

Nome: Barbara Silveira Rodrigues

RA: 2021.1.08.042

Nome: Gustavo Fidelis Camilo

RA: 2021.1.08.043

AEDS I – Relatório do Trabalho Final

Objetivo:

O objetivo desse trabalho é desenvolver um sistema de gerenciamento de arquivos, onde o usuário poderá inserir, remover, adicionar e listar livros no sistema. O programa completo foi feito no replit.com e foi executado com êxito.

Desenvolvimento:

Para poder desenvolver o sistema, foram usadas três bibliotecas: stdio.h, stdlib.h e string.h. Foram criada 6 funções, sendo 5 delas as funções principais do sistema:

- **int buscar (char *titulo)**

```
{  
  
    FILE *arquivo;  
  
    arquivo = fopen("livros.txt", "r");  
  
    se (arquivo == NULL)  
  
    {  
  
        retorna -1;  
  
    }  
  
    char linha[100];  
  
    enquanto(fgets(linha, 100, arquivo) != NULL)  
  
    {  
  
        se(strstr(linha, titulo) != NULL)  
  
        {  
  
            escreva("%s", linha);  
  
            fclose(arquivo);  
  
            retorna 0;  
  
        }  
  
    }  
}
```

```

fclose(arquivo);

retorna -1;

}

```

Para fazer a função de busca, optamos por usar “strstr” para comparar as strings e se for achada a primeira ocorrência, retorna o valor int = 0 e devolve um ponteiro para a ocorrência encontrada e logo após ele imprime na tela o nome do livro buscado. Se não for encontrado, retorna o valor int = -1 e como não foi encontrada nenhuma ocorrência, strstr devolve um ponteiro nulo.

- **void remover(char *títulos)**

```

{

se(buscar(títulos) == -1)

{

    escreva(“titulo não existe”)

    retorna;

}

FILE *arquivo = fopen(“livros.txt”, “r”);

FILE *arquivo_temp = fopen(“livros_temp.txt”, “w”);

char linha[100];

enquanto(fgets(linha, 100, arquivo))

{

    se(strstr(linha, títulos) == NULL)

    {

        fprintf(arquivo_temp, “%s”, linha);

    }

}

fclose(arquivo);

fclose(arquivo_temp);

```

```

remove("livros.txt");

renomeia("livros_temp.txt", "livros.txt");

escreva("o titulo foi removido");

}

```

A função de remover irá escrever todos os livros (com exceção do removido) dentro de um arquivo temporário. Após todos os livros serem inseridos no arquivo temporário, o antigo arquivo será removido e o arquivo temporário será renomeado para assim ser o arquivo principal.

- **void inserir(char *titulo, char *autor, int num_paginas)**

```

{
    se(buscar(titulo) == -1)
    {
        FILE *arquivo;

        arquivo = fopen("livros.txt", "a");

        fprintf(arquivo, "%s - ", titulo);
        fprintf(arquivo, "%s - ", autor);
        fprintf(arquivo, "%d \n ", num_paginas);
        fclose(arquivo);
    }
    else
    {
        escreva("Titulo já existe");
    }
}

```

Para inserir, é usada a função feita anteriormente de buscar, de modo que o título digitado pelo usuário não possa ser colocado duplicadamente. Os inputs digitados pelo usuário fora da função são usados como parâmetro e a função irá colocar tais inputs dentro do arquivo criado. Nesse arquivo, todo livro adicionado será colocado no final do mesmo para que, apenas quando usada a função de listar, eles sejam impressos em ordem alfabética.

- **void listar()**

```
{  
  
    FILE *arquivo  
  
    arquivo = fopen("livros.txt", "r");  
  
    char linha[100];  
  
    char *titulo;  
  
    char *autor;  
  
    int num_paginas;  
  
    int i, a = 0;  
  
    i = contarlinha();  
  
    char dinTitulo[i][100];  
  
    char dinAutor[i][100];  
  
    int dinNum[i];  
  
    char aux[i][100];  
  
    char aux2[i][100];  
  
    int aux3;  
  
    int b, c;  
  
    enquanto(fgets(linha, 100, arquivo) != NULL)  
  
    {  
  
        titulo = strtok(linha, "-");  
  
        strcpy(dinTitulo[a], titulo);
```

```

autor = strtok(NULL, "-");

strcpy(dinAutor[a], autor);

num_paginas = atoi(strtok(NULL, "\n"));

dinNum[a] = num_paginas;

a++

}

fclose(arquivo);


//ordenação

para(b = 0; b < i - 1; b++){

    para(c = 0; c < i - 1 - b; c++){

        se(strcmp(dinTitulo[c], dinTitulo[c+1]) > 0)

        {

            strcpy(aux[c], dinTitulo[c+1]);

            strcpy(dinTitulo[c+1], dinTitulo[c]);

            strcpy(dinautor[c], aux2[c]);


            strcpy(aux2[c], dinAutor[c+1]);

            strcpy(dinAutor[c+1], dinAutor[c]);

            strcpy(dinAutor[c], aux2[c]);


            aux3[c] = dinNum[c+1];

            dinNum[c+1] = dinNum[c];

            dinNum[c] = aux3[c];

        }

    }

}

```

```

para(int l = 0; l < i; l++)
{
    escreva("Titulo: %s – Autor: %s – Numero de paginas: %d", dinTitulo[l],
dinAutor[l], dinNum[l];
}
}

```

Por fim, a função de listar percorre todo o arquivo e o lista em ordem alfabética com a ajuda de variáveis auxiliares e a alocação das mesmas. São usadas matrizes para a alocação e ordenação das variáveis de titulo de modo que também puxe o autor e numero de paginas correspondente ao titulo ordenado. A função faz o uso de strtok, que quebra a str em uma série de tokens. Também é usada a strcpy para copiar a string até o destino escolhido. Para fazer a ordenação, é necessário pegar o número de linhas e, para isso, foi criada a função contarlinha() para percorrer todo o arquivo, pois é possível saber quantas linhas há no arquivo e assim ler todas.

Fora das funções principais que o usuário, temos a seguinte função:

- **int contarlinha()**

```

{
    FILE *arquivo
    arquivo = fopen("livros.txt", "r");
    char linha[100];
    int i = 0;
    enquanto(fgets(linha, 100, arquivo) != NULL)
    {
        i++;
    }
    retorna i;
}

```

A função de contarlinha() foi feita para percorrer todo o arquivo e ler quantas linhas há, afim de que seja possível usar o dado da quantidade de linhas em outras funções, como na função de listar.

Conclusão:

O algoritmo, feito na linguagem C, aborda os conceitos de condições, repetições, alocação, manipulação de arquivos e funções. Durante o desenvolvimento, foi encontrada dificuldade em ordenar os títulos em ordem alfabética, no entanto, foi encontrada a solução fazendo a manipulação de matrizes junto com alocações.

Conclui-se que o programa obteve êxito, pois foi testado tanto no replit.com quanto no Visual Studio Code e todas as funções estão funcionando conforme foi pedido no documento do trabalho dado via moodle.