

Licenciatura em Tecnologias e Sistemas de  
Informação  
2019/2020

**Relatório**

**Ficha de Trabalho Nº2**

Jorge Miguel dos Santos Martins a2019100813

26 de Março de 2020

## Índice

<b>1 INTRODUÇÃO.....</b>	<b>3</b>
<b>2 TRABALHO REALIZADO.....</b>	<b>4</b>
2.1 EXERCÍCIO 1 E 3.....	4
2.2 EXERCÍCIO 2 E 3.....	4
2.3 EXERCÍCIO 3.....	5
2.4 EXERCÍCIO 4.....	5
2.5 EXERCÍCIO 5.....	5
2.6 EXERCÍCIO 6.....	6
2.7 EXERCÍCIO 7.....	7
2.8 EXERCÍCIO 8.....	8
2.9 EXERCÍCIO 9.....	9
<b>3 CONCLUSÃO.....</b>	<b>10</b>
<b>1 REFERÊNCIAS.....</b>	<b>11</b>
<b>2 ANEXOS.....</b>	<b>12</b>

## 1 Introdução

Para a resolução desta ficha de trabalho é nos pedido a capacidade de poder manipular ficheiros, inicialmente é solicitado a escrita e leitura em ficheiros de texto e no final desta ficha de trabalho serão usados os ficheiros binários para a escrita e leitura de dados dos programas realizados.

Alguns do conhecimento utilizado neste trabalho já antecedia do semestre anterior, no caso da escrita e leitura de ficheiros em texto, mas foi inserido um novo conhecimento no que respeita ao uso de ficheiros binários para o mesmo efeito. De notar que foi um desafio devido ainda à diferença no seu uso, tal como na inexperience no manuseamento.

## 2 Trabalho realizado

### 2.1 Exercício 1 e 3

Primeiramente para a criação de um ficheiro com o nome que o utilizador deverá dar, usei uma string com o tamanho de 100, isto porque habitualmente os utilizadores não fazem ficheiros com nomes demasiados extensos.

Após na função de criar o ficheiro do utilizador é deixada uma marca da data ao qual foi registado esse ficheiro, através da função “ctime” da biblioteca “time.h”.

Na criação do ficheiro usei a opção “a” em vez de “w” isto para que se o ficheiro já existisse e o utilizador se enganar e usar um nome de um ficheiro já existente não perder dados que estariam gravados naquele ficheiro. Ou então para caso o utilizador apenas queira voltar a usar aquele ficheiro para continuar a escrever nele.

Para a escrita em si do ficheiro uso uma string com tamanho 5000 em que cada vez que o utilizador carrega “enter” os dados são gravados, em seguida é solicitado se quer continuar a escrever ou então sair. No caso de escolher para sair, será gravado no ficheiro com a função “ctime” o registo da data e hora de quando foi inserida aquela informação.

Na função seguinte serve para retirar a informação contida no ficheiro que o utilizador inseriu e colocar no monitor (stdout) do sistema.

### 2.2 Exercício 2 e 3

Este é um programa para contar as linhas de um ficheiro, número de palavras e número de caracteres. Também com a função de indicar o número de ocorrências de um caractere no ficheiro dado pelo utilizador.

Para contar as linhas usei a cada vez que o fgets retira uma string é contada uma linha, visto que uma linha não deverá ter mais de 100 palavras. Em seguida a string para onde foi colocada a linha retirada do documento é vista e a cada espaço é contabilizada uma palavra até que o ciclo chegue a ‘\0’. Usei também no “if” o “\n” para que este contabilize a ultima palavra da linha.

Para o tamanho de caracteres usados usei a função “strlen” que me dá o tamanho da string, e subtraí um porque o tamanho não é dado de 0 até ao seu final mas sim de 1 ao seu final. O tamanho da string usando esta expressão não contabiliza o ‘\0’.

Para encontrar o número de ocorrências de um caractere dentro ainda do ciclo “while” ao qual estou a fazer a leitura do ficheiro, uso ainda um ciclo que a cada string retirada para contagem do que já foi referenciado este retira o número de ocorrências do caractere dado pelo utilizador.

## 2.3 Exercício 3

Para o exercício 3 foi pedido basicamente para inserir nos exercícios 1 e 2 a biblioteca “time.h” e a quando fosse gravado em ficheiros a informação fosse também registado a hora, data ao qual isso foi realizado.

Para isso basta declarar a variável como “time\_t” e em seguida quando quisermos dar valores a essa variável teremos de inserir time(&variavel). Após a variável receber o valor escrevemos o seu valor no documento como string ao qual declaramos em seguida a variável como “ctime(&variavel).

```
time(&sec);
fprintf(keyOpen, "Escrito no monitor no dia %s \n\n", ctime(&sec));

fclose(keyOpen);
return;
```

*Exemplo de registo*

## 2.4 Exercício 4

Como é pedido no exercício é para criar ficheiros com números aleatórios, um de 1 a 100, outro de 1 a 500 e finalmente de 1 a 1000. Estes ficheiros estarão dispostos de três formas distintas, uma de forma aleatória, crescente, decrescente.

Primeiramente através de vetores de inteiros uso a função rand para criar os números aleatórios, ao qual a condiciono para que nenhum desses números seja 0. Seguidamente uso o algoritmo de inserção linear para os ordenar em ordem crescente. Como está em ordem crescente para o escrever em ordem decrescente basta usar um ciclo em que começo a escrever o vetor do fim para o início, assim poupo código e ciclos desnecessários.

## 2.5 Exercício 5

Neste exercício para retirar os números pedidos, usei ciclos a quando estava a fazer a leitura dos documentos de texto onde os dados estão guardados, assim condicionando a quantidade de inteiros que estava a retirar.

Enquanto o ponteiro “readFile” lê de um documento condicionado o ponteiro “penFile” escreve no ficheiro de nome “ficheiroTotal.txt”, isto é feito para todas as leituras condicionadas e escrita para este ficheiro.

```
void parImpar(){
    int aux, countImpar=0, countPar=0;
    FILE *readFile, *par, *impar;

    readFile=fopen("ficheiroTotal.txt", "r");
    par=fopen("pares.txt", "a");
    impar=fopen("impares.txt", "a");

    if(readFile!=NULL && par!=NULL && impar!=NULL){
        while((fscanf(readFile, "%d", &aux))!=EOF){
            if((aux%2)>0){
                fprintf(impar, "%d\n", aux);
                countImpar++; //aproveito enquanto estou a separar os impares e os pares nos ficheiros de os contar e assim se
            }
            else{
                fprintf(par, "%d\n", aux);
                countPar++;
            }
        }
        printf("No ficheiro total número de pares %d e número de impares %d \n\n", countPar, countImpar);
    }
}
```

### Exemplo de exercício 5

Como aqui podemos ver depois para separar para um ficheiro com os pares e outro para os ímpares, faço uma condição ao qual caso o resto da divisão por 2 seja maior que 0 então o número é ímpar, contrariamente será par. Aproveito que enquanto se faz a escrita no ficheiro posso também usar uma variável de contagem para dar quantos pares e ímpares existiam no “ficheiroTotal.txt”.

## 2.6 Exercício 6

Para este exercício usei a gravação em ficheiros e realizei uma função de “load” que ao iniciar o programa ele vai buscar a informação que necessita dos ficheiros, neste caso nome de alunos, números. Caso não existam ele cria os ficheiros para gravação. Ao ir buscar os dados no ficheiro são carregados nas suas arrays em memória e toda a manipulação é feita em memória, como a adição de alunos e sua remoção. Para o utilizador, listar, adicionar, remover, gravar basta inserir a sua opção num menu, ao qual também tem a opção de encerrar o programa.

Para adicionar alunos usei também a verificação “isalpha” para que o nome do aluno seja mesmo caracteres e não sejam números, esta função faz parte da biblioteca “ctype.h”.

Para a remoção do aluno, como cada aluno tem o seu número, em vez de pedir o nome e verificar pelo nome, peço o número, ao encontrar o número no array para o apagar basicamente copio o valor que está a sua direita. Entretanto ao valor que foi copiado nesse endereço dou o valor de -1, isto é feito sucessivamente até ao fim do vetor.

Para a matriz de Strings, primeiramente limpo o nome deixando aquela linha toda com ‘\0’ e seguidamente faço a mesma técnica que é copiar a linha em baixo.

## 2.7 Exercício 7

No exercício 7 é pedido para que sejam inseridos vários nomes num vetor e em seguida sejam estes guardados num ficheiro de texto. Usando as mesmas técnicas que nos exercícios anteriores, através de um “ponteiro” que é para ficheiros de nome FILE, usando o fopen para abrir o ficheiro e escrevendo nele com fprintf.

Neste exercício o desafio é o uso de ficheiros binários, são abertos e fechados da mesma forma que os ficheiros de texto mas as funções de escrita e leitura são diferentes. Nos binários usamos “fwrite” e “fread” para escrita e leitura nos ficheiros, além de que na abertura do ficheiro teremos de acrescentar “b” no seu modo de inicialização.

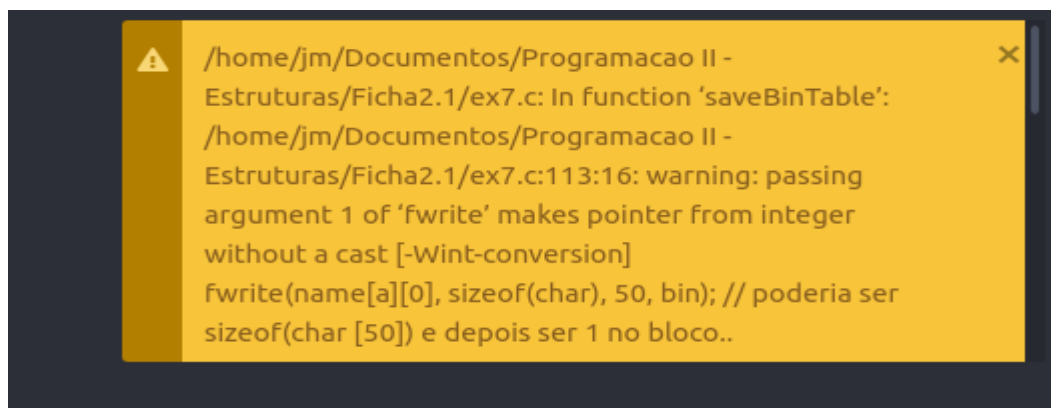
No uso de “fwrite” escrevemos primeiro a matriz de onde vem a informação, o seu tamanho, o número de blocos e o ponteiro que está para o ficheiro a escrever. No caso de “fread” o que muda é que no primeiro lugar será a variável que receberá os dados e não a que está a enviar como é no caso do “fwrite”, isto podendo ser um inteiro, float, char, vetor, array.

Para este exercício para a gravação de ficheiros binários de forma que seja escrita de linha a linha usei um ciclo que corre até chegar ao número da variável counter(usada para saber o número de nomes atual inserido pelo utilizador).

```
while(a<counter){
    fwrite(&name[a][0], sizeof(char), 50, bin); // poderia ser sizeof(char [50]) e depois ser 1 no bloco..
    a++;
}
rewind(bin);
a=0;
while(a<counter){
    fread(&aux2[a][0], sizeof(char), 50, bin);
    a++;
}
for(int i=0; i<counter; ++i){
    printf("string lido: %s\n",aux2[i]);
}
```

*Inserção linha a linha*

Sendo que estou na primeira posição do “fwrite” e “fread” indicando o local exato de onde será escrita a informação sou obrigado a usar o “&” que indica o endereço, pois caso não utilize é me dado o seguinte aviso.



*Aviso vetores sem "&"*

Sei que existem outras formas de poder escrever, mas aproveitei este exercício para explorar algumas formas de o poder fazer.

Para a gravação da tabela num todo usei a expressão na imagem abaixo.

```
fwrite(name, 50* sizeof(char[50]), 1, bin);
rewind(bin);
fread(aux3, 50* sizeof(char[50]), 1, bin);
fclose(bin);
```

*Gravação tabela na integra*

A grande diferença aqui é que indicamos o a matriz e ao dar o seu tamanho é 50 linhas em em seguida o “sizeof” indica que é do tipo “char” com tamanho de 50 caracteres, ao qual será escrito em um só bloco de informação binária.

Ao abrir os ficheiros podemos observar as algumas diferenças, quando gravamos de linha a linha ainda existem alguns caracteres que se conseguem ler, mas quando usamos esta ultima forma de guardar fica tudo completamente ilegível.

## 2.8 Exercício 8

Neste exercício é nos pedido que usemos o que foi feito no exercício 6 e nos dados que guardamos dos nomes dos alunos e dos números sejam em binários. Então apenas mudei a função “saveData” e “loadData” deste programa.

Usei a gravação da matriz logo toda em vez de seguir linha a linha, tal como o vetor de inteiros dos números de alunos também realizei a gravação de o vetor de uma só vez.

```
else{
    fwrite(memoryStudent, 50* sizeof(char[50]), 1, saveStudent);
}
fclose(saveStudent);

veNumber = fopen("SaveNumeros.dat", "wb"); //gravação de numeros

(saveNumber==NULL){
    printf("Nao foi possivel gravar Numeros\n");
    printf("Clique para continuar\n");
    getchar();
    return;
}

se{
    fwrite(memoryNumber, sizeof(int[50]), 1, saveNumber);
```

*Guardar em binário exercício 8*



Aqui já não usei o “&” de endereço pois para a matriz não é necessário, em seguida ao indicar o tamanho pelo que compreendi da indicação da professora, são 50 vezes o tamanho de “char” e este com tamanho de 50, isto gravado em um bloco. Anteriormente eu tentei inserir seguindo a referencia do site

[“https://www.gnu.org/software/libc/manual/pdf/libc.pdf?fbclid=IwAR3i7GPGYKXlJN5ysbnU1xBT\\_Dq5TaQay60CBefnrzNsS5M7J1XAP7cDgcE”](https://www.gnu.org/software/libc/manual/pdf/libc.pdf?fbclid=IwAR3i7GPGYKXlJN5ysbnU1xBT_Dq5TaQay60CBefnrzNsS5M7J1XAP7cDgcE) em que escrevi na parte do sizeof, sizeof(char) \*50 \*50, ao qual o programa corria mas temi que estivesse a interpretar mal o que tinha lido e tivesse a sair fora dos limites da memoria do programa. De referenciar que o que vi se encontra na pagina 311 deste manual.

Para o “fread” usei exatamente a mesma forma de lógica que deixei para a função “fwrite” já anteriormente demonstrada.

## 2.9 Exercício 9

Para este programa resolvi criar um menu ao qual o utilizador pode inserir manualmente as temperaturas, dando inicialmente a data, hora e a temperatura. Estas após serem dadas pelo utilizador são gravadas no ficheiro “temperaturas.dat”, são escritas na ordem data, hora e temperatura.

Então para o calcula da amplitude e da media peço ao utilizador que este me indique a data pelo qual devem ser feitas essas funções, com a função “while” faço a leitura do ficheiro, retirando a informação pela mesma ordem que foi guardada, só depois faço as condições para que sejam feitos os cálculos, condições como verificar se a data do utilizador condiz com a data retirada do documento, e assim sucessivamente.

Como se pode verificar na função “calcamp” para o calculo da amplitude eu uso uma condição para que após a data do utilizador coincidir com a data do documento e seja a primeira a ser encontrada então o mínimo e o máximo terão o exato mesmo valor, iniciando assim um ponto de partida tanto para o mínimo como para o máximo.

```
while((fread(&auxDate, sizeof(auxDate), 1, searchBin))!=0){ // usei assim a leitura da mesma forma como foi
    fread(&auxhour, sizeof(auxhour), 1, searchBin);
    fread(&auxtemp, sizeof(auxtemp), 1, searchBin);
    if(auxDate==date){ //encontrar a data que o utilizador quer para ver a aplitude
        if(counter==0){ //ponto de partida em que dou ao minimo e ao maximo o valor da primeira temperatura reco
            max=auxtemp;
            min=auxtemp;
        }
    }
}
```

*Inicialização do maximo e do minimo para a amplitude*

A lógica usada para ler os dados do ficheiro e condicionar para que sejam os dados corretos é igual na função para calcular a amplitude como para calcular a media.

### 3 Conclusão

Para este relatório diferenciando do que foi feito no anterior, escolhi apenas o que achei mais importante relatar no meu código, deixando também o meu código comentado onde achei essencial explicar as minhas decisões.

Penso que no que toca a manipulação de ficheiros binários ainda tenho muito que explorar as várias formas que podem ser escritos e lidos. É de todo referir que aprendi bastante a importância do uso de ficheiros binários, no que toca a sua precisão e segurança dos dados.

# 1 Referências

Listar consultas usadas

- [https://en.cppreference.com/w/c/io/fread?fbclid=IwAR31yVh6M2HEG41x5F6z6edn9mSnnWtioDsOFZb4HHGRJS-b\\_nItU6SNAWA](https://en.cppreference.com/w/c/io/fread?fbclid=IwAR31yVh6M2HEG41x5F6z6edn9mSnnWtioDsOFZb4HHGRJS-b_nItU6SNAWA)
- [https://www.gnu.org/software/libc/manual/pdf/libc.pdf?fbclid=IwAR3i7GPGYKXIJN5ysbnU1xBT\\_Dq5TaQay60CBEfnrzNsS5M7J1XAP7cDgcE](https://www.gnu.org/software/libc/manual/pdf/libc.pdf?fbclid=IwAR3i7GPGYKXIJN5ysbnU1xBT_Dq5TaQay60CBEfnrzNsS5M7J1XAP7cDgcE)

## 2 Anexos

Lista de ficheiros de código:

ex1\_ex3.c;

ex2\_ex3.c;

ex4.c;

ex5.c;

ex6.c;

ex7.c;

ex8.c;

ex9.c;