

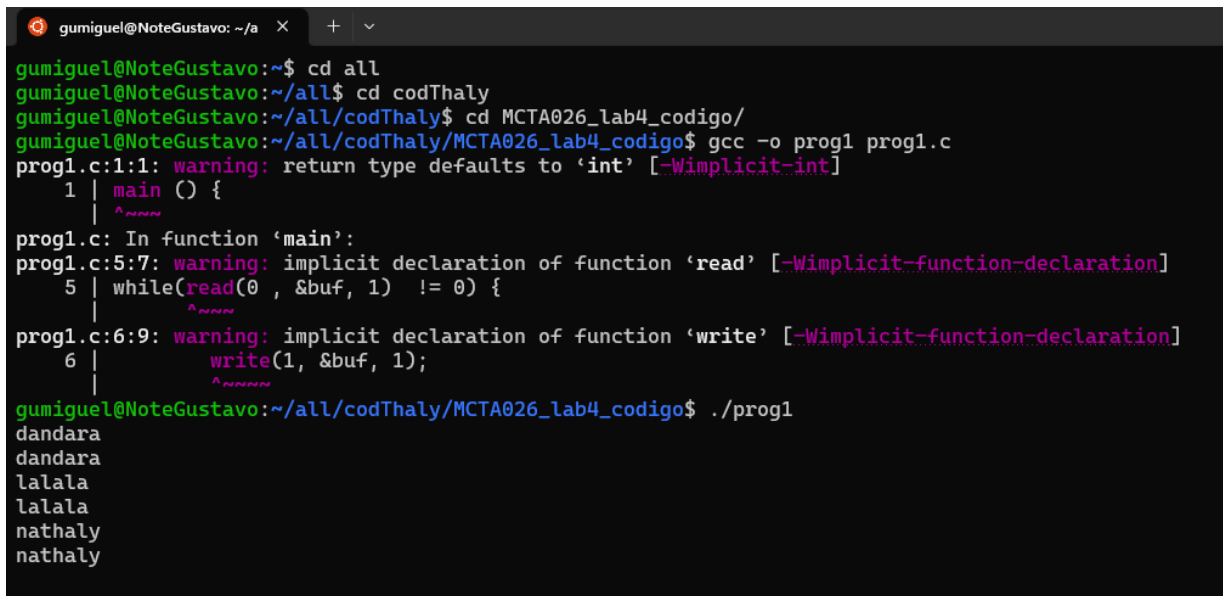
# Sistemas Operacionais

Náthaly Martins De Sá

## **Tópico 04**

### 1. Leitura e escrita em arquivos.

#### 1.1



```
gumiguel@NoteGustavo: ~/a  X  +  v
gumiguel@NoteGustavo:~$ cd all
gumiguel@NoteGustavo:~/all$ cd codThaly
gumiguel@NoteGustavo:~/all/codThaly$ cd MCTA026_lab4_codigo/
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ gcc -o prog1 prog1.c
prog1.c:1:1: warning: return type defaults to 'int' [-Wimplicit-int]
1 | main () {
  | ^~~~~
prog1.c: In function 'main':
prog1.c:5:7: warning: implicit declaration of function 'read' [-Wimplicit-function-declaration]
5 | while(read(0, &buf, 1) != 0) {
  |       ^~~~~
prog1.c:6:9: warning: implicit declaration of function 'write' [-Wimplicit-function-declaration]
6 |     write(1, &buf, 1);
  |     ^~~~~
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ ./prog1
dandara
dandara
lalala
lalala
nathaly
nathaly
```

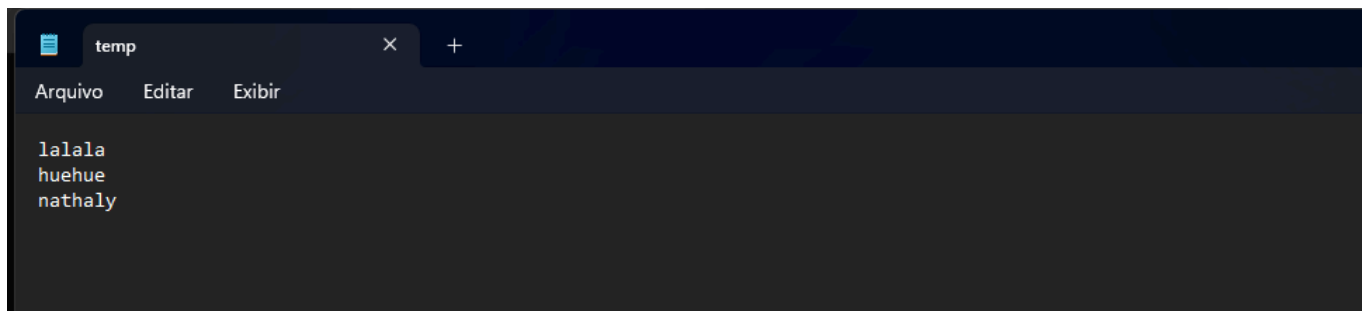
O programa 1 recebe e grava a mensagem, devolvendo a mesma e retornando ao loop em seguida, logo o programa é simples, não sendo necessário incluir outros programas ou funções com o comando include.

E os descritores de entrada e saída padrão são abertos automaticamente pelo sistema operacional quando um programa é executado para que os programas possam interagir com o usuário e outros programas, sendo assim não é necessário abrir esses descritores manualmente no código-fonte do programa.

## 2. Remoção de arquivos.

### 2.1

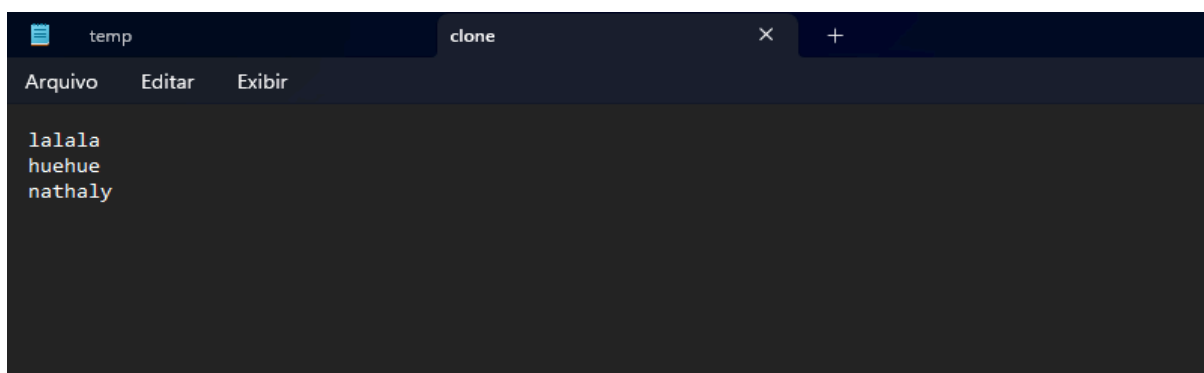
```
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ gcc -o prog2 prog2.c
prog2.c: In function 'main':
prog2.c:14:5: warning: implicit declaration of function 'unlink' [-Wimplicit-function-declaration]
   14 |     unlink("temp.txt");
      |     ^~~~~~
prog2.c:18:5: warning: implicit declaration of function 'lseek'; did you mean 'fseek'? [-Wimplicit-function-declaration]
   18 |     lseek(tmp,0);
      |     ^~~~~~
      |     fseek
prog2.c:20:12: warning: implicit declaration of function 'read'; did you mean 'fread'? [-Wimplicit-function-declaration]
   20 |     while((read(tmp, &c, 1)) > 0) {
      |            ^~~~~~
      |            fread
prog2.c:24:5: warning: implicit declaration of function 'write'; did you mean 'fwrite'? [-Wimplicit-function-declaration]
   24 |     write(tmp,buf,strlen(buf));
      |     ^~~~~~
      |     fwrite
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ ./prog2
Digite unlink para chamada unlink, print para imprimir conteúdo e quit para encerrar o programa: lalala
Grava e retorna ao loop apos comando: lalala
Digite unlink para chamada unlink, print para imprimir conteúdo e quit para encerrar o programa: huehue
Grava e retorna ao loop apos comando: huehue
Digite unlink para chamada unlink, print para imprimir conteúdo e quit para encerrar o programa: nathaly
Grava e retorna ao loop apos comando: nathaly
Digite unlink para chamada unlink, print para imprimir conteúdo e quit para encerrar o programa: █
```



Ao executar o programa 2, o programa grava a mensagem no 'temp.txt' e retoma o loop, ou seja todas as mensagens enviadas pelo usuário serão adicionadas ao arquivo 'temp.txt' logo após o comando, até o programa ser encerrado.

### 2.2

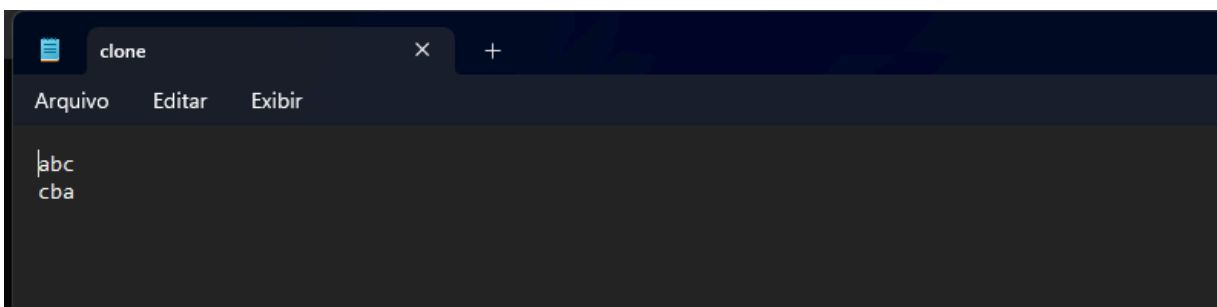
```
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ ln temp.txt clone.txt
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ ls -i *.txt
43063 clone.txt 43063 temp.txt
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$
```



Sim, 'temp.txt' e 'clone.txt' são o mesmo arquivo, pois quando você cria um hard link para um arquivo existente, o novo arquivo criado será essencialmente o mesmo arquivo que o original, já que ambos os arquivos apontam para o mesmo inode e têm os mesmos dados no disco. Em outras palavras, eles são duas entradas diferentes no sistema de arquivos que compartilham o mesmo conteúdo.

## 2.3

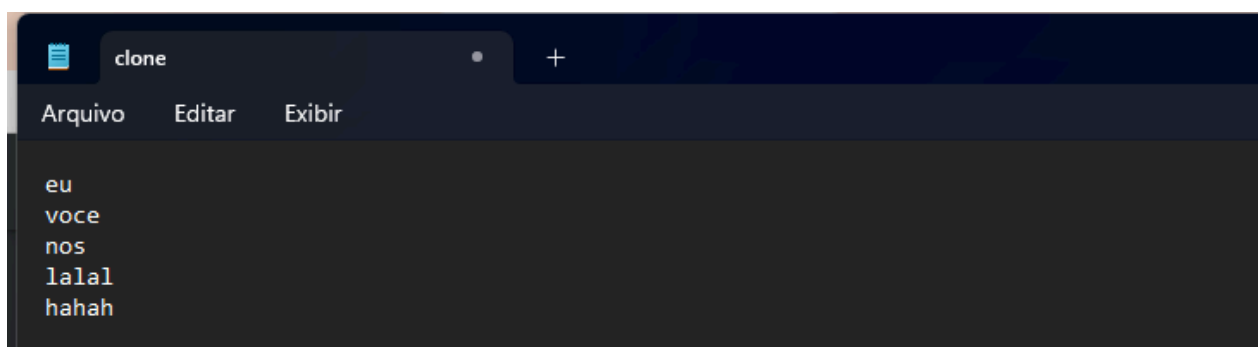
```
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ ./prog2
Digite unlink para chamada unlink, print para imprimir conteúdo e quit para encerrar o programa: abc
Grava e retorna ao loop apos comando: abc
Digite unlink para chamada unlink, print para imprimir conteúdo e quit para encerrar o programa: cba
Grava e retorna ao loop apos comando: cba
Digite unlink para chamada unlink, print para imprimir conteúdo e quit para encerrar o programa: unlink
Unlink executado apos comando: unlink
```



Ao digitar o comando unlink na entrada do programa 2, o arquivo 'temp.txt' é desvinculado e deletado, porém o arquivo 'clone.txt' continua existente e ativo.

## 2.4

```
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ ./prog2
Digite unlink para chamada unlink, print para imprimir conteúdo e quit para encerrar o programa: eu
Grava e retorna ao loop apos comando: eu
Digite unlink para chamada unlink, print para imprimir conteúdo e quit para encerrar o programa: voce
Grava e retorna ao loop apos comando: voce
Digite unlink para chamada unlink, print para imprimir conteúdo e quit para encerrar o programa: nos
Grava e retorna ao loop apos comando: nos
Digite unlink para chamada unlink, print para imprimir conteúdo e quit para encerrar o programa: unlink
Unlink executado apos comando: unlink
Digite unlink para chamada unlink, print para imprimir conteúdo e quit para encerrar o programa: lalal
Grava e retorna ao loop apos comando: lalal
Digite unlink para chamada unlink, print para imprimir conteúdo e quit para encerrar o programa: hahah
Grava e retorna ao loop apos comando: hahah
Digite unlink para chamada unlink, print para imprimir conteúdo e quit para encerrar o programa:
█
```



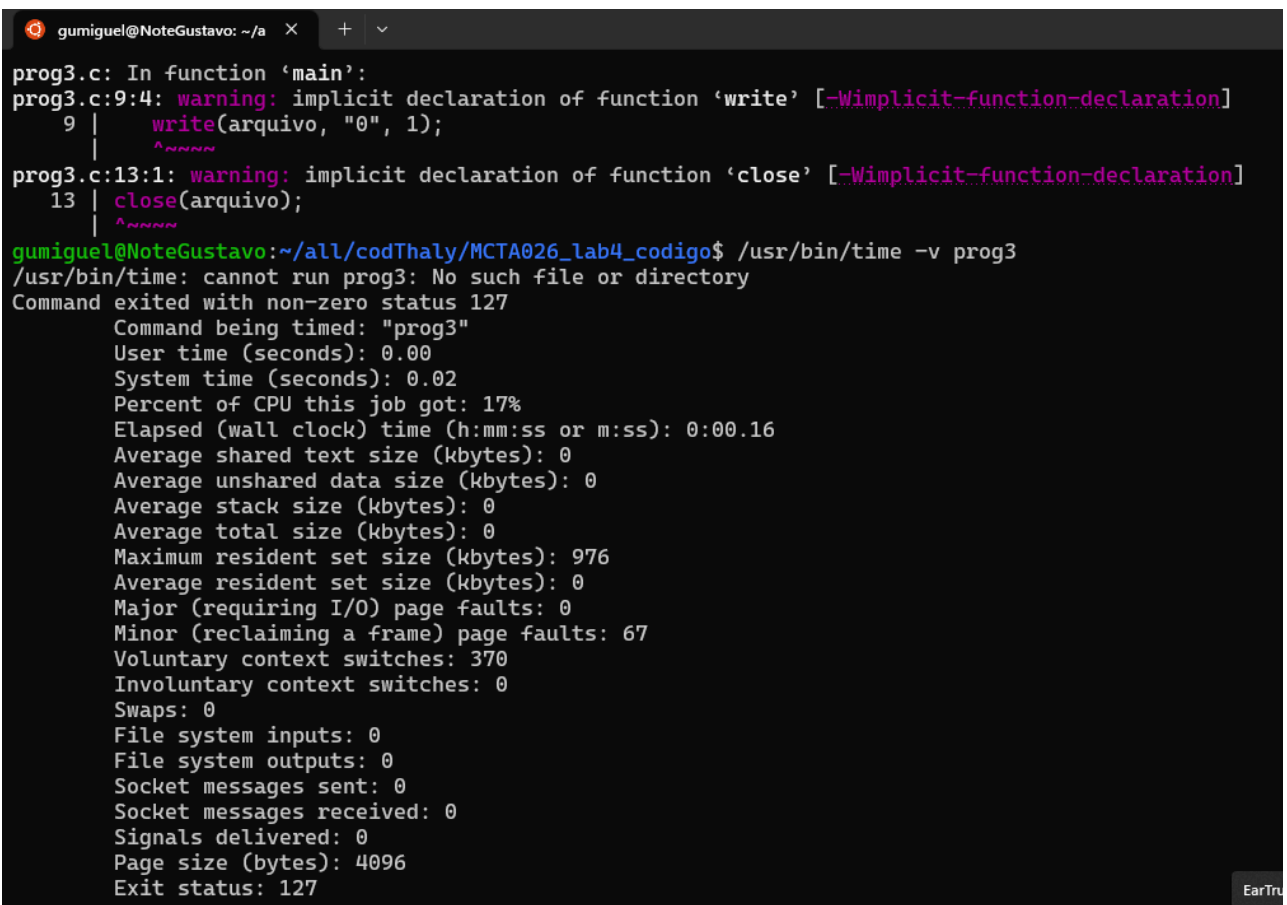
Como citado no tópico anterior o arquivo 'clone.txt' continua existente e ativo, sendo assim ele continua a interagir com o usuário e a ler os comandos do programa. Isso acontece, pois mesmo ao desvincular e deletar o arquivo original, o inode ainda permanece no disco enquanto existir pelo menos um hard link apontando para ele, logo todos os diretórios existentes ainda poderão ser executados, até o inode ser deletado permanentemente.

## 2.5

O comando unlink é usado para remover um link de um arquivo, podendo haver apenas esse link (exclusão permanente do arquivo) ou mais de um link (exclusão apenas do diretório citado). Esse comportamento do unlink seria interessante também para arquivos temporários ou simbólicos, para não ocupar mais memória no sistema, gerenciar as versões desse arquivo, etc.

## 3. Uso de buffer.

### 3.1



```
gumiguel@NoteGustavo: ~/a X + v
prog3.c: In function 'main':
prog3.c:9:4: warning: implicit declaration of function 'write' [-Wimplicit-function-declaration]
  9 |     write(arquivo, "0", 1);
    |     ^~~~~
prog3.c:13:1: warning: implicit declaration of function 'close' [-Wimplicit-function-declaration]
  13 |     close(arquivo);
    |     ^~~~~
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ /usr/bin/time -v prog3
/usr/bin/time: cannot run prog3: No such file or directory
Command exited with non-zero status 127
Command being timed: "prog3"
User time (seconds): 0.00
System time (seconds): 0.02
Percent of CPU this job got: 17%
Elapsed (wall clock) time (h:mm:ss or m:ss): 0:00.16
Average shared text size (kbytes): 0
Average unshared data size (kbytes): 0
Average stack size (kbytes): 0
Average total size (kbytes): 0
Maximum resident set size (kbytes): 976
Average resident set size (kbytes): 0
Major (requiring I/O) page faults: 0
Minor (reclaiming a frame) page faults: 67
Voluntary context switches: 370
Involuntary context switches: 0
Swaps: 0
File system inputs: 0
File system outputs: 0
Socket messages sent: 0
Socket messages received: 0
Signals delivered: 0
Page size (bytes): 4096
Exit status: 127
```

Acima, todos os dados para análise.

## 3.2

```
gumiguel@NoteGustavo: ~/a  X + v
|
|      ^^^^^
prog4.c:10:4: warning: implicit declaration of function 'fsync' [-Wimplicit-function-declaration]
10 |     fsync(arquivo);
|     ^^^^^
|      ^^^^^
prog4.c:14:1: warning: implicit declaration of function 'close' [-Wimplicit-function-declaration]
14 |     close(arquivo);
|     ^^^^^
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ /usr/bin/time -v prog4
/usr/bin/time: cannot run prog4: No such file or directory
Command exited with non-zero status 127
Command being timed: "prog4"
User time (seconds): 0.00
System time (seconds): 0.03
Percent of CPU this job got: 17%
Elapsed (wall clock) time (h:mm:ss or m:ss): 0:00.16
Average shared text size (kbytes): 0
Average unshared data size (kbytes): 0
Average stack size (kbytes): 0
Average total size (kbytes): 0
Maximum resident set size (kbytes): 892
Average resident set size (kbytes): 0
Major (requiring I/O) page faults: 0
Minor (reclaiming a frame) page faults: 65
Voluntary context switches: 370
Involuntary context switches: 0
Swaps: 0
File system inputs: 0
File system outputs: 0
Socket messages sent: 0
Socket messages received: 0
Signals delivered: 0
Page size (bytes): 4096
Exit status: 127
```

Ao comparar os programas 3 (item anterior) e 4 (item atual) podemos observar que o programa 4 possui um gasto de tempo maior que o programa 3, isso porque o 4 apresenta um comando a mais em seu código que faz com que todas as alterações pendentes sejam gravadas no disco, sendo assim o programa 4 também possui número de trocas de contexto voluntárias maior.

## 4. Obtenção dos atributos de um arquivo.

### 4.1

```
gumiguel@NoteGustavo: ~/a  X + v
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ ./prog5 prog4

prog4 e' um arquivo regular

userid 1000, groupid 1000, permissao 100755, link(s) 1
tamanho 16088 (bytes), 32 (blocos), Inode # 43087

Dados de prog4 foram modificados pela ultima vez em Tue Apr 18 13:03:59 2023

Leitura permitida
Escrita permitida
Execucao permitida
```

```
gumiguel@NoteGustavo: ~/a X + v
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ ./prog5 Avaliador

Avaliador e' um
userid 0, groupid 64, permissao 100, link(s) 8
tamanho 0 (bytes), 2 (blocos), Inode # 64

Dados de Avaliador foram modificados pela ultima vez em Wed Dec 31 21:00:00 1969

Leitura proibida
Escrita proibida
Execucao proibida

gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ ./prog5 Treinamento

Treinamento e' um
userid 0, groupid 64, permissao 100, link(s) 8
tamanho 0 (bytes), 0 (blocos), Inode # 64

Dados de Treinamento foram modificados pela ultima vez em Wed Dec 31 21:00:00 1969

Leitura proibida
Escrita proibida
Execucao proibida

gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ ./prog5 Comunicado

Comunicado e' um
userid 0, groupid 64, permissao 100, link(s) 8
tamanho 0 (bytes), 0 (blocos), Inode # 0

Dados de Comunicado foram modificados pela ultima vez em Wed Dec 31 21:00:00 1969

Leitura proibida
```

O programa 5 foi testado com os arquivos prog4, 'Avaliador' (Exel), 'Treinamento' (PowerPoint) e 'Comunicado' (PDF), retornando assim se é um arquivo regular, o id, a quantidade de links que possui, o inode do arquivo, tamanho em bytes, último acesso ao arquivo, além se foi possível executar naquele momento, entre outros.

```
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ ./prog5 Treinamento

Treinamento e' um
userid 0, groupid 64, permissao 100, link(s) 8
tamanho 0 (bytes), 0 (blocos), Inode # 64

Dados de Treinamento foram modificados pela ultima vez em (null)
Leitura proibida
Escrita proibida
Execucao proibida
```

Acima um teste extra com o arquivo 'Treinamento', onde o mesmo estava aberto no momento da execução do comando e é perceptível a mudança que ocorreu na saída.

## 5. Comando stat.

### 5.1

```
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ stat prog4
  File: prog4
  Size: 16088          Blocks: 32          IO Block: 4096   regular file
Device: 840h/2112d    Inode: 43087         Links: 1
Access: (0755/-rwxr-xr-x)  Uid: ( 1000/gumiguel)   Gid: ( 1000/gumiguel)
Access: 2023-04-18 13:03:59.549987913 -0300
Modify: 2023-04-18 13:03:59.549987913 -0300
Change: 2023-04-18 13:03:59.549987913 -0300
 Birth: 2023-04-18 13:03:59.529987748 -0300
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ stat prog4.c
  File: prog4.c
  Size: 197           Blocks: 8           IO Block: 4096   regular file
Device: 840h/2112d    Inode: 43070         Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/gumiguel)   Gid: ( 1000/gumiguel)
Access: 2023-04-18 13:03:59.259985521 -0300
Modify: 2017-04-05 12:58:56.000000000 -0300
Change: 2023-04-18 10:19:52.834107556 -0300
 Birth: 2023-04-18 10:19:52.824107428 -0300
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ stat Comunicado
stat: cannot statx 'Comunicado': No such file or directory
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$
```

Comando stat que proporciona informações similares às apresentadas pelo programa 5.

## 6. Lista de arquivos em um diretório.

### 6.1

```
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ ./prog6 arquivos
Comunicado.pdf (Inode 43079)
Treinamento.ppt (Inode 43083)
. (Inode 42040)
.. (Inode 42012)
Avaliador.xlsx (Inode 42102)
gumiguel@NoteGustavo:~/all/codThaly/MCTA026_lab4_codigo$ ./prog6 cod
prog3 (Inode 43086)
prog1.c (Inode 32375)
prog2 (Inode 43082)
prog4 (Inode 43087)
prog5.c (Inode 43072)
prog5 (Inode 43088)
prog3.c (Inode 43068)
prog1 (Inode 43081)
. (Inode 43085)
prog4.c (Inode 43070)
prog2.c (Inode 43066)
.. (Inode 42012)
```

A organização dos arquivos em um diretório no Linux se dá a partir do seu inode.