

Laboratório de Sistemas Operacionais - Aula Prática 03 – 08.03.2016
Sistema Operacional Linux

Trabalhando com a Estrutura de Arquivos - II

- 1) Como todo Sistema Operacional, o Linux também trabalha com um Sistema de Arquivos para armazenar e organizar todos os arquivos que estão no Sistema. Para isso, ele utiliza uma estrutura de arquivos baseado em pastas e subpastas tornando a manipulação mais fácil para o Usuário. Lembre-se que, a qualquer momento você pode utilizar o comando ***pwd*** para saber em que pasta você está no momento e o comando ***tree*** para listar a estrutura das suas pastas. Esses dois comandos você já estudou na aula 2 passada. Para iniciar, vamos criar o diretório (pasta) ***Aula3***. Digite o comando: ***mkdir Aula3***.
- 2) Verifique se o diretório foi criado corretamente. Para isso, use o comando que mostra o conteúdo de diretórios: ***ls -l***.
- 3) Dê um comando ***pwd*** e note que você está na sua pasta raiz, ou seja, ***/home/bsinnnnnn*** ou ***/home/ecnnnnnn***, onde ***nnnnnn*** é o seu número de matrícula. Este comando mostra o caminho onde você está no Linux. Agora, caminhe para o diretório ***Aula3***. Como você faz isso? Use o comando: ***cd Aula3***. Em seguida, repita o comando ***pwd*** e veja que agora você está em ***/home/bsinnnnnn/Aula3*** ou ***/home/ecnnnnnn/Aula3***.
- 4) Neste ponto, vamos criar o arquivo ***arq1.txt*** dentro de ***Aula3*** com o seguinte conteúdo:

Pedro	27anos	graduado
Carlos	22anos	segundograu
Jorge	26anos	posgraduado
Marcos	24anos	mestrando

Julia 30anos doutora

Obs.: Note que nós desejamos apenas **3 colunas**, então quando digitar o número de anos mais a palavra anos, não dê espaço.

Para facilitar a criação de colunas, por exemplo, para montar a primeira linha, ao final da palavra **Pedro** dê um **tab**, ao final da palavra **27anos** dê um **tab**, e ao final da palavra **graduado** dê um **enter**.

Vamos criar o arquivo? Use o comando **cat > arq1.txt** e digite um nome por linha. Não se esqueça de dar **<enter>** no último nome, senão ele será perdido. No final, fechar o arquivo com **ctrl + c**.

- 5) Você já sabe como olhar o conteúdo de um arquivo? Para fazer isso, basta usar o comando **cat arq1.txt** sem o sinal de **>**. Agora, queremos classificar o arquivo pela coluna 1 (nome). Use o comando **sort arq1.txt**. Sem opção, o **sort** classifica pela coluna 1 (nome).

Veja que o Linux leu seu arquivo, classificou pela coluna 1 (nome) e depois mostrou na tela. O arquivo original não foi afetado.

- 6) Agora queremos classificar o arquivo pela coluna 3 (formação). Você pode fazer de duas formas:

(1) Usando o comando **sort -b +2 arq1.txt**. A opção **+0** classifica pela col. 1; a opção **+1** pela col. 2 e a opção **+ 2** pela col. 3;

(2) Pode usar também o comando **sort -k +3 arq1.txt**.

Veja a diferença no help on line. Como faz isso? **man sort**

- 7) Vamos criar um segundo arquivo: **arq2.txt** com o seguinte conteúdo:

Gato	12anos	angorá
Peixe	07anos	namorado
Cachorro	15anos	dálmata
Pássaro	05anos	sabiá

Criação de arquivo com conteúdo você já sabe. Você usa o comando: **cat > arq2.txt**. No final, fechar com **ctrl + c**. Veja se o conteúdo do seu arquivo está correto. Como fazer isso? Use o comando **cat arq2.txt**.

- 8) Agora você deseja exibir, de forma concatenada, os arquivos **arq1.txt** e **arq2.txt**, nesta ordem. Para tanto, use o comando **cat arq1.txt arq2.txt**.

Entenda o que aconteceu: O Linux primeiro pegou seu arquivo **arq1.txt** e depois pegou seu arquivo **arq2.txt**, e como não havia nenhuma outra opção ele exibiu o conteúdo de ambos os arquivos na sua tela.

Para entender um caso prático, imagine que você deseja ver os arquivos de faturamento de sua empresa, primeiro janeiro e depois fevereiro. Esta é uma boa hora de usar este comando.

- 9) O desafio agora é mostrar na tela os dois arquivos **arq1.txt** e **arq2.txt**, ordenado por tipo (3ª coluna), simultaneamente. Para tanto, use o comando **cat arq1.txt arq2.txt | sort -k +3**.

Veja que neste comando apareceu um novo componente (**|**). Seu nome é **pipe**, e ele serve para transportar o resultado de um comando para o comando seguinte, neste caso, o resultado da concatenação de **arq1.txt** e **arq2.txt**, ao invés de ser enviado para o vídeo, agora será enviado para o próximo comando, no caso um **sort**. Quando o arquivo for classificado pelo **sort**, como não há mais nenhum **pipe** à frente, o resultado será exibido em vídeo.

- 10) Muito bem. Agora gostaríamos de guardar a concatenação dos arquivos e a classificação dentro de um novo arquivo. Este novo arquivo deverá ser **arq3.txt**. Então, use o comando abaixo para fazer isso:

`cat arq1.txt arq2.txt | sort -k +3 > arq3.txt.`

Novamente temos um símbolo novo. Não é tão novo porque você já o usava no comando **`cat`**. Este símbolo é o sinal de **`>`**. Sua função é redirecionar a saída do último comando para um determinado arquivo. Você nunca redireciona saída para arquivo usando **`pipe`**.

- 11) Para ter certeza de que as coisas correram bem, verifique se o arquivo **`arq3.txt`** foi criado em seu diretório **`Aula3`**, e mostre seu conteúdo. Faça isso usando os comando: **`ls -l ; cat arq3.txt.`**

O ponto e vírgula separam comandos dados na mesma linha. Quando você usa ponto e vírgula o Linux entende que um comando nada tem a ver com o comando seguinte. Sua função é bem diferente do pipe (**`|`**).

- 12) Bem, você esteve o tempo todo trabalhando no diretório **`Aula3`**. Agora, desejamos que você volte para o seu diretório raiz, ou seja, **`ecnnnnnn`** ou **`bsinnnnnn`**. Para voltar um passo para traz, ou seja, para voltar para o diretório pai use o comando **`cd ..`**

- 13) Você gostaria de ver toda a estrutura que já existe no seu diretório raiz, e a melhor forma para ver isso é usando a estrutura em forma de árvore. Com o comando **`tree`** para mostrar o conteúdo de todo seu diretório raiz, ou **`tree Aula3`**, para mostrar somente o diretório **`Aula3`**;

- 14) Imagine que você está num cliente seu e que vários arquivos estão no diretório **`Aula3`**, e que você gostaria de levar esta estrutura para analisar melhor em sua empresa. Você poderia fazer isso criando um novo arquivo **`arq4.txt`** com resultado do comando **`tree Aula3`**.

Digite **`tree Aula3 > arq4.txt.`**

15) Mostre o conteúdo do diretório **Aula3**, ou seja, tudo que nele tem. Use o comando **ls -l Aula3**.

16) Neste ponto, você já tem um arquivo chamado de **arq4.txt** que tem a estrutura em árvore do diretório **Aula3**, mas você também gostaria de acrescentar o conteúdo do mesmo diretório, só que agora usando o **ls**. Você poderia usar o comando abaixo:

ls -l Aula3 >> arq4.txt.

Outro símbolo novo (**>>**). Quando este símbolo for usado, ele não destrói o conteúdo do arquivo que está logo à frente dele, mas acrescenta o resultado do comando anterior.

17) Veja como ficou o conteúdo do arquivo **arq4.txt**. Para mostrar seu conteúdo, use o comando **cat arq4.txt**.

18) Vamos aprender a renomear arquivos. Inicialmente, gostaríamos de renomear o **arq4.txt** e chamá-lo de **arq4ren.txt**. Para fazer isso, use o comando **mv arq4.txt arq4ren.txt**.

O nome do comando **mv** significa movimentação, ou seja, recortar e colar. Se a segunda parte do comando fosse outra pasta, o comando recortaria o arquivo da pasta de origem e o moveria fisicamente para a pasta destino.

Agora, se a movimentação for para ser feito dentro da mesma pasta, com o nome de destino diferente o comando **mv** renomeia o arquivo.

19) Agora, desejamos mover fisicamente o arquivo **arq4ren.txt** para o diretório **Aula3**, ou seja, recortar da sua pasta raiz e colar na pasta **Aula3**. Faça isso usando o comando **mv arq4ren.txt Aula3**.

20) Verifique se o arquivo ainda está na sua pasta raiz. Use o comando **ls -l**. Se o comando do item 19 processou corretamente, você não deverá ver o **arq4ren.txt** na sua pasta raiz. Agora veja se ele está na sua pasta **Aula3**. Faça isso usando o comando **ls -l Aula3**.

21) Vamos lembrar alguns comandos já estudados? Procurar quais arquivos tem a palavra **dálmata** em todos os arquivos de todos os diretórios. Ao achar, classificar a busca e o resultado da procura gravar no arquivo **arq5.txt**. Use o comando **grep -iR dálmata * | sort > arq5.txt**.

A letra **R** da opção acima significa recursividade, ou seja, procura no diretório em que está e em todos os subdiretórios.

22) Mostrar apenas as duas ultimas linhas do **arq3.txt**. Use o comando **tail -2 Aula3/arq3.txt**. Vá até o manual (**man tail**) e conheça um pouco mais do comando **tail**.

Note que nós estamos no seu diretório raiz e de lá estamos olhando o conteúdo do diretório **Aula3**. O fato de especificar o diretório no comando mostra que estamos usando o critério de **caminho relativo**.

23) Fechar o sistema/Encerrar a sessão.

Comando: **Exit** ou **logout**;

"Às vezes, quando você inova, comete erros. É melhor admiti-los rapidamente e continuar a melhorar suas outras inovações."

Steve Jobs