

Sistemas Operativos

Grupo 5

Gerado por Doxygen 1.9.8

1 Trabalho Prático	1
1.1 Alunos	1
1.2 Organização	1
2 Índice dos ficheiros	3
2.1 Lista de ficheiros	3
3 Documentação do ficheiro	5
3.1 Referência ao ficheiro <code>acrescentaOrigemDestino.c</code>	5
3.2 Referência ao ficheiro <code>apagaFicheiro.c</code>	5
3.2.1 Descrição detalhada	5
3.2.2 Documentação das funções	5
3.2.2.1 <code>main()</code>	5
3.3 Referência ao ficheiro <code>contaFicheiro.c</code>	6
3.3.1 Descrição detalhada	6
3.3.2 Documentação das funções	6
3.3.2.1 <code>main()</code>	6
3.4 Referência ao ficheiro <code> copiaFicheiro.c</code>	6
3.4.1 Descrição detalhada	7
3.4.2 Documentação das macros	7
3.4.2.1 <code>BUFFER_SIZE</code>	7
3.4.3 Documentação das funções	7
3.4.3.1 <code>main()</code>	7
3.5 Referência ao ficheiro <code>informaFicheiro.c</code>	7
3.5.1 Descrição detalhada	7
3.5.2 Documentação das funções	8
3.5.2.1 <code>main()</code>	8
3.6 Referência ao ficheiro <code>interpretador.c</code>	8
3.6.1 Descrição detalhada	8
3.6.2 Documentação das macros	9
3.6.2.1 <code>MAX_LENGTH</code>	9
3.6.3 Documentação das funções	9
3.6.3.1 <code>main()</code>	9
3.7 Referência ao ficheiro <code>listaDiretoria.c</code>	9
3.7.1 Descrição detalhada	9
3.7.2 Documentação das macros	9
3.7.2.1 <code>PATH_MAX_LEN</code>	9
3.7.3 Documentação das funções	9
3.7.3.1 <code>main()</code>	9
3.8 Referência ao ficheiro <code>mostraFicheiro.c</code>	10
3.8.1 Descrição detalhada	10
3.8.2 Documentação das macros	10
3.8.2.1 <code>BUFFER_SIZE</code>	10

3.8.3 Documentação das funções	10
3.8.3.1 main()	10
3.9 Referência ao ficheiro README.md	11
Índice	13

Capítulo 1

Trabalho Prático

Licenciatura em Engenharia de Sistemas Informáticos (regime pós-laboral) 2023-24

Sistemas Operativos

1.1 Alunos

Número	Nome
39	Fernando Salgueiro
29576	Hélder Costa
30516	Hugo Lopes
30517	Cláudio Fernandes
30518	Nuno Cruz

1.2 Organização

Foram efetuadas as seguintes implementações:

```
Conjunto de comandos para manipulação de ficheiros:
mostra
copia
acrescenta
conta
apaga
informa
lista
```

```
Interpretador de linha de comandos
```

```
Gestão de Sistemas de Ficheiros
```

```
Análise ao Sistemas de Ficheiros
```

Documentação Doxygen: Pasta HTML: index.html

Documentação LaTeX: Pasta latex: refman.pdf

Capítulo 2

Índice dos ficheiros

2.1 Lista de ficheiros

Lista de todos os ficheiros com uma breve descrição:

acrescentaOrigemDestino.c	
Programa para copiar o conteúdo de um ficheiro de origem para um ficheiro de destino, com opção de append	5
apagaFicheiro.c	
Programa para eliminar um ficheiro	5
contaFicheiro.c	
Programa para contar o número de linhas de um ficheiro	6
copiaFicheiro.c	
Programa para copiar o conteúdo de um ficheiro para outro	6
informaFicheiro.c	
Programa para imprimir as informações detalhadas sobre um ficheiro	7
interpretador.c	
Implementação de um interpretador simples que executa comandos básicos	8
listaDiretoria.c	
Programa para listar o conteúdo de uma diretoria	9
mostraFicheiro.c	
Programa para ler o conteúdo de um ficheiro e imprimir no stdout	10

Capítulo 3

Documentação do ficheiro

3.1 Referência ao ficheiro acrescentaOrigemDestino.c

Programa para copiar o conteúdo de um ficheiro de origem para um ficheiro de destino, com opção de append.

```
#include <unistd.h>
#include <fcntl.h>
#include <string.h>
```

Diagrama de dependências de inclusão para acrescentaOrigemDestino.c:

3.2 Referência ao ficheiro apagaFicheiro.c

Programa para eliminar um ficheiro.

```
#include <unistd.h>
#include <stdio.h>
#include <string.h>
```

Diagrama de dependências de inclusão para apagaFicheiro.c:

Funções

- `int main (int argc, char *argv[])`
Função principal do programa.

3.2.1 Descrição detalhada

Programa para eliminar um ficheiro.

Este programa recebe o nome de um ficheiro como argumento e procede à sua eliminação. Se o ficheiro for eliminado com sucesso, é retornada uma mensagem de sucesso. Caso contrário, retorna uma mensagem de erro.

3.2.2 Documentação das funções

3.2.2.1 main()

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

Função principal do programa.

Parâmetros

<i>argc</i>	Número de argumentos passados na linha de comandos.
<i>argv</i>	Vetor de argumentos passados na linha de comandos.

Retorna

Retorna 0 em caso de sucesso, 1 em caso de erro.

3.3 Referência ao ficheiro contaFicheiro.c

Programa para contar o número de linhas de um ficheiro.

```
#include <unistd.h>
#include <fcntl.h>
#include <string.h>
```

Diagrama de dependências de inclusão para contaFicheiro.c:

Funções

- int [main](#) (int argc, char *argv[])

Função principal do programa.

3.3.1 Descrição detalhada

Programa para contar o número de linhas de um ficheiro.

Este programa recebe o nome de um ficheiro como argumento e conta o número de linhas que consta no ficheiro. O programa imprime o número total de linhas no ficheiro no stdout. Se existirem erros durante a abertura, leitura ou fecho do ficheiro, são devolvidas mensagens de erro.

3.3.2 Documentação das funções

3.3.2.1 main()

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

Função principal do programa.

Parâmetros

<i>argc</i>	Número de argumentos passados na linha de comandos.
<i>argv</i>	Vetor de argumentos passados na linha de comandos.

Retorna

Retorna 0 em caso de sucesso, 1 em caso de erro.

3.4 Referência ao ficheiro copiaFicheiro.c

Programa para copiar o conteúdo de um ficheiro para outro.

```
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <string.h>
```

Diagrama de dependências de inclusão para copiaFicheiro.c:

Macros

- #define [BUFFER_SIZE](#) 1024

Funções

- int [main](#) (int argc, char *argv[])

Função principal do programa.

3.4.1 Descrição detalhada

Programa para copiar o conteúdo de um ficheiro para outro.

Este programa recebe um ficheiro como argumento e efetua uma cópia do seu conteúdo para outro ficheiro. Se ocorrerem erros durante a abertura, leitura, escrita ou no fecho dos ficheiros, são retornadas mensagens de erro.

3.4.2 Documentação das macros

3.4.2.1 BUFFER_SIZE

```
#define BUFFER_SIZE 1024
```

3.4.3 Documentação das funções

3.4.3.1 main()

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

Função principal do programa.

Parâmetros

<i>argc</i>	Número de argumentos passados na linha de comandos.
<i>argv</i>	Vetor de argumentos passados na linha de comandos.

Retorna

Retorna 0 em caso de sucesso, 1 em caso de erro.

3.5 Referência ao ficheiro informaFicheiro.c

Programa para imprimir as informações detalhadas sobre um ficheiro.

```
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <pwd.h>
#include <time.h>
#include <stdlib.h>
#include <string.h>
```

Diagrama de dependências de inclusão para informaFicheiro.c:

Funções

- int [main](#) (int argc, char *argv[])

Função principal do programa.

3.5.1 Descrição detalhada

Programa para imprimir as informações detalhadas sobre um ficheiro.

Imprime o tipo, inode, proprietário e datas relevantes (criação, última leitura e última modificação) de um ficheiro.

3.5.2 Documentação das funções

3.5.2.1 main()

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

Função principal do programa.

Parâmetros

<i>argc</i>	Número de argumentos passados na linha de comando.
<i>argv</i>	Vetor de argumentos passados na linha de comando.

Retorna

Retorna 0 em caso de sucesso, 1 em caso de erro.

3.6 Referência ao ficheiro interpretador.c

Implementação de um interpretador simples que executa comandos básicos.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/wait.h>
#include <unistd.h>
#include <errno.h>
```

Diagrama de dependências de inclusão para interpretador.c:

Macros

- #define `MAX_LENGTH` 1024

Funções

- int `main` ()

Função principal do interpretador.

3.6.1 Descrição detalhada

Implementação de um interpretador simples que executa comandos básicos.

Este programa implementa um interpretador que lê comandos do utilizador e executa comandos específicos ou ficheiros executáveis disponíveis no diretório atual. Os comandos disponíveis incluem:

- mostra
- copia
- acrescenta
- conta
- apaga
- informa
- lista
- termina
- help

3.6.2 Documentação das macros

3.6.2.1 MAX_LENGTH

```
#define MAX_LENGTH 1024
```

3.6.3 Documentação das funções

3.6.3.1 main()

```
int main ( )
```

Função principal do interpretador.

Esta função implementa um loop que lê comandos do utilizador, analisa os comandos e executa-os utilizando um processo filho. Comandos reconhecidos são executados a partir de ficheiros executáveis na mesma diretoria ou são comandos internos como "help" e "termina".

Retorna

int Código de retorno do programa: 0 em caso de sucesso.

3.7 Referência ao ficheiro listaDiretoria.c

Programa para listar o conteúdo de uma diretoria.

```
#include <unistd.h>
#include <dirent.h>
#include <stdio.h>
#include <string.h>
#include <sys/stat.h>
```

Diagrama de dependências de inclusão para listaDiretoria.c:

Macros

- #define `PATH_MAX_LEN` 1024

Funções

- int `main` (int argc, char *argv[])
Função principal do programa.

3.7.1 Descrição detalhada

Programa para listar o conteúdo de uma diretoria.

Este programa recebe o caminho de um diretório como argumento da linha de comandos e lista o conteúdo desse diretório, incluindo ficheiros e subdiretórios. Se nenhum caminho for fornecido, o programa lista o conteúdo do diretório atual. O programa exibe o tipo de cada item listado (diretório ou ficheiro) e o seu nome. Se existirem erros durante a abertura ou leitura do diretório, são devolvidas mensagens de erro.

3.7.2 Documentação das macros

3.7.2.1 PATH_MAX_LEN

```
#define PATH_MAX_LEN 1024
```

3.7.3 Documentação das funções

3.7.3.1 main()

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

Função principal do programa.

Parâmetros

<i>argc</i>	Número de argumentos passados na linha de comandos.
<i>argv</i>	Vetor de argumentos passados na linha de comandos.

Retorna

Retorna 0 em caso de sucesso, 1 em caso de erro.

3.8 Referência ao ficheiro mostraFicheiro.c

Programa para ler o conteúdo de um ficheiro e imprimir no stdout.

```
#include <unistd.h>
```

```
#include <fcntl.h>
```

```
#include <string.h>
```

Diagrama de dependências de inclusão para mostraFicheiro.c:

Macros

- `#define BUFFER_SIZE 1024`

Funções

- `int main (int argc, char *argv[])`

Função principal do programa.

3.8.1 Descrição detalhada

Programa para ler o conteúdo de um ficheiro e imprimir no stdout.

Abre um ficheiro inserido como argumento, efetua a leitura do seu conteúdo em blocos e escreve os mesmos no stdout. Caso ocorra algum erro durante a leitura, escrita ou no fecho do ficheiro, é retornada uma mensagem de erro.

3.8.2 Documentação das macros

3.8.2.1 BUFFER_SIZE

```
#define BUFFER_SIZE 1024
```

3.8.3 Documentação das funções

3.8.3.1 main()

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

Função principal do programa.

Parâmetros

<i>argc</i>	Número de argumentos passados na linha de comandos.
<i>argv</i>	Vetor de argumentos passados na linha de comandos.

Retorna

Retorna 0 em caso de sucesso, 1 em caso de erro.

3.9 Referência ao ficheiro README.md

Índice

acrescentaOrigemDestino.c, [5](#)
apagaFicheiro.c, [5](#)
 main, [5](#)

BUFFER_SIZE
 copiaFicheiro.c, [7](#)
 mostraFicheiro.c, [10](#)

contaFicheiro.c, [6](#)
 main, [6](#)
copiaFicheiro.c, [6](#)
 BUFFER_SIZE, [7](#)
 main, [7](#)

informaFicheiro.c, [7](#)
 main, [8](#)
interpretador.c, [8](#)
 main, [9](#)
 MAX_LENGTH, [9](#)

listaDiretoria.c, [9](#)
 main, [9](#)
 PATH_MAX_LEN, [9](#)

main
 apagaFicheiro.c, [5](#)
 contaFicheiro.c, [6](#)
 copiaFicheiro.c, [7](#)
 informaFicheiro.c, [8](#)
 interpretador.c, [9](#)
 listaDiretoria.c, [9](#)
 mostraFicheiro.c, [10](#)
MAX_LENGTH
 interpretador.c, [9](#)
mostraFicheiro.c, [10](#)
 BUFFER_SIZE, [10](#)
 main, [10](#)

PATH_MAX_LEN
 listaDiretoria.c, [9](#)

README.md, [11](#)

Trabalho Prático, [1](#)