

SEG12 - Atividades - Semana 1

Francisco Marcelo, Marcelo Karam e Felipe Scarel

06-08-2018

Introdução ao sistema operacional Linux

1) Identificando bits de permissão

1. Verifique as permissões do diretório `/tmp`. O que você percebe de diferente em relação às permissões de *outros*?
2. Considerando que há permissão de escrita no diretório para todos, o que o impediria de remover um arquivo de outra pessoa?

2) Identificando e entendendo *hard links*

O número de *links* (*link counter*) que apontam para um arquivo é mantido em seu *inode*. Esse contador é utilizado pelo sistema para controlar a liberação dos blocos do disco alocados ao arquivo quando o contador atingir o valor zero, ou seja, quando nenhum outro arquivo estiver apontando para o *inode*.

1. Qual o número de *links* do seu diretório *home*?
2. Crie o arquivo `arqses1ex3` no seu diretório *home*. Utilize o comando `touch`.
3. Verifique o número de *links* do arquivo `arqses1ex3` e anote o resultado. Você pode utilizar o redirecionamento de saída para registrar esse resultado no próprio arquivo criado. Essa informação será necessária para uma atividade posterior.
4. Verifique se mudou o número de *links* do seu diretório *home*.
5. Crie um diretório com o nome de `dirses1ex3`, também no seu diretório *home*.
6. Mais uma vez, verifique o número de *links* do seu diretório *home*. Ele mudou? Você saberia dizer por quê?
7. Qual o número de links do diretório `dirses1ex3`?
8. Verifique qual opção deve ser passada ao comando `ls` para que ele liste as informações do diretório `dirses1ex3` e não o seu conteúdo.
9. Você saberia explicar por que o número de *links* do diretório `dirses1ex3` é maior que um?

3) Conhecendo diferenças entre *hard link* e *symbolic link*

Foi explicada a importância dos *links* criados com o comando `ln`. Para criar um *symbolic link*, a opção `-s` deve ser informada na linha de comando. Consulte as páginas do manual para conhecer outras opções.

1. No seu diretório de trabalho, crie um *hard link* para o arquivo `arqses1ex3`. O nome do arquivo criado deverá ser `hosts.hard`.
2. Verifique agora o número de links do arquivo `arqses1ex3` e compare com aquele obtido na atividade 2. Explique a diferença.
3. Crie um *symbolic link* para o arquivo `arqses1ex3`, que deverá se chamar `hosts.symbolic`.

4. O número de *links* do arquivo `arqses1ex3` aumentou?
5. Caso não tenha aumentado, por que isso aconteceu, considerando que foi criado um *link* para ele?
6. Qual o tamanho do arquivo `hosts.symbolic`?
7. Você percebe alguma correlação entre o tamanho e o arquivo para o qual ele aponta?

4) Trabalhando com *hard link* e *symbolic link*

1. Se o arquivo original `arqses1ex3` fosse removido, o que aconteceria se tentássemos acessá-lo pelo *hard link*? E pelo *symbolic link*?
2. Depois de responder a essas questões, remova o arquivo criado (`arqses1ex3`) e verifique se as suas respostas estão corretas.

5) Conhecendo algumas limitações do *hard link*

1. Crie um arquivo chamado `arqses1ex6`. Em seguida, crie um *hard link* para esse arquivo com o nome `link-arqses1ex6` no diretório `/tmp`. O que aconteceu? Por quê? Como resolver esse problema?

6) Criando *links* para diretórios

Crie, no seu diretório *home*, um *link* simbólico para o diretório `/usr/bin` com o nome de `link-bin`. Com o *link* criado, execute o seguinte:

1. Mude para o diretório `link-bin`.
2. Agora, vá para o diretório pai (utilize a notação `..`). Você saberia explicar por que se encontra no seu diretório *home* e não no diretório `/usr`?

7) Alterando permissões de arquivos e diretórios

O comando `chmod` é utilizado para modificar as permissões de um arquivo. Utilizando a notação octal, execute a seguinte sequência:

1. Modifique a permissão do seu diretório *home* de modo a retirar a permissão de escrita do seu dono.
2. Verifique as permissões associadas ao arquivo `arqses1ex6`. Você tem permissão para escrever nesse arquivo? O grupo tem?
3. Tente remover o arquivo `arqses1ex6`. Você conseguiu? Em caso negativo, você sabe explicar o motivo?
4. Modifique as permissões do arquivo `arqses1ex6` de forma a retirar a permissão de escrita para o dono e colocá-la para o grupo.
5. Com o uso de redirecionamento, tente copiar o conteúdo do seu diretório *home* para dentro do arquivo `arqses1ex6`.

6. Torne a colocar a permissão para escrita no seu diretório *home* para o dono.

8) Atribuindo as permissões padrão

1. Crie arquivos (*arq1ses1ex9*, *arq2ses1ex9*, etc.) e diretórios (*dir1ses1ex9*, *dir2ses1ex9*, etc.) em seu diretório *home*, após definir cada uma das seguintes *umasks*: *000*; *002*; *003*; *023*; *222*; *022*. Em seguida, observe as permissões que foram associadas a cada um dos arquivos e diretórios.

9) Entendendo as permissões padrões

1. Na execução do exercício anterior, você saberia explicar por que, ainda que utilizando a mesma *umask*, as permissões associadas ao arquivo criado diferem das do diretório?

Usuários e grupos

1) Criando contas de usuários

Uma das atividades que fazem parte da rotina diária de um administrador de sistemas é o gerenciamento de contas de usuários. Frequentemente, usuários são criados, modificados, desabilitados ou excluídos do sistema.

1. Descubra se o sistema faz uso de *shadow passwords* ou se ainda utiliza o esquema tradicional.
2. Crie uma conta para você no sistema, seguindo os passos descritos na aula teórica e no material didático.
3. Agora, crie uma conta para o instrutor, utilizando, desta vez, o comando `useradd`. Faça com que a conta criada tenha sete dias de duração e com que o seu diretório de trabalho seja `/NOME`, onde `NOME` é o nome de usuário para o qual a conta deve ser aberta.
4. O comando `useradd` não é uma boa opção para informar a senha do usuário. Por quê?
5. Faça um *script* que simule o comando `newusers`. Para isso, você deve criar um arquivo texto contendo as informações a respeito dos usuários, mantendo o mesmo padrão dos arquivos lidos pelo comando `newusers` (para descobrir o formato, consulte a página de manual: `$ man 8 newusers`). Como este arquivo conterá as senhas dos usuários, é importante removê-lo logo após a criação das contas.

Dica: Utilize a variável de sistema `IFS` (*Internal Field Separator*) em seu *script* para definir o caractere ":" como campo que separa as informações sobre as contas.

2) Verificando e modificando informações de contas de usuário

Após a criação de uma conta, é fundamental que o administrador verifique se ela foi criada corretamente.

1. Entre no sistema com o usuário criado no item 3 da atividade 1 e execute os comandos indicados para verificação de uma conta.
2. Seria possível inserir o número de telefone de trabalho desse mesmo usuário, junto com a informação de quem ele é? Faça isso e torne a checar se a sua mudança surtiu efeito.

3) Criando grupos de usuários

O recurso de grupos de usuários é muito útil para compartilhar informações. No momento em que a conta `instrutor` foi criada, no item 3 da atividade 1 deste roteiro, o grupo primário ficou sendo o seu próprio nome de usuário. Isso ocorre sempre que não é atribuído um valor para o grupo primário, no momento da criação de um novo usuário. Como o usuário criado não faz parte de outro grupo, a não ser do seu próprio, ele somente poderá acessar seus arquivos ou aqueles arquivos para os quais haja permissão de acesso para outros usuários.

1. Use o comando apropriado para criar um grupo chamado **grupoteste**.
2. Liste o arquivo **/etc/group** e anote o **GID** que foi atribuído ao grupo criado.
3. Aproveite para observar, no arquivo **/etc/group**, quais são os outros grupos existentes no sistema. Qual o grupo associado ao usuário **root**?
4. Altere o grupo primário do usuário **instrutor**, de modo que este passe a ser o grupo criado no item 1 da atividade 3, **grupoteste**.
5. Se autentique no sistema utilizando a sua conta e inclua seu usuário como administrador do grupo **grupoteste**. Em seguida inclua o usuário **instrutor** no grupo **grupoteste**. Você conseguiu executar as tarefas propostas? Por quê? Como você deve fazer para realizar as tarefas?
6. Altere novamente o grupo primário do usuário **instrutor** para o grupo **instrutor**.

4) Incluindo usuários em grupos secundários

1. Editando o arquivo **/etc/group**, inclua, no grupo **grupoteste**, o usuário criado no terceiro item da atividade 1 desse roteiro (**instrutor**). Note que o grupo primário do usuário não deve mudar; continua sendo o nome do usuário.
2. Agora, utilize um comando apropriado para inserir nesse mesmo grupo o usuário criado para você no primeiro item da atividade 1.

5) Bloqueando contas de usuários

No Linux, é possível impedir temporariamente o acesso ao sistema mesmo que o usuário esteja utilizando uma conta com acesso liberado a este.

1. Utilizando um comando apropriado, bloqueie a conta criada para o instrutor e teste se obteve sucesso no bloqueio.
2. Agora desbloqueie a conta e faça o teste de acesso para verificar se sua alteração surtiu efeito.

6) Removendo uma conta de usuário manualmente

No Linux, é possível executar uma mesma tarefa de diversas maneiras. Para um administrador de sistemas, é importante conhecer essas alternativas, porque elas podem ser úteis em situações específicas em que não seja possível utilizar um dado recurso ou ferramenta do sistema.

1. Sem utilizar o comando **userdel**, remova a conta criada para você no segundo item da atividade 1.
2. Certifique-se de que esse usuário foi realmente excluído do sistema, utilizando um dos comandos que fornecem informações sobre os usuários.
3. Faça um backup de seus dados de modo que o instrutor possa ter sobre eles o mesmo tipo de acesso que você.

7) Obtendo informações sobre usuários

Muitas vezes, é necessário obter informações sobre os usuários de um sistema. Dois comandos que fornecem informações sobre usuários são `finger` e `id`.

1. Verifique os parâmetros do usuário criado na atividade 1 utilizando esses comandos, e descreva a diferença entre os dois a partir dos resultados obtidos. Consulte as páginas de manual para verificar as opções disponíveis nestes comandos.

8) Removendo contas de usuários

1. Utilizando os comandos apropriados, remova a conta criada para o instrutor. Não se esqueça de que um grupo foi especialmente criado para ele e que ele também possui um grupo secundário.

9) Alterando o grupo a que um arquivo pertence

O arquivo `/etc/passwd` contém informações importantes sobre os usuários do sistema. Esse arquivo pertence ao usuário `root` e ao grupo `root`. As permissões de acesso desse arquivo definem que ele só poderá ser modificado pelo usuário `root`.

1. Faça com que esse arquivo pertença ao grupo `grupoteste`, criado na atividade 3. Com isso, os usuários desse grupo, incluindo o usuário criado na atividade 1 poderão acessar esse arquivo por meio das permissões definidas para os usuários do grupo.

10) Alterando permissões de acesso de arquivos

É muito comum o administrador ter que modificar a permissão de arquivos para possibilitar ou impedir que eles sejam lidos ou modificados por diferentes categorias de usuários. A melhor forma de fazer isso é utilizando o comando `chmod`.

1. O arquivo `/etc/passwd` tem apenas permissão de leitura para os usuários do seu grupo proprietário. Use o comando `chmod` para atribuir permissão de escrita ao grupo proprietário desse arquivo. A permissão de escrita nesse arquivo é inicialmente atribuída apenas ao usuário proprietário do arquivo.
2. O setor de controladoria de uma empresa só possuía um funcionário, que pediu demissão. Como não há um diretório específico para armazenar os arquivos do setor, todos os seus arquivos de trabalho estão armazenados em seu diretório `home`. Que passos você deve fazer para disponibilizar estes arquivos para o novo funcionário que será contratado e para que este tipo de problema não volte a ocorrer?

Observação: Por motivos de segurança, ao final das atividades, retorne a permissão e o grupo do arquivo `/etc/passwd` para os valores originais.

```
# chown root.root /etc/passwd
# chmod 644 /etc/passwd
# ls -lh /etc/passwd
-rw-r--r-- 1 root root 1,7K Ago  7 16:22 /etc/passwd
```


Processos

1) Descobrendo o número de processos em execução

1. Quantos processos estão sendo executados na máquina no momento? Use o comando `wc` para contá-los.
2. Faça um *script* que liste o número de processo que cada usuário está executando.

2) Descobrendo o PID e o PPID de um processo

1. Quais os valores de `PID` e `PPID` do shell que você está utilizando no sistema?
2. Faça um *script* que liste todos os processos que foram iniciados pelo processo `init`. A lista não deve conter mais de uma ocorrência do mesmo processo.

3) Estados dos processos

1. Qual o status mais frequente dos processos que estão sendo executados no sistema? Você saberia explicar por quê?

4) Alternando a execução de processos

1. Execute o comando `$ sleep 1000` diretamente do terminal.
2. Pare o processo e mantenha-o em memória.
3. Liste os processos parados.
4. Coloque-o em *background*.
5. Verifique se o comando `sleep 1000` está rodando.
6. É possível cancelar a execução desse comando quando ele está rodando em *background*? Caso seja possível, faça-o.

5) Identificando o RUID e o EUID de um processo

1. Logado como o usuário `aluno`, execute o comando `passwd` no seu terminal. Antes de mudar a senha, abra uma segunda console e autentique-se como `root`. Verifique o `RUID` e o `EUID` associados ao processo `passwd`. Esses valores são iguais ou diferentes? Você saberia explicar por quê? Por fim, cancele a execução do processo `passwd`.

6) Definindo a prioridade de processos

1. Verifique as opções do comando `nice` e em seguida, execute o comando abaixo, verificando sua prioridade, utilizando o comando `ps`:

```
# nice -n -15 sleep 1000 &  
[1] 2289
```

2. Repita o comando do primeiro item, passando para o comando `nice` o parâmetro `-n -5`. Verifique como isso afeta a prioridade do processo. Ela aumentou, diminuiu ou permaneceu a mesma?

7) Editando arquivos crontab para o agendamento de tarefas

Neste exercício, trabalharemos com o comando `crontab`, utilizado para editar os arquivos `cron` do agendador de tarefas do sistema. Esses arquivos serão verificados pelo *daemon* `cron` periodicamente em busca de tarefas para serem executadas pelo sistema. Para entender o funcionamento do `crontab`, o primeiro passo é ler as páginas do manual relevantes.

Para o comando `crontab` em si, consulte a seção 1 do manual:

```
$ man 1 crontab
```

Para o formato de um arquivo de configuração `crontab`, consulte a seção 5:

```
$ man 5 crontab
```

1. Existe alguma entrada de `crontab` para o seu usuário?
2. Que opção deve ser usada para editar o seu arquivo de `crontab`?

8) Agendando uma tarefa no daemon cron

Neste exercício, será necessário enviar mensagens de correio eletrônico. Para isso, você deverá utilizar o comando `mail`; o instrutor pode fornecer as informações básicas sobre ele. Um exemplo do uso desse comando para enviar uma mensagem ao endereço `fulano@dominio` com o assunto *Mensagem de teste* é:

```
$ mail fulano@dominio -s "Mensagem de teste" < /dev/null
```

1. Configure o `crontab` para que uma mensagem de correio eletrônico seja enviada automaticamente pelo sistema, sem interferência do administrador às 20:30 horas.
2. Como verificar se a configuração foi feita corretamente?
3. Qual o requisito fundamental para garantir que a ação programada será executada?
4. Há como confirmar se a mensagem foi efetivamente enviada, sem consultar o destinatário?
5. Dê dois exemplos de utilização desse mecanismo para apoiar atividades do administrador de

sistemas.

6. Faça um script que liste os arquivos sem dono do sistema e envie a lista por e-mail ao usuário root.
7. Agende no crontab do usuário **root** o script do item 6, de modo que ele seja executado de segunda a sexta às 22:30 horas.

9) Listando e removendo arquivos crontab

1. Liste o conteúdo do seu arquivo de **crontab** e, em seguida, remova-o. Quais as opções utilizadas para executar as ações demandadas?

10) Entendendo o comando exec

1. Execute o comando **\$ exec ls -l**. Explique o que aconteceu.