



Pontifícia Universidade Católica do Rio de Janeiro

Lab 8 - Arquivos e Diretórios

Sistemas Operacionais

Alunos	Tharik Lourenço Suemy Inagaki
Professor	Luiz Fernando Seibel

Rio de Janeiro, 23 de Junho de 2021

Conteúdo

1	Problemas	1
2	Código Fonte	1
3	Resultado da Execução	10
4	Comentários	11

1 Problemas

Utilizando um programa escrito em linguagem C:

1. Criar o 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 nestes arquivos.
2. Exibir todos os atributos de um dos arquivos criados.
3. Buscar um dos arquivos criados a partir do diretório “so”, ler e imprimir o conteúdo do arquivo encontrado.
4. Alterar o conteúdo do arquivo, exibindo o resultado obtido. Usar a primitiva seek() para realizar esta alteração.
5. Mudar a permissão de acesso ao arquivo.
6. Exibir via programa todas as alterações efetuadas. A saída do programa deve ser autoexplicativa (indicar operação a ser efetuada, os valores anteriores e seu resultado – valores posteriores à operação).

Entregar relatório padrão com código fonte e resultados em formato pdf

2 Código Fonte

```
/*  
Lab 8: Arquivos e Diretorios  
Sistemas Operacionais  
Nomes:  
Tharik Lourenco  
Suemy Inagaki  
*/  
#include <dirent.h>  
#include <limits.h>  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <unistd.h>  
#include <sys/types.h>  
#include <sys/stat.h>  
#include <errno.h>  
#include <fcntl.h>
```

```

void criaDiretorios(char *nome);
void criaArquivos(char *nome, char *conteudo);
void mostraAtributos(char *nome);
void encontraArquivo(char *caminho, char *nome);
void alteraArquivo(char *nome, char *conteudo);
void alteraPermissao(char *nome, char *permissao);

int main()
{
    printf("\nInicio do programa\n");
    //questao 1
    printf("\n----- Questao 1 ----- \n");
    printf("\n\nCriando os diretorios\n\n");
    criaDiretorios("so");
    criaDiretorios("so/a");
    criaDiretorios("so/b");
    criaDiretorios("so/c");

    printf("\n\nCriando os arquivos\n\n");
    criaArquivos("so/a/arqa.txt", "Um texto do arqa.txt");
    criaArquivos("so/b/arqb.txt", "Um texto do arqb.txt");
    criaArquivos("so/c/arqc.txt", "Um texto do arqc.txt");
    printf("\n\n");
    //questao 2
    printf("\n----- Questao 2 ----- \n");
    printf("Mostrando os atributos do arquivo arqa.txt\n\n");
    mostraAtributos("so/a/arqa.txt");
    printf("\n\n");
    //questao 3
    printf("\n----- Questao 3 ----- \n");
    char caminho[PATH_MAX + 1];
    if (getcwd(caminho, sizeof(caminho)) == NULL)
    {
        printf("O diretório atual nao pôde ser encontrado\n");
        exit(1);
    }
    printf("Buscando o arquivo arqa.txt...\n");
    encontraArquivo(caminho, "arqa.txt");
    printf("Fim da busca do arquivo\n\n");
    printf("\n\n");
}

```

```

//questao 4
printf("\n----- Questao 4 -----\\n");
alteraArquivo("so/a/arqa.txt", "abc");
encontraArquivo(caminho, "arqa.txt");
printf("\\n\\n");
//questao 5
printf("\\n----- Questao 5 -----\\n");
alteraPermissao("so/a/arqa.txt", "S_IRWXG|S_IRWXO");
printf("\\n\\n");
printf("Fim do programa\\n");
return 0;
}

//Questao 1
void criaDiretorios(char *nome)
{
    struct stat dadosDir;
    if (stat(nome, &dadosDir) == -1)
    {
        if (mkdir(nome, S_IRWXU | S_IRWXG | S_IRWXO) == 0)
        {
            printf("Diretorio %s criado com sucesso\\n", nome);
        }
        else
        {
            printf("Erro ao criar o diretorio %s\\n", nome);
        }
    }
    else
    {
        printf("O Diretorio %s ja existe\\n", nome);
    }
}

void criaArquivos(char *nome, char *conteudo)
{
    int arquivo;
    arquivo = open(nome, O_WRONLY | O_CREAT | O_SYNC, 0700);
    if (arquivo <= 0)
    {
        printf("Erro ao criar/abrir o arquivo %s\\n", nome);
    }
}

```

```

        exit(1);
    }

    if (write(arquivo, conteudo, strlen(conteudo) * sizeof(char)) < 0)
    {
        printf("Erro ao escrever no arquivo %s.\n", nome);
        exit(1);
    }
    printf("Arquivo %s criado com sucesso\n", nome);
}

//Questao 2
void mostraAtributos(char *nome)
{
    struct stat dadosArq;
    if (stat(nome, &dadosArq) == -1)
    {
        printf("O arquivo %s nao existe\n", nome);
        exit(1);
    }
    else
    {
        printf("Exibindo os atributos para o arquivo %s\n\n", nome);
        printf("\n***** Atributos gerais do arquivo *****\n");
        printf("st_dev: %d\n", dadosArq.st_dev);
        printf("st_ino: %lld\n", dadosArq.st_ino);
        printf("st_mode: %hu\n", dadosArq.st_mode);
        printf("st_nlink: %hu\n", dadosArq.st_nlink);
        printf("st_uid (ID Usuario): %u\n", dadosArq.st_uid);
        printf("st_gid (ID Grupo): %u\n", dadosArq.st_gid);
        printf("st_rdev: %d\n", dadosArq.st_rdev);

        printf("\n***** Atributos relacionados aos tamanhos *****\n");
        printf("st_size (Tamanho total): %lld bytes\n", dadosArq.st_size);
        printf("st_blksize: %d\n", dadosArq.st_blksize);
        printf("st_blocks (Numero de blocos de 512Bytes alocados): %lld\n", dadosArq.st_blocks);

        printf("\n***** Atributos relacionados aos ultimos acontecimentos *****\n");
        printf("st_atime (Ultimo acesso): %ld\n", dadosArq.st_atime);
        printf("st_mtime (Ultima modificacao): %ld\n", dadosArq.st_mtime);
        printf("st_ctime (Ultima mudanca de status): %ld\n", dadosArq.st_ctime);
    }
}

```

```

    }
}

```

//Questao 3

```

void encontraArquivo(char *caminho, char *nome)
{
    struct dirent *ent;
    struct stat dadosDiretorio;
    DIR *diretorio = opendir(caminho);
    if (diretorio == NULL)
    {
        printf("Erro ao abrir o diretorio %s\n", caminho);
        exit(1);
    }
    if (stat(caminho, &dadosDiretorio) == -1)
    {
        printf("O arquivo %s nao existe no diretorio %s\n", nome, caminho);
        exit(1);
    }
    ent = readdir(diretorio);
    //Busca o arquivo nos diretorios
    while (ent != NULL)
    {
        if (ent)
        {
            char caminhoArq[PATH_MAX + 1];
            strcpy(caminhoArq, caminho);
            strcat(caminhoArq, "/");
            strcat(caminhoArq, ent->d_name);
            if (stat(caminhoArq, &dadosDiretorio) == -1)
            {
                printf("O arquivo %s nao existe no diretorio %s\n", nome, caminhoArq);
                exit(1);
            }
            if (S_ISDIR(dadosDiretorio.st_mode) &&
                strcmp(ent->d_name, ".") != 0 &&
                strcmp(ent->d_name, "..") != 0)
            {
                encontraArquivo(caminhoArq, nome);
            }
        }
    }
}

```

```

    }
    else
    {
        //Encontrou o arquivo
        if (strcmp(ent->d_name, nome) == 0)
        {

            printf("Arquivo encontrado em: %s\n", caminhoArq);
            char r;
            int arq = open(caminhoArq, O_RDONLY | O_CREAT | O_SYNC, 0700);

            if (arq <= 0)
            {
                printf("Erro ao abrir arquivo\n");
                exit(1);
            }
            //Exibe conteudo do arquivo
            printf("Conteudo do arquivo: ");
            while (read(arq, &r, 1) > 0)
            {
                printf("%c", r);
            }
            printf("\n");
        }
    }
    ent = readdir(diretorio);
}
}

```

//Questao 4

```

void alteraArquivo(char *nome, char *conteudo)
{
    printf("Arquivo a ser alterado: %s\n", nome);

    FILE *arquivo = fopen(nome, "r");
    char *buffer;
    int q = 0;

    if (arquivo == NULL)

```



```

{
    printf("Erro ao abrir arquivo\n");
    exit(1);
}
char c = getc(arquivo);
printf("Conteudo do arquivo: %c", c);
while (c != EOF)
{
    q++;
    c = getc(arquivo);
    printf("%c", c);
}
printf("\n");
buffer = (char *)malloc((q + 1 + strlen(conteudo)) * sizeof(char));
if (buffer == NULL)
{
    printf("Erro de alocao do buffer\n");
    exit(1);
}
printf("Alteracao a ser efetuada: %s\n", conteudo);
fclose(arquivo);
arquivo = fopen(nome, "r");
if (arquivo == NULL)
{
    printf("Erro ao abrir arquivo\n");
    exit(1);
}

int i = 0;
c = getc(arquivo);
//copia o conteudo do arquivo pro buffer
while (c != EOF)
{
    buffer[i] = c;
    i++;
    c = getc(arquivo);
}
int j;
//copia o texto da alteracao pro buffer
for (j = 0; conteudo[j] != '\0'; j++)
{

```

```

        buffer[i] = conteudo[j];
        i++;
    }
    buffer[i] = '\0';
    fclose(arquivo);
    //abre o arquivo para escrever
    arquivo = fopen(nome, "w");

    if (arquivo == NULL)
    {
        printf("Erro ao abrir arquivo\n");
        exit(1);
    }
    //copia o conteudo pro arquivo
    fputs(buffer, arquivo);
    if (fseek(arquivo, strtol(conteudo, NULL, 10), SEEK_SET) < 0)
    {
        printf("Falha no fseek\n");
        exit(1);
    }
    if (fputs(conteudo, arquivo) < 0)
    {
        printf("Erro ao alterar o arquivo\n");
        exit(1);
    }
    fclose(arquivo);
    printf("Alteracao efetuada com sucesso\n");
}

//Questao 5
void alteraPermissao(char *nome, char *permissao)
{
    struct stat dadosArq;
    mode_t permissoes[7] = {
        S_IRWXU,
        S_IRWXG,
        S_IRWXO,
        S_IRWXU | S_IRWXG,
        S_IRWXU | S_IRWXO,
        S_IRWXG | S_IRWXO,
        S_IRWXO | S_IRWXG | S_IRWXU,
    }

```

```

};
int indice;
if (stat(nome, &dadosArq) == -1)
{
    printf("O Arquivo nao existe\n");
    exit(1);
}
printf("Permissao atual: %x\n", dadosArq.st_mode);
printf("Permissao nova: %s\n", permissao);
printf("Alterando a permissao\n");

if (!strcmp("S_IRWXU", permissao))
{
    indice = 0;
}
else if (!strcmp("S_IRWXG", permissao))
{
    indice = 1;
}
else if (!strcmp("S_IRWXO", permissao))
{
    indice = 2;
}
else if (!strcmp("S_IRWXU|S_IRWXG", permissao) ||
        !strcmp("S_IRWXG|S_IRWXU", permissao))
{
    indice = 3;
}
else if (!strcmp("S_IRWXU|S_IRWXO", permissao) ||
        !strcmp("S_IRWXG|S_IRWXU", permissao))
{
    indice = 4;
}
else if (!strcmp("S_IRWXG|S_IRWXO", permissao) ||
        !strcmp("S_IRWXO|S_IRWXG", permissao))
{
    indice = 5;
}
else if (!strcmp("S_IRWXO|S_IRWXG|S_IRWXU", permissao) ||
        !strcmp("S_IRWXG|S_IRWXO|S_IRWXU", permissao) ||
        !strcmp("S_IRWXO|S_IRWXU|S_IRWXG", permissao) ||

```

```

        !strcmp("S_IRWXU|S_IRWXO|S_IRWXG", permissao) ||
        !strcmp("S_IRWXU|S_IRWXG|S_IRWXO", permissao) ||
        !strcmp("S_IRWXG|S_IRWXU|S_IRWXO", permissao))
    {
        indice = 6;
    }
    else
    {
        printf("Essa permissao nao existe\n");
        exit(1);
    }
    chmod(nome, permissao[indice]);
    printf("Permissao depois da alteracao: %x\n", dadosArq.st_mode);
}

```

3 Resultado da Execução

```

Início do programa

----- Questao 1 -----

Criando os diretorios

Diretorio so criado com sucesso
Diretorio so/a criado com sucesso
Diretorio so/b criado com sucesso
Diretorio so/c criado com sucesso

Criando os arquivos

Arquivo so/a/arqa.txt criado com sucesso
Arquivo so/b/arqb.txt criado com sucesso
Arquivo so/c/arqc.txt criado com sucesso

----- Questao 2 -----
Mostrando os atributos do arquivo arqa.txt

Exibindo os atributos para o arquivo so/a/arqa.txt

***** Atributos gerais do arquivo *****
st_dev: 16777224
st_ino: 13978189
st_mode: 33216
st_nlink: 1
st_uid (ID Usuario): 501
st_gid (ID Grupo): 20
st_rdev: 0

```

```

**** Atributos relacionados aos tamanhos ****
st_size (Tamanho total): 20 bytes
st_blksize: 4096
st_blocks (Numero de blocos de 512Bytes alocados): 8

**** Atributos relacionados aos ultimos acontecimentos ****
st_atime (Ultimo acesso): 1624467715
st_mtime (Ultima modificacao): 1624467715
st_ctime (Ultima mudanca de status): 1624467715

----- Questao 3 -----
Buscando o arquivo arqa.txt...
Arquivo encontrado em: /Users/suemyinagaki/Desktop/Lab 8/so/a/arqa.txt
Conteudo do arquivo: Um texto do arqa.txt
Fim da busca do arquivo

----- Questao 4 -----
Arquivo a ser alterado: so/a/arqa.txt
Conteudo do arquivo: Um texto do arqa. t x t
Alteracao a ser efetuada: abc
Alteracao efetuada com sucesso
Arquivo encontrado em: /Users/suemyinagaki/Desktop/Lab 8/so/a/arqa.txt
Conteudo do arquivo: abctexto do arqa.txtabc

----- Questao 5 -----
Permissao atual: 81c0
Permissao nova: S_IRWXG|S_IRWXO
Alterando a permissao
Permissao depois da alteracao: 81c0

Fim do programa

```

4 Comentários

A primeira funcao recebe como parametro o caminho do diretorio a ser criado e, caso o diretorio já exista, exibe uma mensagem, caso contrário, cria o diretório e avisa caso dê tudo certo.

A segunda funcao recebe como parametros o nome do arquivo a ser criado e o conteudo a ser escrito no arquivo.

A terceira funcao, mostraAtributos, recebe o nome de um arquivo e exibe os atributos dele.

A quarta funcao recebe um caminho pro começo da busca e, a partir dele, passa a procurar o arquivo com o nome que foi passado como parametro tambem. Caso encontre o arquivo, exibe o conteudo do mesmo.

A quinta funcao faz uma alteracao em um arquivo. Recebe como parametro o nome do arquivo e o conteudo que deve ser sobrescrito no arquivo.

A ultima questao altera a permissao de um dado arquivo. Recebe como parametro a string reelacionada a nova permissao e o nome do arquivo. Caso a nova permissao exista, ela realiza a alteracao. Caso contrario exhibe uma mensagem.