`
- B) `locate` e `df`



- C) cat e free
- D) free e df
- E) du e free

**16.(COSEAC/UFF - 2019)** No Linux, o comando para listar o conteúdo de um diretório e o comando que mostra o caminho por inteiro do diretório em que o usuário se encontra são, respectivamente:

- A) find e more.
- B) cat e grep.
- C) lpr e cat.
- D) df e finger.
- E) ls e pwd.

**17.(VUNESP/Câmara de Tatuí-SP - 2019)** Considere a execução da sequência de comandos a seguir, em um terminal shell do Linux:

```
# cd /root  
# mkdir -p /root/foo/bar  
# pwd
```

O resultado impresso na tela após a execução do último comando será:

- A) /root
- B) /root/foo/bar
- C) bar
- D) /root/foo
- E) /foo/bar

**18.(CS-UFG/IF Goiano - 2019)** No sistema operacional GNU/Linux, usando a linha de comando, deseja-se executar o seguinte: a) criar um diretório chamado IFG; b) criar um arquivo de texto chamado Concurso.txt; c) apagar o arquivo de texto Concurso.txt; d) apagar o diretório IFG. Qual é a sequência de comandos a ser empregada? Obs.: o sinal “,” é um mero separador entre os comandos



- A) mkdir IFG , touch Concurso.txt , rm Concurso.txt , rmdir IFG
- B) mkdir IFG , create Concurso.txt , kill Concurso.txt , killdir IFG
- C) create IFG , touch Concurso.txt , rm Concurso.txt , rmdir IFG
- D) create IFG , create Concurso.txt , kill Concurso.txt , killdir IFG

## GABARITO



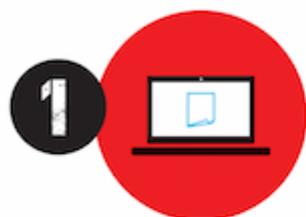
## GABARITO

- |      |            |       |
|------|------------|-------|
| 1- A | 7- C       | 13- E |
| 2- D | 8- E       | 14- B |
| 3- E | 9- D       | 15- D |
| 4- D | 10- Errada | 16- E |
| 5- B | 11- D      | 17- A |
| 6- E | 12- E      | 18- A |



# ESSA LEI TODO MUNDO CONHECE: PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.



**1** Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.



**2** Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).



**3** Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).



**4** Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).



**5** Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).



**6** Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).



**7** Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.



**8** O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.



Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.