

Fundamentos de Software

Lista de Exercícios 4: comandos básicos em ambiente Linux

ORIENTAÇÃO GERAL: os comandos Linux podem ser realizados em distribuição Linux efetivamente instalada na máquina, **WSL com Ubuntu ou outra distribuição**, ambientes web em JS: <http://bellard.org/jslinux/>¹ online, máquinas virtuais, utilitários como o Cygwin: <https://www.cygwin.com/> para Windows e o WSL instalado via Powershell (wsl –install), Windows Subsystem for Linux que é acessível e gratuito via Microsoft Store com distribuição Ubuntu.

Após estar “logado” com o seu usuário, verifique qual a pasta (diretório atual), digite o comando **pwd**. (anote a saída da tela)

1. Verifique a arquitetura do sistema, versão do Linux etc.
uname -a

2. Usando o comando **mkdir** crie a seguinte estrutura de diretórios dentro de sua pasta /home/<username>
 ./ufrn
 ./trabalho
 ./ufrn/textos
3. Retorne à sua pasta home, com o comando **cd ~** _____
4. Usando o comando **cd**, entre na pasta **ufrn** _____
OBS: o shell Linux permite navegar na hierarquia de pastas com os símbolos .. (pontoponto, exemplo: cd ..) retorna ao nível imediatamente anterior. Já cd – (traço)), retorna à pasta antes do último comando cd.
5. Crie um arquivo chamado **numeros.txt** com o comando **cat**, contendo os seguintes números separados por espaço: 10 100 50 25 1 2
cat > numeros.txt
10 100 50 25 1 2
Ao concluir, combine CTRL + D para salvar o conteúdo e voltar ao prompt.
6. Digite **ls -l** para exibir o conteúdo da pasta com todos os detalhes (permissões de acesso, data, hora de criação, tamanho)
Anote a saída da tela:
7. Verifique o conteúdo do arquivo com o comando **cat numeros.txt**
8. Duplicar o arquivo numeros.txt para numeros1.txt
cat numeros.txt > numeros1.txt
9. Abrir o editor de textos **vi** (ou **nano**) para edição do arquivo numeros.txt e incluir na segunda linha os números 50 100 200 300 400

¹ Caso não tenha instalado Linux em disco (hd, ssd) no computador, ou não utilize máquina virtual. Outra opção no Windows é instalar o Cygwin: <https://cygwin.com/install.html> ou wsl com Ubuntu

OBS: o vi possui dois modos: visualizar e edição. Ele entra no modo visualizar e para mudar para o modo edição, a tecla é INS (insert). Vá para a segunda linha, digite os números separados por espaço. No fim, aperte ESC para voltar ao modo visualizar. Digite então **:wq!**

Este comando irá escrever (w, write) o arquivo em disco, e sair (q, quit).

Você pode também editar normalmente o código no VSCODE no Windows e acessar via WSL a pasta no Windows com o comando `cd /mnt/c`

10. Verifique com o comando **cat numeros.txt** se foi salvo corretamente.

11. Adicione ao fim do arquivo numeros.txt o conteúdo de numeros1.txt

cat numeros1.txt >> numeros.txt

12. Copie o arquivo numeros.txt para sua pasta HOME

cp numeros.txt ~

OU

cp numeros.txt /home/<username>

13. Acesse sua pasta HOME e verifique se o arquivo foi copiado corretamente

14. Volte à pasta que estava anteriormente (ufrn). Para isto basta digitar **cd -**

15. Mova o arquivo numeros1.txt para a pasta /home/<username>/trabalhos

mv numeros1.txt ../trabalhos

16. Acesse a pasta trabalhos e verifique se foi movido corretamente. Lembre copiar mantém a origem, mover exclui da origem, equivalente ao RECORTAR.

17. Na pasta trabalhos verifique as permissões para o arquivo numeros1.txt (**ls -l**)

OBS: no sistema de arquivos Linux, as permissões constituem internamente um conjunto de 9 bits, separados por três grupos: USUÁRIO/DONO, GRUPO, RESTO DO MUNDO, ou seja, para cada grupo, 3 bits controlam as permissões READ-WRITE-EXECUTION (r w x). Em termos numéricos, sistema DECIMAL, teríamos para um caso típico de arquivo texto, por exemplo, a permissão 644:

Dono/Criador			Grupo			Resto do Mundo		
r	w	x	r	w	x	r	w	x
1	1	-	1	-	-	1	-	-
6			4			4		

Suponha que eu desejo dar permissão de escrita ao grupo onde o usuário criador está inserido, a permissão passaria à 664, e pode-se alterar com o comando **chmod 664 numeros1.txt** _____

Arquivos que podem ser executados, como arquivos web e códigos compilados precisam ter permissão para execução (x). Por exemplo, para o usuário, grupo e o resto do mundo poderem executar o arquivo, teríamos a permissão 755, e assim por diante. OBSERVE que no CONSOLE, diretórios ficam por padrão em AZUL e arquivos executáveis em VERDE.

18. Vamos criar um **hello.c** no **vi** e compilar vi linha de comando com o **gcc**.

19. Com o arquivo hello.c criado, execute **gcc hello.c -o hello**

20. Isto irá criar o arquivo hello com permissão de execução. Confira e anote as permissões _____

21. Para executar basta digitar **./hello**

22. Renomear o arquivo numeros1.txt para sequencia.txt

mv numeros1.txt sequencia.txt

23. Copiar todos os arquivos para a pasta HOME

cp *.* ~ OU cp * ~

24. Cria na HOME uma pasta **ufrnbackup**

25. Estando na HOME, copie todo o conteúdo da pasta **ufrn** para **ufrnbackup**

cp ufrn/* -r ufrnbackup

OBS: o uso da opção -r no cp, informa que deve ser copiado os arquivos e todos os subdiretórios (-r, recursive)

26. Copie apenas os arquivos .txt para a pasta ufrn/textos e para a pasta ufrnbackup/textos

cp *.txt ufrn/textos

cp *.txt ufrnbackup/textos

27. Deletar arquivos com a extensão .txt da pasta HOME

rm *.txt

28. Apagar o diretório texto dentro da pasta ufrn

rm -r ufrn/textos

29. Ver estrutura de pastas padrão de uma instalação Linux na raiz /, com **ls -l /** e guardar o conteúdo no arquivo listaraiz.txt.

30. Para ver e guardar a lista de comandos disponíveis no console, **digite ls -l /bin**
> listabin.txt

31. Na HOME listar os processos em execução (**ps -a**) e guardar no arquivo listaprocessos.txt. Pode-se também listar os comandos por ordem decrescente de uso da memória com o comando **top**.

32. É muito comum procurar a ocorrência de um texto dentro do resultado de um comando ou arquivo, com o comando **grep**. Por exemplo, vamos procurar o texto **dhcpd** dentro do comando **ps**. Para isto concatenamos os dois comandos com um TUBO (pipe, |). Ou seja, o comando fica

ps -a | grep dhcpd

33. Procurar o número 300 no arquivo numeros.txt: **cat numeros.txt | grep 300**

34. A ideia de processos em background (plano de fundo) e foreground (frente) é muito interessante. Crie um arquivo loop.c com o vi, que simplesmente tenha o seguinte:

```
#include <stdio.h>
int main() {
    while(1) { };
    return 0;
}
```

Compile com o gcc e gere o arquivo loop.o. Execute-o em background com

./loop &

Digite o comando **jobs** para verificar os processos em execução e anote abaixo o resultado:

Suponha que queremos finalizar este processo, bastaria identificar o PID (primeira coluna) e digitar **kill -9 <pid>** ou **kill -kill <pid>**

Se ao invés de matar o processo, quiséssemos trazê-lo para o primeiro plano, bastaria digitar fg %<numero do job>.

35. Na HOME, crie o arquivo learquivo.c com o conteúdo abaixo, compile, execute e analise passo a passo este código, muito importante:

```
int main() {
    FILE *fp;
    char linha[100];
    int vetor[17];

    int L = 1, i = 0, j = 6;
    char *leitura;
    fp = fopen("numeros.txt", "r");
    while (!feof(fp)) {
        leitura = fgets(linha, 100, fp);
        if (leitura) {
            printf("Linha %d: %s", L, linha);
            char *token = strtok(linha, " ");
            while (token != NULL && i < j) {
                vetor[i++] = atoi(token);
                token = strtok(NULL, " ");
            }
            j += i;
        }
        L++;
    }
    i = 0;
    for (; i < 17; i++) printf("%d\t", vetor[i]);
    return 0;
}
```