

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

Professor: Diogo Caetano Garcia

Aluno/Matrícula: Fábio Barbosa Pinto – 11/0116356

Questionário: 05_File_POSIX_2

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

```
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>

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

    fp = open ("ola_mundo.txt", O_WRONLY | O_CREAT, S_IRWXU);
    if(fp==-1)
    {
        printf ("Erro na abertura do arquivo.\n");
        exit (1);
    }

    write(fp, "Ola Mundo!", 10);
    write(fp, "\n", 1);
    close(fp);

    return 0;
}
```

- 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':**

```
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>
```

```
int main()
{
    int fp;
    char nome[100];
    char idade[4];
```

```
char frase[200] = "Nome: ";
char idade_frase[200] = "Idade: ";
int i=0;

printf("Digite seu nome:");
gets(nome);
printf("Digite sua idade:");
gets(idade);
fp = open(nome, O_WRONLY | O_CREAT , S_IRWXU);
if(fp==-1)
{
    printf("Erro na abertura do arquivo.\n");
    exit(-1);
}
write (fp, "Nome: ", 6);
for (i=0;nome[i]; i++)
write (fp, &(nome[i]), 1);
write (fp, "\n", 1);
write (fp, "Idade: ", 7);
for (i=0;idade[i]; i++)
write (fp, &(idade[i]), 1);
write (fp, "\n", 1);
close(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':

```
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>

int main(int argc, char * argv[])
{
    int fp, i;

    fp = open(argv[1] , O_WRONLY | O_CREAT , S_IRWXU);
    if(fp==-1)
    {
        printf("Erro na abertura do arquivo.\n");
        exit(-1);
    }
}
```

```
    }  
    write (fp, "Nome: ", 6);  
    for (i=0;argv[1][i]; i++)  
        write (fp, &(argv[1][i]), 1);  
    write (fp, "\n", 1);  
    write (fp, "Idade: ", 7);  
    for (i=0;argv[2][i]; i++)  
        write (fp, &(argv[2][i]), 1);  
    write (fp, "\n", 1);  
    close(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)  
{  
    int fd;  
    int fileLength = 0;  
    char ch;  
    fd = open(nome_arquivo, O_RDONLY);  
    if(fd < 0)  
    {  
        printf("Erro ao abrir arquivo!\n");  
        return 0;  
    }  
    while(read(fd,&ch,1) > 0)  
        fileLength++;  
    close(fd);  
    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'.

```
char* le_arq_texto(char *nome_arquivo)  
{  
    int fd, fileLength = tam_arq_texto(nome_arquivo);
```

```
char * data;

fd = open(nome_arquivo, O_RDONLY);
if(fd < 0)
{
    printf("Erro ao abrir arquivo!\n");
    return 0;
}
data = malloc(sizeof(char) * fileLength);
if (read(fd, data, fileLength) < 0)
{
    printf("Erro ao ler arquivo!\n");
    return 0;
}
free(data);
return data;
}
```

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':

```
#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;
}

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

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':

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