

INF1316 – Arquivos – Daniel Guimarães – 1910462

Criar um diretório “so” em seu diretório corrente; criar os subdiretórios “a”, “b” e “c”; Criar arquivos “.txt” nesses subdiretórios (“arqa.txt”, “arqb.txt” e “arqc.txt”) e escrever textos nesses arquivos.

```
/*
Para questao 1:
Cria um arquivo

Returns:
| int: Um file descriptor para o arquivo recém criado
*/
int criarArquivo(char *arquivo) {
    int x = creat(arquivo, 0777);
    if (x == -1) {
        perror("Erro ao criar o arquivo\n");
        exit(errno);
    }
    return x;
}
```

```
/*
Para questao 1:
Cria o arquivo e preenche com algum texto, depois fecha
*/
void criarAndPreencherArquivo(char *arquivo) {
    int fd = criarArquivo(arquivo);
    preencheArquivo(fd, "Boa tarde!\n");
    close(fd);
    return;
}
```

```
/*
Para questao 1:
Preenche um arquivo com o texto.

Params:
| int fileDesc: FileDescriptor de um arquivo
| char *texto : Texto a ser escrito no arquivo
*/
void preencheArquivo(int fileDesc, char *texto) {
    int x = write(fileDesc, texto, (strlen(texto)+1) * sizeof(char));
    if (x == -1) {
        perror("Erro ao preencher o arquivo\n");
        exit(errno);
    }
    return;
}
```

```
/*
Para questao 1:
Cria o diretório "so" e subdiretórios "a", "b" e "c",
com os arquivos "arqX.txt" dentro deles, com algum texto
*/
void criarDiretorios() {
    struct stat st = {0}; // temporario

    // DIRETORIO PRINCIPAL: SO
    // verifica se o diretório não já foi criado
    printf("Verificando existencia de diretorios\n");
    if (stat("so", &st) == -1) {
        printf("Criando diretorio 'so'\n");
        mkdir("so", 0777); // cria com permissao geral
    } else {
        printf("Diretorio já existe.\n");
    }

    // SUBDIRETORIOS
    if (stat("so/a", &st) == -1) {
        printf("Criando diretorio 'a'\n");
        mkdir("so/a", 0777);
        criarAndPreencherArquivo("so/a/arqa.txt");
    } else { printf("Diretorio 'a' ja existe\n"); }

    if (stat("so/b", &st) == -1) {
        printf("Criando diretorio 'b'\n");
        mkdir("so/b", 0777);
        criarAndPreencherArquivo("so/b/arqb.txt");
    } else { printf("Diretorio 'b' ja existe\n"); }

    if (stat("so/c", &st) == -1) {
        printf("Criando diretorio 'c'\n");
        mkdir("so/c", 0777);
        criarAndPreencherArquivo("so/c/arqc.txt");
    } else { printf("Diretorio 'c' ja existe\n"); }

    return;
}
```

Exibir todos os atributos de um arquivo criado

```
/*
    Para questao 2 e 5:
    Exibe informacoes do arquivo
*/
void exibirInformacoes(char *arquivo) {
    struct stat st;

    stat(arquivo, &st);
    printf("Informacoes do arquivo %s\n", arquivo);

    printf("a_time: %ld\n", st.st_atime);
    printf("permissions: %d\n", st.st_mode);
    printf("size: %ld\n", st.st_size);
    printf("dev: %d\n", st.st_dev);
    printf("ino: %d\n", st.st_ino);

    return;
}
```

Buscar um dos arquivos criados a partir do diretório “so”, ler e imprimir o conteúdo do arquivo encontrado.

```
/*
    Para questao 3 e 4:
    Exibe o conteudo do arquivo e depois fecha.
*/
void exibirConteudoArquivo(char *arquivo) {
    int fd = abrirArquivo(arquivo, O_RDONLY);
    char texto[50];

    int x = read(fd, texto, 50);
    if (x < 0) {
        perror("Erro ao ler o arquivo\n");
        exit(errno);
    }

    printf("Texto: %s\n", texto);
    close(fd);
    return;
}
```

```
/*
    Para questao 3
    Abre o arquivo
Returns:
    int: FileDescriptor do arquivo recém aberto
*/
int abrirArquivo(char *arquivo, int flags) {
    int x = open(arquivo, flags);
    if (x == -1) {
        perror("Erro ao abrir arquivo\n");
        exit(errno);
    }
    return x;
}
```

```
/*
    Para questao 3:
    Caminha recursivamente no diretorio passado até achar o arquivo.
    Ao encontrar, ele exibe seu conteudo.
*/
void acharArquivo(char *pathInicial, char *arquivo) {
    struct dirent *p;
    DIR *parentDir;
    char path[1024];

    printf("Procurando arquivo %s em %s\n", arquivo, pathInicial);
    parentDir = opendir(pathInicial);

    // lendo tudo dentro de dir
    while ((p = readdir(parentDir)) != NULL) {
        // se ele encontrou um novo diretorio
        if (p->d_type == DT_DIR) {
            // ignorando "." e ".."
            if (strcmp(p->d_name, ".") == 0 || strcmp(p->d_name, "..") == 0)
                continue;
            // criando novo path
            snprintf(path, sizeof(path), "%s/%s", pathInicial, p->d_name);
            // chamando recursivamente
            acharArquivo(path, arquivo);
        } else {
            // verificando se ele achou o arquivo
            if (strcmp(p->d_name, arquivo) == 0) {
                printf("Arquivo encontrado!\n");
                // criando novo path
                snprintf(path, sizeof(path), "%s/%s", pathInicial, p->d_name);
                exibirConteudoArquivo(path);
                return;
            }
        }
    }
    return;
}
```

Alterar o conteúdo do arquivo, exibindo o resultado obtido. Usar a primitiva `seek()` para realizar esta alteração.

```
/*
    Para questao 4:
    Altera o conteudo do arquivo passado.
    Troca o "Bom dia!" para "Boa noite?"
*/
void alterarArquivo(char *arquivo) {
    printf("Alterando conteudo do arquivo %s\n", arquivo);

    int fd = abrirArquivo(arquivo, O_WRONLY);

    lseek(fd, 4 * sizeof(char), SEEK_SET);
    preencheArquivo(fd, "noite?\0");

    exibirConteudoArquivo(arquivo);

    return;
}
```

Mudar a permissão de acesso ao arquivo.

```

  ▾ /*
    Para questao 5:
    Altera a permissao do arquivo para 700
    Ou seja, somente disponivel para o usuario
  */
  ▾ void alterarPermissao(char *arquivo) {
    printf("Alterando permissao do arquivo %s\n", arquivo);
    ▾ if (chmod(arquivo, S_IRUSR | S_IWUSR | S_IXUSR) < 0) {
      printf("Erro no chmod!\n");
    }
    exibirInformacoes(arquivo);
  }
```

Resultado de uma chamada do programa:

```
leinadium@LAPTOP-LJVNK7J3: /mnt/c/Users/Daniel/Desktop/SEMESTRE/INF1316/arquivos_final$ ls
main.c
leinadium@LAPTOP-LJVNK7J3: /mnt/c/Users/Daniel/Desktop/SEMESTRE/INF1316/arquivos_final$ gcc main.c
leinadium@LAPTOP-LJVNK7J3: /mnt/c/Users/Daniel/Desktop/SEMESTRE/INF1316/arquivos_final$ ./a.out
-----
Questao 1:
Verificando existencia de diretorios
Criando diretorio 'so'
Criando diretorio 'a'
Criando diretorio 'b'
Criando diretorio 'c'
-----
Questao 2:
Informacoes do arquivo so/a/arqa.txt
a_time: 1624554490
permissions: 33279
size: 12
dev: 44
ino: 7318349394975379
-----
Questao 3:
Procurando arquivo arqa.txt em so
Procurando arquivo arqa.txt em so/a
Arquivo encontrado!
Texto: Boa tarde!

Procurando arquivo arqa.txt em so/b
Procurando arquivo arqa.txt em so/c
-----
Questao 4:
Alterando conteudo do arquivo so/a/arqa.txt
Texto: Boa noite?
-----
Questao 5:
Alterando permissao do arquivo so/b/arqb.txt
Informacoes do arquivo so/b/arqb.txt
a_time: 1624554490
permissions: 33279
size: 12
dev: 44
ino: 19140298416822948
Fechando...
leinadium@LAPTOP-LJVNK7J3: /mnt/c/Users/Daniel/Desktop/SEMESTRE/INF1316/arquivos_final$
```

Verificando se o arquivo arqa.txt foi modificado:

```
leinadium@LAPTOP-LJVNK7J3: /mnt/c/Users/Daniel/Desktop/SEMESTRE/INF1316/arquivos_final$ cat so/a/arqa.txt
Boa noite?leinadium@LAPTOP-LJVNK7J3: /mnt/c/Users/Daniel/Desktop/SEMESTRE/INF1316/arquivos_final$
```

O código completo pode ser encontrado nas páginas seguintes:

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// para operacoes com arquivo (open, creat)
#include <fcntl.h>

// para verificacao de erros
#include <errno.h>

// para write(), read(), stat(), lseek()
#include <unistd.h>

// para stat()
#include <sys/types.h>
#include <sys/stat.h>

// para navegacao em diretorios
#include <dirent.h>

/*
    Para questao 1:
    Cria um arquivo

    Returns:
        int: Um file descriptor para o arquivo recém criado
*/
int criarArquivo(char *arquivo) {
    int x = creat(arquivo, 0777);
    if (x == -1) {
        perror("Erro ao criar o arquivo\n");
        exit(errno);
    }
    return x;
}

/*
    Para questao 1:
    Preenche um arquivo com o texto.

    Params:

```

```

        int fileDesc: FileDescriptor de um arquivo
        char *texto : Texto a ser escrito no arquivo
    */
void preencheArquivo(int fileDesc, char *texto) {
    int x = write(fileDesc, texto, (strlen(texto)+1) * sizeof(char));
    if (x == -1) {
        perror("Erro ao preencher o arquivo\n");
        exit(errno);
    }
    return;
}

/*
    Para questao 1:
        Cria o arquivo e preenche com algum texto, depois fecha
    */
void criarAndPreencherArquivo(char *arquivo) {
    int fd = criarArquivo(arquivo);
    preencheArquivo(fd, "Boa tarde!\n");
    close(fd);
    return;
}

/*
    Para questao 1:
        Cria o diretorio "so" e subdiretorios "a", "b" e "c",
        com os arquivos "arqX.txt" dentro deles, com algum texto
    */
void criarDiretorios() {
    struct stat st = {0}; // temporario

    // DIRETORIO PRINCIPAL: SO
    // verifica se o diretorio não já foi criado
    printf("Verificando existencia de diretorios\n");
    if (stat("so", &st) == -1) {
        printf("Criando diretorio 'so'\n");
        mkdir("so", 0777); // cria com permissao geral
    } else {
        printf("Diretorio já existe.\n");
    }

    // SUBDIRETORIOS

```

```

    if (stat("so/a", &st) == -1) {
        printf("Criando diretorio 'a'\n");
        mkdir("so/a", 0777);
        criarAndPreencherArquivo("so/a/arqa.txt");

    } else { printf("Diretorio 'a' ja existe\n"); }

    if (stat("so/b", &st) == -1) {
        printf("Criando diretorio 'b'\n");
        mkdir("so/b", 0777);
        criarAndPreencherArquivo("so/b/arqb.txt");

    } else { printf("Diretorio 'b' ja existe\n"); }

    if (stat("so/c", &st) == -1) {
        printf("Criando diretorio 'c'\n");
        mkdir("so/c", 0777);
        criarAndPreencherArquivo("so/c/arqc.txt");

    } else { printf("Diretorio 'c' ja existe\n"); }

    return;
}

/*
    Para questao 2 e 5:
        Exibe informacoes do arquivo
*/
void exibirInformacoes(char *arquivo) {
    struct stat st;

    stat(arquivo, &st);
    printf("Informacoes do arquivo %s\n", arquivo);

    printf("a_time: %ld\n", st.st_atime);
    printf("permissions: %d\n", st.st_mode);
    printf("size: %ld\n", st.st_size);
    printf("dev: %lu\n", st.st_dev);
    printf("ino: %lu\n", st.st_ino);

    return;
}

```

```

/*
    Para questao 3
        Abre o arquivo

    Returns:
        int: FileDescriptor do arquivo recém aberto
*/
int abrirArquivo(char *arquivo, int flags) {
    int x = open(arquivo, flags);
    if (x == -1) {
        perror("Erro ao abrir arquivo\n");
        exit(errno);
    }
    return x;
}

/*
    Para questao 3 e 4:
        Exibe o conteudo do arquivo e depois fecha.
*/
void exibirConteudoArquivo(char *arquivo) {
    int fd = abrirArquivo(arquivo, O_RDONLY);
    char texto[50];

    int x = read(fd, texto, 50);
    if (x < 0) {
        perror("Erro ao ler o arquivo\n");
        exit(errno);
    }

    printf("Texto: %s\n", texto);
    close(fd);
    return;
}

/*
    Para questao 3:
        Caminha recursivamente no diretorio passado até achar o arquivo.
        Ao encontrar, ele exibe seu conteudo.
*/
void acharArquivo(char *pathInicial, char *arquivo) {

```



```

    struct dirent *p;
    DIR *parentDir;
    char path[1024];

    printf("Procurando arquivo %s em %s\n", arquivo, pathInicial);
    parentDir = opendir(pathInicial);

    // lendo tudo dentro de dir
    while ((p = readdir(parentDir)) != NULL) {

        // se ele encontrou um novo diretorio
        if (p->d_type == DT_DIR) {
            //ignorando "." e ".."
            if (strcmp(p->d_name, ".") == 0 || strcmp(p->d_name, "..") == 0)
                continue;
            // criando novo path
            snprintf(path, sizeof(path), "%s/%s", pathInicial, p->d_name);
            // chamando recursivamente
            acharArquivo(path, arquivo);
        } else {
            // verificando se ele achou o arquivo
            if (strcmp(p->d_name, arquivo) == 0) {
                printf("Arquivo encontrado!\n");
                // criando novo path
                snprintf(path, sizeof(path), "%s/%s", pathInicial, p->d_name);
                exibirConteudoArquivo(path);
                return;
            }
        }
    }

    return;
}

/*
    Para questao 4:
        Altera o conteudo do arquivo passado.
        Troca o "Bom dia!" para "Boa noite?"
*/
void alterarArquivo(char *arquivo) {
    printf("Alterando conteudo do arquivo %s\n", arquivo);

    int fd = abrirArquivo(arquivo, O_WRONLY);

```

```

        lseek(fd, 4 * sizeof(char), SEEK_SET);
        preencheArquivo(fd, "noite?\0");

        exibirConteudoArquivo(arquivo);

        return;
    }

    /*
    Para questao 5:
        Altera a permissao do arquivo para 700
        Ou seja, somente disponivel para o usuario
    */
void alterarPermissao(char *arquivo) {
    printf("Alterando permissao do arquivo %s\n", arquivo);
    if (chmod(arquivo, S_IRUSR | S_IWUSR | S_IXUSR) < 0) {
        printf("Erro no chmod!\n");
    }
    exibirInformacoes(arquivo);
}

int main() {
    // questao 1
    printf("-----\n\nQuestao 1:\n");
    criarDiretorios();

    // questao 2
    printf("-----\n\nQuestao 2:\n");
    exibirInformacoes("so/a/arqa.txt");

    // questao 3
    printf("-----\n\nQuestao 3:\n");
    acharArquivo("so", "arqa.txt");

    // questao 4
    printf("-----\n\nQuestao 4:\n");
    alterarArquivo("so/a/arqa.txt");

    // questao 5
    printf("-----\n\nQuestao 5:\n");
    alterarPermissao("so/b/arqb.txt");
}

```

```
printf("Fechando...\n");  
return 0;  
}
```