Disciplina: Fundamentos de Linux	Prof. Joilson Alves Junior

Conteúdo: **Conceitos de Linux**

CONTEXTUALIZAÇÃO

- **Nesta aula teremos como objetivo** aprender o seguinte:
  - Comandos básicos: mkdir, touch, rmdir, rm, cp, mv, cat, find, clear, exit, echo, date, password, wc, more, redirecionadores (|, >), vim, man, apropos, help;
  - Compilar softwares de código aberto.

COMANDOS

mkdir (make directory)

- Cria novos diretórios (vazios).

- Sintaxe básica:
  - $ **mkdir** [caminho1/diretório1] [caminho2/diretório2] ...

- Exemplos
  - 1. Para criar os diretórios “Pasta1" e ´”Pasta2" dentro do diretório /tmp, fazemos:

  - $ **mkdir** /tmp/Pasta1 /tmp/Pasta2

COMANDOS

- Naturalmente, se estivéssemos dentro do diretório /tmp, não seria necessário usar o

  - caminho absoluto:
  - $ pwd
  - /tmp
  - $ **mkdir** Pasta1

- Criando vários diretórios recursivamente (um dentro do outro)

  - $ **mkdir** –p Pasta1/Pasta2/Pasta3/Pasta4

Touch

- Pode ser usado para criar novos arquivos vazios.
- Sintaxe básica:
  - **touch** [opçõess] [arquivo1] [arquivo2] ...
- Exemplos
  - 1. Para criar um arquivo vazio chamado "arquivonovo" no diretório atual, poderíamos usar:
    - $ **touch** arquivonovo

COMANDOS

Opções

- **-t** [[YY]YY]MMDDhhmm[.ss] - Altera a data e hora do arquivo. Para o ano, YYYY, pode-se usar os quatro dígitos ou apenas dois), para o mês MM, para o dia DD, para a hora hh, para o minuto mm e para o segundo ss. Lembrando que as opções de ano e segundo são opcionais (por isso foram colocadas entre colchetes).

- **Exemplos:**
  - 1. Para alterar a data do arquivo “arquivonovo" para o dia 16/11 (16 de novembro), e o horário para 16h11min, usamos:
  - $ **touch** –t 11161611 arquivonovo

- **2. Suponhamos que quiséssemos alterar os segundos também (para 11, por exemplo):**

  - $ **touch** -t 11161611.11 arquivonovo

- **3. Por fim, se quiséssemos que a data do arquivo "arquivonovo" fosse 01/01/2013, com horário 0h0min, rodaríamos o comando da seguinte forma:**

  - $ **touch** -t 201301010000 arquivonovo

Obs: Nas VMs a alteração da hora não vai funcionar, pois por padrão,
os relógios da VM são sincronizados com o relógio HOST em execução e não podem ser alterados independentemente.
A alteração da data vai funcionar normalmente.

rmdir (remove directory)

- **Remove um ou mais diretórios vazios.**
  - Sintaxe básica:
  - $ **rmdir** [caminho1/diretório1] [caminho2/diretório2] ...
  - Exemplos:
    - 1. Para remover os diretórios “Pasta1" e “Pasta2“ em /tmp, poderíamos usar:
    - $ **rmdir** /tmp/Pasta1 /tmp/Pasta2

COMANDOS

rm (remove)

- Remove **arquivos** e **diretórios**.

- **Sintaxe básica:**

  - $ **rm** [opções] [arquivo1] [arquivo2] ...

- **Removendo arquivo :**
  - Criamos um arquivo chamado "teste" no diretório /tmp:
    - $ **touch** /tmp/teste

  - Agora vamos removê-lo:
    - $ **rm** /tmp/teste

COMANDOS

rm (remove)

- **Removendo diretório**
- Opções:
  - **-r:** Opção usada para **remover recursivamente** diretórios e seu conteúdo. Pode ser usada também para remover diretórios vazios.

- Vamos criar um diretório (vazio) chamado "Pasta".
  - $ **mkdir** Pasta

- Se usarmos o seguinte comando para removê-lo, veremos um erro e o diretório não sería removido:

  - $ **rm** Pasta
    - ERRO!

- Para removê-lo, teríamos que fazer:
  - $ **rm -r** Pasta

COMANDOS

cp (copy)

- Este comando serve para copiar arquivos.

- Sintaxe básica:
  - $ **cp** [opçõess] [origem] [destino]

- Exemplos
- 1. Para copiar o arquivo "teste" do /tmp para o diretório home do usuário:
  - $ **cp** /tmp/teste ~

- Opções
- -R : Copia recursivamente os subdiretórios e seu conteúdo.

cp (copy)

- **Exemplo:**

- Suponha que um usuário possui um diretório no /tmp (/tmp/diretorio) e quer copiá-lo para sua home. Suponha ainda que esse diretório a ser copiado não está vazio.

  – $ **cp -R** /tmp/diretorio ~

mv (move)

Move e renomeia arquivos e diretórios.

Sintaxe básica

$ **mv** [opçõess] [origem] [destino]

Exemplos

1. Suponha que um usuário possui um arquivo em sua home chamado arquivo1. Para renomear este arquivo para arquivonovo, supondo que o usuário está em sua home, bastaria usar:

$ **mv** arquivo1 arquivonovo

2. Suponhamos agora que queremos mover o "arquivonovo" para o diretório /tmp (supondo ainda que o usuário está em sua home):

$ **mv** arquivonovo /tmp/

Após a execução desse comando, "arquivonovo" estará no diretório /tmp e não haverá mais uma cópia dele no diretório home do usuário.

Opções

- R : Como outros comandos, essa opção move diretórios e seu conteúdo recursivamente.

cat (concatenate)

Concatena arquivos e imprime o resultado no terminal.

Sintaxe básica:

#Imprime
$ **cat** [arquivo1]

#Imprime e concatena
$ **cat** [arquivo1] [arquivo2] ...

COMANDOS

Exemplos:

curso@curso-desktop:~$ **cat** /etc/resolv.cof
nameserver 192.168.1.10

curso@curso-desktop:~$ **cat** /etc/hostname
Debian

COMANDOS

Exemplos:

curso@curso-desktop:~$ **cat** /etc/resolv.cof /etc/hostname
nameserver 192.168.1.10
Debian

find

O comando find é usado para procurar por diretórios e arquivos no disco. Possui várias opções, mas será ilustrado apenas alguns exemplos simples.

Exemplos

1. Este exemplo procura por um arquivo ou diretório com o nome "Documents" a partir do / :

```
$ find / -name network
```

2. Este outro procura por um arquivo ou diretório com o nome "Music" a partir do diretório home do usuário:

```
$ find ~ -name Documentos
```

COMANDOS

Clear

Limpa o terminal

Exemplo:

$ **clear**

Exit

Este comando serve para sair do shell (interpretador) e para efetuar o log out do usuário no terminal.

Exemplo:

$ **exit**

.

COMANDOS

echo

Mostra um texto na tela.

Exemplo:

$ **echo** mensagem

Pode parecer um comando pouco útil, mas é bastante usado sobretudo **em scripts** para exibir mensagens ao usuário.

- Fazer um exemplo de um script.

date

O comando date imprime ou modifica a data e o horário do sistema. É importante salientar que somente o usuário root e usuários privilegiados podem rodar este comando.

Sintaxe básica:

$ **date** [data]

Exemplos: para visualizar a data e a hora do sistema:

$ **date**

Mon Mar 8 14:45:21 BRT 2019

COMANDOS

- Para alterar a data e a hora do sistema, basta usar o comando da seguinte maneira:

  – $ **date** MMDDhhmm[[YYyy][.ss]]

- Onde MM é o mês, DD é o dia, hh é a hora, mm são os minutos. Opcionalmente, podem ser usados o ano (com 2 ou 4 dígitos) e os segundos (ss).

- Para alterar a data do sistema para o dia 1 de fevereiro e o horário para 14:30, poderíamos fazer:

  – $ **date** 02011430

passwd (password)

Este comando é usado para atualizar a senha dos usuários.

Sintaxe básica

$ **passwd <usuário>**

WC

O comando **wc** é usado para contar linhas, palavras e bytes de um arquivo ou do que for escrito no terminal.

Sintaxe básica

$ wc [opções] [arquivo]

- Opções
- -c: Imprimir a contagem de bytes.
- -l: Imprimir o número de linhas.
- -w: Imprimir o número de palavras.

Exemplos

Vamos usar, para estes exemplos, o conteúdo dos arquivos /etc/passwd

1. Para exibir o número de linhas do arquivo "arquivo1":
$ **wc -l** /etc/passwd

2. Para exibir o número de palavras e de bytes do arquivo:
$ **wc -wc /**etc/passwd

3. Se usássemos o comando **wc** sem nenhuma opção":
$ **wc /**etc/passwd
36 63 2109 **/**etc/passwd

Onde o primeiro número é a contagem de linhas, o segundo, de palavras, e o terceiro, o de bytes.

Consolidação do Conhecimento

Faça os exercícios junto com um relatório para entregar para o professor

1. Vá até seu diretório home e crie um diretório chamado “teste". Use o comando **ls** para ver que o diretório foi criado. Remova o diretório criado e use novamente o comando **ls** para ver que a pasta foi removida.

2. Qual é a diferença entre os resultados produzidos pelos dois comandos a seguir?

$ touch "arquivo novo"

e

$ touch arquivo novo

3. Crie um arquivo chamado "teste" em seu diretório home, usando o comando touch. Use um comando que você já aprendeu para ver a data do novo arquivo criado. Mude a data do arquivo para a data do seu nascimento e verifique a nova data do arquivo.

Consolidação do Conhecimento

4. Crie um diretório chamado "pastateste" dentro do diretório /tmp. Dentro de "pastateste", isto é, no /tmp/Pastateste, crie outa pasta chamada aula. Dentro de aula crie outro diretório chamado linux. Dentro de linux, crie um arquivo chamado "arquivoteste".

a) Remova somente o arquivo "arquivoteste"
b) Remova somente a pasta linux
c) Remova as pastas que sobraram (pastateste e aula) de uma só vez

Atenção: O comando rm é definitivo, ou seja, uma vez que o usuário removeu um arquivo (ou um diretório), este não poderá ser recuperado. Não funciona simplesmente como uma lixeira, mas sim remove definitivamente.

Consolidação do Conhecimento

5. Na pasta atual crie um arquivo chamado teste, copie para a pasta /tmp e depois apague o arquivo.

6. Na pasta atual crie um arquivo chamado Windows, renomeie para Linux , mova para /tmp e apague o arquivo

7. Entre em /etc e veja o conteúdo do arquivo chamado “resolv.conf” com o comando cat.

8. Procure a partir do / um arquivo chamado debian_version. Vá até a pasta onde ele está e veja a versão do seu debian.

9. Limpe a tela, saia do terminal (do root e do usuário comum). Volte para o terminal como root e imprima a palavra VOLTEI no terminal Shell.

10. Altere a data do sistema 01/01/2015 e volte para data atual.

Editor de texto

Vim:

- O Vi é o editor básico do GNU/Linux, está disponível em grande parte das distribuições do GNU/Linux.
- Hoje em dia as distribuições usam uma versão mais completa e com mais recursos do que o Vi, que é o Vim (VI iMproved). Ao invocar o vim, este entra direto para o modo "visualização" onde visualizaremos o arquivo.

- **Exemplo: $ vim testeaula**

- Usam-se os modos texto para a inserção de dados e modo comando para a localização, movimentação, alteração e para salvar e sair do texto.
- Para voltar ao modo de visualização, sempre se usa a tecla ESC.

COMANDOS

Principais comando do VIM (com isso você já se vira)

- Dentro do Vim, pressionar a tecla **i** para passar do Modo de Comandos para o Modo de Inserção;
- Para voltar ao modo de comandos com a tecla ESC;
- Para salvar o conteúdo do arquivo e sair, pressione :wq, pressionando enter logo em seguida.
  - O comando (:wq) é utilizado para salvar (write) o arquivo e sair (quit) do Vim.
- Para sair sem salvar :q!
- Para deletar uma linha, volte ao modo comando (pressione esc), em seguida posicione na linha e depois pressione dd

COMANDOS

- Para deletar um caracter indesejado, posiciona sobre ele e pressione x
- Para copiar uma linha, volte ao modo comando, navegue até a linha que você quer copiar, e pressione yy
- Para colocar a linha, volta ao modo comando, navegue até a linha que você quer colar e pressione p.
- Para selecionar uma parte de uma linha, volte ao modo comando e:
  - Navegue até o conteúdo que você quer copiar;
  - Use a tecla v para selecionar (clique v uma vez e depois use as setas pata selecionar) o conteúdo desejado -Use a tecla y para copiar o conteúdo;
  - Navegue até o local que você precise colar o conteúdo e use a tecla p .

More

- Utilizado para ler arquivos de texto ;
- Use o comando more seguido do caminho e nome do arquivo,
  - **Exemplo: more /var/log/messages**
- Todo conteúdo do arquivo será exibido no terminal, preenchendo a tela com texto.
- Para prosseguir com a leitura, pressione a barra de espaço e, caso precise voltar uma ou mais páginas, use a tecla "b". Se quiser sair antes do fim do arquivo, pressione "q".

COMANDOS

Pipe e redirecionamento

| (Pipe): o pipe (|) é usado para fazer encadeamento de processos, ou seja, faz com que a saída de um comando seja enviada como entrada para o próximo comando.

Exemplo 1:

$ cat /etc/passwd | wc -l

Exemplo 2:

$ ls -1 | wc -l → O "-1" lista os arquivos por linha

Exemplo 3:

$ Dentro de /etc, execute o seguinte comando: ls |grep "localtime"

obs: o grep procura padrões

Pipe e redirecionamento

>

Esta é uma outra forma de direcionar a saída de um comando: diferente do |, que direcionava a saída de um comando para um outro programa ou comando, o > direciona a saída de um comando para um arquivo ou dispositivo.

Exemplo 1 (crie o arquivo1 com algumas linhas) :

$ cat arquivo1 > arquivo2

O >>, assim como o >, também direciona a saída de um comando para um arquivo, a diferença é que ele não substitui o conteúdo do arquivo, mas acrescenta ao final.

Exemplo 2 (crie o arquivo 3 com algo dentro):

$ cat arquivo3 >> arquivo2

Consolidação do Conhecimento

Faça os exercícios junto com um relatório para entregar para o professor

1. Usando o vim, crie o arquivo Linux_teste1 e insira cinco nomes de animais nele (um em cada linha). Saia do arquivo com o comando **wq** e verifique se o conteúdo permanece no arquivo.

2. Usando o vim, crie o arquivo Linux_teste2 e insira cinco nomes de frutas nele (um em cada linha). Saia do arquivo com o comando **wq** e verifique se o conteúdo permanece no arquivo.

3. Crie o arquivo Linux_teste3 e insira cinco nomes de carros nele (um em cada linha). Saia do arquivo com o comando **q!** e verifique se o conteúdo permanece no arquivo.

4. Acrescente o conteúdo do arquivo Linux_teste1 ao conteúdo do arquivo Linux_teste2 utilizando um redirecionador (sem substituir o conteúdo do arquivo Linux_teste2).

5. Substitua o conteúdo do arquivo Linux_teste1 com o conteúdo do arquivo Linux_teste2 utilizando um redirecionador.

Consolidação do Conhecimento

6. Conte o número de linhas do arquivo do arquivo Linux_teste1 utilizando um redirecionador.

7. Com o vim, copie a segunda linha do arquivo Linux_teste1 e cole no lugar da quarta linha.

8. Com o vim, copie a terceira e quarta letras da segunda linha do arquivo Linux_teste1 e cole no lugar da segunda letra da segunda linha, após isso, delete a terceira letra da terceira linha.

9. Pesquise na internet 5 comandos avançados com o vim e utilize (faça exemplos no editor, depois compartilhe com a turma toda).

10. Faça a leitura do arquivo /var/log/messages os comando **cat**, **|**, e **more.**

Obtendo Ajuda

- O que foi apresentado até este momento tem caráter introdutório: com a opção de "ajuda" é possível se aprofundar eachar respostas para problemas não tratados.

man (manual)

- O comando **man** mostra uma página de manual para um determinado comando.
- Sintaxe básica : $ **man** [comando]
- Para visualizar o manual do comando ls, basta usar
  - $ **man ls**
- Para sair de uma pagina de manual, basta digitar q.

COMANDOS

apropos

- Este comando faz buscas de palavras em um banco de dados que contém descrições curtas de comandos e programas.
- Sintaxe básica
  - $ apropos [busca]
- Suponhamos que quiséssemos procurar como remover arquivos. Poderíamos usar
  - $ apropos remove.

--help

- Quase todos os comandos do GNU/Linux possuem a opções help para obter ajuda sobre o comando em questão.
- **Sintaxe básica**
- $ [comando] --help
- Para obtermos ajuda sobre o wc, por exemplo, usamos:
  - **$ wc --help**

Consolidação do Conhecimento

Faça os exercícios junto com um relatório para entregar para o professor

1. Procure sobre a descrição “remove files” (com o “”)

2. Verifique como utilizar o(s) comando(s) que retornaram no exercício anterior com o **man** e **–help**

3. Veja o manual de 5 comandos a sua escolha.

Instalar programas a partir do código fonte

- O código de um programa, cujo desenvolvedor queira distribuir, deve ser armazenado em uma árvore de diretórios hierárquica;

- Essa árvore inclui:
  - O código fonte na linguagem C ;
  - Um Makefile (definir regras de compilação do software);
  - Documentação.

- Para que o código possa ser distribuído, toda arvore de diretórios precisa ser **encapsulada** de forma que seja facilmente armazenada e distribuída.

  → Colocar toda a árvore em um único arquivo.

Instalação de programas

- O encapsulamento geralmente é feito da seguinte forma:
  - Gerar um único arquivo "tarfile" com o utilitário **tar;**
  - Comprimir o arquivo com outro utilitário (gzip, bzip2, etc).
  - O arquivo comprimido resultante é conhecido como ***tarball;***
- Um tarball geralmente é identificado por meio das duas extensões (tar e gz ou bz2);
- Por isso softwares distribuídos como códigos fontes geralmente possuem o seguinte formato:
  - Nome_do_software.tar.gz
  - Nome_do_software.tar.bz2

Abrindo um tarball

- O conteúdo é obtido por meio de duas etapas:
  - Descomprimir o arquivo utilizando o **gzip** (ou outro utilitário adequado);
  - Extrair o arquivo com o **tar**.
- Exemplo:
  - **gzip -d** tarball.tar.gz
  - **tar xvf** tarball.tar
- -d significa modo de descompressão;
- O gunzip também poderia ser utilizado no lugar do gzip –d .

Abrindo um tarball

- É possível enviar a saída do **gzip** direto para o **tar**, utilizando o redirecionador **| (pipe);**
  - Ex**: gzip -d** tarball.tar.gz | tar xv
- Pode-se utilizar também o recurso de descompressão do tar:
  - Ex: **tar zxvf** tarball.tar.gz
- Arquivos bzip2 pode ser aberto assim:
  - **bzip2 -d** tarball.tar.bz2 | tar xv
  - **tar jxvf** tarball.tar.bz2

Compilando um software de código aberto

- Após o código fonte ser extraído, ele está pronto para ser compilado;
- Para compilar, é necessário que o sistema tenha ferramentas disponíveis, tais como o **gcc** e **make**.

Configure:

- Os pacotes de código fonte geralmente incluem um arquivo configure no topo da árvore;
- O configure, ao ser executado, examina o sistema para checar a existência de um compilador, bibliotecas, utilitário, e outros itens necessários para a compilação com sucesso.

Instalação de programas

- O configure utiliza as informações que possui para produzir um arquivo “makefile” personalizado para o software ser instalado no sistema específico;
- Se o configure perceber que algo está faltando, ele vai gerar um erro;
- Se tudo estiver correto, a compilação pode começar.

Make

- Executa tarefas definidas no makefile para compilar o programa a partir de seu código fonte;
  - Compilação é o ato / processo de traduzir um programa feito em uma linguagem de alto nível para uma linguagem de máquina, para que suas instruções sejam executadas pelo processador, ou seja, cria o executável de um programa escrito em uma linguagem de alto nível.

Make install

- O comando make install irá copiar o programa compilado, e suas bibliotecas e documentação para os diretórios corretos.
  - Ex: faz a cópia dos binários para o /bin

Instalação de programas

Informação adicional:

- Em distribuições derivadas do Debian, a principal dependência que pode ser encontrada para o processo de compilação é o **pacote de compiladores** que contém **todas as ferramentas básicas para compilação de um programa**;

- O nome do pacote é **build-essential**;

- Caso tenha algum problema de compilação, ele deve ser instalado primeiro.

  – Instalando: **apt-get install** build-essential

Instalação de programas

Exemplo de compilação

1. Download do arquivo:
2. wget https://www.nano-editor.org/dist/v2.8/nano-2.8.7.tar.gz
   Instalando dependências: **apt-get install** libncurses5.dev libncursesw5.dev

2. **tar –xzvf** nano-2.8.7.tar.gz

3. **cd** nano-2.8.7 (entre no diretório descompactado)
   1. **./configure**
   2. **make**
   3. **make install**

Instalação de programas

Exemplo de desinstalação:

1. Entrar no diretório que a instalação foi feita;

2. make uninstall

3. make clean

Consolidação do Conhecimento

Faça os exercícios junto com um relatório para entregar para o professor

1. Encontre o arquivo hosts, crie uma pasta com o nome da primeira palavra que está escrita dentro desse arquivo na partição onde ficam os dados responsáveis pela inicialização do sistema. Na sequência, renomeie a pasta criada para o seu primeiro nome e copie ela para o diretório home do seu usuário. Limpe a tela e vá para o diretório onde está a pasta com seu nome. Dentro da pasta com seu nome, crie um arquivo chamado aula e outro chamado aula_1. Dentro do primeiro arquivo insira a frase “Curso de Especialização em”, no segundo insira a frase “Desenvolvimento WEB”. Utilizando um redirecionador, insira o conteúdo do arquivo aula dentro do aula_1. Após isso, utilizando outro redirecionador, conte quantas linhas possui a documentação do comando “ls”. Crie, recursivamente, dentro do seu diretório atual, pastas com o nome de cada dígito retornado pela contagem das linhas. Dentro da última pasta criada, procure um software em https://sourceforge.net/, ou outro local de sua escolha, faça o download, a compilação e a instalação. Por fim, veja todas as partições do sistema e a data atual, mude a senha do seu usuário e do usuário root e saia do terminal.

Perguntas ?