

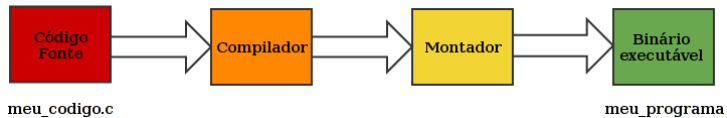
Aula 00: Desenvolvimento no GNU/Linux

A linguagem do computador

- ▶ O computador só entende binário (0 e 1).
- ▶ Humanos não são bons em ler binário. (Pouco intuitivo)
- ▶ Codificamos em linguagens que se aproximam da linguagem natural. (Abstração)
- ▶ **É preciso traduzir a linguagem que usamos para programar para a linguagem que a máquina entende.**



Compilação



Comandos

Os comandos de terminal têm o seguinte formato

```
$ comando [arg1] [arg2] ...
```

- ▶ Parâmetros podem ser opcionais e normalmente são passados após um hífen “-” na forma abreviada ou após dois hífens, na forma completa. Exemplo:

```
$ ls -h  
$ ls --help
```

- ▶ Lembre-se que isto é uma convenção, e muitos programas podem não seguir!

Shell –Principais comandos

- ▶ *ls* : Lista todos os arquivos do diretório atual.
- ▶ *pwd* : Exibe o diretório atual
- ▶ *cd* : Muda o diretório atual
 - ▶ *cd ~* : Muda o diretório atual para a sua home
 - ▶ *cd documents* : Entra no diretório “documents”
 - ▶ *cd ..* : sobe um nível de diretório
- ▶ *man* : Exibe o manfile do programa passado como argumento.
- ▶ *rm meu.arquivo* : Remove o arquivo meu.arquivo

Compilando seu código no GCC

```
$ gcc meu_codigo.c [parametros]
```

- ▶ parâmetros:
 - ▶ `-o meu_programa` : Informa o nome do arquivo de saída.
 - ▶ `-Wall` : Ativa todos os Warnings do compilador.
 - ▶ `-Werror` : Transforma os Warnings em erros.
 - ▶ Existem diversas outras flags úteis: `-g`, `-O3`, `-pedantic`, ...
- ▶ Executando seu programa:

```
$ ./meu_prog
```

Hello World!

Exercício - Escrever, compilar e executar um programa simples no GNU/Linux.

Comparando a saída

- ▶ diff - Compara dois arquivos de entrada.

```
$ diff saida_esperada.txt saida.txt
1c1
< Hello World
--
> Hello Mundo
```

- ▶ Dica: utilizar a opção -u ajuda!

```
$ diff -u saida_esperada.txt saida.txt
-- saida_esperada.txt <DATA E HORA>
+++ saida.txt <DATA E HORA>
@@ -1 +1 @@
-Hello World
+Hello Mundo
```


Aulas de GNU/Linux úteis

- ▶ A tecla Tab auto-completa nome de programas, caminho para arquivos, etc.
- ▶ Existe também o comando *man*. Ele recebe o nome de um comando/função/etc. como argumento e exibe o seu manual.
- ▶ Outra convenção muito útil é implementar textos de ajuda e de versão. Muitos programas implementam isso.
- ▶ O comando *apropos [termo]* pesquisa uma palavra-chave nos arquivos do man.

```
$ ls --help
$ ls --version
$ man ls
$ apropos browser
```

Contato



Gabriel Krisman Bertazi - krisman@libreplanetbr.org
Sergio Durigan Junior - sergiodj@libreplanetbr.org
Martin Ichilevici de Oliveira - martin@libreplanetbr.org
libreplanet-br-sp@libreplanet.org