

Programação 2024/25

LEI, LEI-PL, LEI-CE

Aula Laboratorial 11

Bibliografia:

K. N. King. *C programming: A Modern Approach* (2nd Edition). W. W. Norton: capítulo 22.

Código de apoio para a aula: <https://gist.github.com/FranciscoBPereira>

Ficheiros Binários

Exercícios Obrigatórios

1. Considere a seguinte definição:

```
struct dados{
    int a;
    char b;
};
```

Analise os programas A e B. Ao responder às questões, pode assumir que não existem problemas no processamento do ficheiro.

A

```
#include <stdio.h>

int main() {
    struct dados x = {4, 'k'};
    char c = 'w';
    FILE *f;

    f = fopen("fich.bin", "wb");
    if(f == NULL)
        return 0;
    fwrite(&x, sizeof(struct dados), 1, f);
    fwrite(&c, sizeof(char), 1, f);
    fwrite(&(x.a), sizeof(int), 1, f);
    fclose(f);
    return 0;
}
```

- a) Quando o programa A inicia a sua execução, o ficheiro “fich.bin” tem 4 inteiros armazenados (12, 56, -5, 934). Que informação contém o ficheiro quando termina a execução?

Programação 2024/25

LEI, LEI-PL, LEI-CE

- b) Quantos bytes tem o ficheiro no final da execução do programa A? Escreva a sua estimativa e depois confirme se está correta. Caso não esteja, encontre uma explicação para a diferença.

B

```
#include <stdio.h>

int main() {
    struct dados x;
    char c;
    int i;
    FILE *f;

    f = fopen("fich.bin", "rb");
    if(f == NULL)
        return 0;

    fread(&x, sizeof(struct dados), 1, f);
    fread(&c, sizeof(char), 1, f);
    fread(&i, sizeof(int), 1, f);
    printf("%d - %d - %c - %c\n", x.a, i, x.b, c);
    fclose(f);
    return 0;
}
```

- c) O programa B efetua a leitura do ficheiro anteriormente criado pelo programa A. Repare como a leitura é um espelho perfeito da escrita (tipo e quantidade de dados / ordem pela qual são processados). Que informação é apresentada na consola por este programa?

2. Um ficheiro binário tem vários valores inteiros armazenados.

- a) Escreva uma função em C que calcule a soma e a média de todos os valores armazenados. A função recebe como parâmetro o nome do ficheiro. Devolve a soma e a média como resultado.
- b) Escreva uma função em C que verifique se o ficheiro armazena valores numa ordem estritamente crescente. A função recebe como parâmetro o nome do ficheiro binário. Devolve 1 se a ordem for estritamente crescente, ou 0, caso contrário.

Programação 2024/25

LEI, LEI-PL, LEI-CE

- c) Escreva uma função em C que crie um vetor dinâmico para onde deve copiar todos os valores pares que existem no ficheiro binário. A função tem o seguinte protótipo:

```
int* criaVetor(char *nomeFich, int *tam);
```

A função devolve o endereço do vetor dinâmico como resultado. Recebe como parâmetros o nome do ficheiro binário e o endereço de uma variável inteira onde deve colocar a dimensão do vetor. Se ocorrer algum problema, a função devolve NULL e coloca o valor 0 na variável referenciada pelo segundo parâmetro.

- d) Escreva uma função em C que crie 2 novos ficheiros binários, a partir da informação do ficheiro original: num deles devem ficar os valores superiores ou iguais à média e no outro os valores inferiores à média. O ficheiro original não deve ser alterado. A função recebe os nomes dos 3 ficheiros como parâmetro.

3. Assuma que se pretende gerir percursos efetuados pelos comboios da CP, sabendo que cada um desses percursos é armazenado num ficheiro binário individual. O que caracteriza cada um dos percursos são os seus pontos de paragem. Assim, num ficheiro binário relativo a um determinado percurso estão armazenados os vários pontos de paragem e o tempo que demora a efetuar cada uma das ligações entre duas paragens consecutivas. A informação sobre uma paragem é armazenada numa estrutura do tipo `struct paragem`.

“Percurso_1.dat”

Lisboa	nome
0	minutos
Coimbra	nome
120	minutos
Aveiro	nome
45	minutos
Porto	nome
35	minutos

```
struct paragem{  
    char nome[50];  
    int minutos;  
};
```

O campo `nome` armazena o nome da paragem e o campo `minutos` armazena o tempo que o comboio demora (em média) desde a paragem anterior. As estruturas no ficheiro estão armazenadas por ordem de passagem. A primeira estrutura corresponde ao local de partida do comboio e a última ao local de destino. Na estrutura relativa ao ponto de partida do comboio o campo `minutos` tem o valor 0. Se existir um percurso “Lisboa-Coimbra-Aveiro-Porto” o ficheiro binário correspondente poderia ficar com o formato indicado na figura.

- a) Escreva uma função em C que mostre no monitor os nomes e os minutos de todas as estruturas que fazem parte do ficheiro. O nome do ficheiro é passado como parâmetro da função.
- b) Escreva uma função em C que mostre no monitor os nomes da origem e do destino final do percurso armazenado no ficheiro. O nome do ficheiro é passado como parâmetro da função.

Programação 2024/25

LEI, LEI-PL, LEI-CE

- c) Escreva uma função em C que verifique se uma determinada paragem existe num trajeto. A função recebe o nome do ficheiro e o nome da paragem como parâmetros. Devolve 1 se a paragem existir, ou 0, caso contrário.
- d) Escreva uma função em C que calcule a duração total em minutos do percurso completo armazenado num ficheiro. A função recebe o nome do ficheiro como parâmetro e devolve a duração contabilizada.
- e) Escreva uma função em C que verifique se um percurso de comboio permite efetuar a ligação entre dois locais A e B. A função recebe, como parâmetros, o nome do ficheiro binário onde está armazenado o percurso e os nomes dos locais de início (A) e de final (B) da ligação pretendida. Caso a ligação possa ser efetuada, a função devolve o número de minutos que esta demora. Se a ligação não puder ser efetuada, a função devolve -1.

Nota: os locais A e B não têm de ser adjacentes no percurso do comboio. A única restrição é a de que o percurso do comboio deve passar primeiro em A e depois em B. Por exemplo, considerando o percurso armazenado no ficheiro do exemplo, as ligações Coimbra - Porto ou Lisboa - Aveiro são válidas, enquanto a ligação Aveiro - Coimbra é inválida.

4. Considere as seguintes definições:

```
typedef struct {int h, m;} tempo;

typedef struct Voo voo;
struct Voo{
    char codigo[6];          // código do voo

    char destino[4];         // código do destino
    int emTerra;             // sim = 1; não = 0;
    tempo t;                 // hora de partida
};
```

Para cada dia, o serviço de tráfego aéreo de um aeroporto, regista, num ficheiro binário, os dados dos voos que descolam desse local. O ficheiro é constituído, exclusivamente, por estruturas do tipo *voo*, uma para cada voo. O campo *emTerra* indica se o voo ainda está no aeroporto ou já partiu. No lado direito, pode-se ver um exemplo de um ficheiro com sete voos. As estruturas não estão ordenadas por nenhum critério em particular.

ST123
FNC
1
07 25
AP434
FAO
1
18 35
AP435
FNC
1
09 15
AP535
OPO
1
11 45
RA342
DUB
1
12 15
EJ262
BCN
1
10 25
RA343
DUB
1
11 15

Programação 2024/25

LEI, LEI-PL, LEI-CE

- a) Escreva uma função em C que mostre na consola a informação completa de todos os voos que ainda não partiram. A função recebe o nome do ficheiro binário como parâmetro.
- b) Escreva uma função em C que mostre na consola o código de todos os aeroportos de destino que surgem no ficheiro binário. Cada código só deve surgir uma vez, ou seja, neste exemplo DUB não deverá aparecer 2 vezes na listagem apresentada. A função recebe o nome do ficheiro binário como parâmetro.
- c) Escreva uma função em C que crie um array dinâmico, de estruturas do tipo *voo*, para onde deve copiar todos os voos que tenham como destino um determinado aeroporto. Considerando o ficheiro exemplificado, e assumindo que o aeroporto destino tem código DUB, deveria ser criado um array dinâmico com 2 estruturas (códigos de voo RA342 e RA343). A função recebe, como parâmetros, o nome do ficheiro a atualizar, o código do destino e o endereço de uma variável inteira, onde deve colocar a dimensão do array dinâmico. A função devolve, como resultado, o endereço inicial do array. Em caso de erro, a função devolve NULL.

Exercícios Complementares

5. Considere as seguintes definições:

```
#define TAM 50

typedef struct atg artigo;

struct atg{
    char nome[TAM];
    int peso;                // Em gramas
    int tempo_confecao;      // Tempo de confeção em minutos
    struct{
        int dia, mes, ano;
    }introducao;             //Data de introdução do produto na ementa
};
```

Uma pastelaria mantém informação sobre os produtos que oferece aos seus clientes num ficheiro binário. Dentro do ficheiro binário, a informação está armazenada em estruturas do tipo *artigo*. O ficheiro contém apenas estruturas deste tipo e não está ordenado por nenhum critério em particular.

Programação 2024/25

LEI, LEI-PL, LEI-CE

- a) Escreva uma função em C que contabilize quantos produtos podem ser confeccionados em menos de X minutos. A função recebe, como parâmetros, o nome do ficheiro e o valor X. Devolve o número de produtos que satisfaz a restrição.
- b) Escreva uma função em C que escreva na consola o nome e tempo de confeção de todos produtos que obedecem aos seguintes critérios:
- Foram introduzidos na ementa da pastelaria num determinado ano.
 - Têm um peso igual ou superior à média dos pesos de todos os produtos.

Na consola, a informação de cada produto deve surgir numa única linha. Os vários produtos surgem em linhas consecutivas. A seguir pode consultar um exemplo:

Bola de Berlim	45
Pastel de nata	35
...	

A função recebe, como parâmetros, o nome do ficheiro e o ano a considerar. Devolve 1 se tudo correr bem, ou 0, caso contrário.

6. Pretende-se preencher algumas posições de uma matriz de números reais de precisão simples (`float`). Os valores a utilizar para o preenchimento estão armazenados num ficheiro binário com o seguinte formato: sequências de dois números inteiros X, Y seguidos de um número real Z. Esta informação significa que o valor real Z deve ser utilizado para preencher o elemento da matriz na posição (X, Y). No ficheiro binário existem sucessivas sequências destes conjuntos de 3 números (tantas quantas as posições que devem ser inicializadas). Na figura pode ver um exemplo de um possível ficheiro binário com duas sequências de inicialização:

3
4
12.1
0
1
-2.3

Considerando este exemplo, deveriam ser preenchidas as duas posições da matriz:

- Atribuir ao elemento na linha 3, coluna 4 o valor 12.1
- Atribuir ao elemento na linha 0, coluna 1 o valor -2.3

Escreva uma função em C que efetue o preenchimento da matriz. A função recebe, como parâmetros, as dimensões e endereço inicial da matriz e o nome do ficheiro binário. A função devolve o número de posições da matriz que foram preenchidas com valores lidos do ficheiro. Todas as posições da matriz que não forem preenchidas com valores lidos do ficheiro devem ser inicializadas com o valor 0.

Programação 2024/25

LEI, LEI-PL, LEI-CE

Trabalho Autónomo

7. Os ficheiros binários armazenam de forma eficiente estruturas dinâmicas, permitindo guardar esta informação quando o programa não se encontra em execução ou quando estes dados não estão a ser utilizados.

Escreva 2 funções em C que permitam gerir o armazenamento da matriz dinâmica criada no exercício 5 da aula prática 10:

- i. `guardaMat()` ;
Recebe matriz de inteiros e guarda toda a informação relevante (dimensões e valores armazenados) num ficheiro binário.
- ii. `recuperaMat()` ;
Recupera a estrutura dinâmica a partir da informação armazenada no ficheiro binário. Esta função deve criar a estrutura dinâmica da matriz e preencher com os valores corretos.

Nota: ao resolver este exercício deve definir:

- i. A organização interna do ficheiro binário
- ii. Os detalhes de comunicação (parâmetros / valor devolvido) das funções `guardaMat()` e `recuperaMat()`.