Acesso a ficheiros (streams)

Programação em Sistemas Computacionais

Agenda

• Acesso a ficheiros de texto

Acesso a ficheiros

• FILE: tipo que mantém informação sobre o stream

```
typedef struct { ... } FILE;
```

Abertura e fecho de um stream

```
FILE* fopen(const char* filename, const char* mode);
int fclose(FILE *stream);
```

• mode (man fopen):

```
"r" – leitura (erro se ficheiro não existente; stream posicionado no início)
"w" – escrita (cria ficheiro se não existe; stream posicionado no início)
"a" – acrescenta (cria ficheiro se não existe; stream posicionado no fim)
"r+", "w+", "a+" – modo composto (leitura e escrita)
```

- Funções para leitura e escrita sobre ficheiros:
 - As mesmas funções para aceder aos streams stdin, stdout e stderr com o prefixo f e com o parâmetro adicional FILE*

Exemplos de funções (#include <stdio.h>)

Funções para leitura (...\$ man xxx)

```
int fgetc(FILE* stream);
size_t fread(void* ptr, size_t size, size_t nmemb, FILE* stream);
char * fgets(char* line, int n, FILE* stream);
int fscanf(FILE *stream, const char * frmt, ...);
```

Funções para escrita (...\$ man xxx)

```
int fputc(int c, FILE* stream);
size_t fwrite(const void* ptr, size_t size, size_t nmemb, FILE* stream);
char * fputs(const char* line, FILE* stream);
int fprintf(FILE *stream, const char * frmt, ...);
int fflush(FILE* stream);
```

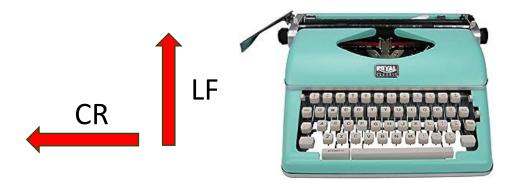
Funções de posicionamento de stream (...\$ man xxx)

```
int fseek(FILE* stream, long offset, int origin);
long ftell(FILE* stream);
```

• Parâmetro origin: SEEK_SET, SEEK_CUR, SEEK_END

Curiosidade: origem da mudança de linha num ficheiro de texto

- Nos sistemas DOS, o caracter '\n' é composto pelos caracteres Line Feed (LF) e Carriage Return (CR)
 - LF = (n) = 10 e CR = (n) = 13
- Porquê?
 - Herança dos sistemas de escrita clássicos: máquinas de escrever, telégrafos, ...







 Nos sistema atuais (Windows/Linux), a mudança de linha é traduzida apenas pelo caracter LF

Exercício

- Adapte o exercício anterior (words_histogram) para receber opcionalmente por parâmetro da aplicação o nome do ficheiro de entrada e o nome do ficheiro de saída através das opções -i e -o, respetivamente. Utilize para processamento das opções a função getopt. Em caso de omissos devem ser usados os streams standard de entrada (stdin) e de saída (stdout)
- Exemplos de chamada da aplicação:
 - \$./words_histogram -i file_in.txt -o file_out.txt
 - \$./words_histogram -o file_out.txt -i file_in.txt
 - \$./words_histogram –i file_in.txt
 - \$./words_histogram -o file_out.txt
 - \$./words_histogram