

Sistemas Operacionais

Atividade Prática de SA

OBS: Cada tela/resposta requisitada tem valor de 1,0.

Estrutura de Diretórios

Conforme discutido em sala, alguns dos principais sistemas de arquivos usados em ambientes Linux são o ext2, ext3, ext4, reiser, entre outros. Apesar destes diferentes sistemas, em geral, os diretórios de um sistema de arquivos no UNIX têm uma estrutura pré-definida comum, com poucas variações:

- **bin** - *contains system binary executables*
- **boot** - *contains files necessary for the system to boot up*
- **dev** - *contains device files which function as an interface to the various hardware drivers.*
These will vary greatly depending on the version of Unix
- **etc** - *contains system configuration settings*
- **home** - *contains the user's home directories (often but not at LSC/ATM Linux network)*
- **mnt** - *mount point for a temporary mounted filesystem*
- **lib** - *contains system libraries for 32-bit applications*
- **lib64** - *contains system libraries for 64-bit applications*
- **opt** - *contains optional applications*
- **proc** - *contains a virtual file system which holds information about running processes and the state of the system*
- **root** - *the root (administrator) user's home directory*
- **sbin** - *contains static binary executables needed for the system*
- **tmp** - *temporary directory used by many applications*
- **usr** - *contains binaries, data and settings for various applications. The structure of /usr mimics the root file system organization*
- **var** - *stores logs, data for services and other transient data*

Uma vez apresentada esta estrutura de diretórios, e considerando alguns dos comandos discutidos em sala, realize as atividades propostas na sequência.

Processos

1) No terminal, vá até o diretório `/proc` e liste seu conteúdo (`ls -l`). Observe que os subdiretórios correspondem aos PIDs dos processos correntes (execute `ps -lax` e verifique isso). **Apresente um printscreen da tela**

O /proc é, por vezes, chamado de "pseudo sistema de arquivos de informações de

processos” (process information pseudo-file system). O diretório não contém “arquivos de verdade”, mas as informações referentes ao seu sistema em tempo de execução (runtime).

Entre as informações disponíveis no /proc, você pode encontrar a quantidade de memória presente no sistema, os dispositivos de armazenamento que estão montados, a configuração atual do hardware, o tempo que o seu dispositivo está ligado etc.

Utilizando `cd/` entra no diretório “proc”, após isso lista o conteúdo com o comando “`ls -l`”

```
rick@PC-Rick: /proc
rick@PC-Rick:/$ cd /proc
rick@PC-Rick:/proc$ ls -l
total 0
dr-xr-xr-x 9 root      root      0 dez 16 18:59 1
dr-xr-xr-x 9 root      root      0 dez 16 18:59 10
dr-xr-xr-x 9 root      root      0 dez 16 18:59 100
dr-xr-xr-x 9 root      root      0 dez 16 18:59 101
dr-xr-xr-x 9 root      root      0 dez 16 18:59 102
dr-xr-xr-x 9 root      root      0 dez 16 18:59 104
dr-xr-xr-x 9 root      root      0 dez 16 18:59 105
dr-xr-xr-x 9 root      root      0 dez 16 18:59 1054
dr-xr-xr-x 9 root      root      0 dez 16 18:59 107
dr-xr-xr-x 9 root      root      0 dez 16 18:59 108
dr-xr-xr-x 9 root      root      0 dez 16 18:59 109
dr-xr-xr-x 9 root      root      0 dez 16 18:59 11
dr-xr-xr-x 9 root      root      0 dez 16 18:59 111
dr-xr-xr-x 9 root      root      0 dez 16 18:59 1116
dr-xr-xr-x 9 root      root      0 dez 16 18:59 116
dr-xr-xr-x 9 root      root      0 dez 16 18:59 12
dr-xr-xr-x 9 root      root      0 dez 16 18:59 120
dr-xr-xr-x 9 root      root      0 dez 16 18:59 121
dr-xr-xr-x 9 rtkit     rtkit     0 dez 16 18:59 1225
dr-xr-xr-x 9 root      root      0 dez 16 18:59 13
dr-xr-xr-x 9 root      root      0 dez 16 18:59 130
dr-xr-xr-x 9 root      root      0 dez 16 18:59 133
dr-xr-xr-x 9 root      root      0 dez 16 18:59 134
dr-xr-xr-x 9 root      root      0 dez 16 18:59 139
dr-xr-xr-x 9 root      root      0 dez 16 18:59 14
dr-xr-xr-x 9 root      root      0 dez 16 18:59 15
dr-xr-xr-x 9 root      root      0 dez 16 18:59 16
dr-xr-xr-x 9 root      root      0 dez 16 18:59 18
dr-xr-xr-x 9 root      root      0 dez 16 18:59 1846
dr-xr-xr-x 9 root      root      0 dez 16 18:59 188
dr-xr-xr-x 9 root      root      0 dez 16 18:59 189
dr-xr-xr-x 9 root      root      0 dez 16 18:59 1895
dr-xr-xr-x 9 root      root      0 dez 16 18:59 19
dr-xr-xr-x 9 root      root      0 dez 16 18:59 190
dr-xr-xr-x 9 root      root      0 dez 16 18:59 191
```

Executa o `ps -lax` e verifica que os subdiretórios correspondem aos PIDs dos processos correntes

```
rick@PC-Rick: /proc
rick@PC-Rick:/proc$ ps -lax
F  UID      PID     PPID  PRI  NI   VSZ   RSS  WCHAN  STAT TTY      TIME COMMAND
4   0         1         0  20   0 168468 13772 -        Ss   ?        0:03 /sbin/init splash
1   0         2         0  20   0    0     0 -        S    ?        0:00 [kthreadd]
1   0         3         2   0 -20   0     0 -        I<   ?        0:00 [rcu_gp]
1   0         4         2   0 -20   0     0 -        I<   ?        0:00 [rcu_par_gp]
1   0         5         2   0 -20   0     0 -        I<   ?        0:00 [netns]
5   0         6         2  20   0    0     0 -        I    ?        0:05 [kworker/0:0-mm_percpu_wq]
1   0         9         2   0 -20   0     0 -        I<   ?        0:00 [kworker/0:1H-kblockd]
1   0        10         2   0 -20   0     0 -        I<   ?        0:00 [mm_percpu_wq]
1   0        11         2  20   0    0     0 -        S    ?        0:00 [rcu_tasks_rude_]
1   0        12         2  20   0    0     0 -        S    ?        0:00 [rcu_tasks_trace]
1   0        13         2  20   0    0     0 -        S    ?        0:00 [ksoftirqd/0]
1   0        14         2  20   0    0     0 -        I    ?        0:00 [rcu_sched]
1   0        15         2 -100  -    0     0 -        S    ?        0:00 [migration/0]
1   0        16         2 -51  -    0     0 -        S    ?        0:00 [idle_inject/0]
1   0        18         2  20   0    0     0 -        S    ?        0:00 [cpuhp/0]
5   0        19         2  20   0    0     0 -        S    ?        0:00 [cpuhp/1]
1   0        20         2 -51  -    0     0 -        S    ?        0:00 [idle_inject/1]
1   0        21         2 -100  -    0     0 -        S    ?        0:00 [migration/1]
1   0        22         2  20   0    0     0 -        S    ?        0:00 [ksoftirqd/1]
5   0        25         2  20   0    0     0 -        S    ?        0:00 [cpuhp/2]
1   0        26         2 -51  -    0     0 -        S    ?        0:00 [idle_inject/2]
1   0        27         2 -100  -    0     0 -        S    ?        0:00 [migration/2]
1   0        28         2  20   0    0     0 -        S    ?        0:00 [ksoftirqd/2]
5   0        31         2  20   0    0     0 -        S    ?        0:00 [cpuhp/3]
1   0        32         2 -51  -    0     0 -        S    ?        0:00 [idle_inject/3]
1   0        33         2 -100  -    0     0 -        S    ?        0:00 [migration/3]
1   0        34         2  20   0    0     0 -        S    ?        0:00 [ksoftirqd/3]
5   0        37         2  20   0    0     0 -        S    ?        0:00 [kdevtmpfs]
1   0        38         2   0 -20   0     0 -        I<   ?        0:00 [inet_frag_wq]
1   0        39         2  20   0    0     0 -        S    ?        0:00 [kauditd]
1   0        40         2  20   0    0     0 -        S    ?        0:00 [khungtaskd]
1   0        41         2  20   0    0     0 -        S    ?        0:00 [oom_reaper]
1   0        42         2   0 -20   0     0 -        I<   ?        0:00 [writeback]
1   0        43         2  20   0    0     0 -        S    ?        0:00 [kcompactd0]
1   0        44         2  25   5    0     0 -        SN   ?        0:00 [ksmd]
1   0        45         2  39  19    0     0 -        SN   ?        0:00 [khugepaged]
```

2) Lembrando que o *bash* é o seu interpretador de comandos, você pode verificar o PID do *bash* executando o comando *ps*. De posse do PID do seu *bash*, entre no subdiretório cujo nome seja o PID do seu *bash*. Ali você encontrará várias informações sobre este processo. Consulte algumas dessas informações para o seu *bash*:

`more (ou cat) /proc/PID/cmdline` // Argumentos da linha de comando.

`more (ou cat) /proc/PID/maps` // Mapas de memória para os executáveis e arquivos da // biblioteca.

`more (ou cat) /proc/PID/stat` // Informações gerais de estado do processo:

```
(1) pid %d
    The process ID.

(2) comm %s
    The filename of the executable, in parentheses.
    This is visible whether or not the executable is
    swapped out.

(3) state %c
    One of the following characters, indicating process
    state:

    R  Running

    S  Sleeping in an interruptible wait

    D  Waiting in uninterruptible disk sleep

    Z  Zombie

    T  Stopped (on a signal) or (before Linux 2.6.33)
        trace stopped

(14) utime %lu
    Amount of time that this process has been scheduled
    in user mode, measured in clock ticks (divide by
    sysconf(_SC_CLK_TCK)). This includes guest time,
    guest_time (time spent running a virtual CPU, see
    below), so that applications that are not aware of
    the guest time field do not lose that time from
    their calculations.

(15) stime %lu
    Amount of time that this process has been scheduled
    in kernel mode, measured in clock ticks (divide by
    sysconf(_SC_CLK_TCK)).
```

Apresente um *printscreen* com o resultado da visualização do *status* do processo.

O comando ps verifica o PID do bash

```
rick@PC-Rick:/proc$ ps
  PID TTY          TIME CMD
  6551 pts/0        00:00:00 bash
  6989 pts/0        00:00:00 ps
```

Argumentos da linha de comando

```
rick@PC-Rick:/proc$ cat /proc/6551/cmdline
bashrick@PC-Rick:/proc$
```

Mapas de memória para os executáveis e arquivos da biblioteca.

```
bashrick@PC-Rick:/proc$ cat /proc/6551/maps
564b5ecbf000-564b5ecf0000 r--p 00000000 08:04 16121951 /usr/bin/bash
564b5ecf0000-564b5edcd000 r-xp 0002f000 08:04 16121951 /usr/bin/bash
564b5edcd000-564b5ee07000 r--p 0010e000 08:04 16121951 /usr/bin/bash
564b5ee07000-564b5ee0c000 r--p 00148000 08:04 16121951 /usr/bin/bash
564b5ee0c000-564b5ee15000 rw-p 0014c000 08:04 16121951 /usr/bin/bash
564b5ee15000-564b5ee20000 rw-p 00000000 00:00 0
564b5f1b5000-564b5f6b0000 r--p 00000000 00:00 0 [heap]
7f67f4c04000-7f67f4c2c000 r--p 00000000 08:04 19008139 /usr/share/locale-langpack/pt_BR/LC_MESSAGES/bash.mo
7f67f4c2c000-7f67f5a3b000 r--p 00000000 08:04 16127292 /usr/lib/locale/locale-archive
7f67f5a3b000-7f67f5a64000 r--p 00000000 08:04 16128268 /usr/lib/x86_64-linux-gnu/libc.so.6
7f67f5a64000-7f67f5bf9000 r-xp 00028000 08:04 16128268 /usr/lib/x86_64-linux-gnu/libc.so.6
7f67f5bf9000-7f67f5c51000 r--p 001bd000 08:04 16128268 /usr/lib/x86_64-linux-gnu/libc.so.6
7f67f5c51000-7f67f5c55000 r--p 00214000 08:04 16128268 /usr/lib/x86_64-linux-gnu/libc.so.6
7f67f5c55000-7f67f5c57000 rw-p 00218000 08:04 16128268 /usr/lib/x86_64-linux-gnu/libc.so.6
7f67f5c57000-7f67f5c64000 rw-p 00000000 00:00 0
7f67f5c64000-7f67f5c72000 r--p 00000000 08:04 16129397 /usr/lib/x86_64-linux-gnu/libtinfo.so.6.3
7f67f5c72000-7f67f5c83000 r-xp 0000e000 08:04 16129397 /usr/lib/x86_64-linux-gnu/libtinfo.so.6.3
7f67f5c83000-7f67f5c91000 r--p 0001f000 08:04 16129397 /usr/lib/x86_64-linux-gnu/libtinfo.so.6.3
7f67f5c91000-7f67f5c95000 r--p 0002c000 08:04 16129397 /usr/lib/x86_64-linux-gnu/libtinfo.so.6.3
7f67f5c95000-7f67f5c96000 rw-p 00030000 08:04 16129397 /usr/lib/x86_64-linux-gnu/libtinfo.so.6.3
7f67f5ca4000-7f67f5cab000 r--s 00000000 08:04 16781553 /usr/lib/x86_64-linux-gnu/gconv/gconv-modules.cache
7f67f5cab000-7f67f5cb4000 rw-p 00000000 00:00 0
7f67f5cb4000-7f67f5cb6000 r--p 00000000 08:04 16127931 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7f67f5cb6000-7f67f5ce0000 r-xp 00002000 08:04 16127931 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7f67f5ce0000-7f67f5ceb000 r--p 0002c000 08:04 16127931 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7f67f5ceb000-7f67f5cee000 r--p 00037000 08:04 16127931 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7f67f5cee000-7f67f5cf0000 rw-p 00039000 08:04 16127931 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7f67f5cf0000-7f67f5cf3000 rw-p 00000000 00:00 0
7ffffa411a000-7ffffa413b000 r--p 00000000 00:00 0 [stack]
7ffffa417c000-7ffffa4180000 r--p 00000000 00:00 0 [vvar]
7ffffa4180000-7ffffa4182000 r-xp 00000000 00:00 0 [vdso]
ffffffffff600000-ffffffffff601000 --xp 00000000 00:00 0 [vsyscall]
```

Informações gerais de estado do processo

```
rick@PC-Rick:/proc$ cat /proc/6551/stat
6551 (bash) S 6531 6551 6551 34816 7064 4194304 3084 31092 3 12 3 3 29 17 20 0 1 0 182868 24346624 2295 18446744073709551615 94881713152000 9488171406
5165 140735946139056 0 0 0 65536 3686404 1266761467 1 0 0 17 3 0 0 0 0 94881714309808 94881714357840 94881718161408 140735946140406 140735946140411
140735946140411 140735946141678 0
```

3) Você consegue encontrar o executável do seu SO? Execute `ls -l` no diretório raiz. Observe que aparece algo assim:

```
lrwxrwxrwx 1 root root 30 jun 29 06:56 vmlinuz -> boot/vmlinuz-4.15.0-54-generic
```

Esse 1º. caractere na linha indica o tipo de arquivo. Neste caso temos `l`, para *link*. Apresente um *printscreen* demonstrando o executável do seu SO.

Entra no raiz e executa o ls -l pra listar

```
rick@PC-Rick: /boot
rick@PC-Rick:/$ cd boot
rick@PC-Rick:/boot$ ls -l
total 159620
-rw-r--r-- 1 root root 261837 out 17 15:36 config-5.15.0-53-generic
-rw-r--r-- 1 root root 261842 nov 22 12:08 config-5.15.0-56-generic
drwxr-xr-x 5 root root 4096 dez 2 21:30 grub
lrwxrwxrwx 1 root root 28 dez 2 21:28 initrd.img -> initrd.img-5.15.0-56-generic
-rw-r--r-- 1 root root 63371904 dez 3 14:35 initrd.img-5.15.0-53-generic
-rw-r--r-- 1 root root 63379357 dez 3 14:35 initrd.img-5.15.0-56-generic
lrwxrwxrwx 1 root root 28 dez 2 21:28 initrd.img.old -> initrd.img-5.15.0-53-generic
-rw-r--r-- 1 root root 182800 fev 6 2022 memtest86+.bin
-rw-r--r-- 1 root root 184476 fev 6 2022 memtest86+.elf
-rw-r--r-- 1 root root 184980 fev 6 2022 memtest86+_multiboot.bin
-rw----- 1 root root 6250186 out 17 15:36 System.map-5.15.0-53-generic
-rw----- 1 root root 6250186 nov 22 12:08 System.map-5.15.0-56-generic
lrwxrwxrwx 1 root root 25 dez 2 21:28 vmlinuz -> vmlinuz-5.15.0-56-generic
-rw----- 1 root root 11548224 out 17 15:41 vmlinuz-5.15.0-53-generic
-rw----- 1 root root 11551392 nov 22 14:07 vmlinuz-5.15.0-56-generic
lrwxrwxrwx 1 root root 25 dez 2 21:28 vmlinuz.old -> vmlinuz-5.15.0-53-generic
```

Tipos de Arquivos

Os sistemas Unix e Linux trabalham com diferentes tipos de arquivos. Os tipos suportados pelo sistema são:

- **Arquivos normais/regulares:** sequências de bytes (texto, binário, executável, etc.)
- **Diretórios:** lista de outros arquivos (nome do arquivo e inode)
- **Arquivos especiais (dispositivos):** interface entre o sistema e dispositivos de entrada e saída. Podem ser dispositivos orientados a caractere ou a bloco
- **Links:** podem ser simbólicos (*soft link*: ponteiro para outro arquivo) ou concretos (*hard link*: atribue mais um nome ao mesmo arquivo que esteja na mesma partição)
- **Sockets e Pipes:** usados para comunicação entre processos (mecanismo para programação)

Arquivos de Dispositivos

No UNIX, tudo é apresentado na forma de arquivos. Ao plugar um pendrive no computador, por exemplo, um arquivo será criado dentro do diretório **/dev** e ele servirá como interface para acessar ou gerenciar o *drive* USB. Nesse diretório, você encontra caminhos semelhantes para acessar terminais e qualquer dispositivo conectado ao computador, como o mouse e até modems.

4) No terminal, vá até o diretório **/dev** e liste seu conteúdo (**ls -l**). Observe que o início de cada linha impressa indica o tipo de arquivo (**c**, **b** ou **d**... eventualmente algum **l**).

Exemplos:

- *disco IDE* **/dev/hda**, **/dev/hdb**, **/dev/hdc**, **/dev/hdd**, ...
- *disco SCSI/SATA* **/dev/sda**, **/dev/sdb**, **/dev/sdc**, **/dev/sdd**, ...
- *partições disco IDE* **/dev/hda1**, **/dev/hda2**, **/dev/hda3**,
- *partições disco SCSI* **/dev/sda1**, **/dev/sda2**, **/dev/sda3**,
- *terminal de controle* **/dev/tty**
- *terminal serial* **/dev/tty1**, **/dev/tty2**, **/dev/tty3**,
- *subdiretório em que são montados os dispositivos USB* **/dev/usb**

Um fato curioso sobre os dispositivos está relacionado a existência de quatro arquivos na pasta **/dev**: **full**, **zero**, **random** e o **null**. Estes arquivos não correspondem a dispositivos de fato. **Você saberia dizer a função de cada um deles?**

```
rick@PC-Rick: /dev
rick@PC-Rick: /dev$ ls -l
total 0
crw-r--r-- 1 root root 10, 235 dez 16 18:59 autofs
drwxr-xr-x 2 root root 900 dez 16 19:00 block
drwxr-xr-x 2 root root 60 dez 16 18:59 bsg
crw----- 1 root root 10, 234 dez 16 18:59 btrfs-control
drwxr-xr-x 3 root root 60 dez 16 18:59 bus
drwxr-xr-x 2 root root 4100 dez 16 18:59 char
crw----- 1 root root 5, 1 dez 16 18:59 console
lrwxrwxrwx 1 root root 11 dez 16 18:59 core -> /proc/kcore
drwxr-xr-x 6 root root 140 dez 16 18:59 cpu
crw----- 1 root root 10, 124 dez 16 18:59 cpu_dma_latency
crw----- 1 root root 10, 203 dez 16 18:59 cuse
drwxr-xr-x 7 root root 140 dez 16 18:59 disk
drwxr-xr-x 2 root root 60 dez 16 18:59 dma_heap
drwxr-xr-x 3 root root 100 dez 16 18:59 dri
crw----- 1 root root 10, 126 dez 16 18:59 ecryptfs
crw-rw---- 1 root video 29, 0 dez 16 18:59 fb0
lrwxrwxrwx 1 root root 13 dez 16 18:59 fd -> /proc/self/fd
crw-rw-rw- 1 root root 1, 7 dez 16 18:59 full
crw-rw-rw- 1 root root 10, 229 dez 16 18:59 fuse
crw----- 1 root root 240, 0 dez 16 18:59 hidraw0
crw----- 1 root root 240, 1 dez 16 18:59 hidraw1
crw----- 1 root root 240, 2 dez 16 18:59 hidraw2
crw----- 1 root root 240, 3 dez 16 18:59 hidraw3
crw----- 1 root root 240, 4 dez 16 18:59 hidraw4
crw----- 1 root root 10, 228 dez 16 18:59 hpet
drwxr-xr-x 2 root root 0 dez 16 18:59 hugepages
crw----- 1 root root 10, 183 dez 16 18:59 hwrng
crw----- 1 root root 89, 0 dez 16 18:59 i2c-0
crw----- 1 root root 89, 1 dez 16 18:59 i2c-1
crw----- 1 root root 89, 2 dez 16 18:59 i2c-2
crw----- 1 root root 89, 3 dez 16 18:59 i2c-3
crw----- 1 root root 89, 4 dez 16 18:59 i2c-4
lrwxrwxrwx 1 root root 12 dez 16 18:59 initctl -> /run/initctl
drwxr-xr-x 4 root root 380 dez 16 18:59 input
crw-r--r-- 1 root root 1, 11 dez 16 18:59 knsg
```

- `/dev/full` é um arquivo especial que sempre retorna o mesmo código de erro ao ser acessado para escrita: `ENOSPC` (Não há espaço disponível no dispositivo) e provê um número infinito de caracteres nulos (`NULL`) a qualquer processo que o acesso para leitura.
- `/dev/zero` é um arquivo especial que fornece quantos caracteres nulos (o `NULL` da tabela ASCII, `0x00`, e não o caractere “dígito zero”, `“0”`, `0x30`) forem lidos dele. O fluxo de caracteres nulos gerado por este dispositivo pode, por exemplo, ser utilizado para sobrescrever informações ou para gerar um arquivo limpo de certo tamanho. Outro uso comum é para fornecer um fluxo de caracteres para inicialização de armazenamento de dados(data storage).
- `/dev/random` é um arquivo especial que serve como gerador de números pseudo-aleatórios. ele permite acesso aos ruídos do ambiente coletados de controladores do dispositivo e outras fontes. Normalmente bloqueia se houver menos entropia disponível do que o solicitado.

5) No terminal, digite:

```
$ echo "Hello World"
```

e depois

```
$ echo "Hello World" > /dev/null
```

... o que aconteceu com a saída do comando? Apresente um *printscreen* com o resultado do comando.


```
rick@PC-Rick:/$ echo "Hello World"
Hello World
rick@PC-Rick:/$ echo "Hello World" > /dev/null
rick@PC-Rick:/$
```

Ele não mostra nada, isso porque o dev/null descarta a informação escrita nele e não retorna nada, por isso a diferença do primeiro comando.

6) No terminal, digite o comando abaixo e observe o resultado.

\$ echo "Hello world" > /dev/full

De forma análoga, você consegue dizer o que está acontecendo? **Apresente um *printscreen* com o resultado do comando.**

```
rick@PC-Rick:/$ echo "Hello world" > /dev/full
bash: echo: erro de escrita: Não há espaço disponível no dispositivo
```

O /dev/full sempre retorna a mesma mensagem de erro(Não há espaço disponível no dispositivo)

Inodes e Atributos de Arquivos

Cada arquivo ou diretório possui um inode associado.

7) No terminal, vá até o diretório HOME (cd ~) e digite **\$ ls -lai**.

Na coluna mais à esquerda, você encontra os números do inode de cada arquivo.

Agora faça a mesma coisa de dentro do diretório raiz. Alguém com o inode 1?

*Nessa distribuição, provavelmente você deve ver o "/" com inode 2. Mas /proc e /sys com inode 1. Isso ocorre na verdade porque esses não são diretórios de fato no sistema de arquivos local. Eles são "montados" (veremos isso daqui a pouco...). Mas se você está curioso, digite no terminal **\$ findmnt***

Como vimos em sala, no inode de cada arquivo estão armazenados diferentes atributos (informações de controle sobre o arquivo). Ali encontramos informações como:

- Tipo de arquivo: por exemplo, regular, diretório, PIPE, *links* simbólicos, arquivos especiais representando dispositivos
- Número de *hard links* apontando p/ o arquivo
- Tamanho (bytes)
- ID do dispositivo
- Número do i-node: Dentro de um mesmo dispositivo, um i-node (arquivo) tem um número único
- UIDs e GIDs do proprietário
 - Quando um arquivo é criado, seu UID é herdado do effective UID do processo criador. Já no caso do GID, depende da versão do UNIX (ex: SVR3: herda o effective GID do processo criador; BSD/Linux: herda o GID do diretório pai).
- *Timestamps* (último acesso, última modificação e última modificação de atributos) • Permissões e *mode flags*
 - read, write, execute ... Acessos divididos por categorias: owner, group, others

Arquivos executáveis têm um atributo especial, o **suid**: quando um usuário executa um arquivo, o *effective* UID do processo correspondente é setado para o UDI do *owner* deste arquivo.

```
rick@PC-Rick:/$ cd ~
rick@PC-Rick:~$ ls -lai
total 164
 49705 drwxr-x-- 31 rick rick 4096 dez 12 09:16 .
 49704 drwxr-xr-x  3 root root 4096 set 28 22:38 ..
 64223 drwxrwxr-x  2 rick rick 4096 out 13 13:41 .android
664799 drwxr-xr-x  4 rick rick 4096 out 20 14:42 aplicativos
 49741 -rw-rw-r--  1 rick rick  104 set 28 21:30 .apport-ignore.xml
 49790 drwxr-xr-x  2 rick rick 4096 out 20 14:42 'Área de Trabalho'
 54222 -rw-----  1 rick rick 6890 dez  8 16:06 .bash_history
 49708 -rw-r--r--  1 rick rick  220 set 25 10:54 .bash_logout
 49707 -rw-r--r--  1 rick rick 3968 set 29 11:15 .bashrc
 49709 drwx----- 23 rick rick 4096 dez  7 20:49 .cache
 49718 drwx----- 28 rick rick 4096 dez  7 21:00 .config
11413  -rw-r--r--  1 rick rick  359 nov 21 16:48 default-soapui-workspace.xml
 49794 drwxr-xr-x  6 rick rick 4096 dez 16 19:24 Documentos
 49791 drwxr-xr-x  7 rick rick 4096 dez  7 17:32 Downloads
3290349 drwxrwxr-x  6 rick rick 4096 dez  7 21:01 .gitkraken
1575487 drwx-----  3 rick rick 4096 out 20 14:42 .gnome
  54670 drwx-----  2 rick rick 4096 dez 12 13:43 .gnupg
3671417 drwxrwxr-x 10 rick rick 4096 out 31 14:33 .gradle
117769 drwxrwxr-x  5 rick rick 4096 set 28 22:39 .icons
 49796 drwxr-xr-x  3 rick rick 4096 set 29 20:56 Imagens
15563  -rw-rw-r--  1 rick rick  135 out 31 14:08 .install4j
 64063 drwxrwxr-x  5 rick rick 4096 out 20 14:43 .java
2497115 drwxrwxr-x  3 rick rick 4096 out 27 15:54 .jdk
 10088 -rw-----  1 rick rick   20 out 20 13:35 .lessht
 49757 drwx-----  3 rick rick 4096 set 28 22:38 .local
 49792 drwxr-xr-x  2 rick rick 4096 set 28 22:39 Modelos
 49795 drwxr-xr-x  2 rick rick 4096 set 28 22:38 Música
661328 drwxrwxr-x  4 rick rick 4096 set 29 11:28 .npm
659800 drwxrwxr-x  8 rick rick 4096 set 29 11:26 .nvm
664753 drwxr-xr-x  3 rick rick 4096 out 20 14:43 .openjfx
 54869 drwx-----  3 rick rick 4096 set 28 22:39 .pki
 49706 -rw-r--r--  1 rick rick  807 set 25 10:54 .profile
 49793 drwxr-xr-x  2 rick rick 4096 set 28 22:38 Público
 49713 drwx----- 12 rick rick 4096 dez  7 21:00 snap
664802 drwxr-xr-x  4 rick rick 4096 out 20 14:42 .soapuios
```



```
rick@PC-Rick:~$ cd /
rick@PC-Rick:/$ ls -lai
total 84
  2 drwxr-xr-x  20 root root  4096 set  28 22:38 .
  2 drwxr-xr-x  20 root root  4096 set  28 22:38 ..
 12 lrwxrwxrwx   1 root root    7 set  25 10:48 bin -> usr/bin
35387 drwxr-xr-x   3 root root  4096 dez   3 14:35 boot
35779 drwxrwxr-x   2 root root  4096 set  28 22:39 cdrom
   1 drwxr-xr-x  20 root root  4760 dez  16 19:00 dev
6029313 drwxr-xr-x 143 root root 12288 dez  14 22:35 etc
 49704 drwxr-xr-x   3 root root  4096 set  28 22:38 home
  13 lrwxrwxrwx   1 root root    7 set  25 10:48 lib -> usr/lib
  14 lrwxrwxrwx   1 root root    9 set  25 10:48 lib32 -> usr/lib32
  15 lrwxrwxrwx   1 root root    9 set  25 10:48 lib64 -> usr/lib64
  16 lrwxrwxrwx   1 root root   10 set  25 10:48 libx32 -> usr/libx32
  11 drwx-----   2 root root 16384 set  25 10:45 lost+found
12451841 drwxr-xr-x   3 root root  4096 set  28 22:14 media
  35738 drwxr-xr-x   2 root root  4096 set  28 22:38 mnt
 8257537 drwxr-xr-x   6 root root  4096 nov  30 16:50 opt
   1 dr-xr-xr-x 318 root root    0 dez  16 18:59 proc
  35739 drwx-----   7 root root  4096 out   3 15:09 root
   1 drwxr-xr-x  39 root root  1160 dez  16 19:22 run
  17 lrwxrwxrwx   1 root root    8 set  25 10:48 sbin -> usr/sbin
 35326 drwxr-xr-x  25 root root  4096 out   5 14:40 snap
786433 drwxr-xr-x   2 root root  4096 ago   9 08:48 srv
   1 dr-xr-xr-x  13 root root    0 dez  16 18:59 sys
4718593 drwxrwxrwt  22 root root  4096 dez  16 19:48 tmp
16121857 drwxr-xr-x  14 root root  4096 ago   9 08:48 usr
  35780 drwxr-xr-x  15 root root  4096 set  29 11:11 var
```

```
rick@PC-Rick:/$ findmnt
TARGET                                SOURCE                                FSTYPE    OPTIONS
-/sys                                 /dev/sda4                                ext4      rw,relatime,errors=remount-ro
-/sys/kernel/security                 sysfs                                     sysfs     rw,nosuid,nodev,noexec,relatime
-/sys/fs/cgroup                       cgroup2                                 cgroup2   rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot
-/sys/fs/pstore                       pstore                                  pstore    rw,nosuid,nodev,noexec,relatime
-/sys/fs/bpf                           bpf                                      bpf        rw,nosuid,nodev,noexec,relatime,mode=700
-/sys/kernel/debug                    debugfs                                  debugfs    rw,nosuid,nodev,noexec,relatime
-/sys/kernel/tracing                  tracefs                                  tracefs    rw,nosuid,nodev,noexec,relatime
-/sys/fs/fuse/connections              fusectl                                  fusectl    rw,nosuid,nodev,noexec,relatime
-/sys/kernel/config                   configfs                                 configfs    rw,nosuid,nodev,noexec,relatime
-/proc                                 proc                                     proc       rw,nosuid,nodev,noexec,relatime
-/proc/sys/fs/binfmt_misc              systemd-1                                autofs     rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=1786
-/proc/sys/fs/binfmt_misc              binfmt_misc                              binfmt_misc rw,nosuid,nodev,noexec,relatime
-/dev                                  udev                                     udev       rw,nosuid,nodev,noexec,relatime
-/dev/pts                             devpts                                   devpts     rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000
-/dev/shm                             tmpfs                                    tmpfs      rw,nosuid,nodev,inode64
-/dev/mqueue                          mqueue                                  mqueue     rw,nosuid,nodev,noexec,relatime
-/dev/hugepages                       hugetlbfs                               hugetlbfs  rw,relatime,pagesize=2M
-/run                                  tmpfs                                    tmpfs      rw,nosuid,nodev,noexec,relatime,size=1173852k,mode=755,inode64
-/run/lock                             tmpfs                                    tmpfs      rw,nosuid,nodev,noexec,relatime,size=5120k,inode64
-/run/credentials/systemd-sysusers.service none                                     ramfs      ro,nosuid,nodev,noexec,relatime,mode=700
-/run/snapd/ns                         tmpfs[/snapd/ns]                       tmpfs      rw,nosuid,nodev,noexec,relatime,size=1173852k,mode=755,inode64
-/run/snapd/ns/firefox.mnt              nsfs[mnt:[4026532311]]                  nsfs       rw
-/run/snapd/ns/docker.mnt               nsfs[mnt:[4026532547]]                  nsfs       rw
-/run/snapd/ns/snapd-desktop-integration.mnt nsfs[mnt:[4026532548]]                  nsfs       rw
-/run/snapd/ns/snap-store.mnt            nsfs[mnt:[4026532550]]                  nsfs       rw
-/run/user/1000                        tmpfs                                    tmpfs      rw,nosuid,nodev,relatime,size=1173848k,nr_inodes=293462,mode=700,uid=1000,gid=1000
-/run/user/1000/gvfs                   gvfsd-fuse                              fuse.gvfsd-fus rw,nosuid,nodev,relatime,user_id=1000,group_id=1000
-/run/user/1000/doc                    portal                                   fuse.portal rw,nosuid,nodev,relatime,user_id=1000,group_id=1000
```

8) No terminal, digite **\$ stat NOME_DO_ARQUIVO**. Faça isso para diferentes tipos de arquivos (um arquivo de texto, um dispositivo no **/dev**, um diretório). Observe os campos “Blocos” e “bloco de E/S” (Obs.: podem aparecer em inglês). **Apresente um *printscreen* com os resultados dos comandos.**

Arquivos do tipo *link*

O *link* é um mecanismo que faz referência a outro arquivo ou diretório em outra localização. Os *links* são arquivos especiais e podem ser identificados com um "l" quando executado o comando **ls -la**.

```
rick@PC-Rick: ~/Documentos/S0$ stat texto.txt
Arquivo: texto.txt
Tamanho: 6          Blocos: 8          bloco de E/S: 4096  arquivo comum
Dispositivo: 804h/2052d Inode: 655829  Links: 1
Acesso: (0664/-rw-rw-r--)  Uid: ( 1000/ rick)  Gid: ( 1000/ rick)
Acesso: 2022-12-16 20:24:09.897051044 -0300
Modificação: 2022-12-16 20:24:09.913050589 -0300
Alteração: 2022-12-16 20:24:36.116305533 -0300
Criação: 2022-12-16 20:24:09.897051044 -0300
```

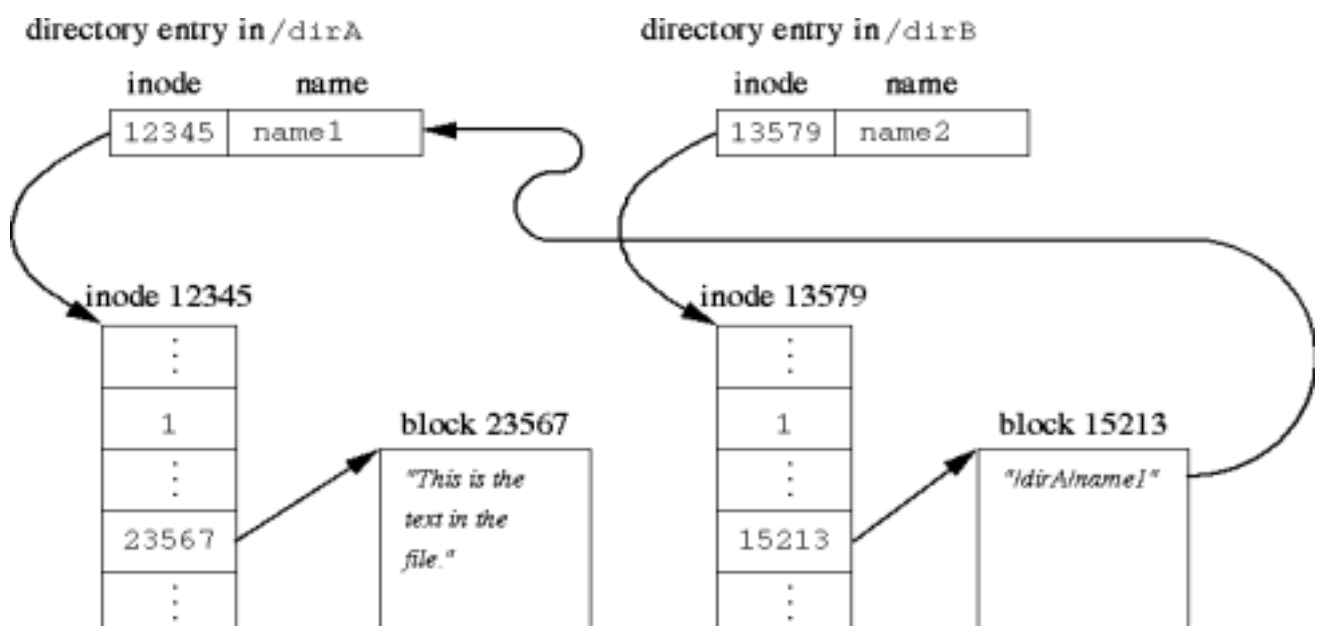
```
rick@PC-Rick: ~/Documentos/S0$ stat /dev/full
Arquivo: /dev/full
Tamanho: 0          Blocos: 0          bloco de E/S: 4096  arquivo especial de caractere
Dispositivo: 5h/5d   Inode: 8          Links: 1          Tipo de dispositivo: 1,7
Acesso: (0666/crw-rw-rw-)  Uid: ( 0/ root)  Gid: ( 0/ root)
Acesso: 2022-12-16 18:59:32.127821066 -0300
Modificação: 2022-12-16 18:59:32.127821066 -0300
Alteração: 2022-12-16 18:59:32.127821066 -0300
Criação: -
```

```
rick@PC-Rick: ~$ stat /home/rick/Documentos
Arquivo: /home/rick/Documentos
Tamanho: 4096       Blocos: 8          bloco de E/S: 4096  diretório
Dispositivo: 804h/2052d Inode: 49794      Links: 6
Acesso: (0755/drwxr-xr-x)  Uid: ( 1000/ rick)  Gid: ( 1000/ rick)
Acesso: 2022-12-16 20:23:16.350577803 -0300
Modificação: 2022-12-16 19:24:23.198350226 -0300
Alteração: 2022-12-16 19:24:23.198350226 -0300
Criação: 2022-09-28 13:16:26.082620884 -0300
rick@PC-Rick: ~$
```

Symbolic links

No *link* tipo simbólico, o *link* é um arquivo especial de disco do tipo *link*, que tem como conteúdo o caminho para chegar até o arquivo alvo. As principais características são:

- Pode-se fazer *links* simbólicos em arquivos e diretórios;
- O *link* simbólico e o arquivo alvo não precisam estar na mesma partição de disco;
- Se o *link* simbólico for apagado/movido. Somente o *link* será apagado/movido;
- Qualquer usuário pode criar/ desfazer um *link* simbólico (respeitando as permissões).



Como criar: **ln -s path1_alvo_do_link path2_nome_do_arquivo_link**

Como visualizar o *link* criado: **ls -l**

- 9) No terminal, crie um *link* simbólico usando **ln -s** e depois verifique o resultado usando **ls -l**. Apresente um *printscreen* com o resultado do comando.

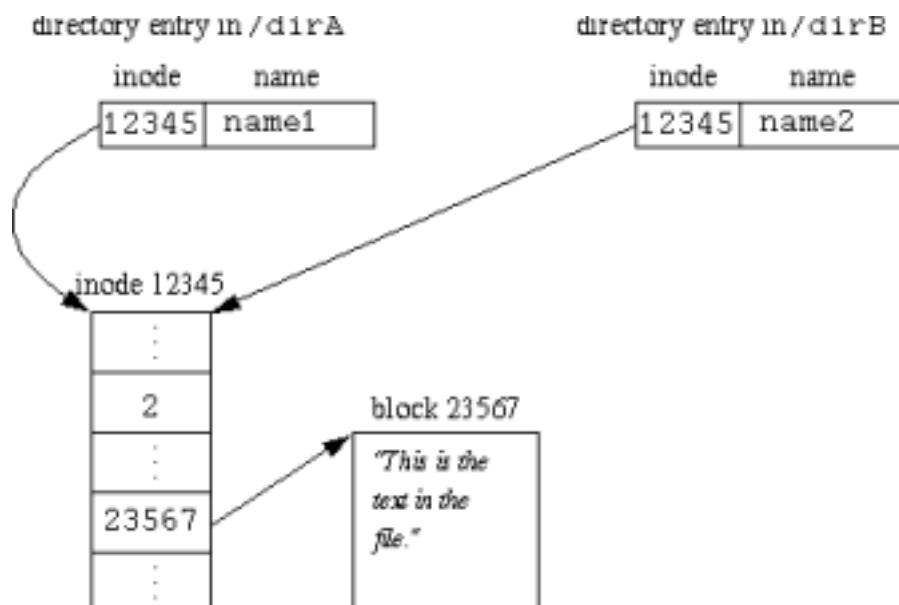
Hard links

No *link* tipo *hardlink*, o *link* é apontado para o mesmo inode do arquivo alvo, sendo assim, os dois arquivos serão o mesmo. As principais características são:

- Não é possível fazer um *hardlink* para um diretório;
- Somente é possível fazer *hardlink* em arquivos que estejam em uma mesma partição de disco;
- Se o *hardlink* for apagado/movido, você estará apagando/movendo o arquivo alvo;
- O usuário deve ter permissão de RW no arquivo destino

Como

criar: **ln path1_alvo_do_Link path2_nome_do_arquivo_link**



```
rick@PC-Rick: ~  
rick@PC-Rick:~$ sudo ln -s /home/rick/Documents/SO  
[sudo] senha para rick:  
rick@PC-Rick:~$ ls -l  
total 48  
drwxr-xr-x 4 rick rick 4096 out 20 14:42 aplicativos  
drwxr-xr-x 2 rick rick 4096 out 20 14:42 'Área de Trabalho'  
-rw-r--r-- 1 rick rick 359 nov 21 16:48 default-soapui-workspace.xml  
drwxr-xr-x 6 rick rick 4096 dez 16 19:24 Documentos  
drwxr-xr-x 7 rick rick 4096 dez 7 17:32 Downloads  
drwxr-xr-x 3 rick rick 4096 set 29 20:56 Imagens  
drwxr-xr-x 2 rick rick 4096 set 28 22:39 Modelos  
drwxr-xr-x 2 rick rick 4096 set 28 22:38 Música  
drwxr-xr-x 2 rick rick 4096 set 28 22:38 Público  
drwx----- 12 rick rick 4096 dez 7 21:00 snap  
lrwxrwxrwx 1 root root 24 dez 16 20:34 SO -> /home/rick/Documents/SO  
-rw-r--r-- 1 rick rick 2321 nov 21 16:48 soapui-settings.xml  
drwxr-xr-x 2 rick rick 4096 set 28 22:39 Videos  
rick@PC-Rick:~$ sudo ln -s /home/rick/Documents/SO/chocolate.txt chokito.txt  
rick@PC-Rick:~$ ls -l  
total 48  
drwxr-xr-x 4 rick rick 4096 out 20 14:42 aplicativos  
drwxr-xr-x 2 rick rick 4096 out 20 14:42 'Área de Trabalho'  
lrwxrwxrwx 1 root root 38 dez 16 20:36 chokito.txt -> /home/rick/Documents/SO/chocolate.txt  
-rw-r--r-- 1 rick rick 359 nov 21 16:48 default-soapui-workspace.xml  
drwxr-xr-x 6 rick rick 4096 dez 16 19:24 Documentos  
drwxr-xr-x 7 rick rick 4096 dez 7 17:32 Downloads  
drwxr-xr-x 3 rick rick 4096 set 29 20:56 Imagens  
drwxr-xr-x 2 rick rick 4096 set 28 22:39 Modelos  
drwxr-xr-x 2 rick rick 4096 set 28 22:38 Música  
drwxr-xr-x 2 rick rick 4096 set 28 22:38 Público  
drwx----- 12 rick rick 4096 dez 7 21:00 snap  
lrwxrwxrwx 1 root root 24 dez 16 20:34 SO -> /home/rick/Documents/SO  
-rw-r--r-- 1 rick rick 2321 nov 21 16:48 soapui-settings.xml  
drwxr-xr-x 2 rick rick 4096 set 28 22:39 Videos  
rick@PC-Rick:~$
```

10) No terminal, crie um *hard link* usando **ln** e depois verifique o resultado usando **ls -l**. Tente acessar o arquivo (**more** se for um arquivo ASCII) via o *link* criado. Apresente um **printscreen** com o resultado do comando.

```
rick@PC-Rick: ~  
rick@PC-Rick:~$ sudo ln /home/rick/Documents/SO/chocolate.txt /home/rick/Documents/SO2  
ln: falha ao acessar '/home/rick/Documents/SO/chocolate.txt': Arquivo ou diretório inexistente  
rick@PC-Rick:~$ sudo ln /home/rick/Documents/SO/teste.txt /home/rick/Documents/SO2  
ln: falha ao acessar '/home/rick/Documents/SO/teste.txt': Arquivo ou diretório inexistente  
rick@PC-Rick:~$ sudo ln /home/rick/Documents/SO/texto.txt /home/rick/Documents/SO2  
rick@PC-Rick:~$ ls -l  
total 48  
drwxr-xr-x 4 rick rick 4096 out 20 14:42 aplicativos  
drwxr-xr-x 2 rick rick 4096 out 20 14:42 'Área de Trabalho'  
lrwxrwxrwx 1 root root 38 dez 16 20:36 chokito.txt -> /home/rick/Documents/SO/chocolate.txt  
-rw-r--r-- 1 rick rick 359 nov 21 16:48 default-soapui-workspace.xml  
drwxr-xr-x 7 rick rick 4096 dez 16 20:39 Documentos  
drwxr-xr-x 7 rick rick 4096 dez 7 17:32 Downloads  
drwxr-xr-x 3 rick rick 4096 set 29 20:56 Imagens  
drwxr-xr-x 2 rick rick 4096 set 28 22:39 Modelos  
drwxr-xr-x 2 rick rick 4096 set 28 22:38 Música  
drwxr-xr-x 2 rick rick 4096 set 28 22:38 Público  
drwx----- 12 rick rick 4096 dez 7 21:00 snap  
lrwxrwxrwx 1 root root 24 dez 16 20:34 SO -> /home/rick/Documents/SO  
-rw-r--r-- 1 rick rick 2321 nov 21 16:48 soapui-settings.xml  
drwxr-xr-x 2 rick rick 4096 set 28 22:39 Videos  
rick@PC-Rick:~$ sudo ln /home/rick/Documents/SO/texto.txt /home/rick/Documents/SO2  
ln: falha ao criar link absoluto '/home/rick/Documents/SO2/texto.txt': Arquivo existe  
rick@PC-Rick:~$ ls -l /home/rick/Documents/SO2/  
total 4  
-rw-rw-r-- 2 rick rick 6 dez 16 20:24 texto.txt  
rick@PC-Rick:~$ more /home/rick/Documents/SO2/texto.txt  
teste  
rick@PC-Rick:~$
```

11) Você consegue criar um *hardlink* para um diretório? Apresente um **printscreen** com o resultado do comando.

```
rick@PC-Rick:~$ sudo ln /home/rick/Documents/SO/ /home/rick/Documents/SO2  
ln: /home/rick/Documents/SO/: não são permitidos links absolutos para diretórios
```

Como deu pra perceber, não dá pra criar hard link em diretórios