

## EEL170 – Computação 1 – Prática 1

### Comandos básicos de Linux/Unix

- Comandos básicos para gerenciamento de arquivos e diretórios:

Comando	Função
<b>cd</b>	Muda do diretório atual para o outro listado.
<b>cp</b>	Copia arquivo.
<b>ls</b>	Lista o conteúdo (arquivos e subdiretórios) de um diretório.
<b>file</b>	Determina o tipo do arquivo.
<b>grep</b>	Procura padrão em um arquivo texto e, caso o encontre, retorna a linha do arquivo contendo o padrão procurado.
<b>mv</b>	Move ou renomeia arquivo ou diretório.
<b>pwd</b>	Exibe caminho absoluto do diretório atual.
<b>rm</b>	Remove arquivo ou diretório.
<b>mkdir</b>	Cria diretório.
<b>chmod</b>	Muda as permissões de um arquivo ou diretório.
<b>touch</b>	Cria um arquivo vazio.
<b>rmdir</b>	Remove diretório vazio.
<b>man</b>	Exibe o manual de um dado comando.

Exemplos de uso:

```
maquina:~> cd meudiretorio
maquina:~/meudiretorio> ls
arquivovelho.txt
maquina:~/meudiretorio> cp arquivovelho.txt arquivonovo.txt
maquina:~/meudiretorio> ls
arquivonovo.txt arquivovelho.txt
maquina:~/meudiretorio> file arquivonovo.txt
arquivonovo.txt: ISO-8859 text
maquina:~/meudiretorio> grep "nome do aluno" arquivonovo.txt
nome do aluno e o restante da linha.
maquina:~/meudiretorio> mkdir diretorionovo
maquina:~/meudiretorio> ls
arquivonovo.txt arquivovelho.txt diretorionovo
maquina:~/meudiretorio> cd diretorionovo
maquina:~/meudiretorio/diretorionovo> pwd
/home/usuario/meudiretorio/diretorionovo
```

```

maquina:~/meudiretorio/diretorionovo> mv ../arquivonovo.txt .
maquina:~/meudiretorio/diretorionovo> ls
arquivonovo.txt
maquina:~/meudiretorio/diretorionovo> ls ..
arquivovelho.txt diretorionovo
maquina:~/meudiretorio/diretorionovo> rm arquivonovo.txt
maquina:~/meudiretorio/diretorionovo> ls
maquina:~/meudiretorio/diretorionovo> cd ..
maquina:~/meudiretorio> rmdir diretorionovo
maquina:~/meudiretorio> ls
arquivovelho.txt
maquina:~/meudiretorio> touch arquivonovo.txt
maquina:~/meudiretorio> ls
arquivonovo.txt arquivovelho.txt

```

- Comandos de edição de texto:

Comando	Função
<b>vi</b>	Editor de texto.
<b>vim</b>	Editor de texto vi melhorado.

**Exercício 1:** Crie um subdiretório próprio (utilize seu nome.sobrenome para nomeá-lo) no diretório public da máquina ou em outro indicado pelo professor. Em seguida, crie um arquivo de texto (extensão .txt).

**Exercício 2:** Liste todos os arquivos do diretório atual. Experimente o uso da opção -l. Quais as diferenças? *Dica: Para verificar as opções de qualquer comando do Linux, visite o manual do comando.*

**Exercício 3:** Crie uma cópia do arquivo de texto criado no Exercício 1. Em seguida, renomeie a cópia para que fique claro a partir do nome que se trata de um backup do arquivo original.

### Comandos básicos do vi/vim

- Comandos de inserção:

Comando	Função
<b>i</b>	Inserir texto antes da posição do cursor.
<b>a</b>	Inserir texto após a posição do cursor.
<b>R</b>	Substituir o texto na posição atual do cursor.
<b>A</b>	Inserir o texto no final da linha atual do cursor.
<b>o</b>	Adicionar uma linha abaixo da linha atual do cursor.

<b>O</b>	Adiciona uma linha acima da linha atual do cursor.
<b>ESC</b>	Muda para o modo de visualização.

- Comandos para alteração do texto:

<b>Comando</b>	<b>Função</b>
<b>x</b>	Deleta o caractere que está sob o cursor.
<b>dw</b>	Deleta a palavra, da posição atual do cursor até o final.
<b>dd</b>	Deleta a linha atual e copia o conteúdo para área de transferência.
<b>u</b>	Desfaz a última modificação.
<b>yy</b>	Copia 1 linha para a área de transferência.
<b>Nyy</b>	Copia N linhas para a área de transferência.
<b>p</b>	Cola o conteúdo da área de transferência.
<b>Np</b>	Cola N vezes o conteúdo da área de transferência.
<b>Ndd</b>	ndd - Deleta o conteúdo de N linhas a partir da linha atual.

- Comandos para busca de expressões, escrita do arquivo editado e saída do editor:

<b>Comando</b>	<b>Função</b>
<b>/palavra</b>	Busca por "palavra" no texto.
<b>:wq ou :x</b>	Salva o arquivo e saem do editor.
<b>:w</b>	Salva o arquivo.
<b>:q</b>	Sai do editor.

Para editar um arquivo com o vim, basta passar o nome do arquivo como argumento na linha de comando. Caso o arquivo não exista, o vim abre um arquivo com o nome dado. Exemplo:

```
maquina:~/meudiretorio> vim arquivovelho.txt
(** Editor é aberto ocupando toda a janela do terminal **)
```

**Exercício 4:** Abra o arquivo arquivovelho.txt no editor de texto vim e insira 4 nomes de colegas, cada um em uma linha diferente. Salve o arquivo e saia do editor.

**Exercício 5:** Abra novamente o arquivovelho.txt no vim e verifique a diferença entre todos os comandos de inserção. Após fazer alterações, salve o arquivo modificado com outro nome usando comandos do vim. Por exemplo, salve o arquivo modificado com o nome arquivomodificado.txt.

**Exercício 6:** Abra o arquivo modificado e use o comando dd com o cursor sobre uma das linhas. Desfaça a modificação com o u. Em seguida, copie e mova linhas usando os comandos de alteração de texto.

### Primeiro programa em C

- Compilação e execução:

Comando	Função
gcc	Compila programas em C.
./programa	Executa o "programa".

O comando para compilação usando o gcc usa como entrada o arquivo contendo o código-fonte e gera como saída o arquivo executável correspondente. Caso um erro seja detectado durante o processo de compilação, o procedimento é interrompido.

**Exercício 7:** Abra um arquivo com o vim e salve-o com o nome helloworld.c. Insira o código a seguir no arquivo, salve, compile e execute. Veja a saída do programa e examine os arquivos gerados no diretório atual.

```
#include <stdio.h>

int main () {
    printf ("Hello World!\n");
    return 0;
}
```