

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de
Computação



Laboratório de Software Básico (EA872)

Relatório 5 Sistema de Arquivos

Aluna: Gabriela de Castro Surita
gabsurita@gmail.com

Professor: Christian Rodolfo Esteve Rothenberg

1 Exercícios

1.1 (d) (1,0) Duplicação de descritores

(d.1) Após a compilação, execute o programa com o comando ./d teste.d (o arquivo teste.d é fornecido) e documente a saída.

```
$ ./d teste.d
1a. Leitura:
0123456789
0123456789
0123456789
0123456789
01
2a. Leitura:
23456789
0123456789
0123456789
0123456789
0123
3a. Leitura:
456789
0123456789
0123456789
0123456789
012345
```

(d.2) Explique a saída do programa a partir da função desempenhada pelas variáveis i e j.

O comando abre um arquivo por meio de um descritor i, cria uma cópia j do descritor (como alias) e faz 3 leituras de 50 bytes do arquivo contendo linhas da forma 0123456789M\$, sendo a primeira com o descritor i e as demais com o descritor j. Como o descritor j é um alias para o descritor i, o ponto em que i termina sua leitura é onde j começa a sua.

1.2 (e) (1,0) Redireção de arquivos

(e.1) Após a compilação, execute o programa com o comando ./e teste1.e ls -s e documente a saída.

```
$ ./e teste1.e ls -s
```

```
[castro@localhost arqlab5]$ cat teste1.e
total 160
12 a
  4 a.c
  4 a.txt
12 b
12 b1
  4 b1.c
12 b2
  4 b2.c
12 c
  4 c.c
12 d
  4 d.c
12 e
  4 e.c
  4 g.c
  4 h.c
  0 teste1.e
  4 teste.a
  4 teste.b
  8 testec1.txt
20 testec2.txt
  4 teste.d
```

(e.2) Explique o que faz o comando executado.

O comando cria um arquivo com o nome passado como primeiro parâmetro e redireciona para ele saída padrão de um programa que é passado a seguir, no caso, a listagem dos arquivos e o número de blocos ocupados por cada um.

(e.3) Em seguida, execute o comando `ls -s & teste2.e` e documente a saída.

```
$ cat teste2.e
total 164
12 a
  4 a.c
  4 a.txt
12 b
12 b1
  4 b1.c
12 b2
  4 b2.c
```

```
12 c
  4 c.c
12 d
  4 d.c
12 e
  4 e.c
  4 g.c
  4 h.c
  4 teste1.e
  0 teste2.e
  4 teste.a
  4 teste.b
  8 testec1.txt
20 testec2.txt
  4 teste.d
```

(e.4) Explique o que faz o comando executado.

O comando executa a listagem de diretório e encaminha para um arquivo de saída.

(e.5) Compare os conteúdos dos arquivos teste1.e e teste2.e e explique suas diferenças e o funcionamento do programa.

No programa e.c, o arquivo de saída é criado antes de executar o comando de listagem, e portanto o próprio arquivo é apresentado na lista também.

```
$ diff teste1.e teste2.e
1c1
< total 160
---
> total 164
18c18,19
<  0 teste1.e
---
>  4 teste1.e
>  0 teste2.e
```

1.3 (f) (5,0) Projeto Servidor WEB

Código fonte

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <fcntl.h>

#define NOT_FOUND 404
#define FORBIDDEN 403
#define SERVER_ERROR 500

int get_resource(char* webspace_path, char* resource_path){

    char* full_path = (char*)malloc((strlen(webspace_path)
                                     +strlen(resource_path)+1)*sizeof(char));

    // 1. Concatena caminho
    sprintf(full_path, "%s%s", webspace_path, resource_path);

    struct stat path_status;

    // 2. Busca estado do caminho
    if(stat(full_path, &path_status) < 0)
        return NOT_FOUND;

    // 3. Checa permissao de leitura
    if(!(path_status.st_mode & S_IRUSR)){
        return FORBIDDEN;
    }

    // 4.1 Caminho e um arquivo
    if(S_ISREG(path_status.st_mode)){
        int file_desc;

        // 4.1.1 Abre arquivo com open()
        if((file_desc = open(full_path, O_RDONLY)) == -1){
            return SERVER_ERROR;
        }

        char buf[2048];
        int n;

        // 4.1.2 Escreve na stdout

```

```

    while((n = read(file_desc,buf,sizeof(buf))) != 0){
        fflush(stdout);
        write(1,buf,n);
    }
}
// 4.2 Caminho e um diretorio
else{

    // 4.2.1 Verifica se diretorio permite varredura
    if(!(path_status.st_mode & S_IXUSR)){
        return FORBIDDEN;
    }

    // Monta caminho do index.html
    char* index_path = (char*)malloc((strlen(full_path)
        +strlen("index.html")+1)*sizeof(char));

    // Monta caminho do welcome.html
    char* welcome_path = (char*)malloc((strlen(full_path)
        +strlen("welcome.html")+1)*sizeof(char));

    sprintf(index_path, "%sindex.html", full_path);
    sprintf(welcome_path, "%swelcome.html", full_path);

    struct stat path_status;

    // 4.2.2 Verifica se os arquivos existem
    if(stat(index_path, &path_status) < 0){
        if(stat(welcome_path, &path_status) < 0)
            return NOT_FOUND;
    }

    int file_desc;

    // 4.2.3 Tenta ler index.html
    if((file_desc = open(index_path, O_RDONLY)) == -1){

        // Se nao existe index.html, verifica welcome.html
        if((file_desc = open(welcome_path, O_RDONLY)) == -1)
            return FORBIDDEN;
    }

    char buf[2048];
    int n;

```

```

// 4.2.4 Imprime o conteudo
while((n = read(file_desc,buf,sizeof(buf))) != 0){
    fflush(stdout);
    write(1,buf,n);
}

free(index_path);
free(welcome_path);
}

free(full_path);
return 0;
}

int main(int argc, char** argv){

    printf("Status_%.d\n", get_resource(argv[1], argv[2]));
    return 0;
}

```

Ambiente de testes

```

$ tree test_webpace/
test_webpace/
  index.html
  readable_dir
    welcome.html
  readable_dir2
    index.html
  readable_dir3
    welcome.html
  readable_file.html
  unreadable_dir
  unreadable_file.html

4 directories, 6 files

$ ls -lR test_webpace/
test_webpace/:
total 28
-r--r--r-- 1 castro castro   31 Sep 29 17:44 index.html

```

```
drwxrwxrwx 2 castro castro 4096 Sep 29 17:26 readable_dir
drwxrwxr-x 2 castro castro 4096 Sep 29 17:43 readable_dir2
drwxrwxr-x 2 castro castro 4096 Sep 29 17:47 readable_dir3
-r--r--r-- 1 castro castro 31 Sep 29 17:12 readable_file.html
dr--r--r-- 2 castro castro 4096 Sep 29 17:13 unreadable_dir
----- 1 castro castro 31 Sep 29 17:12 unreadable_file.html
```

test_webospace/readable_dir:

total 4

```
-r--r--r-- 1 castro castro 31 Sep 29 17:11 welcome.html
```

test_webospace/readable_dir2:

total 4

```
----- 1 castro castro 31 Sep 29 17:10 index.html
```

test_webospace/readable_dir3:

total 4

```
----- 1 castro castro 31 Sep 29 17:46 welcome.html
```

test_webospace/unreadable_dir:

ls: cannot access test_webospace/unreadable_dir/welcome.html: Permission denied

total 0

```
-????????? ? ? ? ? ? welcome.html
```

Casos de testes

```
echo "-----"
```

```
echo " _TESTES_DE_ACESSO"
```

```
echo "-----"
```

```
echo ""
```

```
echo "Teste_1:_acesso_em_arquivo_legivel"
```

```
./webospace test_webospace/ readable_file.html
```

```
echo ""
```

```
echo "Teste_2:_acesso_em_diretorio_com_index.html"
```

```
./webospace test_webospace/ /
```

```
echo ""
```

```
echo "Teste_3:_acesso_em_diretorio_com_welcome.html"
```

```
./webospace test_webospace/ readable_dir/
```

```
echo ""
```

```
echo "Teste_8:_acesso_em_caminho_inexistente"
```



```

./webpace test_webpace/ index2.html
echo ""

echo "-----"
echo "__TESTES_DE_PERMISSAO"
echo "-----"
echo ""

echo "Teste_4:_acesso_em_arquivo_ilegivel"
./webpace test_webpace/ unreadable_file.html
echo ""

echo "Teste_5:_acesso_em_diretorio_sem_permissao_de_execucao"
./webpace test_webpace/ unreadable_dir/
echo ""

echo "Teste_6:_acesso_em_diretorio_com_index.html_ilegivel"
./webpace test_webpace/ readable_dir2/
echo ""

echo "Teste_7:_acesso_em_diretorio_com_welcome.html_ilegivel"
./webpace test_webpace/ readable_dir3/
echo ""

```

Saída dos testes

```

$ sh tests.sh
-----
TESTES DE ACESSO
-----

Teste 1: acesso em arquivo legivel
<!DOCTYPE HTML>
<html>
</html>
Status 0

Teste 2: acesso em diretorio com index.html
<!DOCTYPE HTML>
<html>
</html>
Status 0

Teste 3: acesso em diretorio com welcome.html

```

```
<!DOCTYPE HTML>
<html>
</html>
Status 0
```

Teste 8: acesso em caminho inexistente
Status 404

TESTES DE PERMISSAO

Teste 4: acesso em arquivo ilegivel
Status 403

Teste 5: acesso em diretorio sem permissao de execucao
Status 403

Teste 6: acesso em diretorio com index.html ilegivel
Status 403

Teste 7: acesso em diretorio com welcome.html ilegivel
Status 403