

# Relatório 01

Vinícius de Oliveira Peixoto Rodrigues (245294)

Agosto de 2022

## Questão (d)

### Item 1

A saída do programa é:

```
1a. Leitura:
9876543210
9876543210
9876543210
9876543210
98
2a. Leitura:
76543210
9876543210
9876543210
9876543210
9876
3a. Leitura:
543210
9876543210
9876543210
9876543210
987654
```

### Item 2

A variável `i` é um file descriptor para o arquivo indicado pelo `argv[1]` (primeiro argumento do programa), no caso o `teste.d`; a variável `j` é um file descriptor obtido a partir da syscall `dup(int oldfd)`, de modo que `i` e `j` compartilham do mesmo status e file offset. Por isso, quando são lidos 50 bytes pelo fd `i`, na próxima chamada do `read` (onde é usado o fd `j`), é feita a leitura a partir de onde se parou no `i` (visto que ambos compartilham o mesmo file offset).

## Questão (e)

### Itens 1 e 2

```
> ./e teste1.e ls -s
> cat teste1.e
total 2108
 16 a
  4 a.c
 16 b1
  4 b1.c
 16 b2
  4 b2.c
 16 c
  4 c.c
 16 d
  4 d.c
 16 e
  4 e.c
  4 f.c
  4 g.c
  4 h.c
  4 teste.a
  4 teste.b
  4 teste.d
  0 teste1.e
  8 testec1.txt
1956 testec2.txt
```

O programa cria um arquivo com o nome dado em `argv[1]` (primeiro argumento do programa); em seguida, duplica o file descriptor e atribui o número de `fd 1` (equivalente ao `stdout`). Em seguida, fecha o descritor original e executa a syscall `execvp(char *file, char *argv[])`, passando como `file` o `argv[2]` (`ls`) e como argumentos o `&argv[2]` (`["ls", "-s"]`).

Essencialmente, o programa faz uma redireção da saída (redireciona o output de `argv[2]` para o arquivo `argv[1]`).

### Itens 3, 4 e 5

```
> ls -s > teste2.e
> cat teste2.e
total 2112
 16 a
  4 a.c
 16 b1
  4 b1.c
```

```

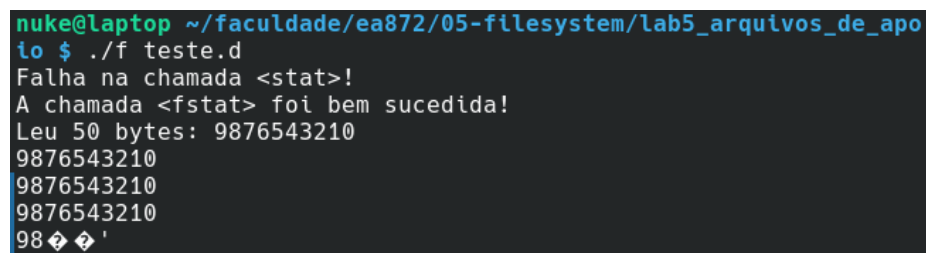
16 b2
  4 b2.c
16 c
  4 c.c
16 d
  4 d.c
16 e
  4 e.c
  4 f.c
  4 g.c
  4 h.c
  4 teste.a
  4 teste.b
  4 teste.d
  4 teste1.e
  0 teste2.e
  8 testec1.txt
1956 testec2.txt

```

O comando acima faz a mesma coisa: redireciona o output do comando "ls -s" para um arquivo (no caso `teste2.e`). Ele funciona da mesma forma que o programa `e.c`.

## Questão (f)

### Item 1



```

nuke@laptop ~/faculdade/ea872/05-filesystem/lab5_arquivos_de_apo
io $ ./f teste.d
Falha na chamada <stat>!
A chamada <fstat> foi bem sucedida!
Leu 50 bytes: 9876543210
9876543210
9876543210
9876543210
98 '

```

### Item 2

Porque o hard link `tmp` havia sido excluído por meio da chamada `unlink` (de modo que foi excluído o hard link em si, mas não o arquivo para o qual ele apontava).

A chamada `fstat` foi feita usando o número do file descriptor do arquivo (`argv[1]`) em si, obtido enquanto o hard link `tmp` ainda estava vivo.

### Item 3

O programa falhará na chamada `open("tmp", ...)`, visto que não há um arquivo (ou hard link) com esse nome.

### Item 4

A syscall `read` não coloca um null terminator (`'\0'`) após copiar o conteúdo do buffer, de modo que o `printf` vai ler lixo até encontrar um byte zero na pilha do programa. Para evitar isso, a forma mais simples seria escrever um `'\0'` no final do buffer:

```
buf[i] = 0;
```

## Questão (g)

### Itens 1 e 2

```
nuke@laptop ~/faculdade/ea872/05-filesystem/lab5_arquivos_de_apolo $ ./g $(pwd)
/home/nuke/faculdade/ea872/05-filesystem/lab5_arquivos_de_apolo e' um arquivo-diretorio
user-id 1001, group-id 1001, permissao 40775, link(s) 2
tamanho 4096 (bytes), 8 (blocos), Inode # 4981325
Dados de /home/nuke/faculdade/ea872/05-filesystem/lab5_arquivos_de_apolo foram modificados
pela ultima vez em Tue Sep 27 21:58:33 2022
Leitura permitida
Escrita permitida
Execucao permitida
```

```
nuke@laptop ~/faculdade/ea872/05-filesystem/lab5_arquivos_de_apolo $ ./g $(pwd)/teste.d
/home/nuke/faculdade/ea872/05-filesystem/lab5_arquivos_de_apolo/teste.d e' um arquivo regular
user-id 1001, group-id 1001, permissao 100644, link(s) 1
tamanho 226 (bytes), 8 (blocos), Inode # 4981336
Dados de /home/nuke/faculdade/ea872/05-filesystem/lab5_arquivos_de_apolo/teste.d foram modificados
pela ultima vez em Fri Mar 14 12:24:06 2003
Leitura permitida
Escrita permitida
Execucao proibida
```

```
nuke@laptop ~/faculdade/ea872/05-filesystem/lab5_arquivos_de_apolo $ ./g /dev/null
/dev/null e' um arquivo especial (character)
user-id 0, group-id 0, permissao 20666, link(s) 1
tamanho 0 (bytes), 0 (blocos), Inode # 5
Dados de /dev/null foram modificados pela ultima vez em Fri Sep 23 10:45:46 2022
Leitura permitida
Escrita permitida
Execucao proibida
```

O programa usa a syscall **stat** para obter os atributos de um arquivo e os imprime no console.

## Questão (h)