

AULA 00: Ambiente Linux

Sumário

1. Apresentação do curso.....	2
1.1. A Banca.....	3
1.2. Metodologia das aulas.....	3
2. Conteúdo programático e planejamento das aulas (Cronograma).....	5
3. Linux.....	6
3.1. Introdução e histórico.....	6
3.2. Conceitos básicos.....	7
3.3. Estrutura de diretórios.....	7
3.4. Permissões.....	8
3.5. Questões Comentadas.....	12
3.6. Boot do Sistema.....	17
3.7. Administração de Usuários.....	21
3.8. Configurações de Rede - Suporte a TCP/IP.....	23
3.9. IPTABLES (Firewall).....	26
3.10. Logs do Sistema.....	28
3.11. Superusuário: su e sudo.....	30
3.12. Gerenciamento de Processos.....	31
3.13. Visualizando os estados dos processos.....	32
4. Programação bash.....	38
5. Sistemas de arquivos.....	47
5.1. Sistema de arquivos EXT4.....	47
6. Conclusão.....	53

Saudações, concurseiros!

Como é de praxe, vou me apresentar para que vocês me conheçam um pouco. Sou Analista no Serpro (Serviço Federal de Processamento de Dados), e atuo na área que faz gestão de Segurança da Informação na empresa. Antes disso, trabalhei 4 anos como Técnico em Redes, também no Serpro, com suporte e administração em ambientes Linux e Windows, servidores, virtualização e rede, entre outras coisas. Fui duas vezes finalista da Maratona de Programação (ACM ICPC). Possuo as certificações LPI (Administrador Linux) e MCSO (Security Officer). A minha experiência como professor e instrutor foi adquirida dentro da empresa em que trabalho ministrando cursos, e na preparação de turmas para competições de programação na universidade. Aqui no TI Para Concursos, trabalho principalmente com Sistemas Operacionais Linux, Windows e Virtualização.

Gosto de trabalhar na preparação de alunos, principalmente de alunos interessados em aprender, o que certamente é o seu caso! :) Por isso, não tenha medo de me mandar questões para resolução, dúvidas ou questionamentos sobre a melhor forma de aprender algum tópico, pois este é um diferencial disponível para quem adquire um curso no TIParaConcursos.net.

1. Apresentação do curso

Neste curso, trabalharemos vários assuntos: Windows Server, Linux, Virtualização, Servidores, Backup e Arquitetura e Organização de Computadores. Enfatizaremos os principais tópicos, mas este curso não tem a pretensão de esgotar todo o mundo de conhecimento sobre estes assuntos, pois é impossível. Iremos então enfatizar os tópicos mais importantes, citados pelo edital, e dar maior importância às questões que vêm sendo cobradas nas bancas.

Este curso é classificado como teórico e exercícios, porém com ênfase maior nos exercícios. Nosso objetivo é trabalhar com um nível interme-

diário (também resolvemos algumas questões difíceis) uma grande gama de questões e assuntos em uma quantidade de aulas que torne o material com um excelente custo-benefício. Como vocês podem observar, não forçamos uma quantidade grande de aulas mesmo com muitos assuntos em algumas aulas.

Para esta aula demonstrativa, trouxemos conceitos básicos e noções de administração do Linux, conforme pedido no programa.

1.1. A Banca

Este nosso curso será voltado para o CESPE, a banca prevista para o nosso concurso, e além disso é a banca que mais realiza concursos de TI. Alguns raros concurseiros falam mal desta banca, mas eu lhes tranquilizo logo dizendo que na minha opinião é a melhor (ou menos pior) de todas. O CESPE é uma banca bem profissional, responde os recursos na maioria das vezes de maneira justa e age de maneira responsável na maioria das vezes quando comete erros. Isso dá uma tranquilidade maior ao concurseiro, ao saber que será julgado na maior parte pelo seu próprio mérito.

O outro ponto importante do CESPE é o seu sistema de questões Certo/Errado, onde uma cancela outra. Isso exige uma atenção especial do candidato. Muitos concurseiros de primeira viagem chegam à prova sem nem saber deste “detalhe”, e outros marcam todas questões inadvertidamente, sem ter nenhuma estratégia definida previamente. Isto certamente não é a melhor maneira. Mas também certamente não é o caso de vocês. A minha dica é: procurem por artigos sobre como resolver questões CESPE e trace sua própria estratégia antes de ir para a prova. De preferência testando ela antes em um ou mais simulados. Isto certamente te trará resultados equivalentes a algumas dezenas de horas de estudos.

1.2. Metodologia das aulas

Temos também que explicar algumas coisas que parecem básicas e parecem chover no molhado, mas que temos sim que esclarecer pois é nosso papel para com os alunos:

- a) A abordagem do curso, como vocês já devem conhecer, deve ser informal e em forma de prosa de conversa. Estamos seguindo uma definição dos nossos coordenadores pedagógicos, que são profissionais de grande experiência, caras gabaritados, de alto garbo e elegância, que sabem que as aulas em texto não podem nem devem ser cansativas despejando somente conteúdo. O clima mais leve, de conversa, facilita a concentração e diminuiu o cansaço da rotina de estudos que por vezes é um pouco extenuante. E também, como sou cearense, me reservo também ao direito de contar piada de vez em quando. :)
- b) A matéria tem que começar do básico. Sim amigos, para algumas pessoas que já tem algum conhecimento avançado da matéria, podem achar alguns conceitos simples, mas nós temos que abordar, seguindo a premissa de que o aluno não sabe ou mesmo que já faz tempo que viu esse conteúdo e tem que revisitá-lo. No entanto, você que já é “o bonitão das tapiocas” e já sabe bem determinadas partes da matéria, não se desespere! Este material servirá sim para você, pois abordaremos muita resolução de questões. Cabe ressaltar que muitas questões cobram um bom entendimento de conceitos básicos.
- c) O curso deve focar no que cai na prova. Temos a tendência de querer estudar muitos conteúdos e todos em profundidade, ou estudar mais assuntos que gostamos. Corrigir esse desvio é um grande passo para a aprovação. Portanto ao estudar por um material orientado, como esse do TIParaConcursos.net, você já está “de quebra” a um passo a frente dos concorrentes. Ah, e por favor, adquiram os mate-

riais de forma correta! Os materiais já estão sendo disponibilizados por um preço acessível, e o trabalho dos elaboradores deve ser recompensado. Além do mais, alguns concursos fazem sindicância da vida pregressa do candidato e não pega bem verem que o indivíduo tá respondendo um processo por violação de direitos autorais, né?

- d) Resolução de questões: Daremos bastante foco em resolução de questões, pois é o que dá musculatura cerebral para que vocês pon-tuem mais nas provas. Portanto isso será uma parte considerável do nosso curso.

2. Conteúdo programático e planejamento das aulas (Cronograma)

O Conteúdo programático está dividido conforme os grupos de conteúdo do edital, no entanto poderemos dar mais ênfase em determinados assuntos que têm maior probabilidade de cair, mas abordaremos todos assuntos que estão no programa destas áreas. Enfatizar os assuntos corretos é uma questão estratégica e é uma característica do material que você está adquirindo.

Aula	Conteúdo a ser trabalhado
Aula 00 Ambiente Linux	Parte I (Demonstrativa) <ul style="list-style-type: none">• Conceitos básicos.• Noções de administração.
Aula 01 29/07	Ambiente Windows e AD: <ul style="list-style-type: none">• Sistemas operacionais Windows. Conceitos básicos. Noções de administração. Serviços de diretório: Active Directory e LDAP. Interoperabilidade.
Aula 02 07/08	Virtualização, Datacenters e Cloud: <ul style="list-style-type: none">• Virtualização.• Tecnologias e arquitetura de Datacenter. Conceitos básicos.• Cloud Computing.
Aula 03 20/08	Alta Disponibilidade, Balanceamento e desempenho: <ul style="list-style-type: none">• Topologia típica de ambientes com alta disponibilidade e escalabilidade.

	<ul style="list-style-type: none">• Balanceamento de carga, fail-over e replicação de estado.• Técnicas para detecção de problemas e otimização de desempenho.
Aula 04 27/08	<ul style="list-style-type: none">• Servidores de aplicação JEE.<ul style="list-style-type: none">◦ Conceitos básicos.◦ Noções de administração.
Aula 05 04/09	Armazenamento de Dados: <ul style="list-style-type: none">• Serviços de armazenamento, padrões de disco e de interfaces.• RAID.• Tecnologias de armazenamento DAS, NAS e SAN.• Tecnologias de backup. Deduplicação.• ILM - Information Lifecycle Management.

3. Linux

3.1. Introdução e histórico

O Linux é um sistema operacional de código aberto desenvolvido por um pesquisador finlandês chamado Linus Torvalds. Ele procurava fazer um clone do MINIX, um sistema operacional de outro pesquisador universitário que implementava alguns conceitos dos sistemas UNIX. O grande diferencial do Linux foi exatamente a sua abertura desde o início, com o seu criador tornando o código do mesmo disponível para que todos contribuissem e melhorassem o seu funcionamento.

Aconteceu então algo bastante interessante, o Linux acabou se encontrando com outro projeto de sistema operacional livre, o projeto GNU (GNU is Not Unix), que visava projetar um sistema operacional de código livre que fosse uma alternativa aos Unices proprietários da época. O projeto GNU já tinha vários programas utilitários funcionais, mas não tinha um kernel. Foi aí onde o Linux se encaixou perfeitamente, passando a integrar o projeto GNU, formando uma parceria ideal.

O Linux passou ao longo dos anos a conquistar uma grande estabilidade, vindo a ser apoiado por grandes empresas como a IBM, e gerando outras grandes empresas de serviços baseados nele, como a Red Hat.

Linux, como falamos na historinha é o kernel, o núcleo do sistema operacional. Os demais aplicativos como interface gráfica, interpretador de comandos, processador de texto, etc são outros softwares. Quando falamos de Linux, a rigor, estamos falando do núcleo do sistema operacional, e os outros softwares são outros projetos.

Atualmente, as distribuições (versões) mais populares de Linux no mercado são o Ubuntu, versão voltada para a facilidade de uso; Red Hat, uma versão voltada para o mercado corporativo, com grande estabilidade e suporte profissional; SUSE, que engloba características do Ubuntu e Red Hat, e o Debian, uma versão feita pela comunidade na qual se baseiam

um bom número de distribuições, inclusive o Ubuntu. Todas estas citadas, exceto a Debian, possuem empresas que dão apoio e oferecem serviços baseados nelas.

Caso ainda não tenha se convencido da importância deste negócio, vai mais uma: o Linux hoje é o sistema que roda a maioria dos servidores da internet, e também é o kernel no qual o Android foi baseado. Android, como vocês sabem, é o sistema operacional que roda na maioria dos smartphones do mundo atualmente.

3.2. Conceitos básicos

Apresentaremos a seguir os conceitos principais do Linux, muitos deles são características em comum com os sistemas Unix em geral.

3.3. Estrutura de diretórios

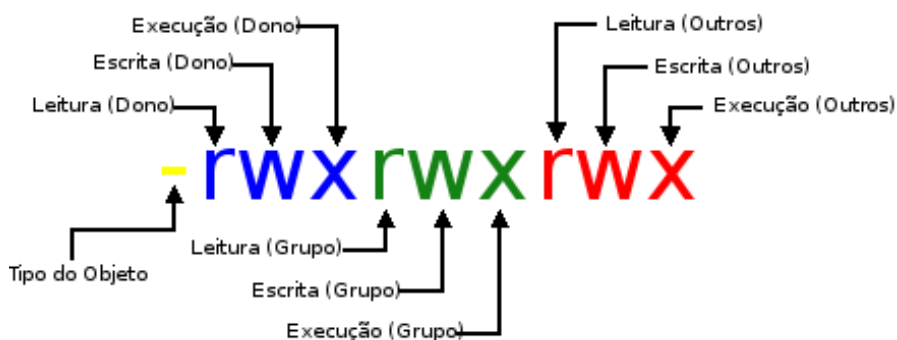
O Linux (e os Unices em geral) possui(em) uma estrutura básica de diretórios chamada Filesystem Hierarchy Standard (FHS). Nela está contida toda a estrutura lógica de informações do sistema:

/	Diretório raiz do sistema de arquivos. É abaixo dele que se situam todos os outros
/bin	Contém programas executáveis do sistema que são usados com frequência pelos usuários
/boot	Contém arquivos necessários para a inicialização do sistema
/dev	Contém arquivos especiais usados para acessar dispositivos (periféricos) existentes no computador
/etc	Arquivos de configuração de computador local
/home	Diretórios dos arquivos pessoais dos usuários
/lib	Bibliotecas essenciais compartilhadas pelos programas do sistema e módulos do kernel
/media	Ponto de montagem para mídia removível
/mnt	Ponto de montagem para sistemas de arquivos montados temporariamente
/opt	Pacotes de aplicações adicionais, geralmente instaladas pelo usuário

/proc	Sistema de arquivos de informações do kernel, processos e sistemas de arquivo
/root	Diretório do super-usuário
/sbin	Diretório de programas executáveis usados pelo superusuário (root) para administração e controle do funcionamento do sistema
/tmp	Diretório para armazenamento de arquivos temporários criados por programas.
/usr	Contém a maior parte dos programas (pacotes) oficiais da distribuição (executáveis, documentos, bibliotecas)
/usr/bin	Contém programas executáveis do que são usados com frequência pelos usuários
/usr/lib	Bibliotecas compartilhadas pelos programas do sistema.
/var	Contém a maior parte dos arquivos que são gravados com frequência pelos programas do sistema (logs), e-mails, spool de impressora, cache, etc.

3.4. Permissões

Nos sistemas UNIX e UNIX like (Linux por exemplo), cada arquivo tem permissões associadas a ele. Estas permissões são descritas em 10 caracteres, que descrevem o tipo de objeto (1), as permissões do proprietário do arquivo (3), as permissões dos membros grupo ao qual o proprietário pertence (3), e as permissões para qualquer usuário.



Para ilustrar, vamos utilizar o exemplo abaixo, exibindo informações sobre o arquivo “teste”:

```
$ ls -l teste
```

```
-rw-r--r-- 1 victor grupo 0 2013-06-04 09:26 teste
```

O resultado do comando é a listagem dos detalhes do arquivo teste, com as permissões dele aparecendo no começo.

Tipo do objeto	User (usuário dono)	Group (grupo)	Others (outros usuários)
-	rw-	r--	r--

O tipo de objeto pode ser os seguintes:

- d diretório
- b arquivo de bloco
- c arquivo especial de caractere
- p canal
- s socket
- arquivo "normal"

As permissões r, w e x significam:

- r significa permissão de leitura (read)
- w significa permissão de gravação (write)
- x significa permissão de execução (execution)
- significa permissão desabilitada

Para visualizarmos os "rwx" em forma binária, cada caractere dos r,w,x corresponde a um bit "ligado". A representação decimal da configuração permissão equivale à conversão deste número binário.

Permissão	Binário	Decimal
---	000	0

--x	001	1
-w-	010	2
-wx	011	3
r--	100	4
r-x	101	5
rw-	110	6
rwX	111	7

O entendimento das permissões é importante para que a gente saiba usar o comando **chmod** (**ch**ange **mo**de), que altera as permissões dos arquivos.

A sintaxe dele é:

```
chmod [opções] alteração arquivo
```

Algumas opções:

-R: altera recursivamente as permissões de um diretório

-c: informa quais arquivos/diretórios estão tendo permissões alteradas

-v: informa quais arquivos/diretórios estão sendo processados

Alguns exemplos de uso

```
chmod ug+rw teste.txt
```

→ Está adicionando (+) ao dono (u - user) e ao seu grupo (g) permissões de escrita e leitura.

```
chmod g=rwx teste
```

→ Está “setando” no arquivo permissão de escrita, leitura e execução a todos membros do grupo.

```
chmod 600 notas.txt
```

→ Conforme a tabela que acabamos de mostrar, equivale a atribuir a permissão.

“- rw- --- ---”

3.5. Questões Comentadas

Vamos resolver algumas questões?

CESPE - 2012

No Linux, o diretório /bin contém programas do sistema que são utilizados pelos usuários, não sendo necessário, para que esses programas sejam executados, que eles possuam a extensão .exe.

Certo () Errado ()

Comentário:

O diretório /bin contém programas do sistema que são utilizados pelo usuário. O /usr contém programas que são instalados pelo usuário, como pacotes da distribuição Linux, um processador de texto por exemplo.

No Linux, os executáveis não precisam ter extensões em seu nome.

Resposta: Certo.

CESPE - 2012 - TCE-ES - Auditor de Controle Externo

Com relação aos sistemas operacionais Linux e Windows e aos programas de navegação na Internet, julgue os itens a seguir.

No Linux, o diretório padrão de usuários é o /home. Nesse diretório, o usuário tem permissão de escrita, de execução e de leitura.

Certo () Errado ()

Comentário:

Uma questão gaiata. Vamos prestar bem atenção nela. O diretório /home é o diretório definido no padrão FHS como a localização dos arquivos pessoais dos usuários. Ao criarmos um usuário Zezim, por exemplo, o Linux cria um diretório /home/zezim. Nesse diretório, o usuário tem permissão de criar arquivos, acessar o diretório (é o que significa permissão de execução sobre um diretório), e de listar o conteúdo deste (é o que significa permissão de leitura para um diretório). O diretório /home é onde ficam os vários diretórios dos usuários, não sendo portanto permitido a um usuário criar diretamente arquivos lá, acessar ou apagar os outros diretórios que lá se encontra, como um diretório /home/maria, por exemplo. No Windows também temos algo parecido, apenas mudam os nomes dos diretórios.

Resposta: E

CESPE - 2011 - EBC - Analista - Administração de Sistemas

Em ambiente Unix, o diretório raiz é representado por uma barra em posição normal (/).

Certo () Errado ()

Comentário:

No sistema UNIX e nos sistemas Unix-like (derivados ou inspirados no Unix), o diretório raiz é o /. Para quem não tem familiaridade, é como se fosse um "C:" do Windows.

Resposta: C

CESPE - 2011 - Correios - Analista de Correios - Analista de Sistemas - Desenvolvimento de Sistemas

Com relação aos sistemas operacionais Unix e Linux, julgue os itens que se seguem.

No Unix, os drivers de dispositivos podem aparecer como arquivos, e os canais de comunicação entre processos ou conexões de rede, para o usuário, também se assemelham a arquivos.

Certo () Errado ()

Comentário:

Há uma máxima que diz que “no Linux tudo é arquivo”. Tudo pode ser entendido como arquivo, desde os arquivos de dados propriamente ditos, até os dispositivos, alguns são interpretados pelo sistema como arquivos!

Isso vem da filosofia do UNIX, que foi desenvolvido com essa ideia, de tratar tudo como arquivo.

Relembrando, ao listarmos os arquivos com o comando “ls -l” um arquivo pode aparecer como :

d diretório (arquivo diretório? Foi mantido aqui para manter a consistência da explicação)

b arquivo de bloco

c arquivo especial de caractere

p canal

s socket

- arquivo "normal"

Resposta: C

CESPE - 2010

Com relação aos sistemas operacionais Windows e Linux, julgue os itens subsequentes.

No diretório Linux, os arquivos relacionados a dispositivos de hardware estão armazenados no diretório /lib.

Certo () Errado ()

Comentário:

O diretório relacionado a dispositivos é o /dev. O /lib contém bibliotecas essenciais do sistema e módulos do kernel.

Resposta: Errado.

CESPE - 2012

Suponha que, após a execução do comando `ls -l`, seja apresentada a seguinte saída em tela.

```
-rwxr----- 1 root gerentes 12148 Jul 14 2011 mensagens
```

Nessa situação, é correto afirmar que os usuários que integram o grupo gerentes somente podem ler o conteúdo do arquivo mensagens.

Certo () Errado ()

Comentário:

Relembrando, as permissões do arquivo são divididas em três grupos, tirando o primeiro caractere, que é de uso especial para descrever se o objeto é um arquivo ou diretório, os outros nove caracteres são divididos em três grupos:

User: O próprio usuário

Group: O grupo do usuário proprietário

Others: Qualquer outro usuário.

Então para nossa questão, ficaria:

U: rwx → permissão total, de leitura (Read), escrita (Write)

G: r-- → Permissão de leitura

O: --- → Nenhuma permissão

O que o enunciado diz? Que os usuários do grupo gerentes podem ler o arquivo. Na saída do comando vemos (1) que o grupo do dono do arquivo é o gerentes, e (2) que há permissão de leitura para o grupo. Tudo certo.

Resposta: C

CESPE - 2011

Considere que o resultado a seguir tenha sido obtido após a execução do comando `$ ls -ld ch3 test`.

```
-rw-rw-r-- 1 corr vend 4983 Jan 18 22:13 ch3
```

```
drwxr-xr-x 2 corr vend 1024 Jan 24 13:47 test
```

Nesse caso, é correto afirmar que o arquivo `ch3` tem permissão de leitura para todos os usuários do sistema operacional e de escrita apenas para os usuários `owner` e `group`; e que, para o diretório `test`, o usuário `owner` pode adicionar, mudar e apagar arquivos.

Certo () Errado ()

Comentário:

Na 1ª afirmação, ele diz o arquivo `ch3` tem permissão de leitura (`r`) para todos os grupos e escrita para `owner` (dono) e `group` (grupo). Podemos

verificar exatamente isto pois há 'r' no O,G,U, e 'w' apenas no O e G.

Na 2ª também está correto afirmar que o dono (**d****rw**xr-x) tem permissões de leitura, escrita e mudar e apagar. Um detalhe interessante nessa questão é que o 'x' se comporta de acordo com o que é o arquivo, se for um arquivo comum ele dá permissão de executar, enquanto se for um diretório (veja o 'd' no começo), ele dá permissão de mudar e apagar o diretório.

Resposta: Certo.

3.6. Boot do Sistema

O Boot do Linux segue de forma resumida a seguinte rotina:

1. A BIOS da máquina procura a MBR uma indicação para o gerenciador de boot;
2. O gerenciador de boot (Lilo ou Grub comumente) é carregado;
3. O gerenciador de boot sabe onde se encontra o kernel e o carrega;
4. O kernel vai para o nível de execução (runlevel) apropriado e inicia os processos do sistema, o init, os drivers, e demais serviços.

Runlevels

Os principais runlevel, ou níveis de execução, utilizados no Linux são:

- 1 (modo monousuário - de manutenção);
- 2 (modo multiusuário sem rede);
- 3 (modo multiusuário com rede), e;
- 5 (modo multiusuário com rede e ambiente gráfico X).

O nível 0 é utilizado para desligar (halt) o sistema, enquanto o runlevel 6 reinicia (reboot) o mesmo.

Questões Comentadas

CESPE - 2012 - FNDE - Especialista em Financiamento e Execução de Programas e Projetos Educacionais

Os sistemas operacionais Linux podem ser inicializados tanto no modo automático como no modo manual. No modo automático, o sistema executa o procedimento de inicialização completo por sua conta, sem qualquer comando externo. No modo manual, o sistema segue o procedimento automático até determinado ponto, a partir do qual, antes de a maioria dos scripts de inicialização ter sido executada, passa o controle ao operador da máquina.

Certo () Errado ()

Comentário:

O processo de boot do Linux segue os seguintes passos:

- A BIOS da máquina procura a MBR uma indicação para o gerenciador de boot;
- O gerenciador de boot é carregado (Lilo, Grub, etc);
- O gerenciador de boot carrega o kernel;
- O kernel vai para o nível de execução (runlevel) apropriado, inicia os processos do sistema, drivers e demais serviços.

Os principais runlevel utilizados no Linux são o 1 (modo monousuário - de manutenção), 2 (modo multiusuário sem rede), 3 (modo multiusuário com rede), e 5 (modo multiusuário com rede e ambiente gráfico X).

Esses 4 passos descrevem o “caminho feliz” de um processo de boot. Quando é feita uma instalação padrão, geralmente são configurados os parâmetros para que o sistema seja inicializado automaticamente.

No entanto, podemos deixar essa configuração de forma manual, apontar o gerenciador de boot para carregar um kernel diferente, ou, após isso, quando o Linux não é configurado para ir automaticamente para o runlevel 5 (multiusuário, rede, gráfico) por algum motivo (ex.: é um servidor e não carrega ambiente gráfico por padrão), essa ação pode ser feita manualmente, através do comando “init #”, onde # é o número do runlevel.

Resposta: C

3.7. Administração de Usuários

CESPE - 2008 - SERPRO - Analista - Suporte Técnico

Na administração de contas de usuários, pode-se informar o nome do diretório home de um usuário quando o comando `useradd` é executado; há instalações do Linux em que o arquivo `login.defs` tem os valores default a serem usados quando da criação de novas contas; é possível alterar dados da conta de um usuário via `usermod`; deve ser executado `userdel -r jose`, caso se queira remover a conta do usuário `jose`, sem remover o diretório home desse usuário.

Certo () Errado ()

Comentário:

Vamos quebrar em partes.

1. Quando vamos criar usuários, o comando realmente é o “`useradd`” e podemos informar o diretório home desejado através da opção “`-b`”. Correto.
2. O arquivo `login.defs` define configurações específicas do arquivo `shadow`, que é onde são armazenadas as senhas criptografadas dos usuários. Essas são configurações “default” quando da criação das contas. Correto.
3. É possível alterar uma conta de usuário com o comando `usermod`. Correto.
4. Caso queiramos manter o diretório home do usuário quando removemos a conta com o `userdel`, basta não colocarmos nenhuma opção. A opção “`-r`” é para remover, ao invés de manter como a questão fala. Errado.

Resposta: Errado

3.8. Configurações de Rede – Suporte a TCP/IP

Antes de abordarmos as configurações de rede no Linux, precisamos revisar alguns conceitos. O TCP/IP é um conjunto de protocolos que define padrões para comunicação entre diferentes computadores. Enquanto o IP (Internet Protocol) fornece endereçamento lógico para as máquinas (hosts), o TCP é o protocolo que garante que os pacotes de uma mensagem sejam transmitidos corretamente de fim-a-fim, fazendo controle de erros e da ordem de transmissão dos pacotes, entre outras coisas.

Para ter conectividade internet, o sistema precisa de: endereçamento - IP, máscara de rede; de um gateway, que é o computador que possui saída para internet; e de um DNS, que é o servidor que fará a resolução dos endereços em forma de nomes endereços IPs. Estas configurações, no Linux, são feitas em vários arquivos, e o principal deles é o `/etc/network/interfaces` (pode variar de localização dependendo da distribuição). Nele podemos configurar estes três parâmetros. Vejamos um exemplo de arquivo de configuração:

```
auto eth0
#iface eth0 inet dhcp
iface eth0 inet static
    address 192.168.0.130
    netmask 255.255.255.0
    network 192.168.0.0
    broadcast 192.168.0.255
    gateway 192.168.0.1
    dns-nameservers 192.168.0.1
```

Este é um exemplo de configuração de endereço estático (repare que a linha `dhcp` está comentada), que é a configuração usada em servidores.

Em estações de trabalho, geralmente não configuramos este arquivo diretamente, e sim através do “cara” que escreve sobre ele, que é um programa gráfico ou um utilitário de linha de comando. No Ubuntu, por exemplo, há um aplicativo que fica na barra de notificação, o NetworkManager. Este programa cuida de toda configuração para nós quando o endereço é adquirido por DHCP, mas também facilita a edição no caso de configuração manual/estática, não necessitando que o usuário edite os arquivos de configuração na “telinha preta” do terminal, e sim pela interface gráfica.

É importante se atentar para este fato, pois em alguns casos, ao configurarmos diretamente no arquivo de configuração, esta configuração realizada pode ser sobrescrita pelo gerenciador de conexão, pois é assim que o sistema foi projetado.

Voltando para o lado servidor, podemos ainda configurar estes parâmetros através de comandos individuais. Há um comando para cada um destes parâmetros.

Para alterar o IP:

```
ifconfig <interface> <enderecoip>
```

Exemplo

```
ifconfig eth0 192.168.0.102 netmask 255.255.255.0
```

Para alterar o gateway padrão:

```
route add default gw 192.168.0.1
```

Para alterar o DNS, maneira 1 – edição manual do /etc/resolv.conf:

```
nameserver <endereco_DNS1>  
nameserver <endereco_DNS2>
```

Para alterar o DNS, maneira 2:

Em alguns sistemas o /etc/resolv.conf é gerado automaticamente a partir do programa resolvconf, que é chamado em rotinas de configuração de endereço no DHCP ou PPPD (conexão ADSL). Neste caso, para alterar o DNS, é necessário incluir um arquivo contendo os endereços dentro do diretório /etc/resolvconf/resolv.conf.d/ que vai ficar como /etc/resolvconf/resolv.conf.d/<seuarquivodeDNS>

Ao ser chamado o resolvconf, ele irá incluir as entradas no arquivo que você criou, o “seuarquivodeDNS”. É importante ressaltar que é recomendável executar as configurações através dos comandos invés de editar diretamente o arquivo /etc/network/interfaces, a não ser que se saiba bem o que está fazendo, pois a edição direta é mais suscetível a erros do que os comandos, que acusam o erro quando os parâmetros estão errados. No entanto, a configuração estática é mais usada em servidores, pois nestes a configuração não muda muito. Para inibir a atualização automática feita pelo resolvconf, basta apagar o arquivo /etc/resolv.conf que é um

link simbólico para o arquivo gerado pelo resolvconf, e criar ele como um arquivo regular, e assim a configuração do DNS nunca será sobrescrita por esta rotina.

Quanto ao comando route, é importante ressaltar que ele não serve apenas para definir o gateway. O route é o comando que edita as tabelas de roteamento do sistema. Com ele, é possível definir o roteamento dos pacotes, classificando-os por origem, como faixa de IPs ou sub-redes.

Alguns outros comandos para resolução de problemas no TCP/IP:

dig

O dig consulta o servidor DNS exibindo detalhes das respostas, facilitando a resolução de problemas em resolução de nomes

ifconfig

O ifconfig permite exibir e editar configurações da interface de rede.

netstat

O netstat exibe o status das conexões de rede.

ping

Serve para testar se um host remoto pode está acessível.

route

Permite ler e configurar informações de gateway padrão e rotas estáticas.

tcpdump

Escreve informações de tráfego de pacotes de rede na tela ou em arquivo.

traceroute

Exibe informações da rota percorrida na rede pelo pacote até chegar no host fim. Exibe cada ponto (hop) que está no meio do caminho.

3.9. IPTABLES (Firewall)

No Linux há uma enorme facilidade de configurarmos diretamente as regras do firewall (chains). O iptables é um módulo que está dentro do kernel Linux para a filtragem de tráfego Ipv4. O ip6tables é o módulo congênere que faz o mesmo para o Ipv6, e o arptables é o módulo que gerencia o tráfego ARP.

Obs.: Cuidado! iptables também é o nome da aplicação em *userspace* na qual configuramos as regras do firewall.

Questões Comentadas

CESPE - 2013 - SERPRO - Analista - Redes

Acerca de uso do protocolo SSH, julgue os itens seguintes.

Em sistemas do tipo Linux, por padrão, a porta 22 com protocolo TCP é utilizada pelo SSH. Entretanto, em configurações avançadas, essa porta pode ser alterada para qualquer porta TCP acima de 1024.

Certo () Errado ()

Comentário:

A porta padrão do serviço SSH é a 22. É comum em servidores mais visados, como uma medida de segurança, alterar esta porta para outra menos comum, o que retarda os ataques de scripts que varrem as portas e quando encontram um serviço ssh, tentam realizar login por força bruta. Há ainda outras medidas de segurança do ssh, como coibir o login no usuário root, e limitar o número de tentativas de acesso seguidas.

Resposta: C

CESPE - 2013 - SERPRO - Analista - Redes

O Kernel do Linux, na versão 2.6 ou superior, suporta nativamente a capacidade de filtro de pacotes TCP/IP por meio do sistema SELINUX.

Certo () Errado ()

Comentário:

O Kernel do Linux suporta nativamente a filtragem de pacotes, porém

isso é desde bem antes da versão 2.6. O outro erro desta questão é dizer que o SELINUX que habilita este recurso. O SELINUX não está relacionado à filtragem de pacotes, ele é uma implementação de recursos adicionais de segurança no Linux, é basicamente uma camada adicional de segurança no esquema de permissões de acesso a arquivos no Linux.

Resposta: E

3.10. Logs do Sistema

Os Logs são os melhores amigos do administrador de sistemas. Com eles, podemos ter uma boa direção para detectar onde estão acontecendo os erros, e assim identificar quais são as medidas corretivas necessárias. Em sistemas críticos, que ocasionam prejuízo grande a cada minuto que estão parados, isso pode ser uma habilidade muito útil.

No Linux, os logs do sistema ficam todos organizados em um diretório padrão, o /var/log. Algumas aplicações, como o httpd (Apache) armazenam seus logs dentro de um diretório em /var/log. Os logs do Linux são coletados por um serviço chamado rsyslog. O arquivo de configuração deste serviço fica em /etc/rsyslog.conf

Tá e o que é armazenado nesses logs?

Ok, vamos citar alguns exemplos:

- /var/log/messages - Contém mensagens globais do sistema, incluindo as mensagens enviadas durante o boot do sistema. Alguns exemplos de coisas que são logadas no messages são: mail, cron, daemon, kern e auth.
- /var/log/dmesg - Esse arquivo contém o chamado “kernel ring buffer”. Valha-me deus! Calma! O Kernel do Linux (e de outros SOs certamente) armazena informações sobre os dispositivos físicos do computador, quando este é iniciado ou quando um dispositivo novo é conectado. Estas informações de identifica-

ção dos dispositivos ficam armazenadas neste no buffer do kernel. Quando alguma mensagem nova atualiza o estado do buffer, ele é sobrescrito. Podemos visualizar este log através do comando dmesg.

- /var/log/auth.log - Contém várias informações de autorização, como logins de usuários e os mecanismos de autenticação usados.
- /var/log/boot.log - Contém informações que são logadas quando o sistema faz boot.
- /var/log/daemon.log - Contém informações logadas por vários daemons (processos de sistema).
- /var/log/dpkg.log - Log do gerenciador de pacotes DPKG, o gerenciador de instalação de programas na distribuição Debian e suas derivadas.
- /var/log/kern.log - Contém informação logada pelo Kernel. Pode ser muito útil para tentar debugar problemas gerados por um Kernel customizado.
- /var/log/mail.log - Contém informações de log do servidor de e-mail do sistema.
- /var/log/Xorg.x.log - Mensagens de log do X, o servidor gráfico do Linux.
- /var/log/cups - Logs do Cups, o gerenciador de impressão do Linux.
- /var/log/faillog - Contém tentativas falhas de login.

Obviamente que não iríamos conseguir exaurir completamente esta lista aqui, porém aqui temos os principais logs do sistema e sua utilidade.

3.11. Superusuário: su e sudo

O comando **su** serve para mudar de usuário ou se tornar o usuário root. Veja a sintaxe dele:

```
$ su [opções] [usuario]
```

Para mudar para um usuário

```
$ su joao
```

Para se tornar super-usuário

```
$ su
```

Para se tornar super-usuário com o mesmo ambiente do usuário root:

```
$ su -
```

O comando **sudo** (**super user do**) permite que o usuário execute comandos como super-usuário sem a necessidade de mudar para um terminal com acesso de super-usuário (o terminal de root fica com “#” em vez de “\$” na linha de comando do usuário normal). A sintaxe dele é bastante simples:

```
$ sudo comando
```

Este é um recurso de segurança para proteger o ambiente (e também a saúde mental do administrador), uma vez que não é recomendável permanecer na linha de comando como usuário *root*, onde qualquer erro pode gerar uma catástrofe no sistema, ou mesmo facilitar surgimento de problemas de segurança. Já vi caras teoricamente experientes e com grandes certificações em segurança ficando logados em root quando poderiam utilizar o sudo, talvez por inabilidade com o sistema. Então amigos, guardem o sudo no coração. Bora usar o sudo. Sudo usem o sudo!

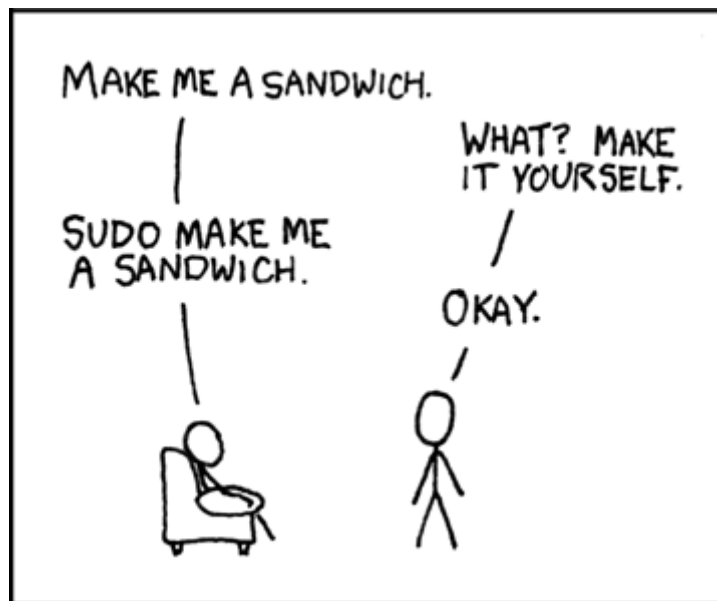


Figura 1: Ilustração (XKCD). Se você não riu desta charge, volte e releia o tópico do sudo.

Para poder utilizar o comando sudo, o usuário deve ter autorização para tal. A autorização se dá pela inclusão do nome do usuário no arquivo /etc/sudoers. Esta inclusão pode ser manual ou pelas interfaces gráficas de administração dos usuários, que incluem automaticamente a autorização no /etc/sudoers quando o perfil do usuário é “administrador”. O importante é que o usuário tenha esta permissão.

3.12. Gerenciamento de Processos

Nos sistemas operacionais, um processo é a forma de representar um programa em execução. É o processo que utiliza os recursos do computador - processador, memória, etc - para a realização das tarefas para as quais a máquina é destinada.

3.12.1. Estados dos processos

Processos podem estar nos seguintes estados:

D: Descansando enquanto espera por outra ação (geralmente E/S), sem possibilidade de interrupção.

R: Executando (Running).

S: Descansando enquanto espera por outra ação (esperando algum evento ser completado), com possibilidade de interrupção.

T: Parado, suspenso.

Z: Zumbi. O processo foi terminado mas não foi removido do sistema.

3.13. Visualizando os estados dos processos

Comando **ps** - Listar processos

Lista os processos em execução, apresentando o PID e outras informações sobre o processo, como o comando executado (CMD) e estado atual do processo (STAT).

```
$ ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME
COMMAND
root           1  0.0  0.0  27064  2836 ?        Ss   20:17   0:00
/sbin/init
root           2  0.0  0.0      0     0 ?        S    20:17   0:00
[kthreadd]
root           3  0.0  0.0      0     0 ?        S    20:17   0:02
[ksoftirqd/0]
root           5  0.0  0.0      0     0 ?        S<   20:17   0:00
[kworker/0:0H]
root           7  0.0  0.0      0     0 ?        S<   20:17   0:00
[kworker/u:0H]
...
```

Comando **top** - Lista processos em tempo real

```
$ top

top - 23:30:27 up 3:13, 2 users, load average: 0,60, 0,56, 0,52
Tarefas: 224 total, 1 running, 221 sleeping, 0 stopped, 2 zombie
%Cpu(s): 3,0 us, 1,5 sy, 0,0 ni, 95,0 id, 0,4 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem: 8062600 total, 5763844 used, 2298756 free, 336052 buffers
KiB Swap: 4194300 total, 0 used, 4194300 free, 1716672 cached

  PID USER      PR  NI  VIRT  RES  SHR S  %CPU  %MEM    TIME+  COMMAND
 2813 victor    20   0 2920m 1,9g  54m S   8,3  25,3   39:00.65 firefox
```

```
2222 victor      20    0 1496m 130m  39m S   3,7  1,7   9:23.10 compiz
1260 root        20    0  240m  75m   14m S   3,0  1,0  12:41.05 Xorg
3341 victor      20    0 1450m 292m  80m S   2,7  3,7   6:50.40 soffice.bin
8446 victor      20    0  587m  20m   12m S   1,3  0,3   0:00.34 gnome-termi-
nal
3112 victor      20    0  729m  74m   30m S   0,7  0,9  11:00.93 plugin-con-
tainer
...
```

Comando pstree - Lista os processos em forma de árvore

```
$ pstree
init--NetworkManager--dhclient
                        --dnsmasq
                        --3*[{NetworkManager}]
--accounts-daemon--2*[{accounts-daemon}]
--acpid
--aptd--{aptd}
--at-spi-bus-laun--dbus-daemon
                  --3*[{at-spi-bus-laun}]
--at-spi2-registr--{at-spi2-registr}
--avahi-daemon--avahi-daemon
--bamfdaemon--3*[{bamfdaemon}]
--bluetoothd
--colord--{colord}
--console-kit-dae--64*[{console-kit-dae}]
--cron
...
```

Observe que todos processos vêm do init.

Veja o que acontece quando listamos os processos com o comando ps -e

```
$ ps -e
PID TTY          TIME CMD
  1 ?            00:00:01 init
  2 ?            00:00:00 kthreadd
  3 ?            00:00:00 migration/0
  4 ?            00:00:02 ksoftirqd/0
  5 ?            00:00:00 watchdog/0
 27 ?            00:00:01 events/0
 35 ?            00:00:00 cpuset
...
```

O Process ID (PID) do init é o 1! Questão de prova! O init é o primeiro processo invocado pelo kernel quando do boot do sistema, e todos são filhos dele.

3.13.1. Sinais de processos

O SIGKILL (9) é o sinal que mata o processo, sem hesitação do sistema. Este sinal tem que ser usado com cuidado, pois pode gerar processos zumbis ao matar um processo pai deles.

O SIGTERM (15) é recebido pelos programas dizendo-os para finalizar de forma normal.

O SIGSTOP (19) é o sinal que suspende os programas.

Para enviar estes sinais aos processos nós utilizamos os comandos kill e killall.

Comando kill - envia um sinal ao processo (PID = Nº de id. do Processo)

```
$ kill [-sinal] PID
```

Comando killall

```
$ killall [-sinal] nomedoprocesso
```

3.13.2. Realocação de prioridades de processos

Para configurar as prioridades de CPU dos processos, nós utilizamos os comandos **nice** e **renice**. Com o nice nós executamos programas com prioridade modificada. Já o renice serve para modificar prioridade em um processo que já está rodando.

No Linux, as prioridades vão de -20 a +19. As prioridades menores (negativas) são as mais importantes.

```
$ nice [opções] [comando]
```

```
$ renice -n prioridade -p PID
```

Questões Comentadas

CESPE - 2013 - Telebras - Especialista em Gestão de Telecomunicações - Analista de TI

A respeito dos sistemas operacionais Windows e Linux, julgue os itens a seguir.

Para obter uma lista dos usuários logados no sistema operacional Linux, é necessário executar o comando top.

Certo () Errado ()

Comentário:

O comando que exibe os usuários logados é o who, não o top. O comando top exibe os processos em execução.

Resposta: Errado.

CESPE - 2012 - Banco da Amazônia - Técnico Científico - Suporte Técnico

No que se refere à instalação, configuração e administração do sistema operacional Linux, julgue os itens que se seguem.

Os processos em execução em primeiro plano (foreground) são exibidos após a execução do comando jobs.

Certo () Errado ()

Comentário:

Um processo em execução no prompt do shell pode ser enviado para o plano de fundo (background), liberando assim o shell para outros co-

mandos, através das opções:

1) quando o criamos, através do caractere “&” usado no final do comando:

[usuario@maquina](#):\$ comando &

2) Através da combinação de teclas Ctrl+Z enquanto o comando está rodando, e em seguida pelo comando **bg**. Isso irá respectivamente parar o processo liberando a linha de comando, e executá-lo em background.

Ambas opções trazem um processo em execução. Para retornar o comando para o plano principal (**foreground**), utilizamos o comando “**fg** comando”. Para visualizarmos os comandos rodando em background ou parados, utilizamos o comando **jobs**.

Resposta: Errado

CESPE - 2010

A respeito de sistemas operacionais Linux e Unix, julgue os itens que se seguem.

No sistema operacional UNIX, o comando nice faz que um processo seja disparado com alta prioridade.

Certo () Errado ()

Comentário:

O comando nice permite que nós iniciemos um processo definindo a prioridade dele. A prioridade no Linux varia de -20 (maior prioridade) a 19 (menor prioridade). A sintaxe do comando é nice [opção] [programa], onde em [opção] colocamos a prioridade de -20 a 19 e em [programa] informamos o nome do comando, serviço ou aplicativo que iremos chamar. Portanto, podemos utilizar o nice para chamar um processo com prioridade baixa, tornando a afirmativa errada. Se ele falasse algo do tipo “o co-

mando nice permite”, estaria correto, porém da maneira como está escrito restringe a utilidade do nice ao que está escrito. Errado.

Resposta: E

4. Programação bash

O shell é uma camada de interação entre o kernel do sistema operacional e o usuário. Esta interface pode ser por linha de comando (CLI) ou via gráfica (GUI). Quando falamos de shell linux, estamos falando de linha de comando. O shell é um processador de comandos em modo texto. Vários shells foram especificados no mundo Unix, entre eles o Korn shell (ksh) e o C shell (csh). O Bash, ou Bourne-again shell é o shell que abordaremos, e surgiu como uma alternativa livre, para o sistema GNU, ao Bourne shell e também veio com alguns melhoramentos.

Vamos ver alguns dos principais comandos. O símbolo \$ denota que estamos num terminal como um usuário normal, o \$ não faz parte do comando. Parâmetros entre “[]” são opcionais, enquanto os sem estes símbolos são obrigatórios para que o comando funcione.

```
$ pwd
```

O comando pwd mostra o diretório atual.

```
$ ls
```

O ls (list) equivale ao “dir” do DOS, ele mostra os arquivos de um determinado diretório.

```
$ ls -a
```

Lista todos os arquivos (all)

```
$ ls -l
```

Lista todos arquivos mostrando tamanho e permissões dos usuários

```
$ cd
```

Muda o diretório para o seu diretório de usuário

```
$ cd nomedir
```

Muda para o diretório “nomedir”, a partir do diretório atual

```
$ cd /diretorio1/subdiretorio
```

O mesmo que o comando anterior, mas está utilizando o caminho completo desde o diretório raiz (“/”)

```
$ cd ..
```

Muda para um diretório acima

```
$ tar -cvf nomedoarquivo.tar nomedodiretorio
```

Cria um arquivo contendo todos arquivos do diretório “nomedodiretorio” (Concatenate, Verbose, File)

```
$ tar -zcvf nomedoarquivo.tar.gz nomedodiretorio
```

O mesmo do comando anterior, mas compacta o diretório com o gunzip

```
$ tar -xvf nomedoarquivo.tar.gz [caminhododiretorio]
```

Extrai os arquivos compactados no diretório. Caso a última opção seja deixada em branco, será extraído no diretório atual.

```
$ head [-número] nomedoarquivo
```

Este comando é utilizado para mostrar as 10 primeiras linhas (head = cabeçalho) de um arquivo texto ou um fluxo de texto. Se usado com a opção -número, exibe as “número” primeiras linhas.

```
$ tail
```

Este comando é utilizado para mostrar as 10 últimas (tail = calda) linhas de um arquivo texto ou um fluxo de texto. Se usado com a opção -número, exibe as “número” últimas linhas.

```
$ man nomedocomando
```

Este é o comando mais importante de todos. Com ele você consulta o manual dos outros comandos.

```
$ grep expressão arquivo
```

Este comando é utilizado para filtrar uma determinada expressão em um arquivo ou fluxo de texto.

```
$ sort [opções] arquivo
```

Lista ordenada alfabeticamente as linhas de texto de um arquivo. Pode listar a ordem de forma reversa ou de outras formas, de acordo com as opções.

Vamos fazer um pequeno exemplo prático? Vamos fazer um Shell Script de uma única linha. Digamos que você está num servidor, num diretório que armazena dados de usuários e deseja ver quais diretórios estão consumindo mais espaço em disco. Para listar isto, usaremos o comando **du** (disk usage), que mostra o uso de disco de cada diretório seus subdiretórios. Para não listar o uso de cada sub-diretório, somente 1 nível de diretório, iremos utilizar também a opção **-max-depth=1**. A maneira de listar o tamanho em formato legível para nós humanos é com o parâmetro **-h**.

Usarei meu diretório como exemplo, mas podemos imaginar em vez dos diretórios listados abaixo, nomes ou matrículas de usuários no lugar deles.

```
$ du -h --max-depth=1
2,0M  ./purple
36K   ./dbus
1,8G  ./Imagens
1,7G  ./Área de Trabalho
272K  ./java
504K  ./Documentos
61M   ./mozilla
41M   ./android
72K   ./winff
... (muitas linhas)
```

Ao executar este comando, muitos diretórios são listados de forma desordenada. Uma quantidade tão grande que chega a poluir a tela e não há ordenação de tamanho. O que fazer? Precisamos ordenar. Jogaremos a saída deste comando direto em outro comando, o **sort**. Para jogar a saída de um comando em outro, nós utilizamos o genial pipe, o símbolo “|”, que conecta um comando com o outro.

```
$ du -h --max-depth=1 | sort
4,0K  ./gnome2_private
4,0K  ./gphoto
4,0K  ./gvfs
4,0K  ./Modelos
4,0K  ./nvidia
4,0K  ./Ubuntu One
41M   ./android
4,2M  ./josm
4,3M  ./multisystem
... (muitas linhas)
```

Porém eu utilizei a opção `-h` no comando `du`, que está alterando o tamanho dos arquivos para torná-lo mais fácil de ler pra mim e está alterando a ordenação. Devo utilizar a opção `-h` no `sort` também.

```
$ du -h --max-depth=1 | sort -h
... (muitas linhas)
4,2M  ./josh
4,3M  ./multisystem
4,8M  ./workspace
6,3M  ./xmind
16M   ./thunderbird
25M   ./nv
41M   ./android
... (muitas linhas)
```

Viram que mágica? Agora ele está ordenando! Porém quero ordenar de forma decrescente com o parâmetro “`-r`” (r de reverse), pois quero listar primeiro os diretórios maiores. Após isto quer filtrar somente os 10 primeiros maiores diretórios. Irei jogar a saída do **`sort -hr`** no comando **`head`**.

```
$ du -h --max-depth=1 | sort -hr | head
111G  .
35G   ./Público
20G   ./Downloads
16G   ./VirtualBox VMs
15G   ./Vídeos
8,7G  ./Dropbox
4,7G  ./PlayOnLinux
3,8G  ./local
2,9G  ./tmp
1,8G  ./Imagens
```

Pronto! Agora tenho a lista decrescente dos 10 diretórios que mais consomem armazenamento no meu disco gerado pelo comando que acabei de fazer.

Para gerarmos um script a partir do comando que a gente usou, basta criar um arquivo “`lista_espaco_diretorios.sh`” com o conteúdo abaixo.

```
#!/bin/bash
du -h --max-depth=1 | sort -hr | head
```

Depois de salvo, é necessário atribuir permissão de execução a este script.

```
$ chmod +x lista_espaco_diretorios.sh
```

Para usá-lo, basta entrar no diretório no qual você deseja listar o tamanho dos subdiretórios e executar o script que você acabou de criar:

```
$ sh caminho-onde-foi-salvo-oscript/lista_espaco_diretorios.sh
```

Amigos, obviamente shell script não é só isso, mas a filosofia é essa, ir tratando os dados que você deseja com uma sucessão de comandos, conforme suas necessidades. Um shell script mais complexo envolveria também algumas estruturas de repetição e condicionais (for, while, if), que são muito parecidas com as usadas nas linguagens de programação, e também várias ferramentas de expressões regulares como o grep e o awk. O objetivo do shell script é facilitar e até automatizar tarefas de administração do sistema, como ao listar os diretórios que estão consumindo mais espaço, que poderia ser utilizada em uma área de armazenamento de dados de uma aplicação, por exemplo.

Vamos ver mais alguns comandos.

mv	Move arquivos ou pastas
rm	Remove arquivos ou pastas. Opção -r remove recursivamente.
uniq	Lista as entradas únicas em uma sequência de texto, eliminando as linhas
wc	Word Counter, lista números de linhas, palavras, bytes de cada arquivo
wc -l	Lista a quantidade de linhas de um arquivo
cat	Exibe o conteúdo de um arquivo
sudo comando	Serve para executar o “comando” como super-usuário. Falaremos dele mais adiante.

Questões Comentadas

CESPE - 2012

Após a execução do comando `cat /var/log/messages > /tmp/lista`, nenhuma informação retorna em tela, sendo disponibilizado o prompt para novos comandos. O comando redireciona a saída padrão para o arquivo `/tmp/lista`.

Certo () Errado ()

Comentário:

Vamos relembrar. O comando `cat` exibe o conteúdo de um arquivo no terminal. O caractere “>” dentro dos principais shells é o redirecionador, usado para redirecionar a saída do comando que iria para a saída padrão (a tela) para ir para outro lugar, no exemplo o arquivo `/tmp/lista`. Bate com o que o enunciado falou? Sim. Nenhuma informação é retornada em tela pois foi redirecionada para o arquivo. O prompt é liberado para novos comandos. E por fim a saída é redirecionada para o arquivo usado no comando.

Resposta: C

CESPE - 2011 - EBC - Analista - Administração de Sistemas

Julgue os itens subsequentes, a respeito dos ambientes Windows e Unix.

Quando a instrução `cat arquivo | grep sol` é executada em ambiente Unix, o comando `cat` tem a função de exibir o conteúdo do arquivo e o pipe (|), de repassar a saída do comando `cat` para a entrada do comando `grep`.

Certo () Errado ()

Comentário:

O pipe, caractere “|”, serve para ligar a saída de um comando à entrada de outro. Ao utilizarmos ele no comando acima, estamos jogando o conteúdo do “arquivo”, que seria jogado na saída padrão, o terminal, na entrada do comando grep. O parâmetro usado no comando grep faz com que sejam filtradas todas as linhas que contenham a string “sol”.

Resposta: C

CESPE - 2011 - MEC - Administrador de Redes

Julgue os próximos itens, relativos aos sistemas operacionais Windows, Unix e Linux.

No Unix, os scripts shell são arquivos de texto que possuem permissão de execução; a primeira linha de um script shell deve-se iniciar com #\$. .

Certo () Errado ()

Comentário:

Esta questão é para pegar quem não tem vivência com shell script. A primeira linha de um script shell sempre inicia com **#!** e em seguida o caminho do interpretador de shell que será utilizado. Se utilizarmos o bash, a primeira linha de um shell script será :
#!/bin/bash

Caso tenha dificuldade de lembrar, lembre-se do [Barrabin Barrabash, o grande](#).

Resposta: Errado

CESPE - 2009 - ANAC - Técnico Administrativo - Informática

Um arquivo shell script pode ser usado quando a sequência de comandos que se necessita repetir com frequência pode ser colocada em um arquivo que, ao ser executado, ocasiona o disparo de cada um dos comandos da sequência, na ordem indicada.

Certo () Errado ()

Comentário:

Um script serve exatamente para o que o elaborador falou: para facilitar a vida do administrador quando é necessário executar uma série de atividades repetitivas. Imagine renomear 200 arquivos? É um trabalho muito maçante e ineficiente fazer isto manualmente, um por um. Os scripts facilitam a vida do administrador e dão sentido à utilização de uma estrutura de computadores num ambiente de TI.

Resposta: C

CESPE - 2008 - SERPRO - Técnico - Operação de Redes

É recomendado que todos os scripts tenham uma especificação de caminho de busca, por exemplo `PATH=/usr/ucb:/usr/bin:/bin; export PATH`. O Bourne Shell tem como padrão exportar variáveis de ambiente para scripts filhos. Nesse caso, a utilização do comando `export` não é necessária para este efeito.

Certo () Errado ()

Comentário:

Há várias maneiras de uma variável PATH estar definida quando o script executa e utiliza esta variável. A primeira é definindo globalmente para **todo** o sistema através do arquivo /etc/environment. A segunda é definir a variável para o usuário, fazendo a configuração nos arquivos .bashrc ou .bash_profile que ficam no diretório home do usuário. A terceira é exatamente a abordada pela questão, utilizando o comando export. Ao utilizarmos o export, todos os sub-shells (sim, um script pode chamar sub-shells!) herdarão a definição da variável do shell pai. Portanto é errado dizer que a utilização do comando export não é necessária para isso, pois é exatamente este comando que faz isto.

Resposta: E

5. Sistemas de arquivos

5.1. Sistema de arquivos EXT4

O sistema de arquivos ext4 é o sucessor do ext3 como principal sistema de arquivos do Linux, e traz novos aprimoramentos em recursos, escalabilidade, confiabilidade e desempenho.

Nas características de funcionalidade, para facilitar a migração, o ext4 suporta o ext3. Explicando melhor, é possível montar um FS ext3 como ext4.

Na questão de escalabilidade, o ext4 aumentou seu suporte de volumes de sistema de arquivos, tamanhos de arquivos e limites de subdiretórios. O ext4 suporta sistemas de arquivos de até 1 exabyte, arquivos de até 16 TB. Vale lembrar que este limite de 1 exabyte é teórico do filesystem, pois na prática é necessário muito processamento para gerenciar as informações sobre um sistema de arquivos de 1 exabyte.

Em matéria de confiabilidade, o ext4 implementa verificação de erros (checksum) no journal (log de modificações do sistema de arquivos), evitando assim problemas mesmo na probabilidade remota de o journal ser corrompido em caso de algum erro. A desfragmentação on-line é outro recurso que melhora também o tempo de verificação de erros (fsck) no sistema de arquivos.

Questões Comentadas

CESPE - 2013

No que se refere ao ambiente Linux e ao sistema de arquivos

EXT4, julgue o item a seguir.

O sistema de arquivos EXT4 do Linux possui suporte a ACLs (access control list) POSIX.

Certo () Errado ()

Comentário:

ACLs no filesystem são recursos controle de acesso aos arquivos além das permissões que apresentamos (-rwxrwx---), com listas de controles de acessos para vários usuários, por exemplo. Isto é um recurso opcional suportado sim no EXT4.

Resposta: Certo.

CESPE - 2012

Para criar o sistema de arquivos do tipo ext3 no dispositivo /dev/sdd1, deve-se executar o comando mkfs -t ext3 /dev/sdd1.

Certo () Errado ()

Comentário:

O comando mkfs cria um sistema de arquivos. A opção -t especifica o tipo. O tipo ext3 e o caminho /dev/sdd1 são os demais parâmetros do comando para que ele funcione.

Resposta: Certo

CESPE - 2012 - TJ-AC - Técnico Judiciário - Informática

Tanto no Linux quanto no Windows 7 a organização de diretórios e arquivos é realizada por meio de estruturas em árvore. A diferença é que no Windows 7 com NTFS um diretório somente pode estar hierarquicamente associado a no máximo um diretório superior, enquanto no Linux com ext4, não há esta restrição.

Certo () Errado ()

Comentário:

Os sistemas de arquivos atuais são todos baseados em árvores. É a estrutura de dados utilizada para montar a relação entre os arquivos e diretórios do sistema de arquivos, e para executar a busca por estes. Há até um sistema de arquivos chamado btrfs que utiliza o nome da estrutura de dados: B-Tree File System. Portanto, como sabemos um nó em uma árvore não tem como estar associados a mais de um superior hierárquico. Este conceito é bem diferente e não é utilizado nem no NTFS nem no EXT4.

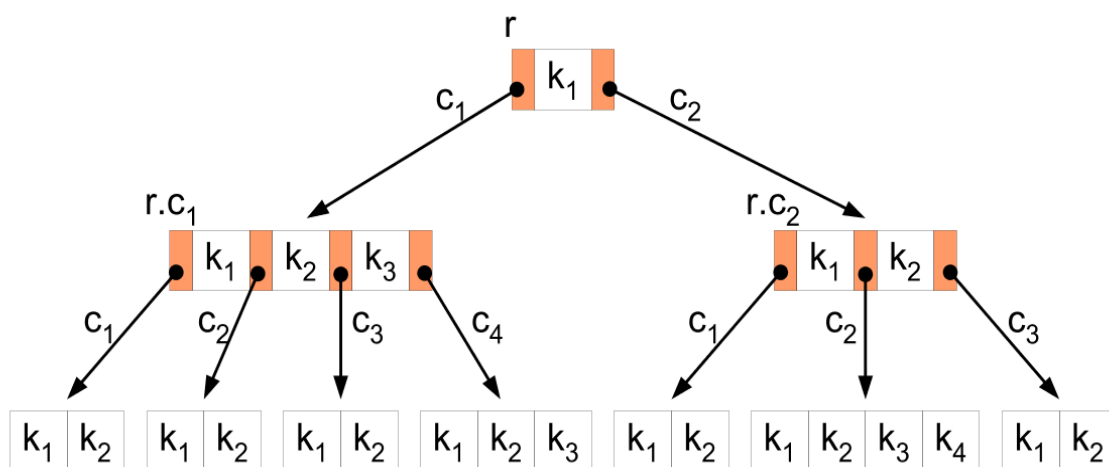


Ilustração: uma árvore B.

Resposta: E

CESPE - 2012 - FNDE - Especialista em Financiamento e Execução de Programas e Projetos Educacionais

Julgue os itens a seguir, relativos aos sistemas operacionais Linux e Windows.

No Linux, o sistema de arquivos pode ter um formato arbitrariamente longo. Entretanto, cada componente de um nome de caminho deve ter, no máximo, 4.095 caracteres.

Certo () Errado ()

Comentário:

Essa questão é interessante porque amedronta o candidato com detalhes técnicos, mas . Todo sistema de arquivos tem um limite máximo de volume, número máximo de arquivos. Aqui já podemos afirmar que a questão está errada, sem precisar saber detalhes técnicos.

Mesmo os sistemas de arquivos mais recentes, como o ext4, possuem um limite. Para este filesystem, os limites de tamanho máximo de volume é de 1 EB (exabyte), e o tamanho de número de arquivos é 4 bilhões. Mesmo não sendo facilmente atingível, há um limite.

O limite máximo do tamanho de arquivos no ext3 e ext4 é de 255 caracteres.

Resposta: E

CESPE - 2011

Acerca dos sistemas operacionais Windows 2003, Windows 2008

Server, Linux RedHat e Suse, julgue os itens de 36 a 40.

O sistema operacional Red Hat Enterprise Linux 6 suporta o gerenciamento e armazenamento de dados da ordem de 15 TB, em sistema de arquivos nativo ext4.

Certo () Errado ()

Comentário:

Resposta Certa. O elaborador tirou a pergunta direto do manual, com uma pequena pegadinha para confundir. Anote aí. O limite do sistema de arquivos EXT4 no RHEL 6 é 16 TB, que é o limite recomendado tecnicamente. Como a questão foi escrita com a expressão “da ordem”, a afirmação é correta.

CESPE - 2011 - FUB - Analista de Tecnologia da Informação - Específicos

Com relação à interoperação de sistemas operacionais, família Windows e sistemas Unix, julgue os itens subsequentes.

Para que o Linux possa fazer a leitura de partições NTFS, é imprescindível o módulo de núcleo ntfs.ko.

Certo () Errado ()

Comentário:

O NTFS é um sistema de arquivos produzido pela Microsoft, para o Windows. O suporte dele foi inicialmente provido através de algumas ferra-

mentas, até que foi desenvolvido o módulo para suporte pelo kernel, o ntfs.ko.

“.ko” é uma das extensões utilizadas para denotar módulos do kernel.

Resposta: C

6. Conclusão

Prezados, espero reencontrá-los por aqui nas próximas aulas. Tere-mos muito conteúdo para trabalhar e espero trazer um conteúdo que seja rico pra vocês, além de muitas questões comentadas.

Gostaria de lembrá-los que vocês que adquirirem o curso terão du-rante o período até a prova a possibilidade de enviar dúvidas e questões para resolução, então aproveitem, pois este é um diferencial que pode ajudá-los a acertar mais questões na prova. Aguardo também através do e-mail victor.carvalho@tiparaconcursos.net sugestões que vocês acharem pertinentes.

Até a próxima e bons estudos!