

Disciplina: Sistemas Operacionais Embarcados **Código:** 120961 **Turma:** A

Professor: Diogo Caetano Garcia

Aluno/Matrícula: Fábio Barbosa Pinto – 11/0116356

Questionário: 04_File_stdio

1) Crie um código em C para escrever "Ola mundo!" em um arquivo chamado 'ola_mundo.txt'.

Arquivo ola_mundo.txt:
Ola Mundo!

Arquivo ola_mundo.c:

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, const char * argv[]) {
    FILE *fp;
    char c;
    fp = fopen("ola_mundo.txt", "r"); /* Arquivo ASCII, para leitura */
    if(!fp)
    {
        printf( "Erro na abertura do arquivo");
        exit(-1); }
    while((c = getc(fp) ) != EOF) /* Enquanto não chegar ao final do arquivo */
        printf("%c", c); /* imprime o caracter lido */
    printf("\n");
    fclose(fp);
    return 0;
}
```

Comando terminal:

```
$ gcc ola_mundo.c -o ola_mundo
$ ./ola_mundo
$ Ola mundo!
```

2) Crie um código em C que pergunta ao usuário seu nome e sua idade, e escreve este conteúdo em um arquivo com o seu nome e extensão '.txt'. Por exemplo, considerando que o código criado recebeu o nome de 'ola_usuario_1':

```
$ ./ola_usuario_1
$ Digite o seu nome: Eu
$ Digite a sua idade: 30
$ cat Eu.txt
$ Nome: Eu
$ Idade: 30 anos
```

Arquivo ola_usuario_1.c:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(int argc, const char * argv[]) {
FILE *fp;
char nome[20];
char a[20];
char idade[5];

printf("Digite seu nome:");
scanf("%s",nome);
strcpy(a, nome);
printf("Digite sua idade:");
scanf("%s", idade);
strcat(nome, ".txt");
fp = fopen(nome, "w+");

if(!fp) printf("Erro de escrita");

fprintf(fp, "Nome: %s\n", a);
fprintf(fp, "Idade: %s\n", idade);

fclose(fp);
return 0;
```

3) Crie um código em C que recebe o nome do usuário e e sua idade como argumentos de entrada (usando as variáveis argc e *argv[]), e escreve este conteúdo em um arquivo com o seu nome e extensão '.txt'. Por exemplo, considerando que o código criado recebeu o nome de 'ola_usuario_2':

```
$ ./ola_usuario_2 Eu 30
$ cat Eu.txt
$ Nome: Eu
$ Idade: 30 anos
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(int argc, char * argv[]) {
FILE *fp;
char name[20];
strcpy(name, argv[1]);
strcat(name, ".txt");
fp = fopen(name, "w+");
```

```
if(!fp) printf("Erro de escrita");
fprintf(fp, "Nome: %s\n", argv[1]);
fprintf(fp, "Idade: %s\n", argv[2]);

fclose(fp);
return 0;
}
```

4. Crie uma função que retorna o tamanho de um arquivo, usando o seguinte protótipo: `int tam_arq_texto(char *nome_arquivo);` Salve esta função em um arquivo separado chamado 'bib_arqs.c'. Salve o protótipo em um arquivo chamado 'bib_arqs.h'. Compile 'bib_arqs.c' para gerar o objeto 'bib_arqs.o'.

```
$ gcc -c bib_arqs.c
int tam_arq_texto(char *nome_arquivo)
{
    FILE *fp;
    int fileLength = 0;
    char str[1000];
    fp = fopen(nome_arquivo, "r");
    if(!fp)
    {
        printf("Erro ao abrir arquivo!\n");
        return 0;
    }
    while(fgets(str, 1000, fp) != NULL)
        fileLength += strlen(str);
    fclose(fp);
    return fileLength;
}
```

5. Crie uma função que lê o conteúdo de um arquivo-texto e o guarda em uma string, usando o seguinte protótipo: `char* le_arq_texto(char *nome_arquivo);` Repare que o conteúdo do arquivo é armazenado em um vetor interno à função, e o endereço do vetor é retornado ao final. (Se você alocar este vetor dinamicamente, lembre-se de liberar a memória dele quando acabar o seu uso.) Salve esta função no mesmo arquivo da questão 4, chamado 'bib_arqs.c'. Salve o protótipo no arquivo 'bib_arqs.h'. Compile 'bib_arqs.c' novamente para gerar o objeto 'bib_arqs.o'.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(int argc, char * argv[]) {
    FILE *fp;
    char c;
    char a[1];
```

```
char text[100];
char arquivo[50];
printf("Nome do arquivo .txt: ");
scanf("%s", arquivo);
fp = fopen(arquivo,"r"); /* Arquivo ASCII, para leitura */
if(!fp)
{
    printf( "Erro na abertura do arquivo");
    exit(-1);
}
while((fgets(text, 100, fp)) != NULL)
printf("%s", text);
fclose(fp);
return 0;
}
```

6) Crie um código em C que copia a funcionalidade básica do comando cat: escrever o conteúdo de um arquivo-texto no terminal. Reaproveite as funções já criadas nas questões anteriores. Por exemplo, considerando que o código criado recebeu o nome de 'cat_falsificado':

```
$ echo Ola mundo cruel! Ola universo ingrato! > ola.txt
$ ./cat_falsificado ola.txt
$ Ola mundo cruel! Ola universo ingrato!
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "bib_arqs.h"

int main(int argc, char const *argv[])
{

    printf("%s\n", le_arq_texto(argv[1]));
    return 0;
}
```

7) Crie um código em C que conta a ocorrência de uma palavra-chave em um arquivo-texto, e escreve o resultado no terminal. Reaproveite as funções já criadas nas questões anteriores. Por exemplo, considerando que o código criado recebeu o nome de 'busca_e_conta':

```
$ echo Ola mundo cruel! Ola universo ingrato! > ola.txt
$ ./busca_e_conta Ola ola.txt
$ 'Ola' ocorre 2 vezes no arquivo 'ola.txt'.
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "bib_arqs.h"

int main(int argc, char const *argv[])
{
    char *str = le_arq_texto((char *) argv[2]);
    char *srch = (char *) argv[1];
    int len = strlen(srch);
    int count = 0;
    char *where = str;

    if (len)
    while ((where = strstr(where, srch))) {
        where += len;
        count++;
    }
    printf("%s' ocorre %d vezes no arquivo %s\n", argv[1],count,argv[2]);
    return 0;
}
```