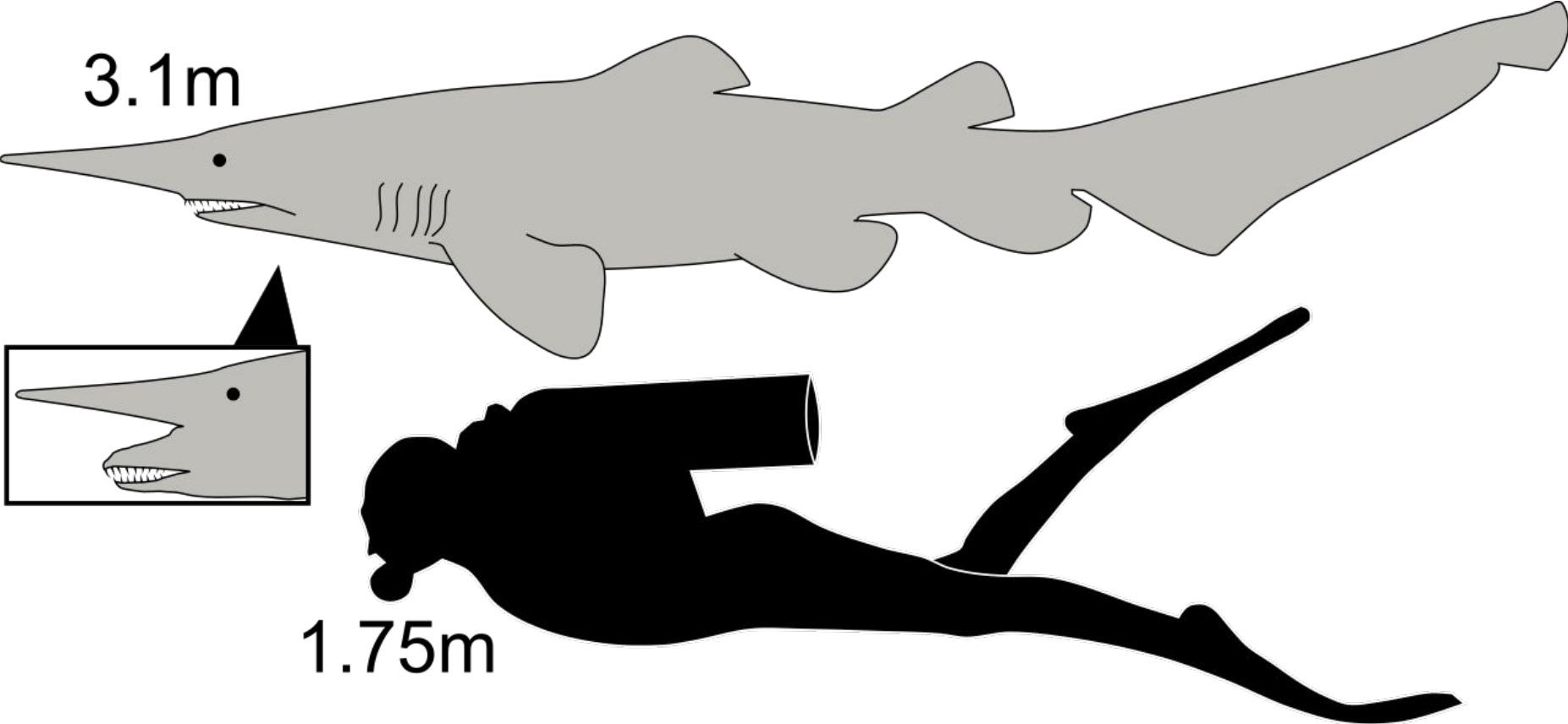


Goblin shark



1300 m

Goblin shark



Goblin shark



The ampullae of Lorenzini



https://en.wikipedia.org/wiki/Ampullae_of_Lorenzini



Devices and drivers

Unix design

every physical device connected to the computer is represented as

?

Unix design

every physical device connected to the computer is represented as

a file

Devices

```
$ ls -l /dev
```

```
brw-rw---- 1 root disk      8,    0 maj 27 09:50 sda

lrwxrwxrwx 1 root root      15 maj 27 09:50 stderr -> /proc/self/fd/2
lrwxrwxrwx 1 root root      15 maj 27 09:50 stdin  -> /proc/self/fd/0
lrwxrwxrwx 1 root root      15 maj 27 09:50 stdout -> /proc/self/fd/1

crw-rw-rw- 1 root tty       5,    0 maj 27 09:50 tty
crw--w---- 1 root tty       4,    0 maj 27 09:50 tty0
crw--w---- 1 root tty       4,    1 maj 27 09:50 tty1
crw--w---- 1 root tty       4,   10 maj 27 09:50 tty10

crw-r----- 1 root kmem     1,    1 maj 27 09:50 mem
crw-rw-rw- 1 root root      1,    3 maj 27 09:50 null
crw-rw-rw- 1 root root      1,    5 maj 27 09:50 zero
crw-rw-rw- 1 root root      1,    8 maj 27 09:50 random
```


Devices

```
$ ls -l /dev
```

Where are the network devices?

```
brw-rw---- 1 root disk      8,    0 maj 27 09:50 sda

lrwxrwxrwx 1 root root      15 maj 27 09:50 stderr -> /proc/self/fd/2
lrwxrwxrwx 1 root root      15 maj 27 09:50 stdin  -> /proc/self/fd/0
lrwxrwxrwx 1 root root      15 maj 27 09:50 stdout -> /proc/self/fd/1

crw-rw-rw- 1 root tty       5,    0 maj 27 09:50 tty
crw--w---- 1 root tty       4,    0 maj 27 09:50 tty0
crw--w---- 1 root tty       4,    1 maj 27 09:50 tty1
crw--w---- 1 root tty       4,   10 maj 27 09:50 tty10

crw-r----- 1 root kmem     1,    1 maj 27 09:50 mem
crw-rw-rw- 1 root root      1,    3 maj 27 09:50 null
crw-rw-rw- 1 root root      1,    5 maj 27 09:50 zero
crw-rw-rw- 1 root root      1,    8 maj 27 09:50 random
```

Unix design

every physical device connected to the computer is represented as

~~a file~~

a file descriptor

For example: network devices are not represented as files
but
we can *read* and *write* when maintaining a socket connection.

Devices

```
$ ls -l /dev
```

```
brw-rw---- 1 root disk      8,    0 maj 27 09:50 sda


lrwxrwxrwx 1 root root      15 maj 27 09:50 stderr -> /proc/self/fd/2
lrwxrwxrwx 1 root root      15 maj 27 09:50 stdin  -> /proc/self/fd/0
lrwxrwxrwx 1 root root      15 maj 27 09:50 stdout -> /proc/self/fd/1

crw-rw-rw- 1 root tty       5,    0 maj 27 09:50 tty
crw--w---- 1 root tty       4,    0 maj 27 09:50 tty0
crw--w---- 1 root tty       4,    1 maj 27 09:50 tty1
crw--w---- 1 root tty       4,   10 maj 27 09:50 tty10

crw-r----- 1 root kmem     1,    1 maj 27 09:50 mem
crw-rw-rw- 1 root root      1,    3 maj 27 09:50 null
crw-rw-rw- 1 root root      1,    5 maj 27 09:50 zero
crw-rw-rw- 1 root root      1,    8 maj 27 09:50 random
```

Devices

```
$ ls -l /dev
```

										name
brw-rw----	1	root	disk	8,	0	maj	27	09:50	sda	
lrwxrwxrwx	1	root	root	15	maj	27	09:50	stderr	-> /proc/self/fd/2	
lrwxrwxrwx	1	root	root	15	maj	27	09:50	stdin	-> /proc/self/fd/0	
lrwxrwxrwx	1	root	root	15	maj	27	09:50	stdout	-> /proc/self/fd/1	
crw-rw-rw-	1	root	tty	5,	0	maj	27	09:50	tty	
crw--w----	1	root	tty	4,	0	maj	27	09:50	tty0	
crw--w----	1	root	tty	4,	1	maj	27	09:50	tty1	
crw--w----	1	root	tty	4,	10	maj	27	09:50	tty10	
crw-r-----	1	root	kmem	1,	1	maj	27	09:50	mem	
crw-rw-rw-	1	root	root	1,	3	maj	27	09:50	null	
crw-rw-rw-	1	root	root	1,	5	maj	27	09:50	zero	
crw-rw-rw-	1	root	root	1,	8	maj	27	09:50	random	

Devices: sda

Which one?

Which
partition?




sda2

SCSI device

Devices

```
$ ls -l /dev
```

										name
brw-rw----	1	root	disk	8,	0	maj	27	09:50	sda	
lrwxrwxrwx	1	root	root	15	maj	27	09:50	stderr	-> /proc/self/fd/2	
lrwxrwxrwx	1	root	root	15	maj	27	09:50	stdin	-> /proc/self/fd/0	
lrwxrwxrwx	1	root	root	15	maj	27	09:50	stdout	-> /proc/self/fd/1	
crw-rw-rw-	1	root	tty	5,	0	maj	27	09:50	tty	
crw--w----	1	root	tty	4,	0	maj	27	09:50	tty0	
crw--w----	1	root	tty	4,	1	maj	27	09:50	tty1	
crw--w----	1	root	tty	4,	10	maj	27	09:50	tty10	
crw-r-----	1	root	kmem	1,	1	maj	27	09:50	mem	
crw-rw-rw-	1	root	root	1,	3	maj	27	09:50	null	
crw-rw-rw-	1	root	root	1,	5	maj	27	09:50	zero	
crw-rw-rw-	1	root	root	1,	8	maj	27	09:50	random	

Devices: tty

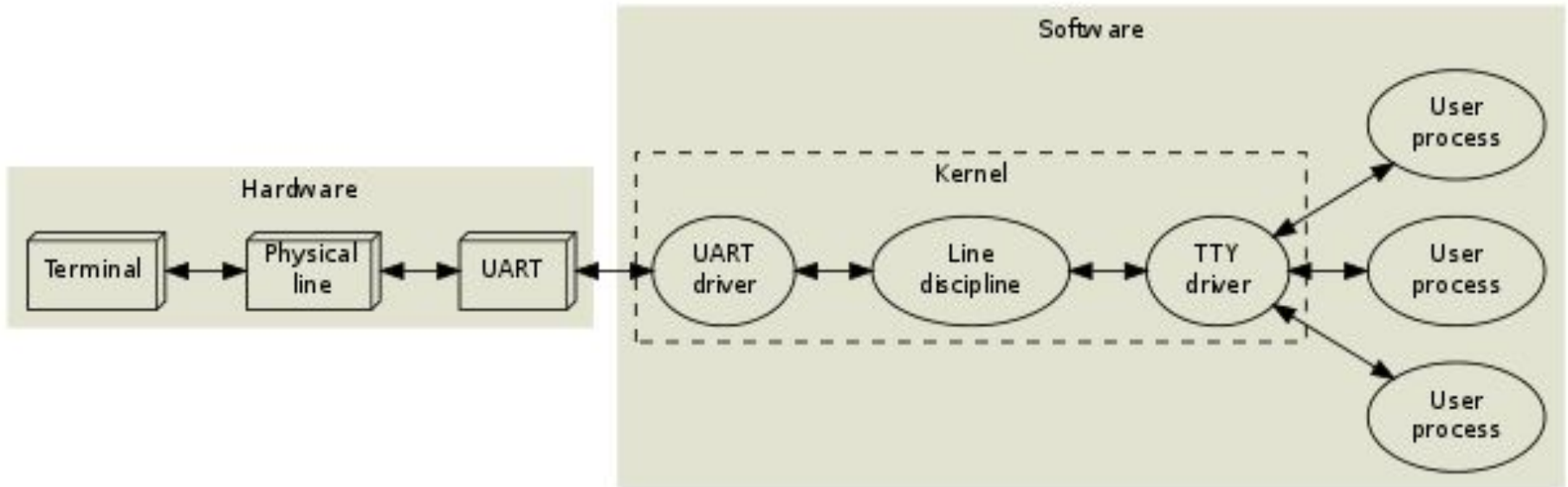


<http://www.cl.cam.ac.uk/~djc11/howcomputerswork/teletypeclose.jpg>

Teletypewriter

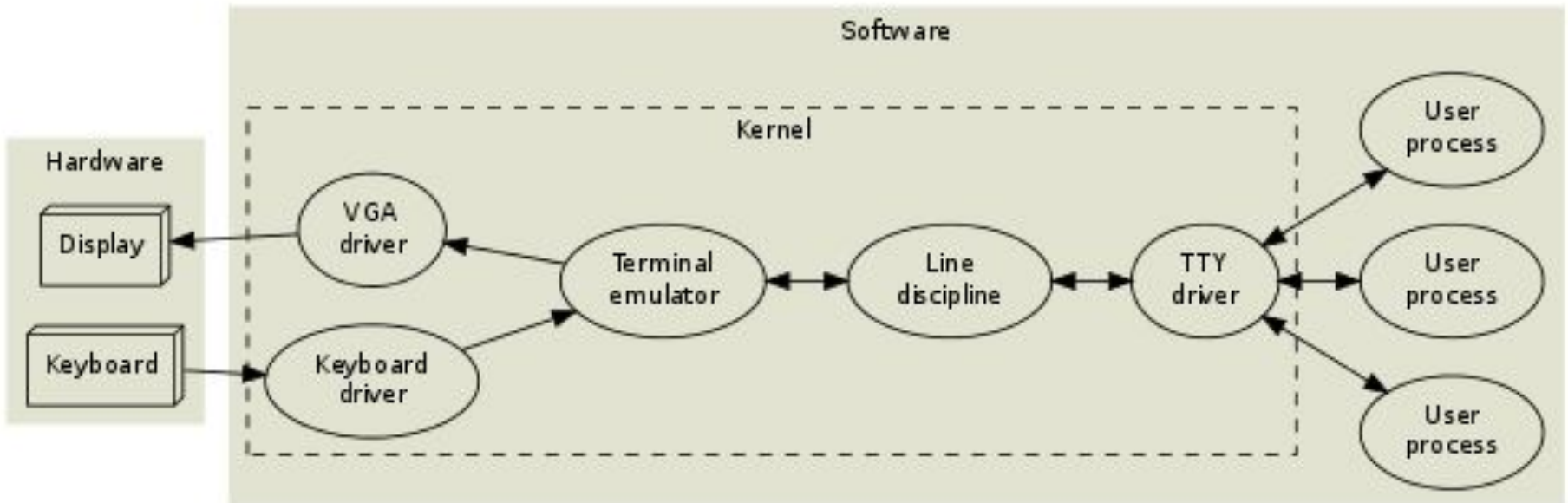
Devices: tty

Before




Devices: tty

Now



Devices

```
$ ls -l /dev
```

										name
brw-rw----	1	root	disk	8,	0	maj	27	09:50	sda	
lrwxrwxrwx	1	root	root	15	maj	27	09:50	stderr	-> /proc/self/fd/2	
lrwxrwxrwx	1	root	root	15	maj	27	09:50	stdin	-> /proc/self/fd/0	
lrwxrwxrwx	1	root	root	15	maj	27	09:50	stdout	-> /proc/self/fd/1	
crw-rw-rw-	1	root	tty	5,	0	maj	27	09:50	tty	
crw--w----	1	root	tty	4,	0	maj	27	09:50	tty0	
crw--w----	1	root	tty	4,	1	maj	27	09:50	tty1	
crw--w----	1	root	tty	4,	10	maj	27	09:50	tty10	
crw-r-----	1	root	kmem	1,	1	maj	27	09:50	mem	
crw-rw-rw-	1	root	root	1,	3	maj	27	09:50	null	
crw-rw-rw-	1	root	root	1,	5	maj	27	09:50	zero	
crw-rw-rw-	1	root	root	1,	8	maj	27	09:50	random	

Pseudo-devices



Pseudo-devices

device node *not necessarily* physical device



device node *sometimes* OS functionality



Pseudo-devices

device node *not necessarily* ↔ physical device

device node *sometimes* ↔ OS functionality

pseudo-device	writing to	reading from
<code>/dev/null</code>	accepts and discards all input	always returns EOF

Pseudo-devices

device node *not necessarily* ↔ physical device

device node *sometimes* ↔ OS functionality

pseudo-device	writing to	reading from
<code>/dev/null</code>	accepts and discards all input	always returns EOF
<code>/dev/zero</code>	accepts and discards all input	produces a continuous stream of zeroes

Pseudo-devices

device node *not necessarily* ↔ physical device

device node *sometimes* ↔ OS functionality

pseudo-device	writing to	reading from
<code>/dev/null</code>	accepts and discards all input	always returns EOF
<code>/dev/zero</code>	accepts and discards all input	produces a continuous stream of zeroes
<code>/dev/full</code>	returns “disk full” message	produces a continuous stream of NULL

Pseudo-devices

device node *not necessarily* ↔ physical device

device node *sometimes* ↔ OS functionality

pseudo-device	writing to	reading from
<code>/dev/null</code>	accepts and discards all input	always returns EOF
<code>/dev/zero</code>	accepts and discards all input	produces a continuous stream of zeroes
<code>/dev/full</code>	returns “disk full” message	produces a continuous stream of NULL
<code>/dev/random</code> <code>/dev/urandom</code>	update the entropy pool https://security.stackexchange.com/a/69433	produces a stream of pseudo-random numbers

Btw: randomness

events that are non-deterministic
and hard for an outside observer to measure



entropy pool (random values)

Btw: randomness

events that are non-deterministic
and hard for an outside observer to measure

/dev/random



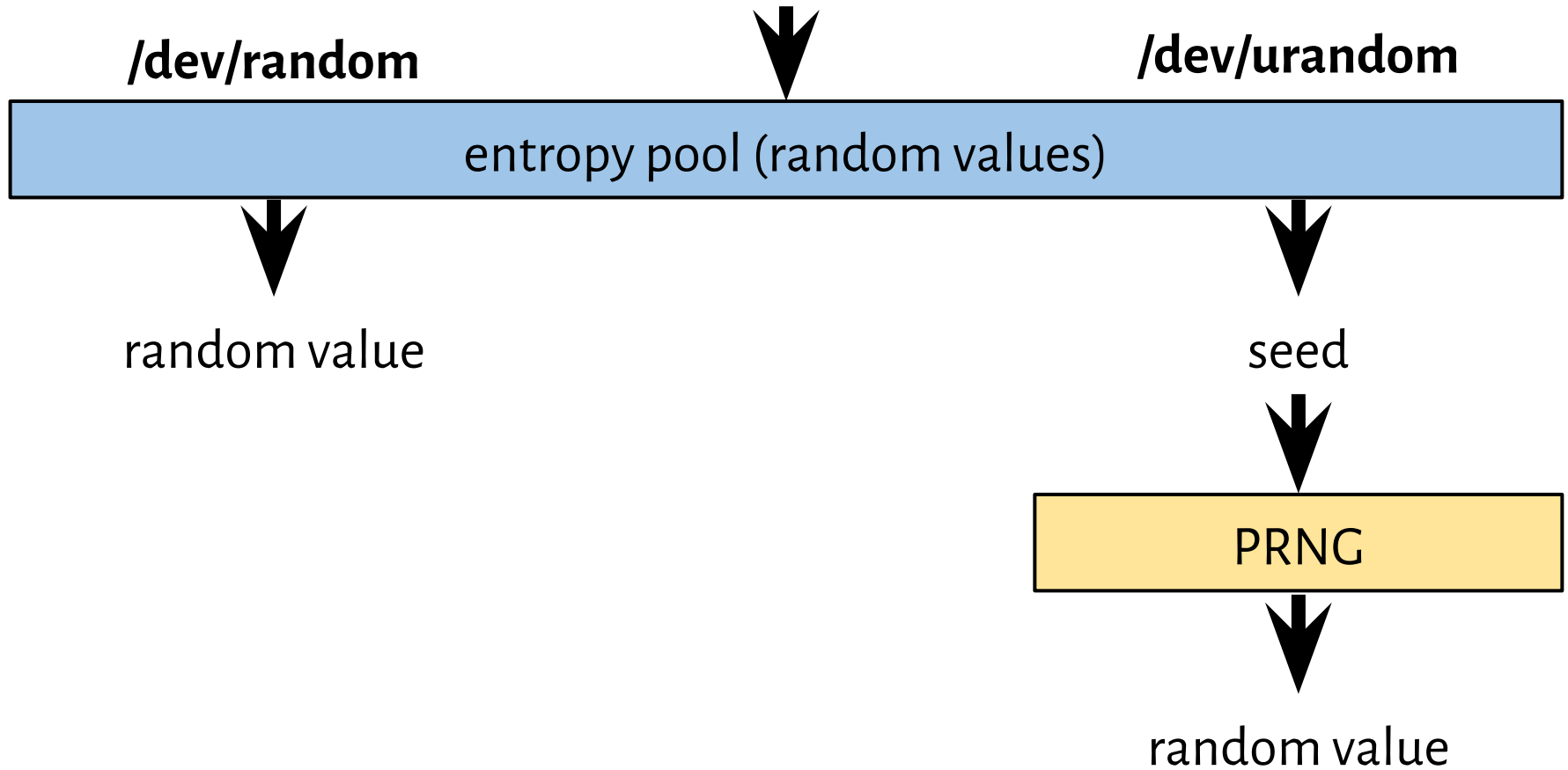
entropy pool (random values)



random value

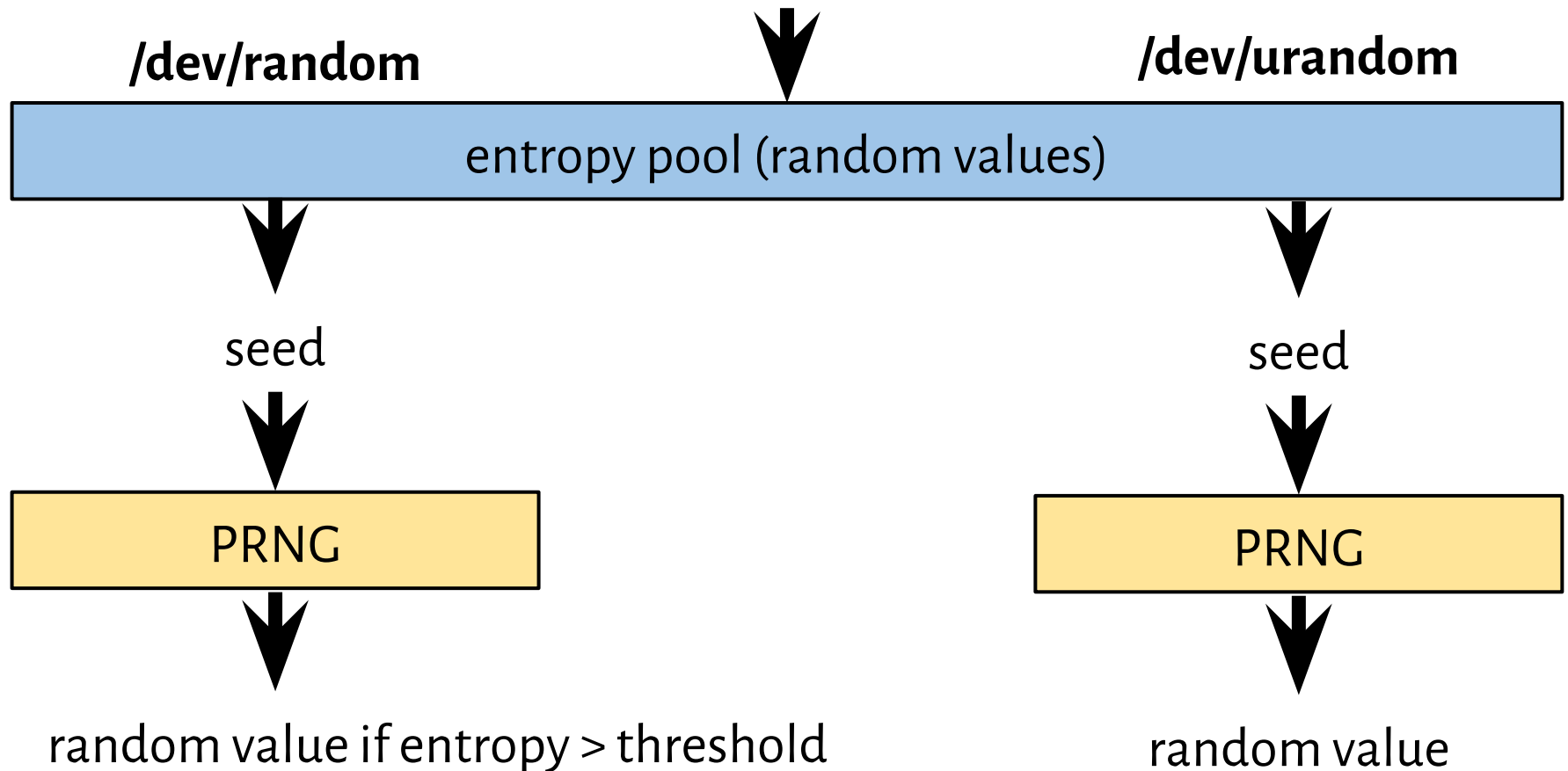
Btw: randomness

events that are non-deterministic
and hard for an outside observer to measure



Btw: randomness

events that are non-deterministic
and hard for an outside observer to measure



Devices

```
$ ls -l /dev
```

```
brw-rw---- 1 root disk      8,    0 maj 27 09:50 sda

lrwxrwxrwx 1 root root      15 maj 27 09:50 stderr -> /proc/self/fd/2
lrwxrwxrwx 1 root root      15 maj 27 09:50 stdin  -> /proc/self/fd/0
lrwxrwxrwx 1 root root      15 maj 27 09:50 stdout -> /proc/self/fd/1

crw-rw-rw- 1 root tty       5,    0 maj 27 09:50 tty
crw--w---- 1 root tty       4,    0 maj 27 09:50 tty0
crw--w---- 1 root tty       4,    1 maj 27 09:50 tty1
crw--w---- 1 root tty       4,   10 maj 27 09:50 tty10

crw-r----- 1 root kmem     1,    1 maj 27 09:50 mem
crw-rw-rw- 1 root root      1,    3 maj 27 09:50 null
crw-rw-rw- 1 root root      1,    5 maj 27 09:50 zero
crw-rw-rw- 1 root root      1,    8 maj 27 09:50 random
```

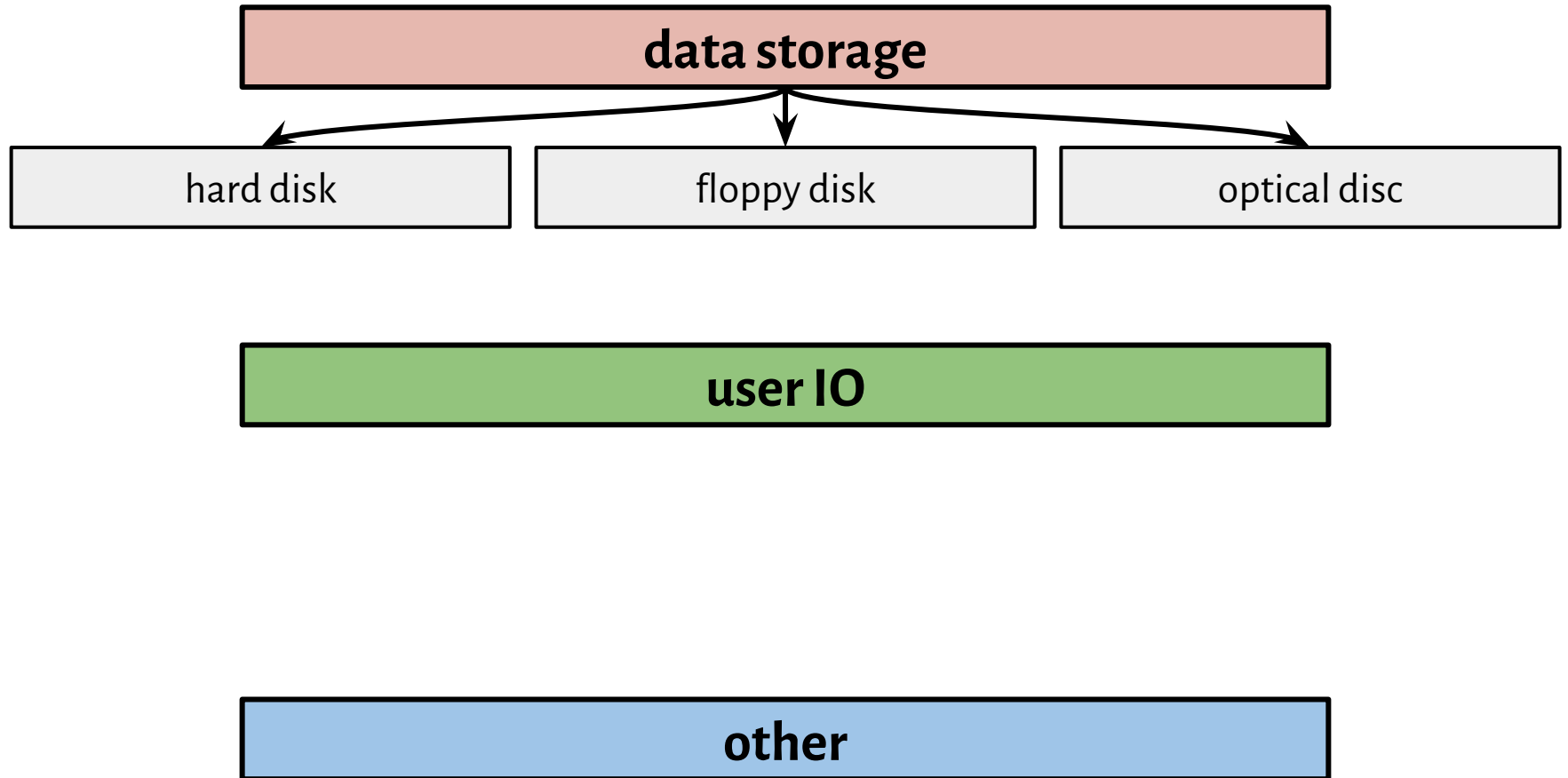
Devices - examples

data storage

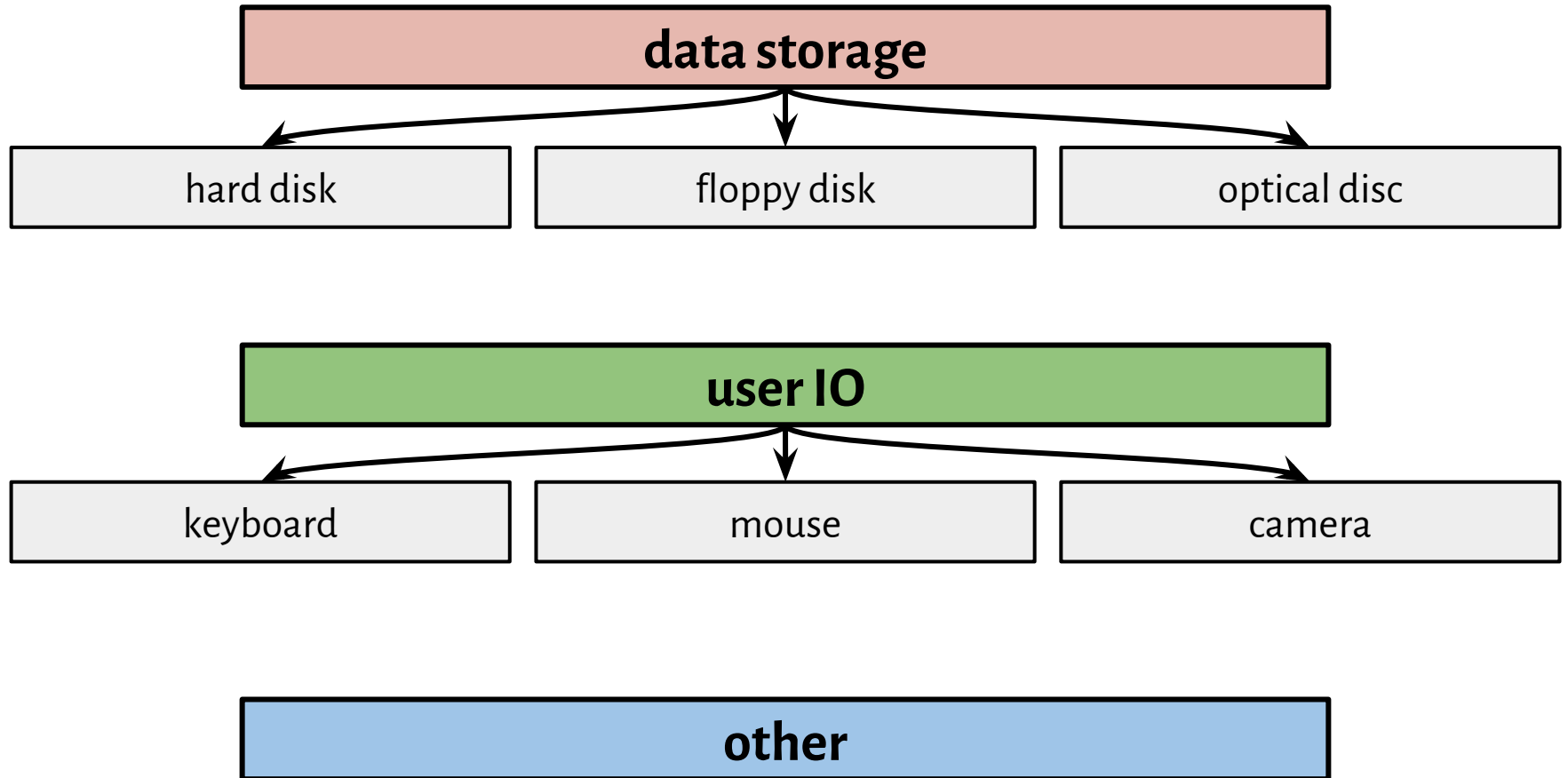
user IO

other

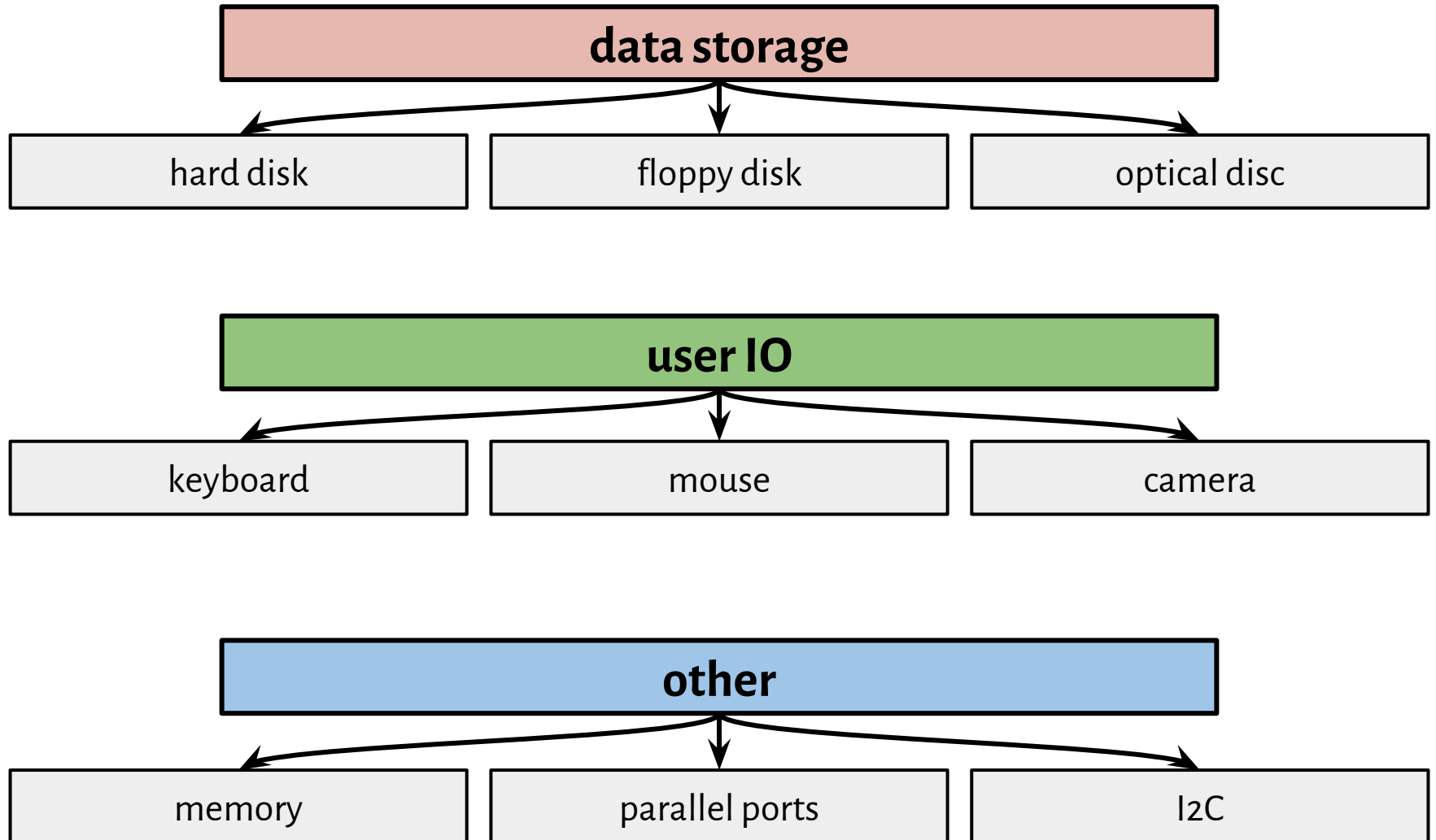
Devices - examples



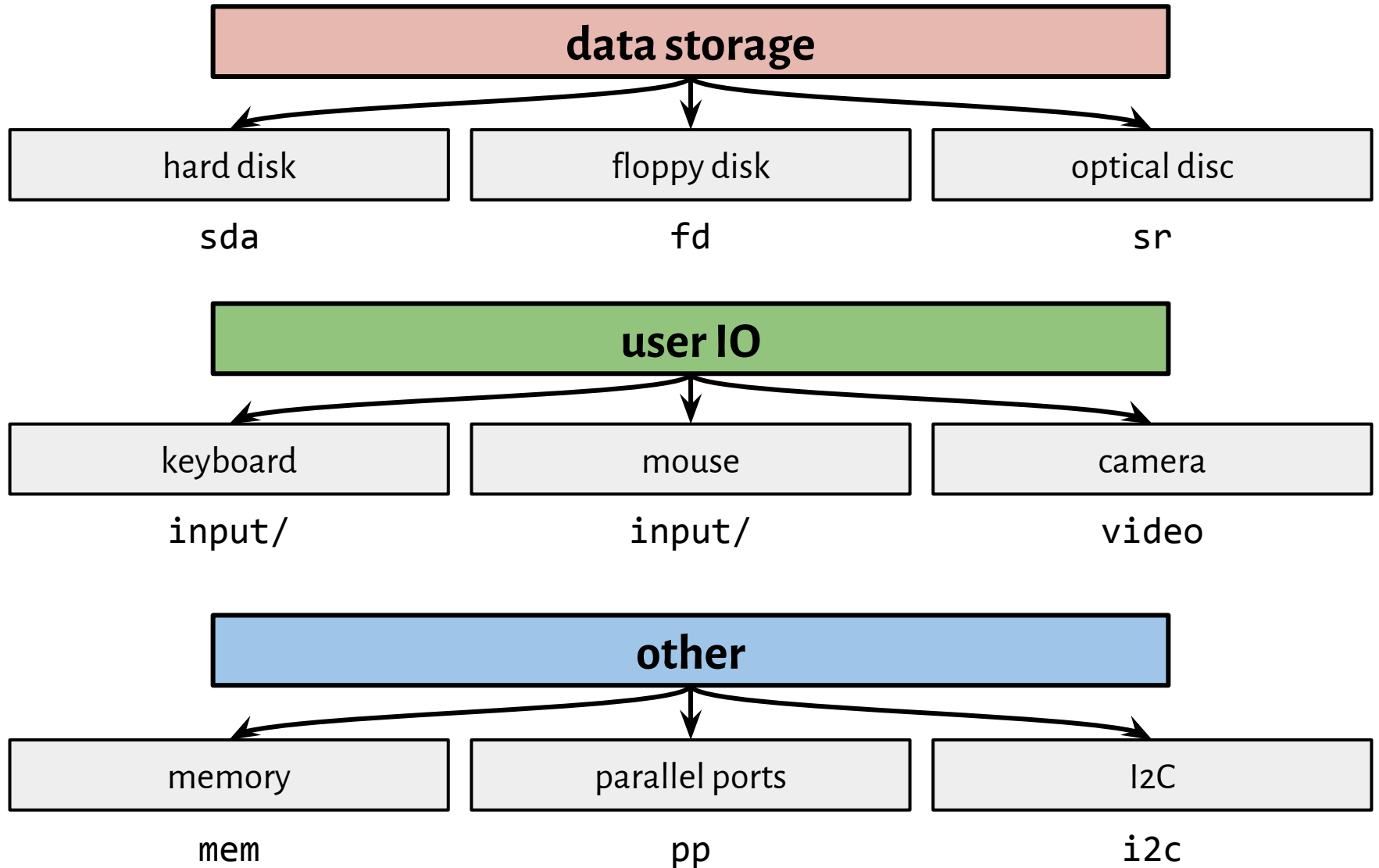
Devices - examples



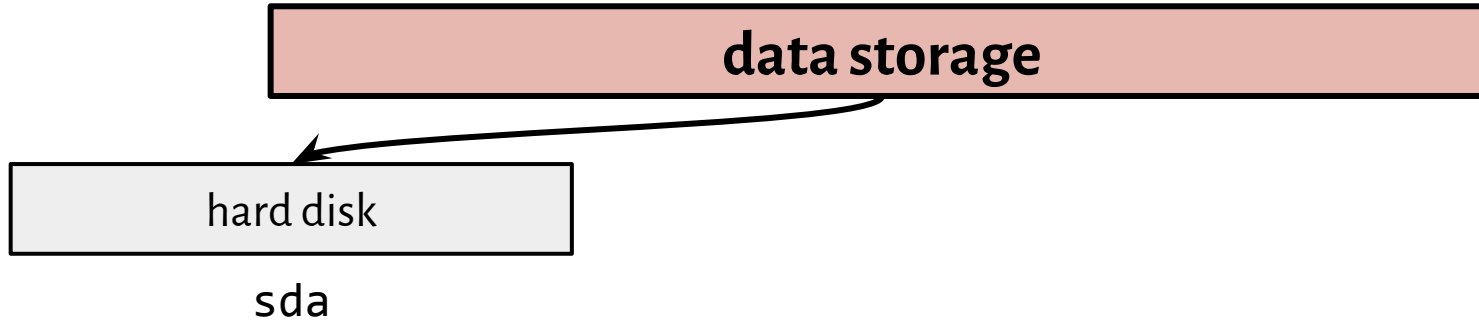
Devices - examples



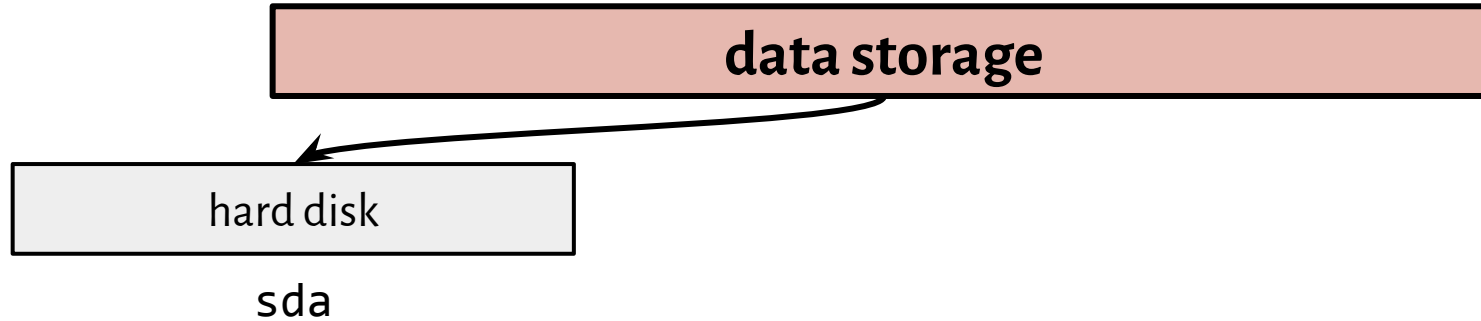
Devices - examples



Devices - symlinks



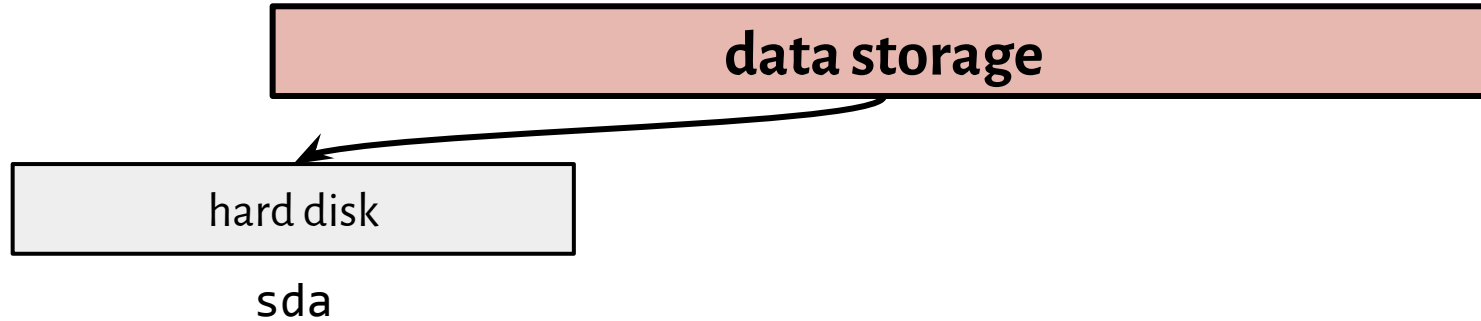
Devices - symlinks



```
$ ls /dev/sda*
```

```
/dev/sda /dev/sda1 /dev/sda2 /dev/sda3
```

Devices - symlinks



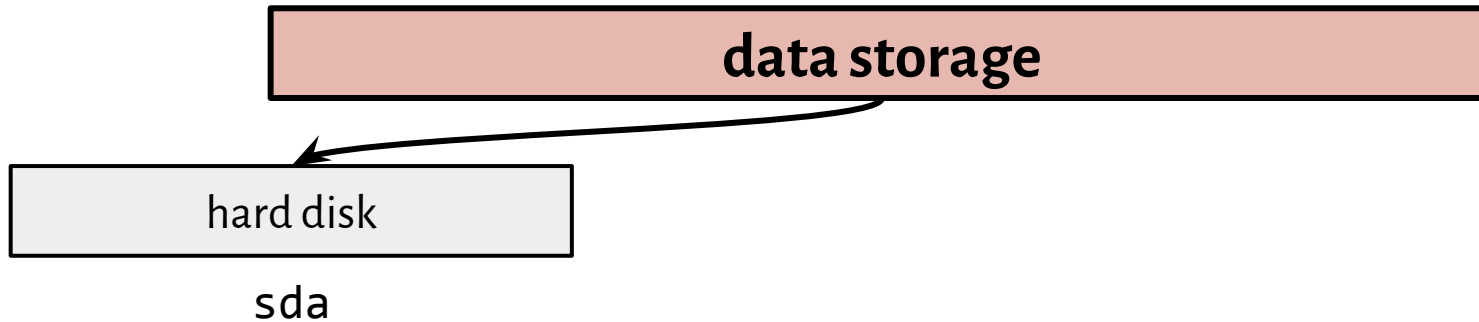
```
$ ls /dev/sda*
```

```
/dev/sda /dev/sda1 /dev/sda2 /dev/sda3
```

```
$ ls /dev/disk
```

```
by-id/ by-partlabel/ by-partuuid/ by-path/ by-uuid/
```

Devices - symlinks



```
$ ls /dev/sda*
```

```
/dev/sda /dev/sda1 /dev/sda2 /dev/sda3
```

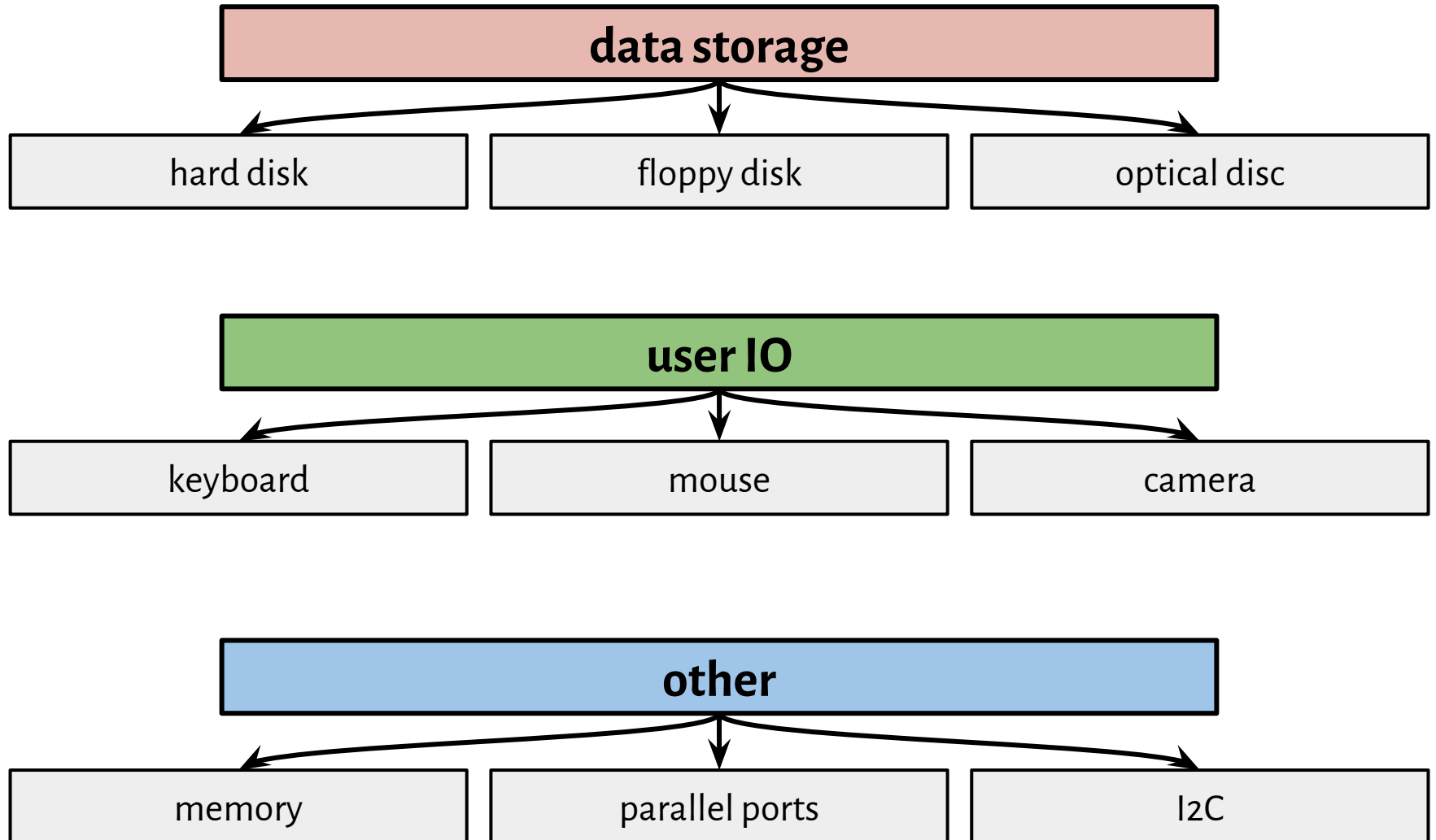
```
$ ls /dev/disk
```

```
by-id/ by-partlabel/ by-partuuid/ by-path/ by-uuid/
```

```
$ ls -l /dev/disk/by-path
```

```
...rwx 1 root root  9 maj 29 10:33 pci-0000:00:1f.2-ata-1 -> ../../sda
...rwx 1 root root 10 maj 29 10:33 pci-0000:00:1f.2-ata-1-part1 -> ../../sda1
...rwx 1 root root 10 maj 29 10:33 pci-0000:00:1f.2-ata-1-part2 -> ../../sda2
...rwx 1 root root 10 maj 29 10:33 pci-0000:00:1f.2-ata-1-part3 -> ../../sda3
...rwx 1 root root  9 maj 29 10:33 pci-0000:00:1f.2-ata-2 -> ../../sr0
```

Devices - examples



Devices - kinds

Block devices

it is possible to read any byte on a disk at any time

Character devices

it is only possible to read the next byte

Devices - kinds

Block devices

it is possible to read any byte on a disk at any time

reads are cached, writes are buffered

Character devices

it is only possible to read the next byte

but: can be applied to a piece of block-based hardware

Devices - kinds

Block devices are gone (FreeBSD)

it is possible to read any byte on a disk at any time

reads are cached, writes are buffered

<https://www.freebsd.org/doc/en/books/arch-handbook/driverbasics-block.html>

Character devices

it is only possible to read the next byte

but: can be applied to a piece of block-based hardware

Devices - kinds

Block devices are gone (FreeBSD)

it is possible to read any byte on a disk at any time

reads are cached, writes are buffered

<https://www.freebsd.org/doc/en/books/arch-handbook/driverbasics-block.html>

Character devices are not efficient (Linux)


it is only possible to read the next byte

but: can be applied to a piece of block-based hardware

https://en.wikipedia.org/wiki/Raw_device

Devices

```
$ ls -l /dev
```

kind									name
 brw-rw----	1	root	disk	8,	0	maj	27	09:50	sda
lrwxrwxrwx	1	root	root	15	maj	27	09:50		stderr -> /proc/self/fd/2
lrwxrwxrwx	1	root	root	15	maj	27	09:50		stdin -> /proc/self/fd/0
lrwxrwxrwx	1	root	root	15	maj	27	09:50		stdout -> /proc/self/fd/1
crw-rw-rw-	1	root	tty	5,	0	maj	27	09:50	tty
crw--w----	1	root	tty	4,	0	maj	27	09:50	tty0
crw--w----	1	root	tty	4,	1	maj	27	09:50	tty1
crw--w----	1	root	tty	4,	10	maj	27	09:50	tty10
crw-r-----	1	root	kmem	1,	1	maj	27	09:50	mem
crw-rw-rw-	1	root	root	1,	3	maj	27	09:50	null
crw-rw-rw-	1	root	root	1,	5	maj	27	09:50	zero
crw-rw-rw-	1	root	root	1,	8	maj	27	09:50	random

Devices

```
$ ls -l /dev
```

kind			major number						name
brw-rw----	1	root	disk	8,	0	maj	27	09:50	sda
lrwxrwxrwx	1	root	root	15	maj	27	09:50		stderr -> /proc/self/fd/2
lrwxrwxrwx	1	root	root	15	maj	27	09:50		stdin -> /proc/self/fd/0
lrwxrwxrwx	1	root	root	15	maj	27	09:50		stdout -> /proc/self/fd/1
crw-rw-rw-	1	root	tty	5,	0	maj	27	09:50	tty
crw--w----	1	root	tty	4,	0	maj	27	09:50	tty0
crw--w----	1	root	tty	4,	1	maj	27	09:50	tty1
crw--w----	1	root	tty	4,	10	maj	27	09:50	tty10
crw-r-----	1	root	kmem	1,	1	maj	27	09:50	mem
crw-rw-rw-	1	root	root	1,	3	maj	27	09:50	null
crw-rw-rw-	1	root	root	1,	5	maj	27	09:50	zero
crw-rw-rw-	1	root	root	1,	8	maj	27	09:50	random

Devices

```
$ ls -l /dev
```

kind		major number	minor number	name
brw-rw----	1 root disk	8,	0 maj 27 09:50	sda
lrwxrwxrwx	1 root root	15 maj 27 09:50	stderr -> /proc/self/fd/2	
lrwxrwxrwx	1 root root	15 maj 27 09:50	stdin -> /proc/self/fd/0	
lrwxrwxrwx	1 root root	15 maj 27 09:50	stdout -> /proc/self/fd/1	
crw-rw-rw-	1 root tty	5,	0 maj 27 09:50	tty
crw--w----	1 root tty	4,	0 maj 27 09:50	tty0
crw--w----	1 root tty	4,	1 maj 27 09:50	tty1
crw--w----	1 root tty	4,	10 maj 27 09:50	tty10
crw-r-----	1 root kmem	1,	1 maj 27 09:50	mem
crw-rw-rw-	1 root root	1,	3 maj 27 09:50	null
crw-rw-rw-	1 root root	1,	5 maj 27 09:50	zero
crw-rw-rw-	1 root root	1,	8 maj 27 09:50	random

Drivers

Drivers

a computer program that

Drivers

a computer program that
operates or controls

Drivers

a computer program that

operates or controls

a particular type of device that is attached to a computer

Drivers

a computer program that

operates or controls

a particular type of device that is attached to a computer

- ★ Drivers communicate with devices directly (Linux) or via microkernel (Minix).
- ★ Drivers are **hardware dependent** and **operating-system-specific**.

Preparation

- ★ Open the scenario:

```
home/students/inf/PUBLIC/SO/scenariusze/11
```

- ★ Find bonus slides:

```
www.mimuw.edu.pl~/inga/SO2018/
```

- ★ Run MINIX on qemu

<http://wiki.minix3.org/doku.php?id=developersguide:driverprogramming>