



Ficha Prática 1 SO 2022/2023

Utilização do Sistema Operativo e Redireccionamento

Revisão. Redireccionamento do stdin, stdout e stderr e pipes.
Exemplificação com o comando cat. Exercício de Programação:
Criação dum comando tipo "cat" usando as funções do standard library (stdio.h)
Ver : <http://www.di.ubi.pt/~operativos/praticos/pdf/2-utilizacao.pdf>
<https://www.di.ubi.pt/~operativos/unixtut/unix4.html>

1 Revisão: Redireccionamento

a. Descreva o significados dos símbolos

- < _____
- > _____
- >> _____
- 2> _____

b. Investigue o programa **cat**

- Pelo linha de comando : `man cat` ou `info cat`
- Internet <http://www.computerhope.com/unix/ucat.htm>
- Experimente : `cat < cat /etc/passwd < cat < /etc/passwd < cat -n /etc/passwd <`

c. Quantos parâmetros são passados para o programa **cat** nos seguintes casos

- `cat` 1 - Um Programa recebe sempre o seu nome próprio !
- `cat -t /etc/passwd` _____
- `cat /etc/passwd` _____
- `cat -t < /etc/passwd` _____

2 Exercício com o comando Linux sort

- Execute o comando **sort** para ordenar uma lista de números introduzidos pelo teclado
- Criar o ficheiro `frutas.txt` que represente o numero de identificação, nome da fruta e quantidade

```
1:lima:236
4:maranja:355
8:maca:370
12:uva:5400
13:limão:350
```

 - Criar comandos para Ordenar o ficheiro pelo (i) id, (ii) nome e (iii) quantidade enviando o resultados para o ficheiro `res.txt`

3 Exercícios com o comando Linux grep

- Utilizando o comando **grep** crie um comando para encontrar todas as linhas do ficheiro de `/etc/passwd` que contém users com bash Shell por defeito.
- Utilizando o comando **grep** crie um comando para listar todos os ficheiros de inclusão referenciado dentro de ficheiros dum diretório

4 Pipe revisão

```
$ curl https://www.rtp.pt/noticias/rss/desporto > tmp ; grep benfica tmp ; rm tmp
$ curl https://www.rtp.pt/noticias/rss/desporto | grep benfica
```

- Explique o funcionamento e diferença entre as duas linhas com comandos em cima.

5 Programação em C

Escrever um comando chamado `mostrar` cuja funcionalidade seja igual ao comando `cat`.
De seguida apresenta-se o ficheiro `mostrar.c` para servir de base ao seu desenvolvimento.

`mostrar.c`

```
#include <stdio.h>

void streamCopy ( FILE * entrada, FILE * saida);

int main( int argc, char *argv[] ){

    if ( 1 == argc )
        streamCopy ( stdin,  stdout);
    //else abrir ficheiros - exercício
    return 0;
}

void streamCopy ( FILE * entrada,  FILE * saída ){
    int ch;
    while ( (ch=fgetc(entrada)) != EOF ){
        fputc( ch, saida);
    }
}
```

- a. Escrever e compilar o programa `mostrar.c` e efetuar os seguintes testes

```
T1: mostrar
T2: mostrar < frutas.txt
T3: mostrar frutas.txt
(comparar com cat, cat </etc/passwd e cat /etc/passwd)
```

Terminada ?: O seu Programa efectua corretamente os testes ? Repare que o T3 não irá mostrar o ficheiro `/etc/passwd` portanto o teste passe se nada acontece ☺

- b. Inserir código no `mostrar.c` de modo a que a aplicação daí resultante permita mostrar o conteúdo de ficheiros passados para a aplicação como parâmetros.
- Evitar erros fatais no programa, verifique fazendo **testes**. Efetuar os testes T1,T2 e T3 e adicionalmente

```
T4: mostrar frutas.txt mostrar.c
T5: mostrar frutas.txt naoexiste
```

Para abrir um ficheiro utilize a função `fopen` do `<stdio.h>` . Faça “**man 3 fopen**” para ver o seu sintaxe
Quando for detectado um ficheiro que não existe o programa deverá imprimir no canal de erros padrão (stderr) a mensagem `fprintf(stderr,“%s : Fatal Error”` onde `%s` é o nome do ficheiro e depois chamar a função `perror(NULL)`

- c. Implementa a opção `-T`. Esta opção faz que caracteres de controlo (`\t`) são representados pelo simbolo `^I`

```
T6: mostrar -T
T7: mostrar -T frutas.txt mostrar.c
```

- d. Implementa a opção `-n`. Esta opção faz que no inicio da cada linha é imprimido o numero da linha do input

```
T8: mostrar -n
T9: mostrar -n frutas.txt mostrar.c
```

Notas:

- Sintaxe: `mostrar [-T] [-n] ficheiros`
- `FILE *stdin, *stdout, *stderr`. Estão declaradas no ficheiro `<stdio.h>` que habitualmente está na pasta `/usr/include`
- Estes ficheiros estarão abertas automaticamente no inicio da execução do programa. A linkagem é automática com a biblioteca standard (`libc.xx`). The standard static, ‘`libc.a`’, ou dynamic ‘`libc.so`’ C library is automatically linked into your programs by the compiler control program

Entregar no moodle : código da solução do exercício 5 (`mostrar.c`) e num ficheiro de texto (`solution.txt`) o output dos testes efetuados.

Sugestões

Tratamento das Opções. Sugestão

Syntax mostrar [-n] [-T] [files]

Utilizar “Global Variable(s)” para indicar se uma opção estiver ligada ou desligada.

```
int optiont=0;
```

```
int optionn=0;
```

```
int main (int argc, char *argv[])
{
    while (argc>1 && argv[1][0]!='-')
    {
        switch (argv[1][1])
        {
            case 'T':
                optiont=1;
                argc--;
                argv++;
                break;
            case 'n':
                optionn=1;
                argc--;
                argv++;
        }
    }
}
```

//resto do programa mantêm-se

Tratamento da nova linha na função de streamcopy

```
int nlinha=1;          //numero da linha atual
int inicioDaLinha=1; //1 true 0 false
while ((ch = fgetc (entrada)) != EOF)
{
    if ( optionn && inicioDaLinha ) {
        fprintf(saida,"%6d\t", nlinha++);
        inicioDaLinha = 0;
    }

    fputc (ch, saida);

    if (ch=='\n') inicioDaLinha =1;
}
```

Notas:

Fim do Ficheiro EOF (End Of File): CTR+d (unix) CTR+z (windows) : ↵ O simbolo para carrger no “enter”