



---

# RELATÓRIO

---

## Trabalho Prático da Unidade Curricular de Sistemas Operativos

Engenharia de Sistemas Informáticos

30 de Abril de 2022

Projeto realizado por:

Henrique Neto  
(16626)

João Moreira  
(23522)

João Araújo  
(23103)

Bruna Paço  
(24073)

Rogério Filho  
(21868)

Rodrigo Monteiro  
(23514)

## Índice

## Introdução

O seguinte relatório da unidade curricular de Sistemas Operativos do curso de Engenharia de Sistemas Informáticos do Instituto Politécnico do Cávado e do Ave, têm como objetivo dar a conhecer o trabalho prático sobre os conceitos de gestão de processos e de ficheiros, assim como a aplicação da comunicação entre processos.

Para isso, realizou-se um programa na linguagem C através da implementação de comandos baseados nas funções de chamada ao sistema, em inglês, system calls.

Tendo em conta o programa desenvolvido através da linguagem referida anteriormente, constam as seguintes funcionalidades:

1. Mostrar Ficheiro
2. Copiar Ficheiro
3. Acrescentar Origem Destino
4. Contar Linhas
5. Apagar Ficheiro
6. Informar Ficheiro
7. Lista de Ficheiros e Pastas de uma diretoria específica

## Curta demonstração do conteúdo do programa

### ➤ Funções no Header File

```
void printMenu();
void mostraFicheiro(char *fileName);
int tamanhoString(char *name);
bool verificarFicheiroExiste(char *fileName);
int bytesFicheiro(char *fileName);
void deletarFicheiro(char *fileName);
void copiarFicheiro(char *fileName);
void concatenarString(char *str1, char *str2, char *newString);
int tamanhoString(char *name);
void acrescentaDestino(char *fileOrigem, char *fileDestino);
void contarLinhas(char *fileName);
void listarDiretorio(char *diretoria);
```

### ➤ Menu

```
void printMenu() {
    printf("-----\n");
    printf("----- MENU -----\n");
    printf("-----\n");
    printf(" [1] -> Mostrar Ficheiro\n");
    printf(" [2] -> Copiar Ficheiro\n");
    printf(" [3] -> Acrescenta Origem Destino\n");
    printf(" [4] -> Conta Linhas Ficheiro\n");
    printf(" [5] -> Apaga Ficheiro\n");
    printf(" [6] -> Informa Ficheiro\n");
    printf(" [7] -> Lista Diretoria\n");
    printf(" [0] -> SAIR\n");
    printf("opcao: ");
}
```

```
-----
----- MENU -----
-----
[1] -> Mostrar Ficheiro
[2] -> Copiar Ficheiro
[3] -> Acrescenta Origem Destino
[4] -> Conta Linhas Ficheiro
[5] -> Apaga Ficheiro
[6] -> Informa Ficheiro
[7] -> Lista Diretoria
[0] -> SAIR
```

## ➤ Makefile

```
M makefile
1  all: build run
2
3  ✓ build:
4      gcc ./src/*.c -o main
5
6  ✓ run:
7      ./main
8
9  ✓ clean:
10     rm main
```

## 1. Mostrar Ficheiro

Assim que o utilizador clicar na tecla 1 no menu que surgirá assim que o projeto for executado, aparecerá todo o conteúdo existente que consta no ficheiro de texto que foi referido como parâmetro.

É importante reconhecer que caso o ficheiro não exista, o utilizador será avisado do sucedido.

A porção de código escrito tendo em base uma função para mostrar determinado ficheiro segue-se na imagem:

```
void mostraFicheiro(char *fileName)
{
    char opcao;
    int file;

    printf("-----\n");

    //Abrir arquivo
    char readBuffer[bytesFicheiro(fileName)];
    file = open(fileName, O_RDONLY);

    read(file, readBuffer, sizeof(readBuffer));
    write(STDIN_FILENO, readBuffer, sizeof(readBuffer));
    close(file);

    printf("\n-----\n");
}
```

## 2. Copiar Ficheiro

Esta funcionalidade tem como finalidade fazer a cópia de um ficheiro onde por conseguinte será criado um novo ficheiro designado por “ficheiro.copia”. Os dados do ficheiro são copiados na sua totalidade tendo como referência o ficheiro passado como parâmetro.

Também para esta função, caso o ficheiro seja inexistente, o utilizador será alertado do ocorrido.

```
do {  
    printf("> Insira o caminho do ficheiro a ser copiado: ");  
    scanf("%s", fileName);  
  
    if (verificarFicheiroExiste(fileName) == FALSE) {  
        perror("    > Erro ao ler ficheiro!");  
    }  
} while (verificarFicheiroExiste(fileName) == FALSE);
```

A parte do código desenvolvido referente à funcionalidade de copiar um determinado ficheiro pode ser vista na imagem:

```
void copiarFicheiro(char *fileName) {  
  
    int file, newFile;  
    char readBuffer[bytesFicheiro(fileName)];  
    char opcao;  
  
    char *aux = ".copia";  
    char newString[tamanhoString(fileName) + tamanhoString(aux)];  
  
    concatenarString(fileName, aux, newString);  
  
    file = open(fileName, O_RDONLY);  
    newFile = open(newString, O_RDWR|O_CREAT, S_IRUSR | S_IWUSR);  
  
    read(file, readBuffer, sizeof(readBuffer));  
    int result = write(newFile, readBuffer, sizeof(readBuffer));  
    close(file);  
    close(newFile);  
  
    if (result < 0) {  
        perror("> Nada foi escrito!");  
    } else {  
        printf("> Ficheiro copiado com sucesso!\n");  
        printf("    > Foram copiados: %d bytes para o ficheiro '%s'\n", result, newString);  
    }  
  
    printf("-----\n");  
}
```



### 3. Acrescentar Origem Destino

De facto, para que o utilizador tenha a possibilidade de acrescentar, no menu terá de pressionar a tecla 3 e de seguida já vai poder acrescentar a informação que consta em origem para o final, em destino.

Ainda em relação a esta função, se o utilizador for em busca de um ficheiro que é inexistente, o mesmo será avisado.

Para que fosse possível concretizar o funcionamento deste comando, foi escrita a seguinte porção de código:

```
void acrescentaDestino(char *fileOrigem, char *fileDestino) {  
  
    int file1, file2;  
    char readBuffer[bytesFicheiro(fileOrigem)];  
    char opcao;  
  
    file1 = open(fileOrigem, O_RDONLY);  
    file2 = open(fileDestino, O_WRONLY | O_APPEND);  
  
    read(file1, readBuffer, sizeof(readBuffer));  
    int result = write(file2, readBuffer, sizeof(readBuffer));  
    close(file1);  
    close(file2);  
  
    if (result == -1) {  
        perror("> Nada foi escrito!\n");  
    } else {  
        printf("> Ficheiro copiado com sucesso!\n");  
    }  
  
    printf("\n-----\n");  
}
```

## 4. Contar Linhas

A partir desta função, o utilizador terá a oportunidade de saber quantas linhas contém o ficheiro de texto. As linhas serão contadas e a quantidade das mesmas vai ser visível.

Se o ficheiro não estiver explícito, a função não é executada.

A função contém a seguinte informação:

```
void contarLinhas(char *fileName)
{
    int fd, count=0;
    int size = bytesFicheiro(fileName);
    char readBuffer[bytesFicheiro(fileName)];
    char *apont;
    char opcao;

    fd = open(fileName, O_RDONLY);

    read(fd, readBuffer, sizeof(readBuffer));

    apont = strtok(readBuffer, "\n");

    while(apont != NULL)
    {
        count ++;
        apont = strtok(NULL, "\n");
    }

    close(fd);

    printf("Numero de linhas: %d\n", count);

    printf("\n-----\n");
}
```

## 5. Apagar Ficheiro

Através desta funcionalidade, o utilizador vai definir e digitar o nome de um ficheiro que deseja eliminar após clicar na tecla 5 do menu.

O ficheiro será removido a partir da função “unlink”, isto é, uma entrada de diretório que se refere a um ficheiro

Caso o ficheiro não exista, nada acontecerá.

```
void deletarFicheiro(char *filename)
{
    char opcao;

    unlink(filename);
    printf("Ficheiro Deletado com sucesso!!\n");

    printf("-----\n");
}
```

## Conclusão

Para concluir, foi uma mais-valia realizar este projeto pois foi relevante para termos a perceção acerca de como se realiza a gestão de processos e de ficheiros bem como a aplicação da comunicação entre processos através deste trabalho prático realizado tendo em base vários recursos como a linguagem C onde, também, através da mesma foi possível aprimorar os conhecimentos da realização de um programa.

## **Bibliografia**

<https://social.microsoft.com/Forums/pt-BR/home>

<https://stackoverflow.com/>

**Recursos da unidade curricular presentes no Moodle**