Programação / Programação I
LEI/1, LTSI/1, LMAT/1

Hugo Pedro Proença

Universidade da Beira Interior
Departamento de Informática

Resumo

- Interacção com ficheiros binários
  - fwrite()
  - fread()
  - fseek()
  - rewind()
  - sizeof()
- Exercícios

Ficheiros Binários

- Tal como referido anteriormente, a unidade atómica de informação neste tipo de ficheiros é o “bit”.
  - Nível mais baixo que os ficheiros de texto
  - Menor percepção ser humano
  - Melhor desenpenho

Ficheiros Binários

- As fases exigidas na interação com ficheiros de texto permanecem válidas neste tipo de ficheiros:
  - Declaração da stream.
  - Abertura da stream
  - Leitura/Escrita
  - Fecho da stream

Ficheiros Binários : Abertura

- A função “fopen()” é também utilizada na abertura de streams associadas a ficheiros binários.

- Exemplo:

  FILE *f=fopen(“dados.dat”,”wb”);

- Principais modos de abertura:
  - “wb”: abre um ficheiro binário para escrita.
  - “rb”: abre um ficheiro binário para leitura.
  - “ab”: abre um ficheiro binário em modo “append”.
  - “r+b”: abre um ficheiro binário em modo de escrita e leitura.

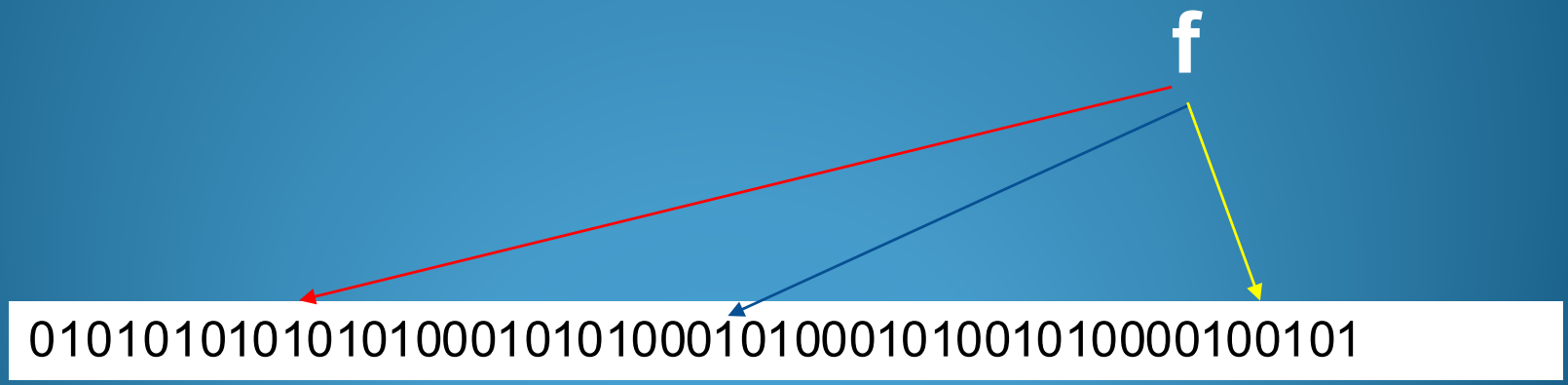
Ficheiros Binários : Posicionamento

- **É importante notar que ao interagir (escrever / ler) com um ficheiro a variável "stream" aponta sempre para uma posição específica do ficheiro:**

```
FILE * f=fopen(...);
...
```

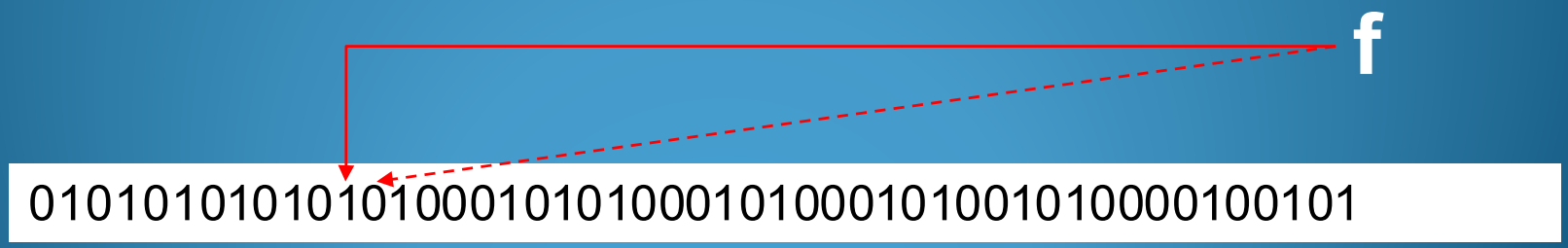
Ficheiros Binários: Posicionamento

- Cabe ao programador gerir a posição de/para onde ler ou escrever no ficheiro.
  - Opção 1: Utilização das funções de Escrita/Leitura.
  - Opção 2: Posicionamento explícito.



Ficheiros Binários: Posicionamento

- **Opção 1: ao aceder a uma posição do ficheiro (seja para leitura ou escrita), a posição da stream é automáticamente actualizada para o espaço imediatamente à direita.**
  - **No exemplo abaixo ilustrado, caso fosse lido o bit "1" indicado pela seta contínua, a posição da stream sería actualizada para a seta tracejada.**



Ficheiros Binários: Escrita

- A função fwrite() permite a escrita de sequências binárias de tamanho arbitrário.
- Protótipo:

```
size_t  fwrite(const void *variavel, size_t num_bytes, size_t count, FILE *f);
```

- O valor de retorno é o número de bytes efectivamente escritos em ficheiro.
- "variavel" é um apontador para o espaço de memória que se deseja copiar para ficheiro.
- "num_bytes" define o número de bytes a escrever por cada elemento.
- "count" define o número total de elementos a escrever.
- "f" específica o ficheiro onde se deseja escrever.

fwrite(): exemplos

```
int x,tot;
Pessoa p;
Pessoa v[10];
FILE *f=fopen("exemplos.dat","w");
...
tot=fwrite(&x,sizeof(int),1,f);
tot=fwrite(&p,sizeof(Pessoa),1,f);
tot=fwrite(v,sizeof(Pessoa),10,f);
```

Os exemplos acima ilustrados permitem respectivamente escrever...

- 1 inteiro...
- 1 elemento do tipo Pessoa...
- 1 vector de 10 elementos do tipo Pessoa...

para o ficheiro binário "exemplos.dat".

Função sizeof()

A função sizeof() permite é uma função avaliada em tempo de compilação que permite devolver o espaço ocupado por um tipo de dados.

Proporciona independência relativamente a diferentes platafortmas e/ou sistemas operativos.

sizeof(int) → Devolve o número de bytes que um inteiro ocupa.

sizeof(Pessoa)→Devolve o número de bytes que um elemento do tipo “Pessoa” ocupa.

Ficheiros Binários: Leitura

- A função fread() permite a leitura de sequências binárias de tamanho arbitrário.
- Protótipo:

```
size_t  fread(void *variavel, size_t num_bytes, size_t count, FILE *f);
```

- O valor de retorno é o número de bytes efectivamente lidos de ficheiro.
- "variavel" é um apontador para o espaço de memória onde se deseja copiar a informação lida de ficheiro.
- "num_bytes" define o número de bytes a ler por cada elemento.
- "count" define o número total de elementos a ler.
- "f" específica o ficheiro de onde se deseja ler.

fwrite(): exemplos

```
int x,tot;
Pessoa p;
Pessoa v[10];
FILE *f=fopen("exemplos.dat","r");
...
tot=fread(&x,sizeof(int),1,f);
tot=fread(&p,sizeof(Pessoa),1,f);
tot=fread(v,sizeof(Pessoa),10,f);
```

Os exemplos acima ilustrados permitem respectivamente ler...

- 1 inteiro...
- 1 elemento do tipo Pessoa...
- 1 vector de 10 elementos do tipo Pessoa...

do ficheiro binário "exemplos.dat".

Posicionamento explícito

- **Sempre que não é desejado ler/escrever de forma sequêncial, podem-se usar 2 funções de posicionamento da stream:**

- **Protótipos:**

```
void rewind(FILE *f);
int fseek(FILE *fp, long numBytes, int origem);
```

Posicionamento explícito

- **Sempre que não é desejado ler/escrever de forma sequêncial, podem-se usar 2 funções de posicionamento da stream:**

- **Protótipos:**

```
void rewind(FILE *f);
int fseek(FILE *fp, long numBytes, int origem);
```

fseek()

- A função “fseek()” permite definir o local onde se deseja posicionar a stream de acesso ao ficheiro.
- Exemplos:

```
X=fseek(f,100,SEEK_SET);   //Posiciona a stream 100 bytes à direita do ínicio do ficheiro
X=fseek(f,15,SEEK_CUR);    //Avança 15 bytes a partir da posição actual
X=fseek(f,20,SEEk_END);    //Posiciona a stream 15 bytes antes do final do ficheiro
```

rewind()

- A função rewind() permite posicionar a stream no ínicio do ficheiro.

```
FILE *f=fopen("exemplo.dat","rb"); //stream no ínicio
fread(...)
fread(...)                  //stream avançou no ficheiro
rewind(f);      //Reposiciona a stream no ínicio do ficheiro.
fread(...)      //Leitura a partir do ínicio do ficheiro
```

Ficha prática 6: Exercício

•Implemente um programa em linguagem C que efectue a gestão do valor de penalização pela assiduidade de cada um dos alunos de Programação.

Para tal, utilize o seguinte tipo de dados:

```
typedef struct{
        int numeroAluno;
        float penalização;
}Penalizacao;
```

Por forma a aumentar a segurança da aplicação, toda a informação é registada em ficheiros binários, procurando-se minimizar a quantidade de informação em memória. implemente as seguintes funções:

•void adicionaFalta(FILE *f, int numAluno);
    //"f" é uma stream aberta para leitura e escrita,
    //"numAluno" corresponde ao aluno a que se deseja marcar falta.

•float devolvePenalizacao(FILE *f, int numAluno);
    ////"f" é uma stream aberta para leitura e escrita,
    //"numAluno" corresponde ao aluno de que se deseja obter informação
    //É devolvido o valor de penalização do aluno, ou "-1" caso o aluno não exista.