

Ficha Prática 0

Sistemas Operativos 2023

Consultar Sites da Disciplina:

- <http://www.di.ubi.pt/~operativos/>
- <http://www.di.ubi.pt/~operativos/praticos/praticos.html>
- [Moodle Balcão Virtual](#)
- Linux Tutorial <http://www.di.ubi.pt/~operativos/unixtut/>

Programação em C

1 Crie um ficheiro chamado “ola.c” com seguinte código usando um editora simples.

```
main() { printf("\nBem Vindo\n") ; }
```

- formatar o ficheiro usando o comando “indent” > indent -npsl -bl ola.c
- voltar a ver as diferenças no ficheiro > cat ola.c

Compile com as opções seguintes: `cc -Wall -o ola ola.c` e a seguir alterar o código fonte para eliminar os avisos da compilação do programa. Tem de inserir a linha `#include <stdio.h>` no início do ficheiro. Tentar executar o comando “ola” escrevendo

> ./ola e > ola

- Se tiver de escrever ./ola modifique a variável PATH temporariamente para incluir o directório actual
> PATH=\$PATH:.
- E tentar executar novamente usando > ola
- Criar o directório ~/bin com o comando > mkdir ~/bin
- Mover o executável ola para o directório ~/bin com o comando > mv ola ~/bin
E executar > ola
- Conseguir executar o comando ola sem escrevendo o caminho completo ? (deverá ver qual o valor do seu path?)
- Se for necessário, alterar o path para incluir o directório ~/bin e execute o comando novamente ola !

2 Escreva o programa *argCount.c* cuja objectivo é contar o numero de argumentos passado ao executável e imprimir o nome do executável e numero de argumentos e listar os argumentos.

```
#include <stdio.h>
int main( int argc, char *argv[] ){
    printf("Nome próprio é %s e n° de argumentos %d\n",*argv,argc) ;
    int i=0;
    while ( *argv != NULL )
        printf("%d : %s\n",i++,*argv++);
    return 0;
}
```

- Compilar: `cc -Wall -o argCount argCount.c`
- Executar: `./argCount ola` Executar: `./argCount 1 2 three`
 - Faça uma digrama para explicar as variáveis do tipo inteiro e do apontador e a exposição de memória.
- Pode-se substituir o ciclo while por um ciclo for e usar a notação da indexação vetorial.
`for (i=0;i<argc;i++) printf("%d : %s\n",i, argv[i]);`

Notas:

- `char *argv[]` é equivalente a `char **argv`
- Utilizar o utilitário **indent** para formatar o seu código > indent -npsl -bl argCount.c

Configuração da Shell Bash

Fazer : Ficha <http://www.di.ubi.pt/~operativos/praticos/pdf/1-configuracao.pdf>

1.1 Criar o ficheiro `.bash_aliases` com aliases (atalhos) já definidos. Deverá incluir os seguintes

```
casa em alternativa aos comandos cd $HOME;date;pwd
cdstdc em alternativa ao comando cd /usr/include
ll em alternativa ao comando ls -alt
alias indent='indent -npsl -bl'
alias cc='cc -Wall'
```

Indent sugestão do formato para um ficheiro C. **Compilador** ativa sempre todas os avisos.

1.2 Alterar o ficheiro `.bashrc` (ou `.bash_profile`)

- (i) Para incluir os atalhos
(nota o seu ficheiro de configuração pode ter já estas linhas)

```
if [ -f ~/.bash_aliases ]; then
    source ~/.bash_aliases
fi
```

- (ii) Mudar a prompt por defeito (se desejar)

```
PS1="SO >"
```

- (iii) Acrescentar a diretória “\$HOME/bin” e “directoria atual” à variável `PATH`.

```
PATH=$PATH:$HOME/bin:•
export PATH
```

Sumário

- O Sistema Operativo : Linux GUI vs CLI -- Bash Interpretador
- Comandos sobre ficheiros e diretórios : ls, file, pwd, mkdir, cd, mv
- Comandos de visualização dos ficheiros cat, head/tail less/more od (octal dunn)
- O comando echo
- Editoras
- Variáveis do sistema Shell HOME, PATH, PS1
- Atalhos (ficheiro `.alias` ou `.bash_aliases`) e configuração do Shell (`.bashrc`)
- Compilação em c : opções -o e -Wall
- Indentação de ficheiros de código c

Nota: A variável `PATH` é uma variável de ambiente Linux que contém uma lista ordenada que o sistema operativo pesquisa para encontra uma executável quando um comando é emitido.

Repl.it users : show hidden files, add to repl.it.nix file, pkgs.bashInteractive, to use indent, awk etc