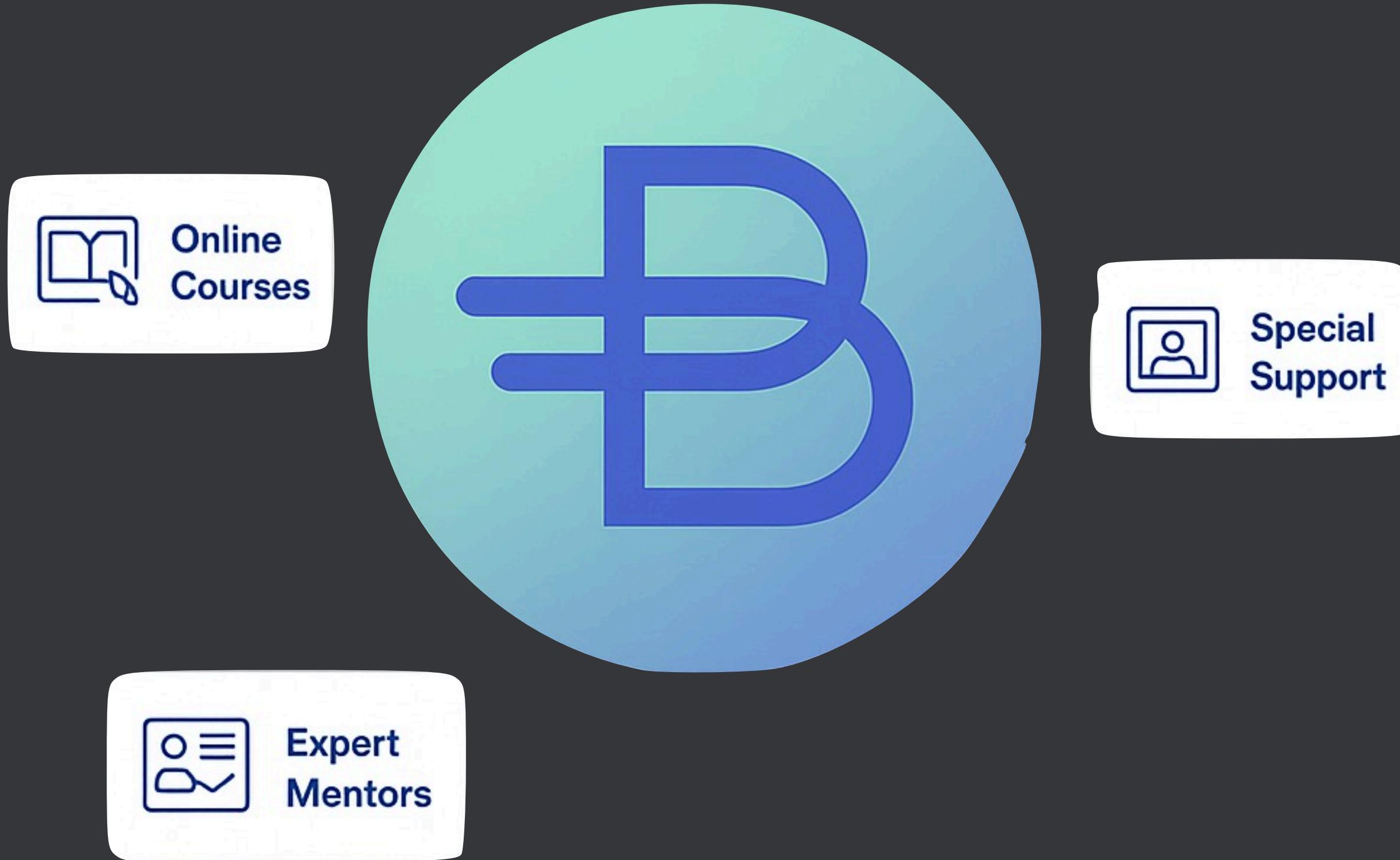
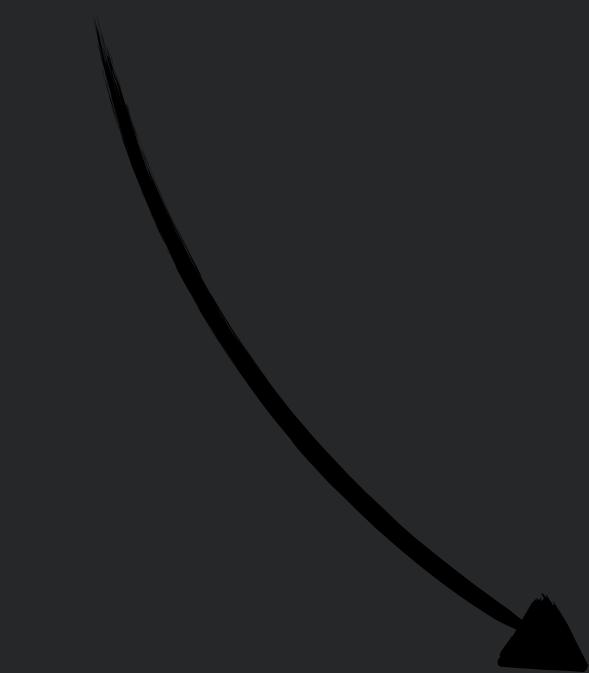


Bullet's Embedded Linux Level 1



Presented by: Hazem Khaled

Session Flow



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Introduction to Linux
Stacks

File system Stack

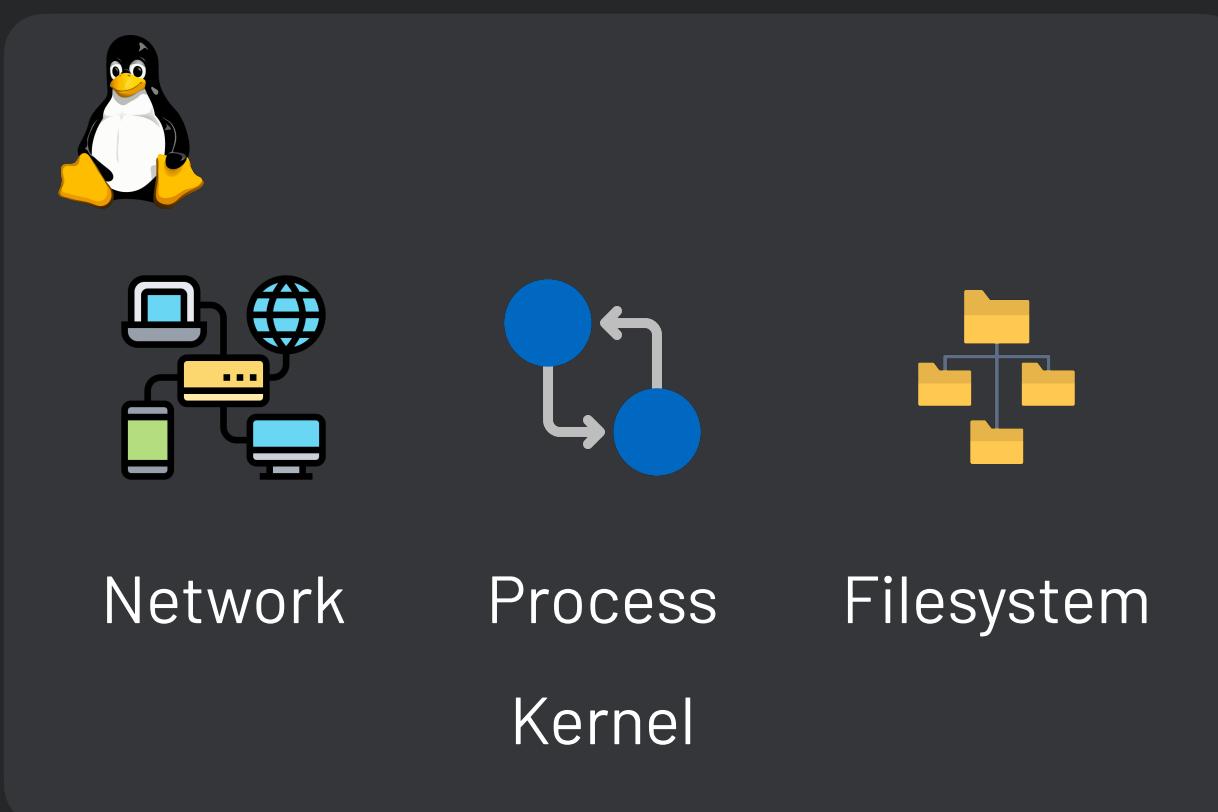
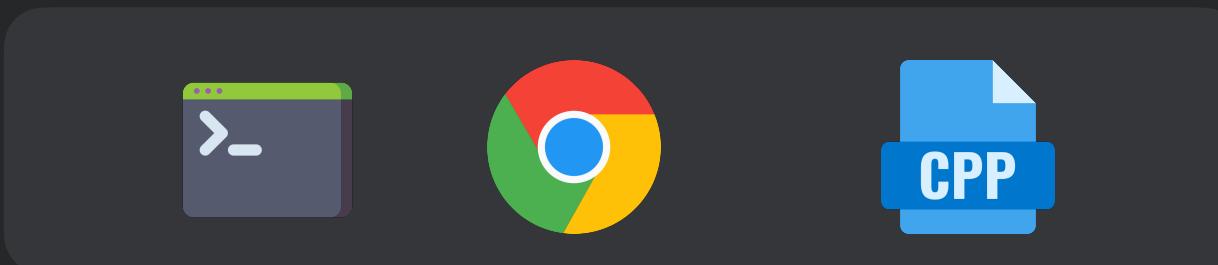
In session exercises

Process Management
Stack

In session exercises

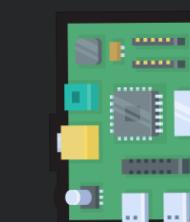


Is a software
→



Operating system = Userspace +
Kernel

Device Drivers



Introduction to Linux
Stacks

File system Stack

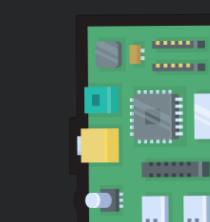
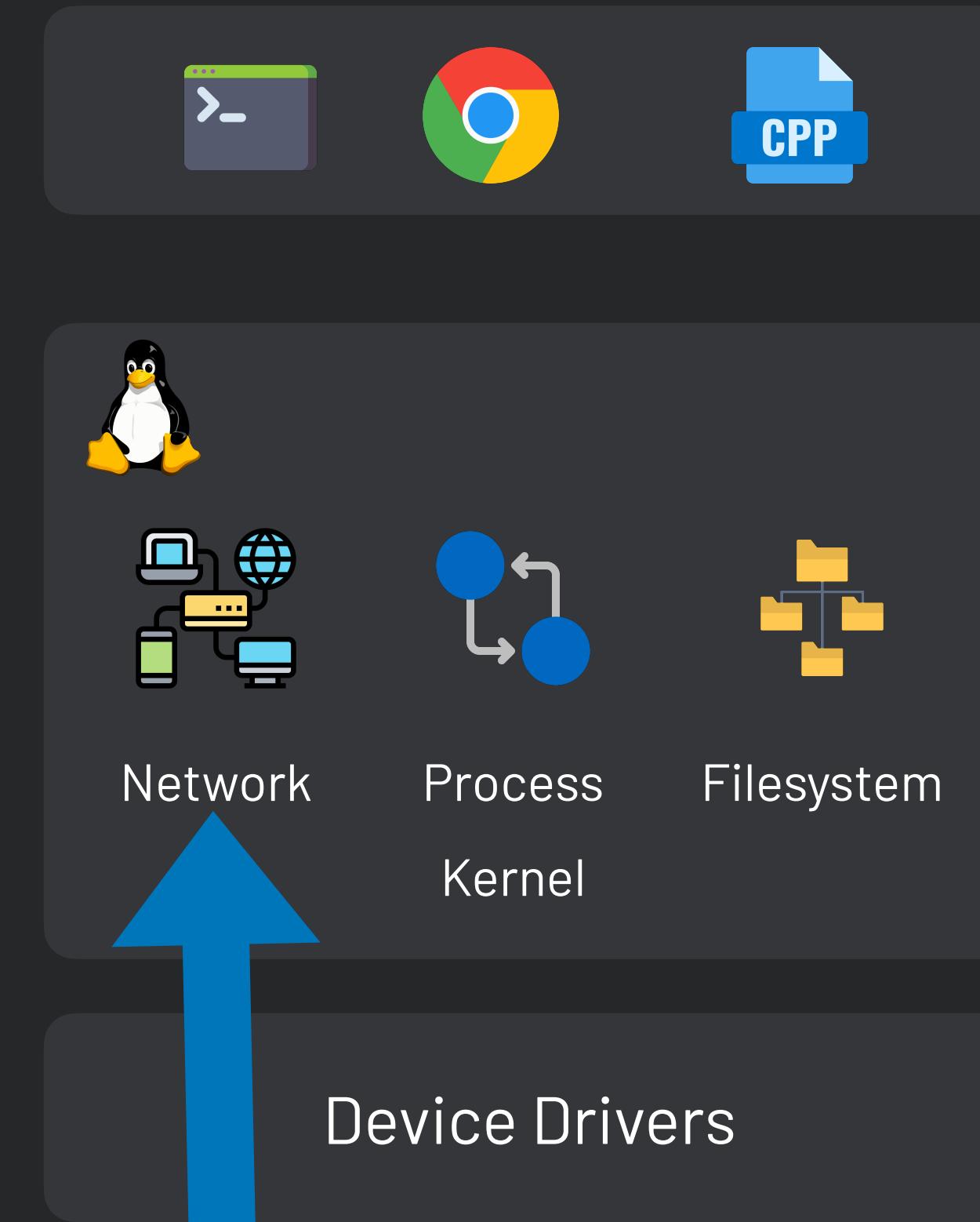
In session exercises

Process Management
Stack

In session exercises



Is a software
→



Operating system = Userspace +
Kernel

Introduction to Linux
Stacks

File system Stack

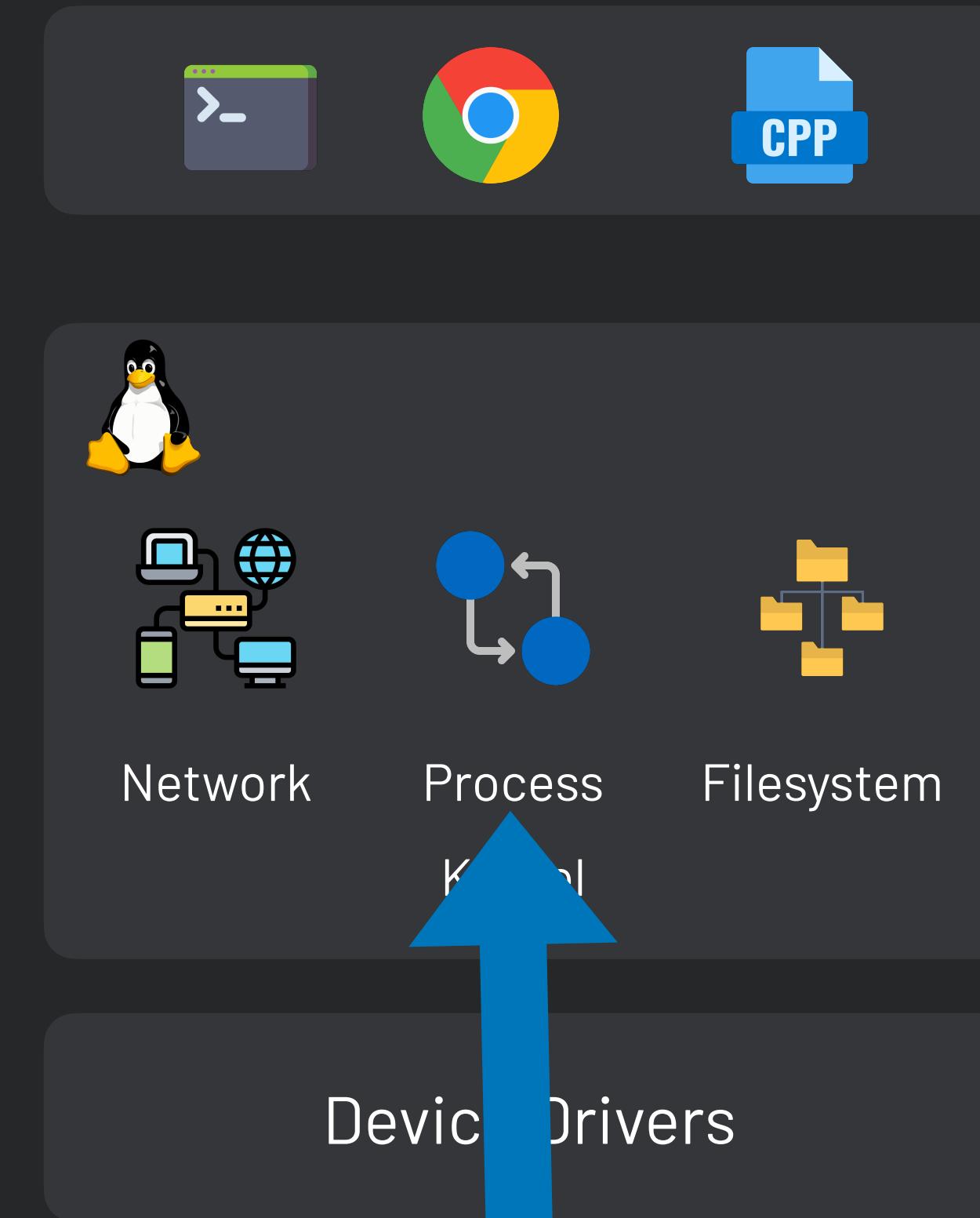
In session exercises

Process Management
Stack

In session exercises



Is a software
→



Operating system = Userspace +
Kernel

Introduction to Linux
Stacks

File system Stack

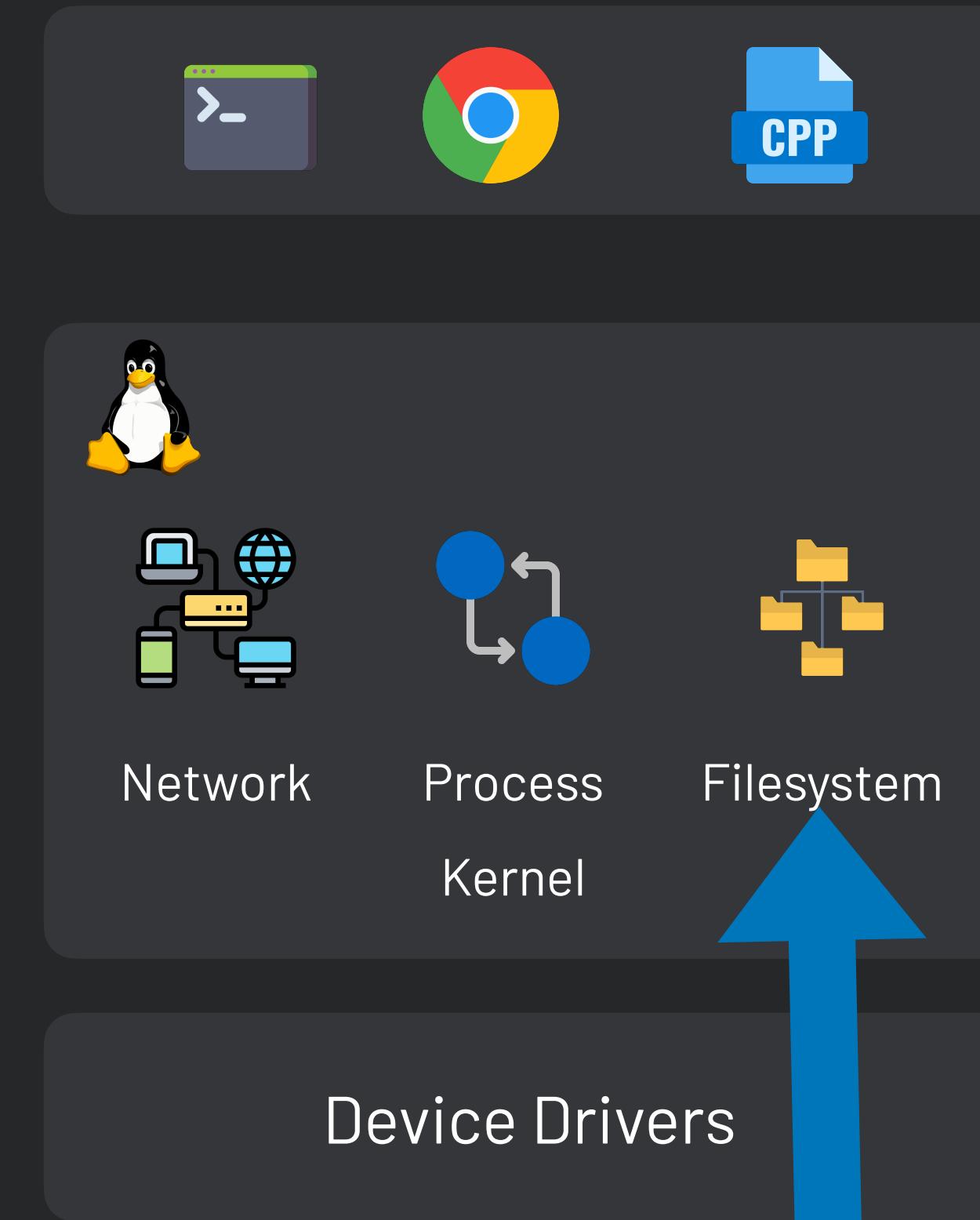
In session exercises

Process Management
Stack

In session exercises



Is a software
→



Operating system = Userspace +
Kernel

Introduction to Linux
Stacks

File system Stack

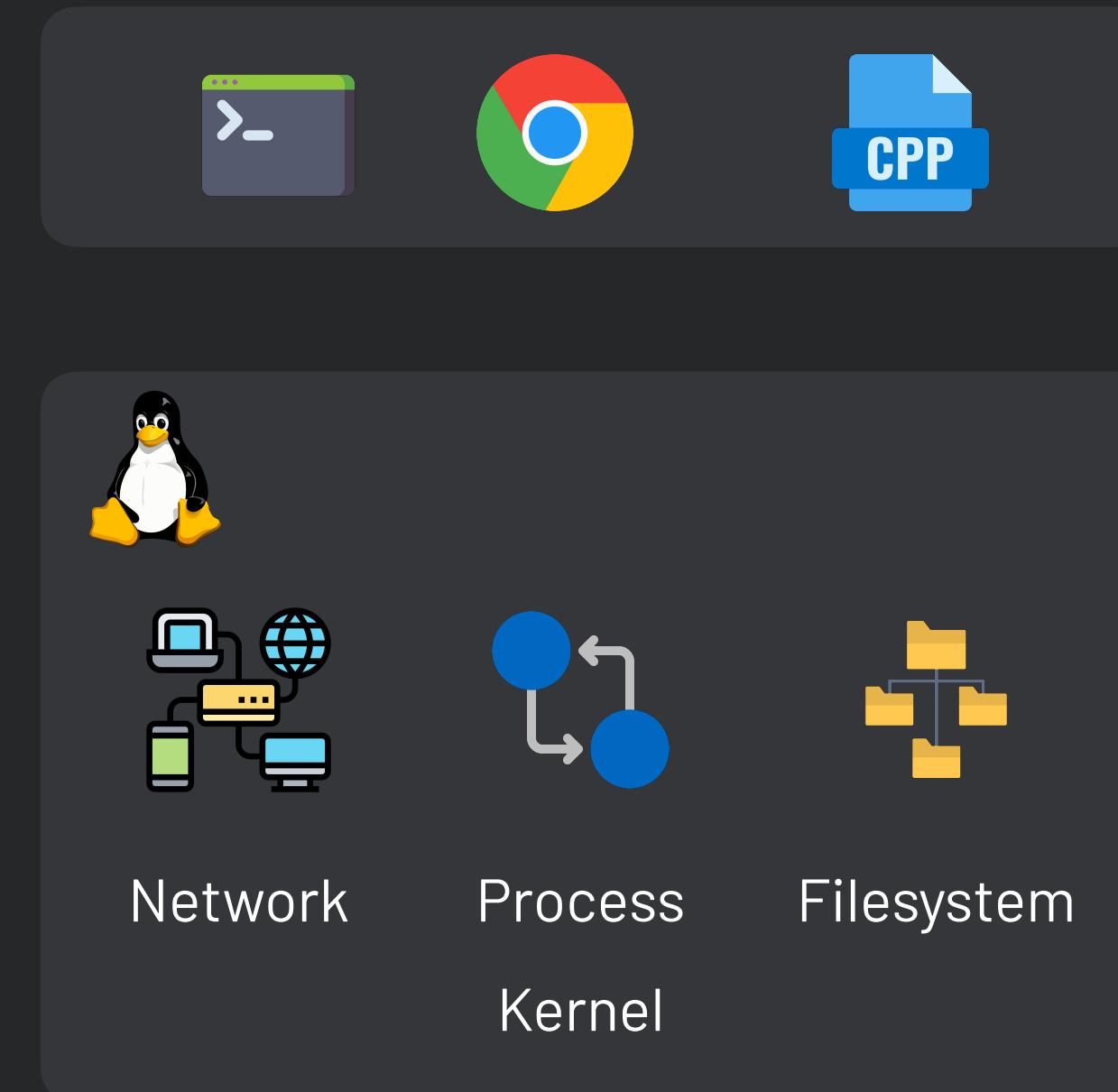
In session exercises

Process Management
Stack

In session exercises



Is a software
→



Operating system = Userspace +
Kernel

Introduction to Linux
Stacks

File system Stack

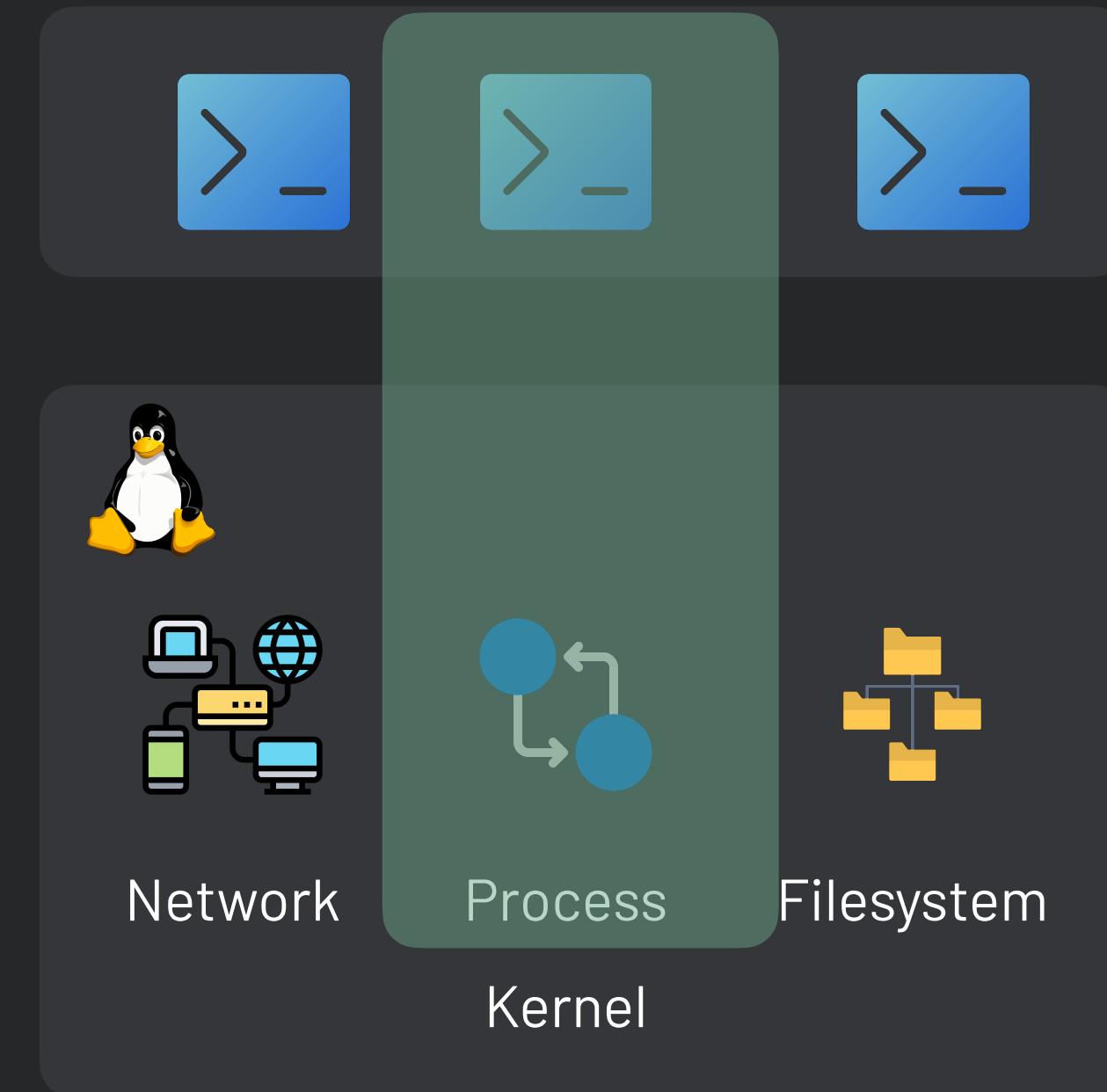
In session exercises

Process Management
Stack

In session exercises

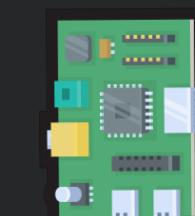


Is a software
→



Operating system = Userspace +
Kernel

Device Drivers



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

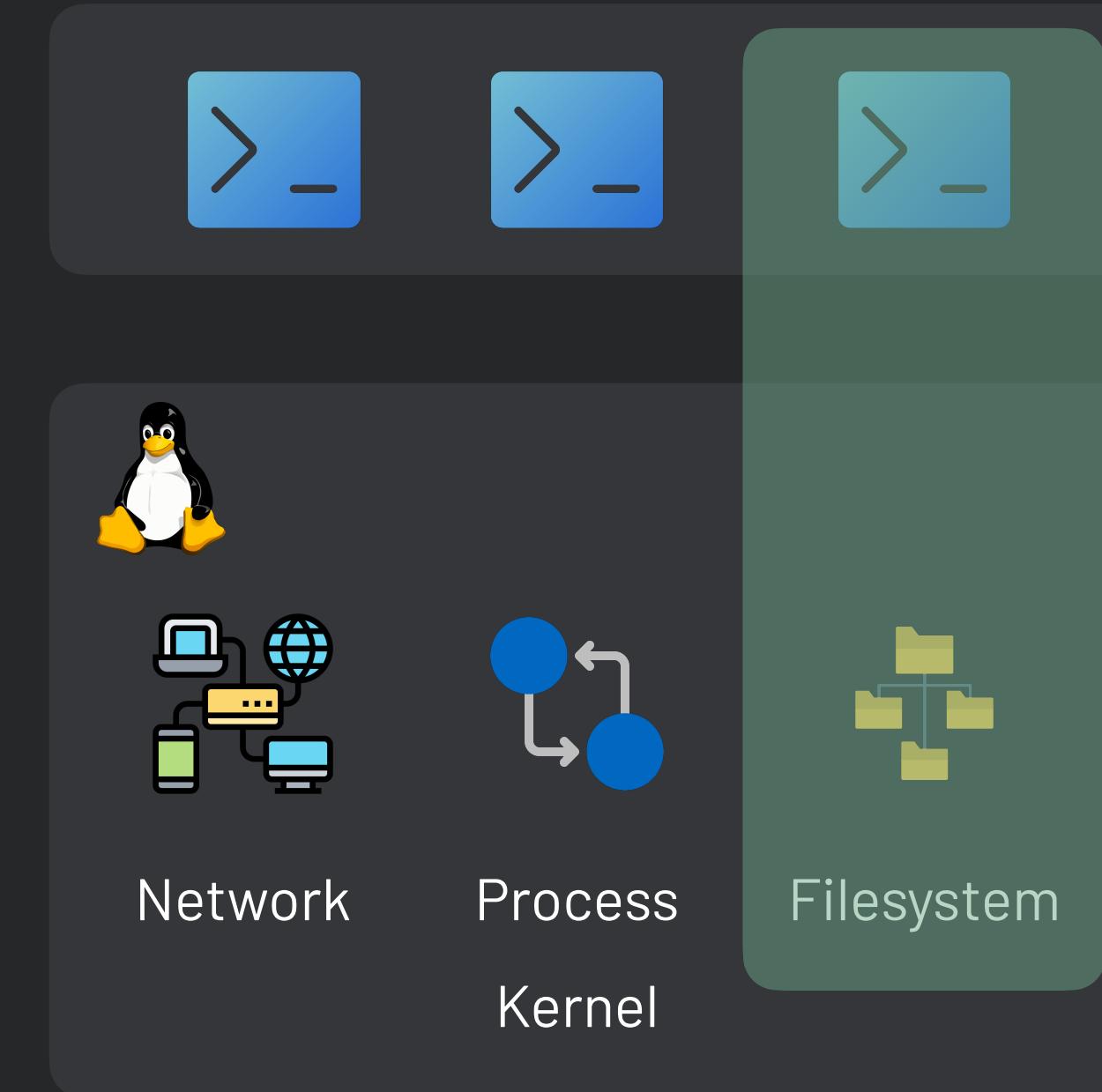
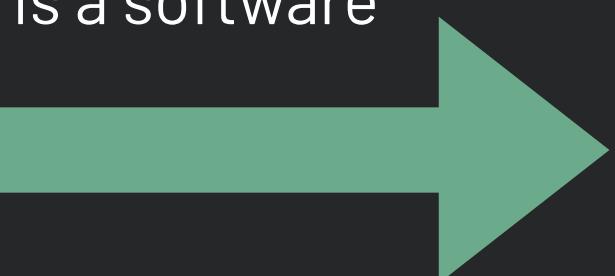
In session exercises

1

Each stack has associated commands
in user-space



Is a software



Device Drivers

Operating system = Userspace +
Kernel

Introduction to Linux
Stacks

File system Stack

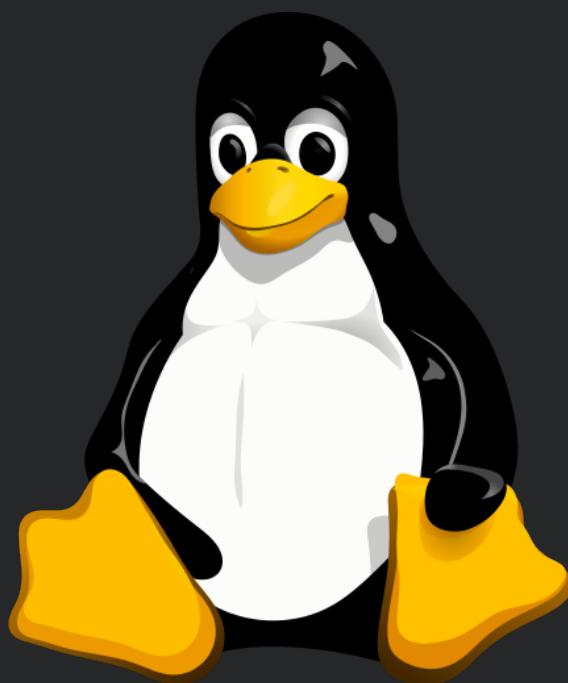
In session exercises

Process Management
Stack

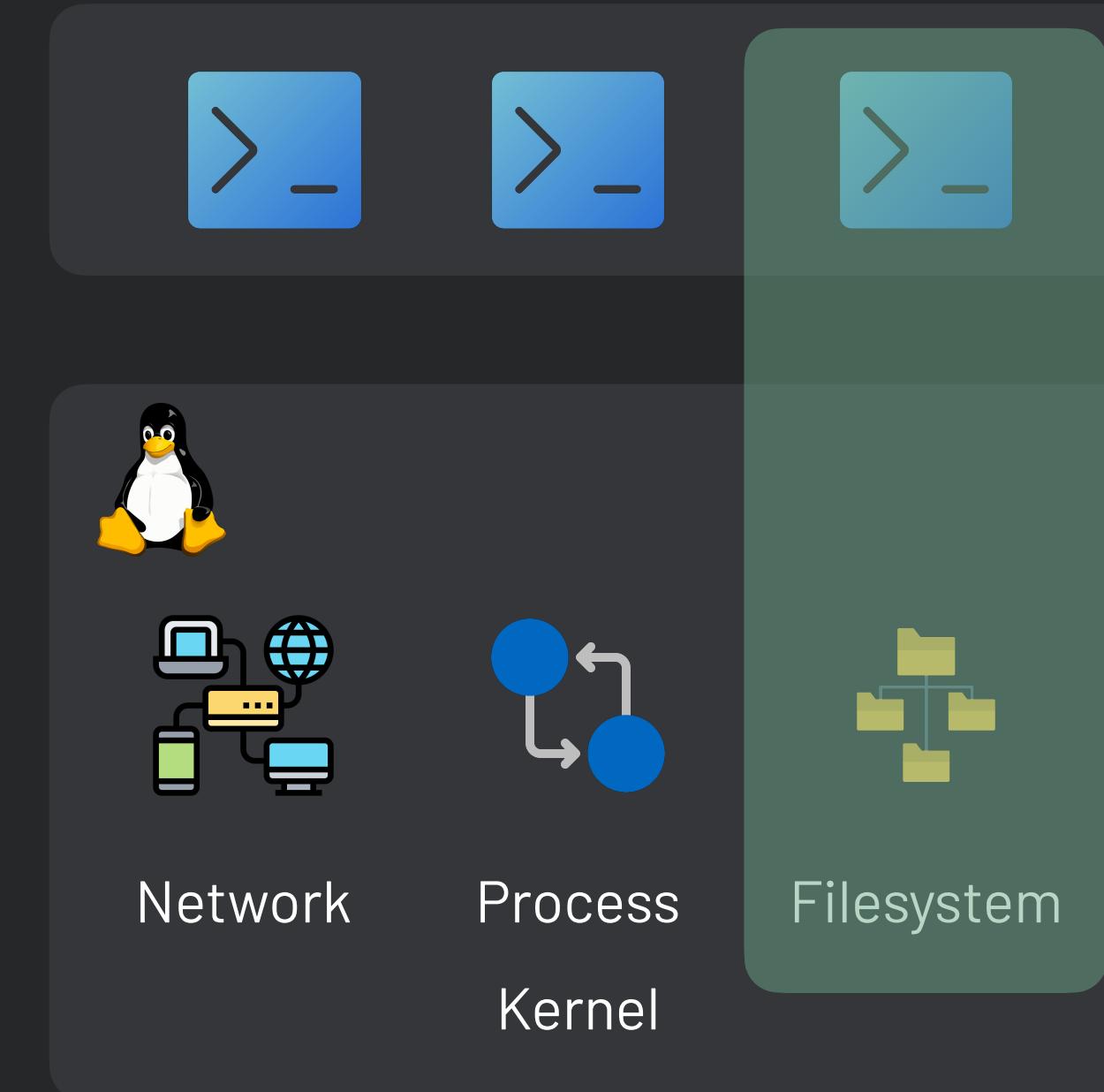
In session exercises

1

Each stack has associated commands
in user-space

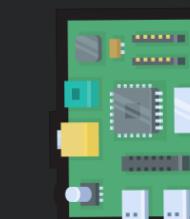


Is a software



Operating system = Userspace +
Kernel

Device Drivers



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

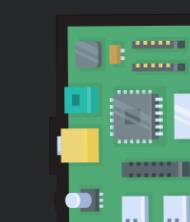
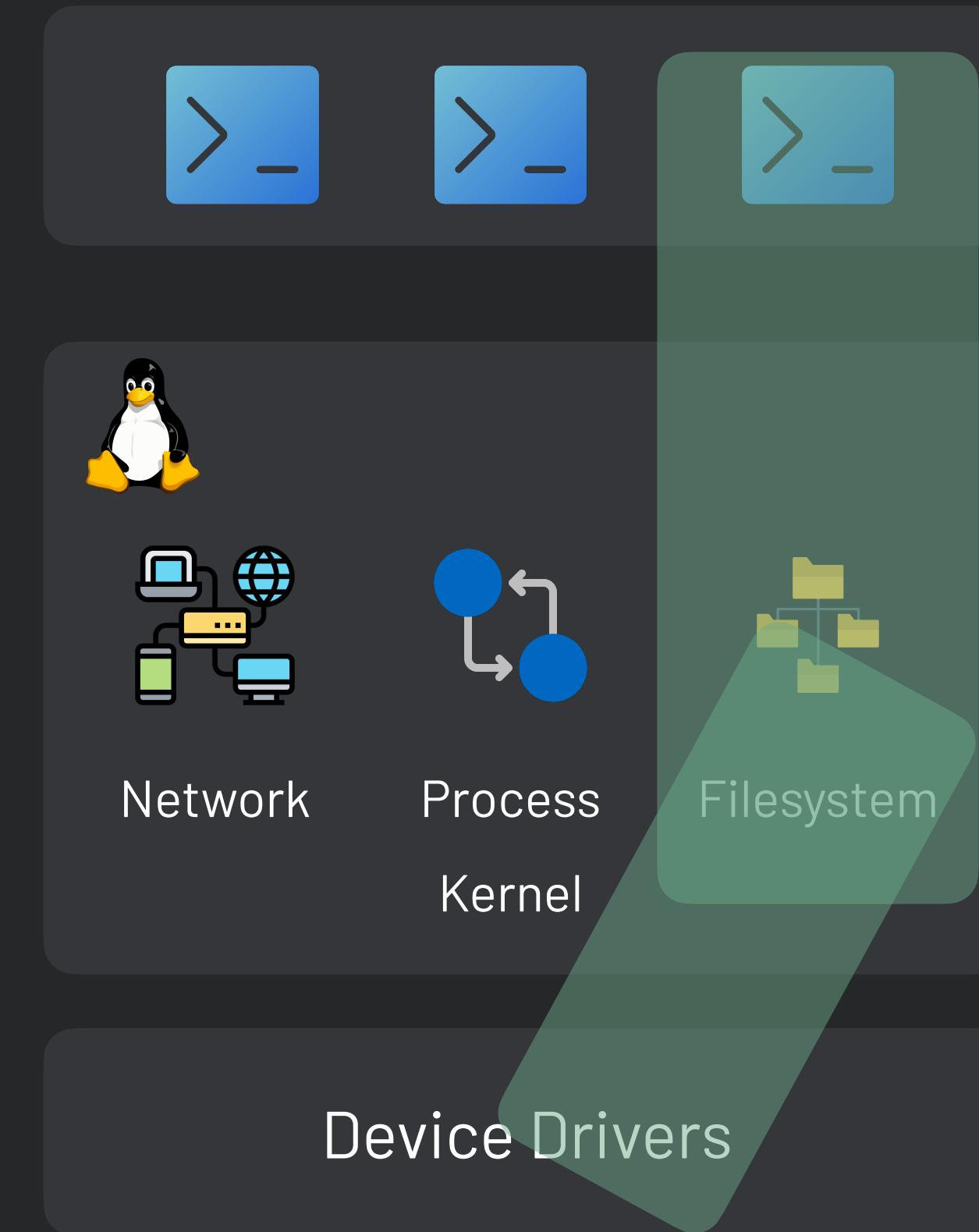
In session exercises

2

Each stack attached with Hardware
Driver for Hardware Interaction



Is a software
→



Introduction to Linux
Stacks

File system Stack

In session exercises

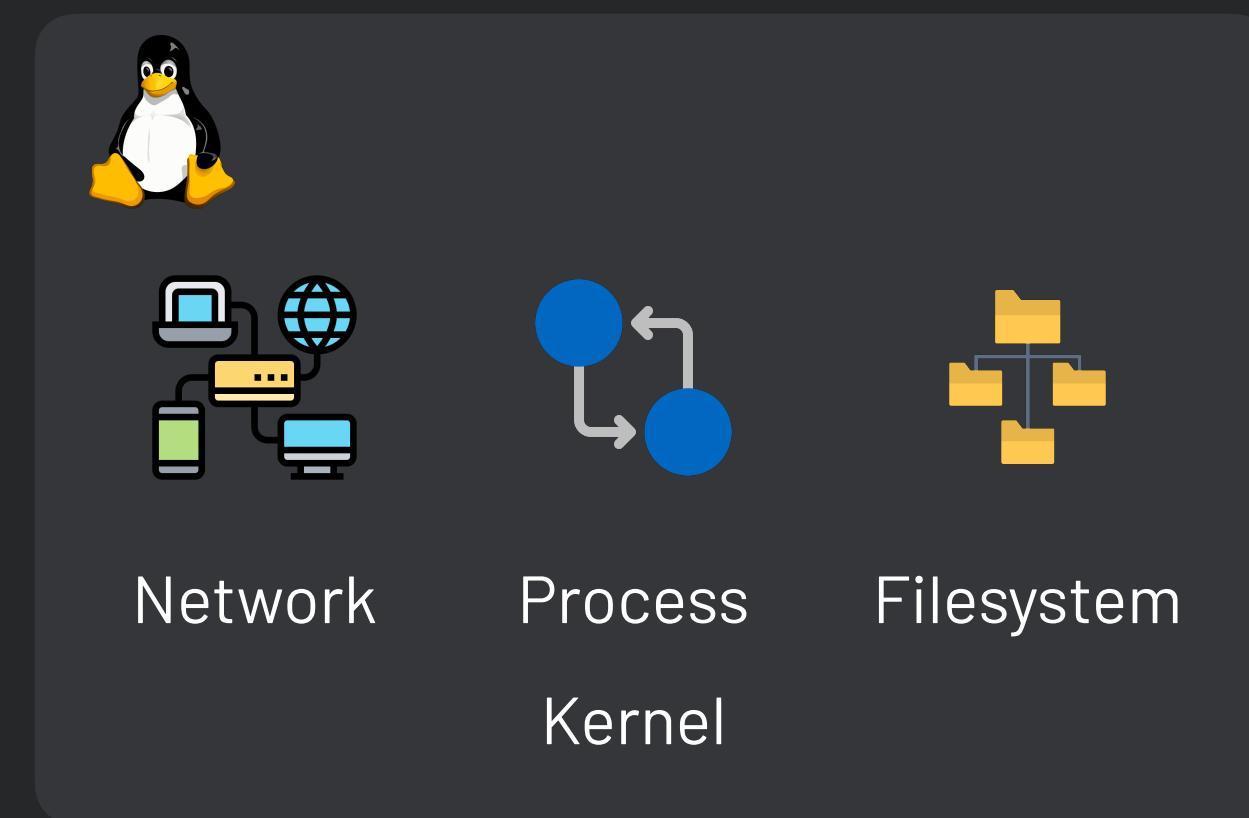
Process Management
Stack

In session exercises

Command Execution flow



Is a software
→



Device Drivers

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

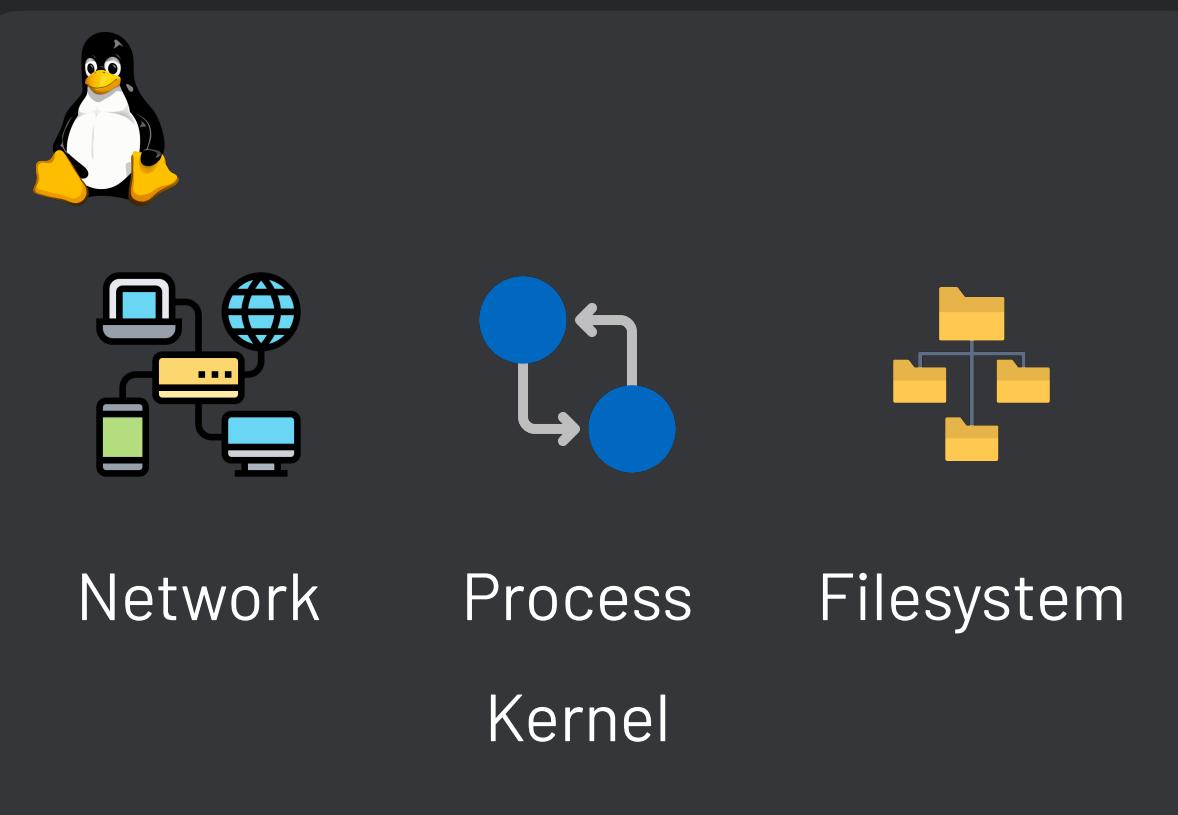
In session exercises

Command Execution flow

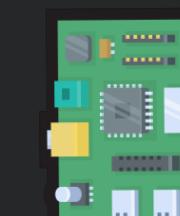


ls a software
→

ls



Device Drivers



Introduction to Linux
Stacks

File system Stack

In session exercises

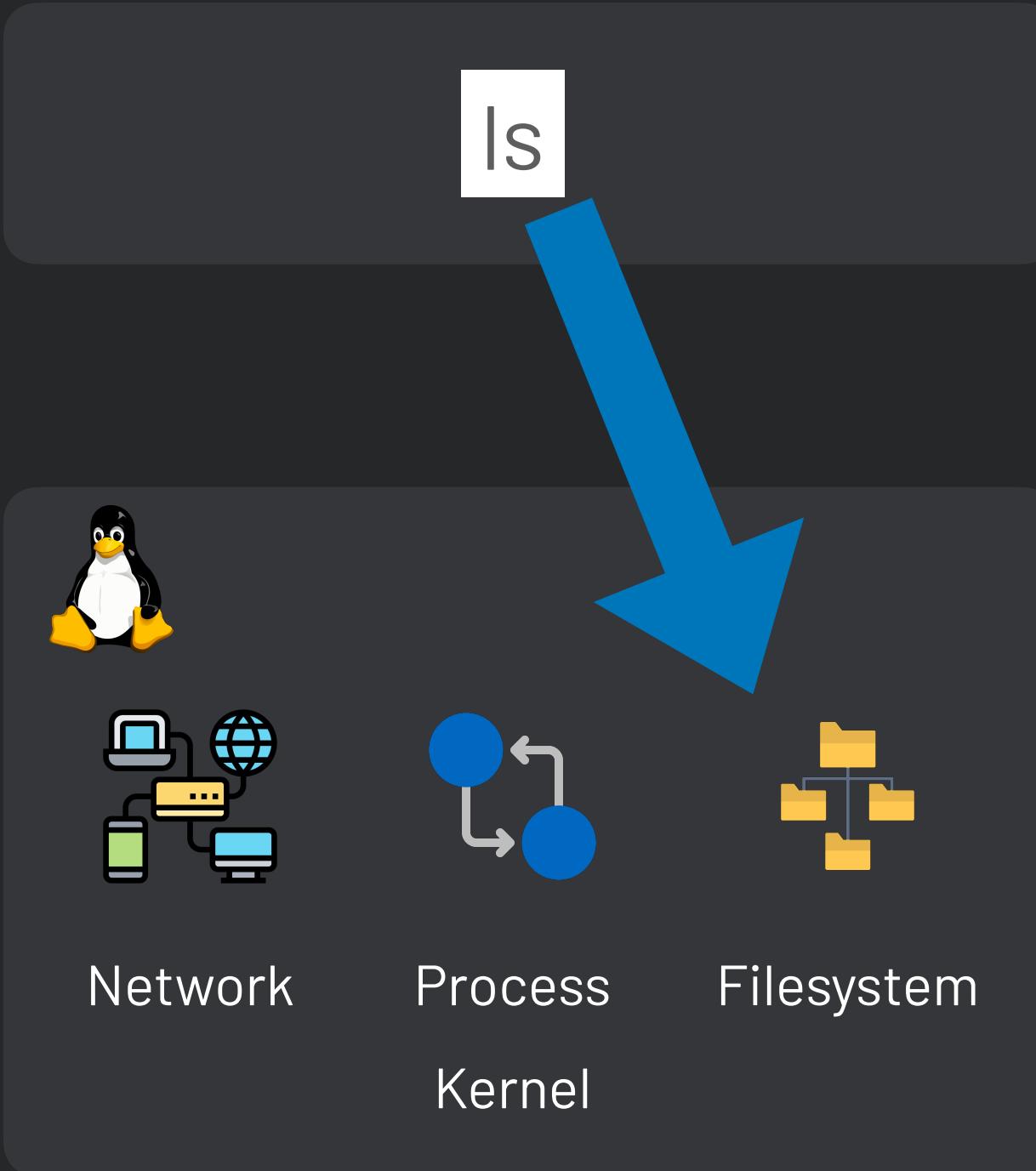
Process Management
Stack

In session exercises

Command Execution flow



ls a software
→



Device Drivers

Introduction to Linux
Stacks

File system Stack

In session exercises

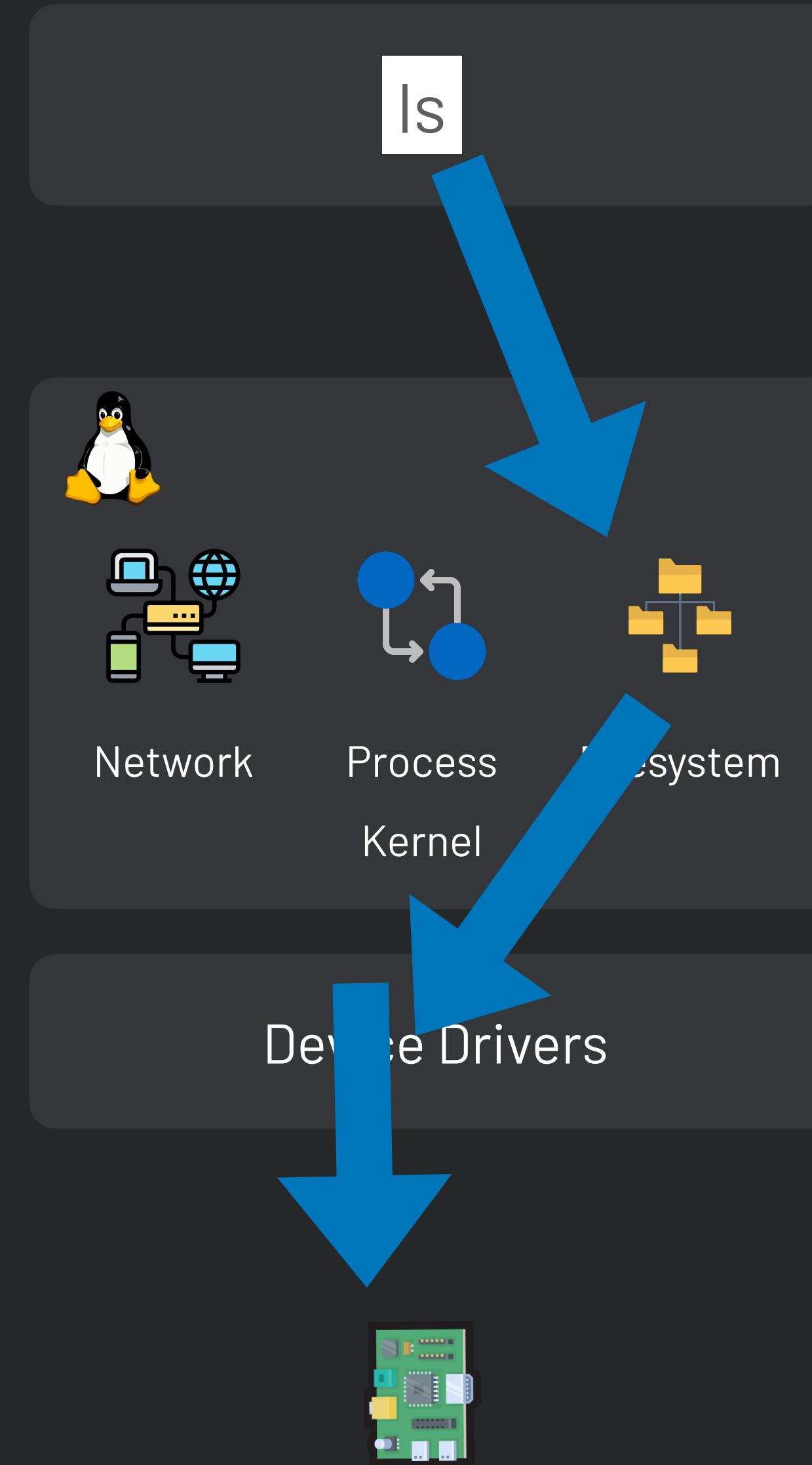
Process Management
Stack

In session exercises

Command Execution flow



ls a software
→



Introduction to Linux
Stacks

File system Stack

In session exercises

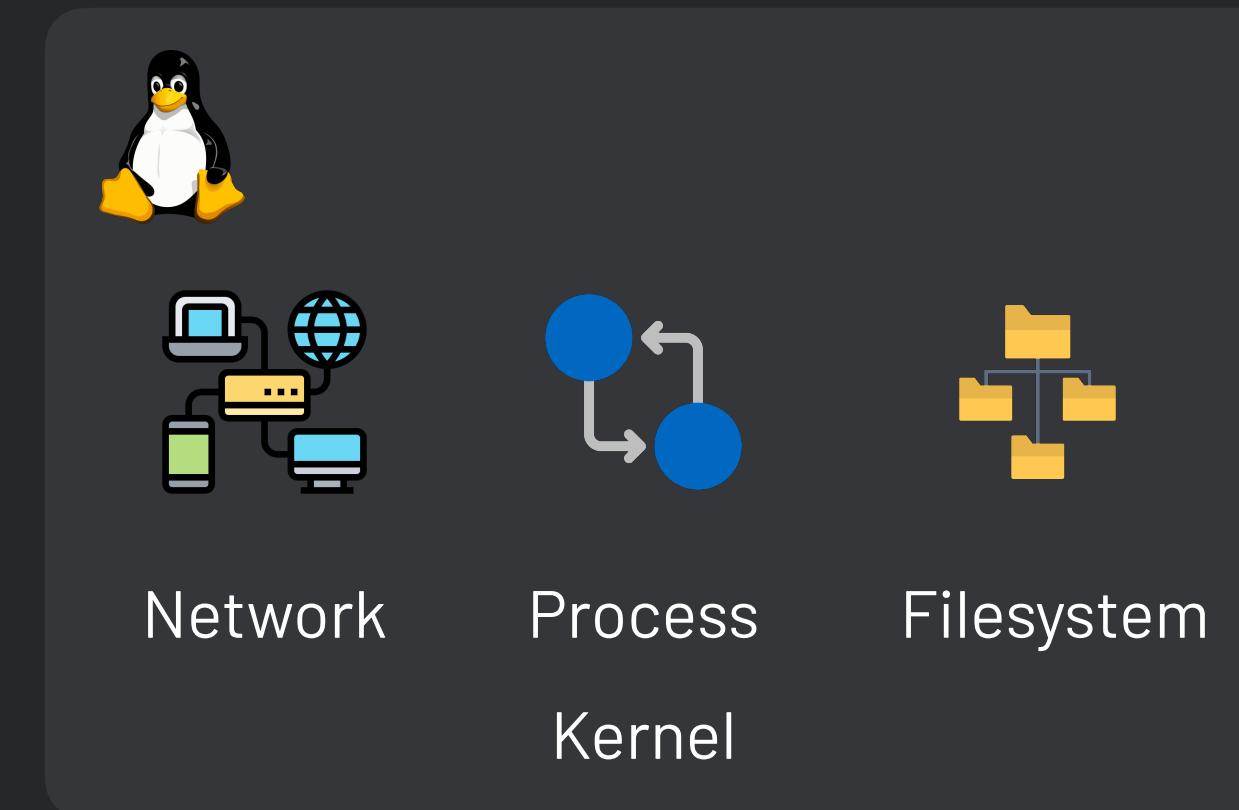
Process Management
Stack

In session exercises

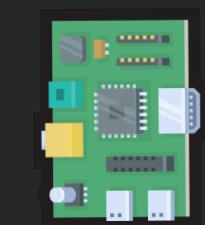
Command Execution flow



ls a software
→



Device Drivers



Introduction to Linux
Stacks

File system Stack

In session exercises

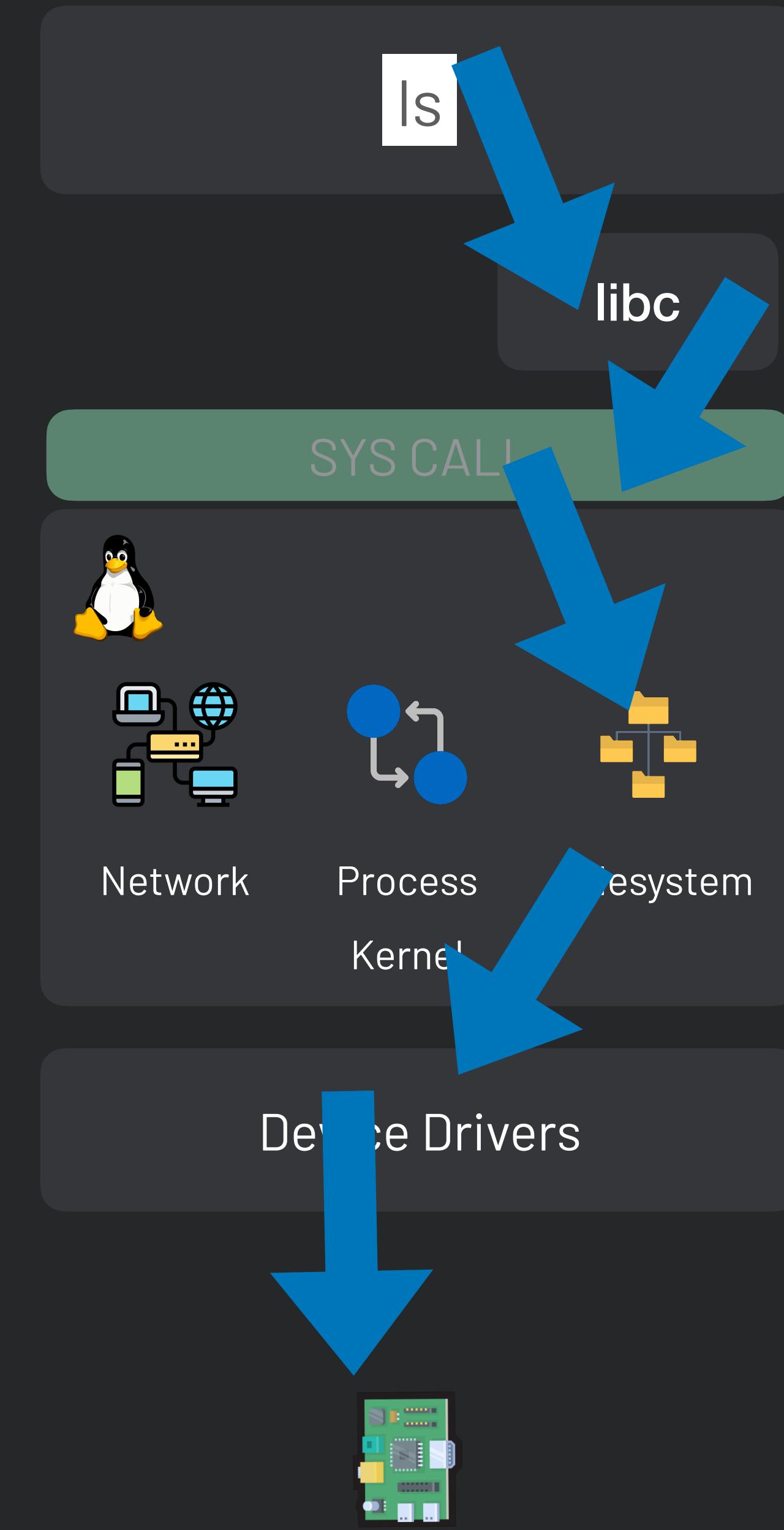
Process Management
Stack

In session exercises

Command Execution flow



ls a software
→



Introduction to Linux
Stacks

File system Stack

In session exercises

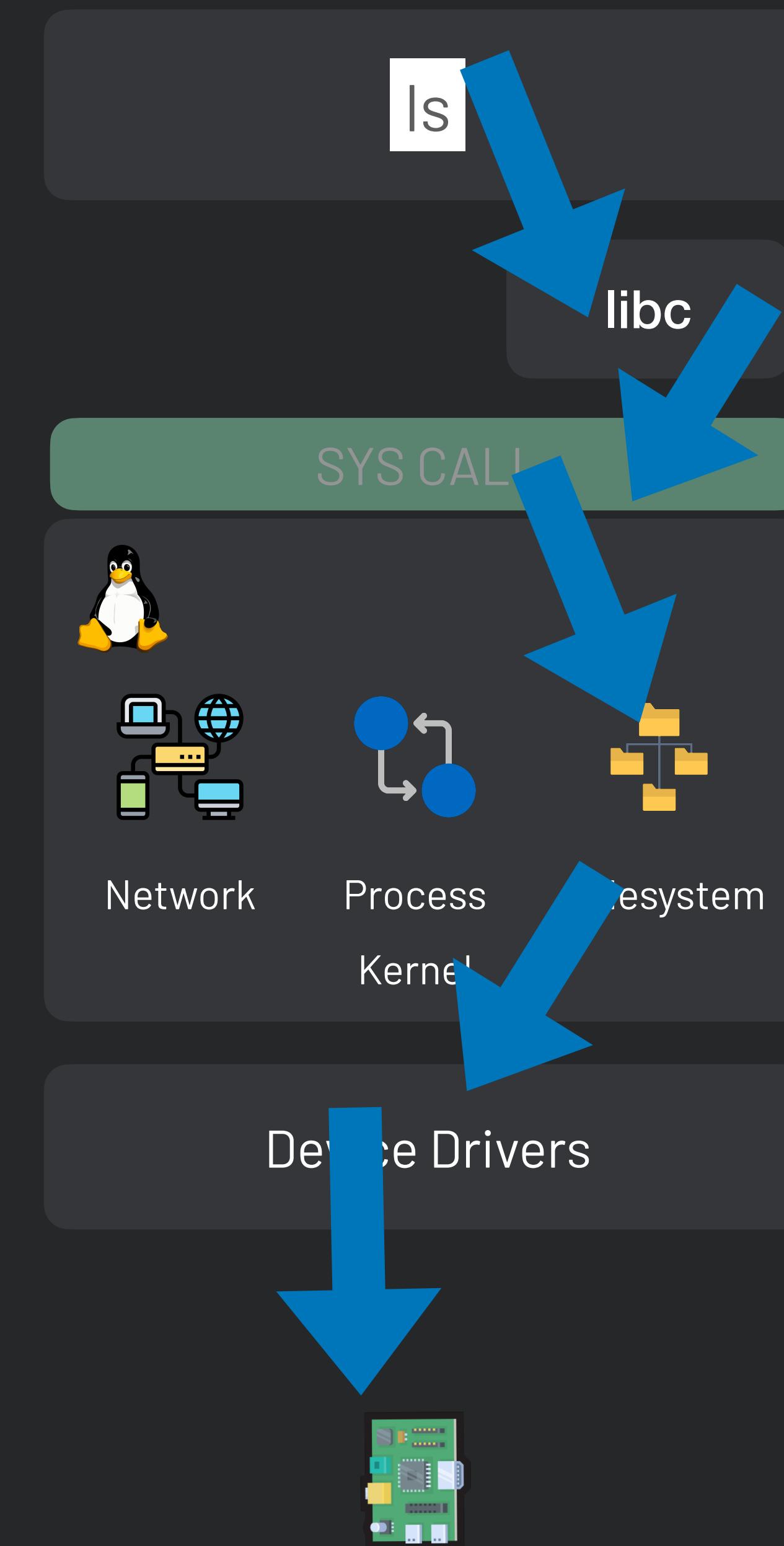
Process Management
Stack

In session exercises

Command Execution flow



ls a software
→



→ System programming Engineer

Introduction to Linux
Stacks

File system Stack

In session exercises

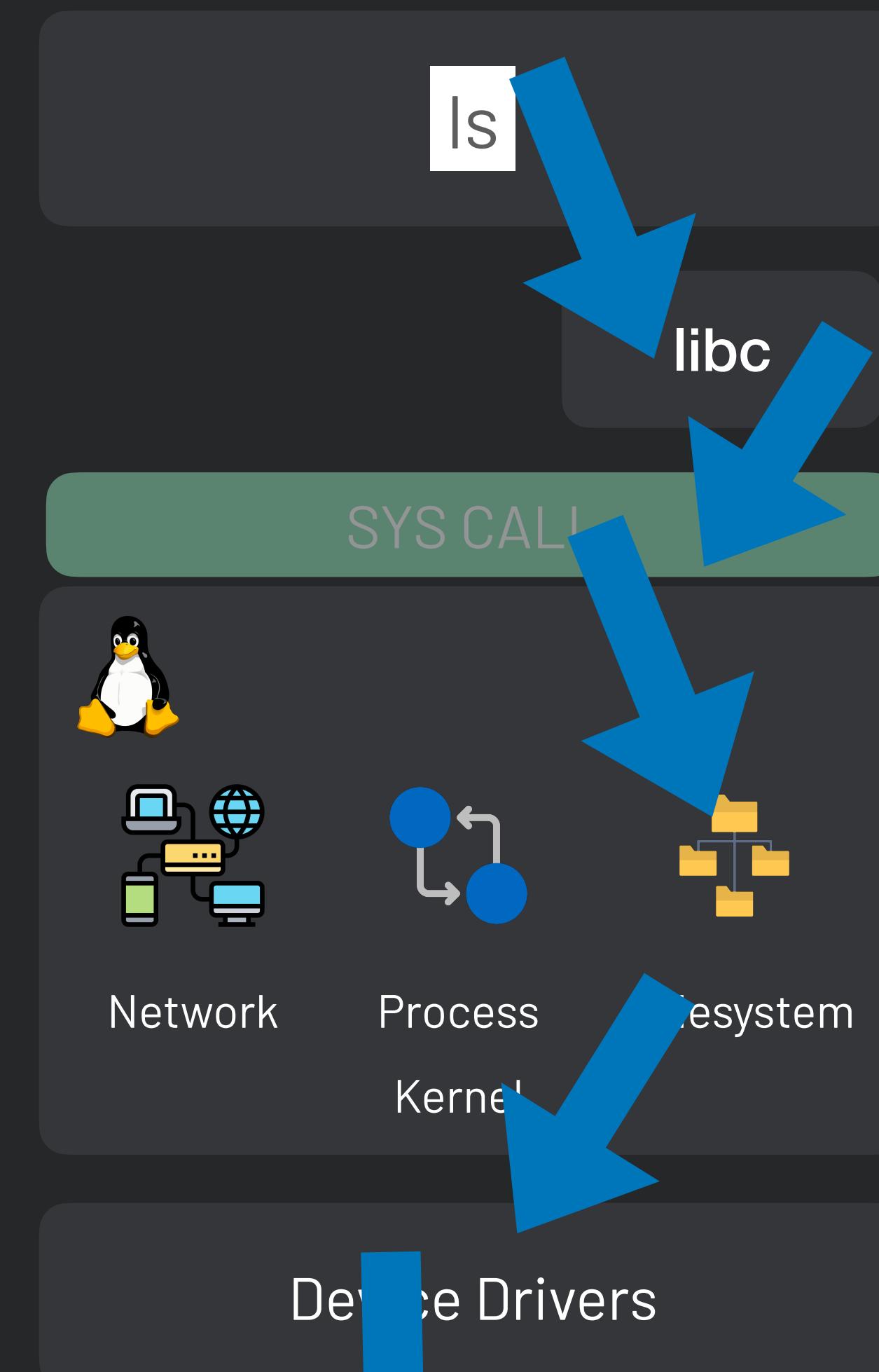
Process Management
Stack

In session exercises

Command Execution flow



ls a software
→



→ System programming Engineer

→ Debugging skills

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

For each stack :

**Stack Definition /
Functions**

Commands

Interaction Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Stack
Functions

Commands

Interaction
Hardware



Directories

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Stack
Functions

Commands

Interaction
Hardware



Introduction to Linux
Stacks

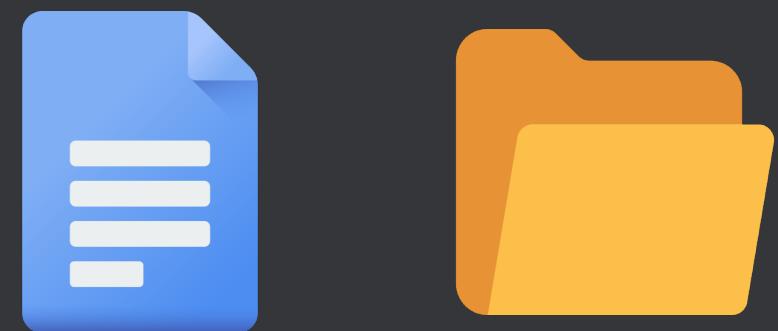
File system Stack

In session exercises

Process Management
Stack

In session exercises

Userspace



Stack
Functions

Commands

Interaction
Hardware



Introduction to Linux
Stacks

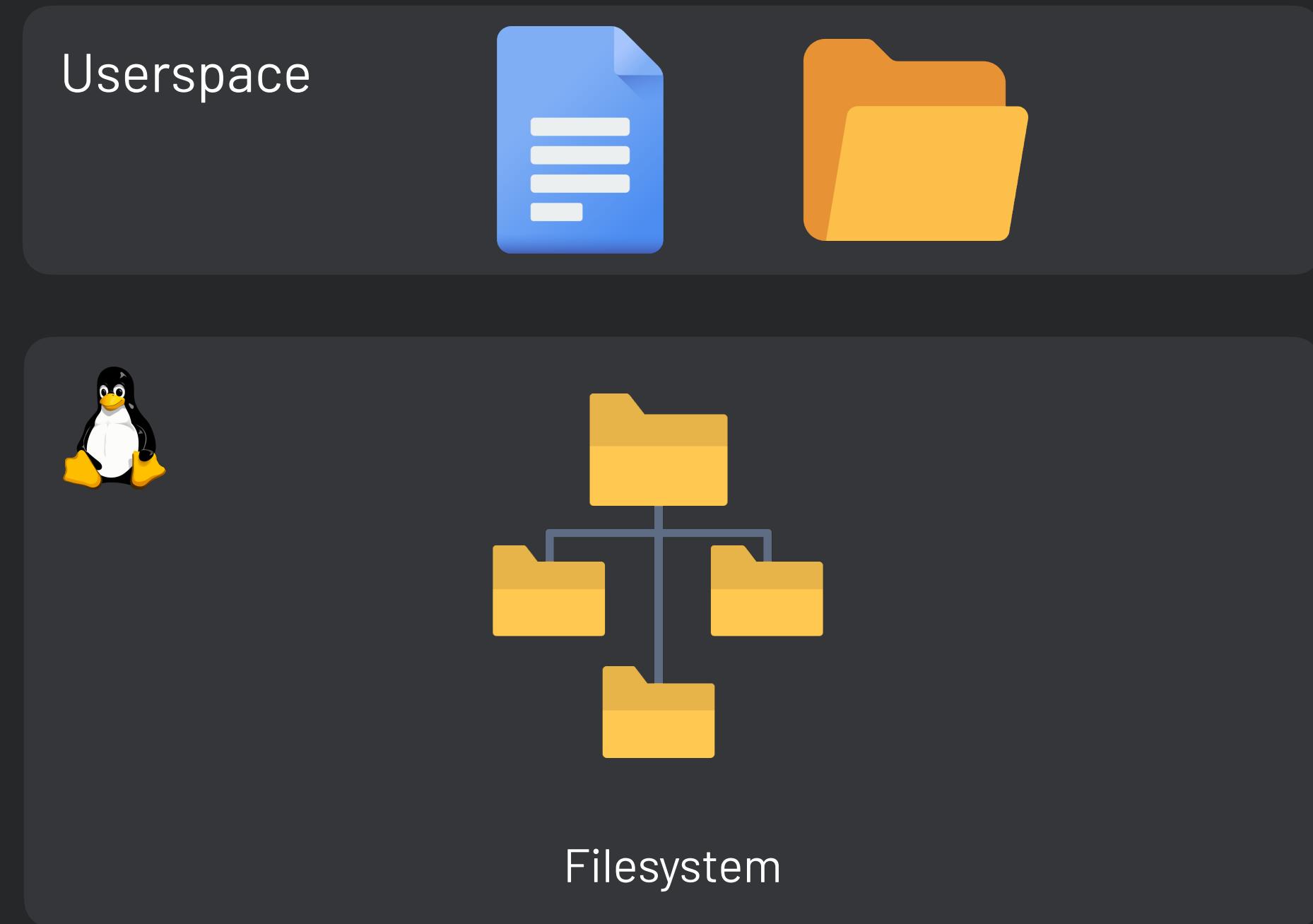
File system Stack

In session exercises

Process Management
Stack

In session exercises

Change view from user space /
storage



Stack
Functions

Commands

Interaction
Hardware



Introduction to Linux
Stacks

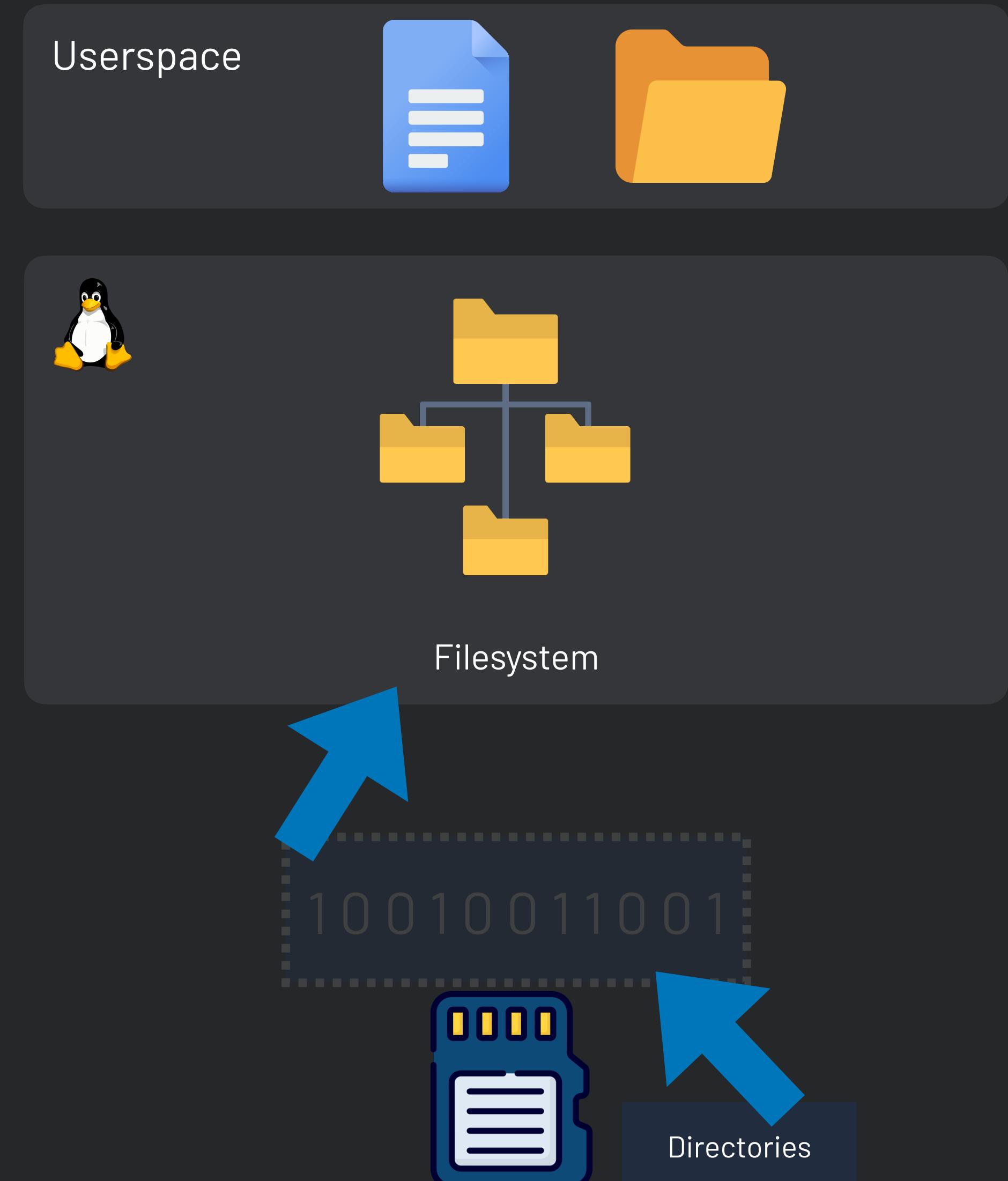
File system Stack

In session exercises

Process Management
Stack

In session exercises

Change view from user space /
storage



Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

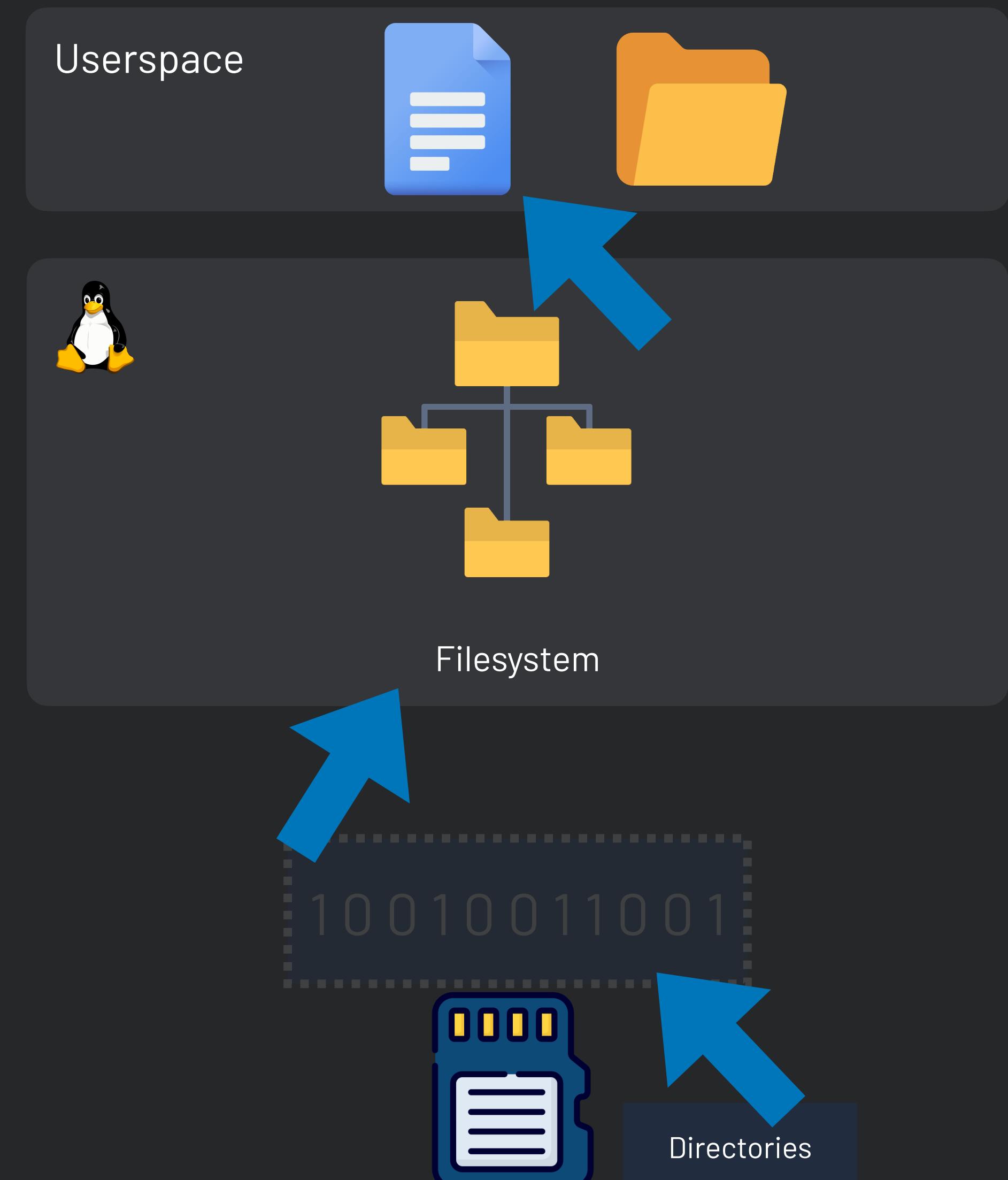
Stack
Functions

Commands

Interaction
Hardware

1

Change view from user space /
storage



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

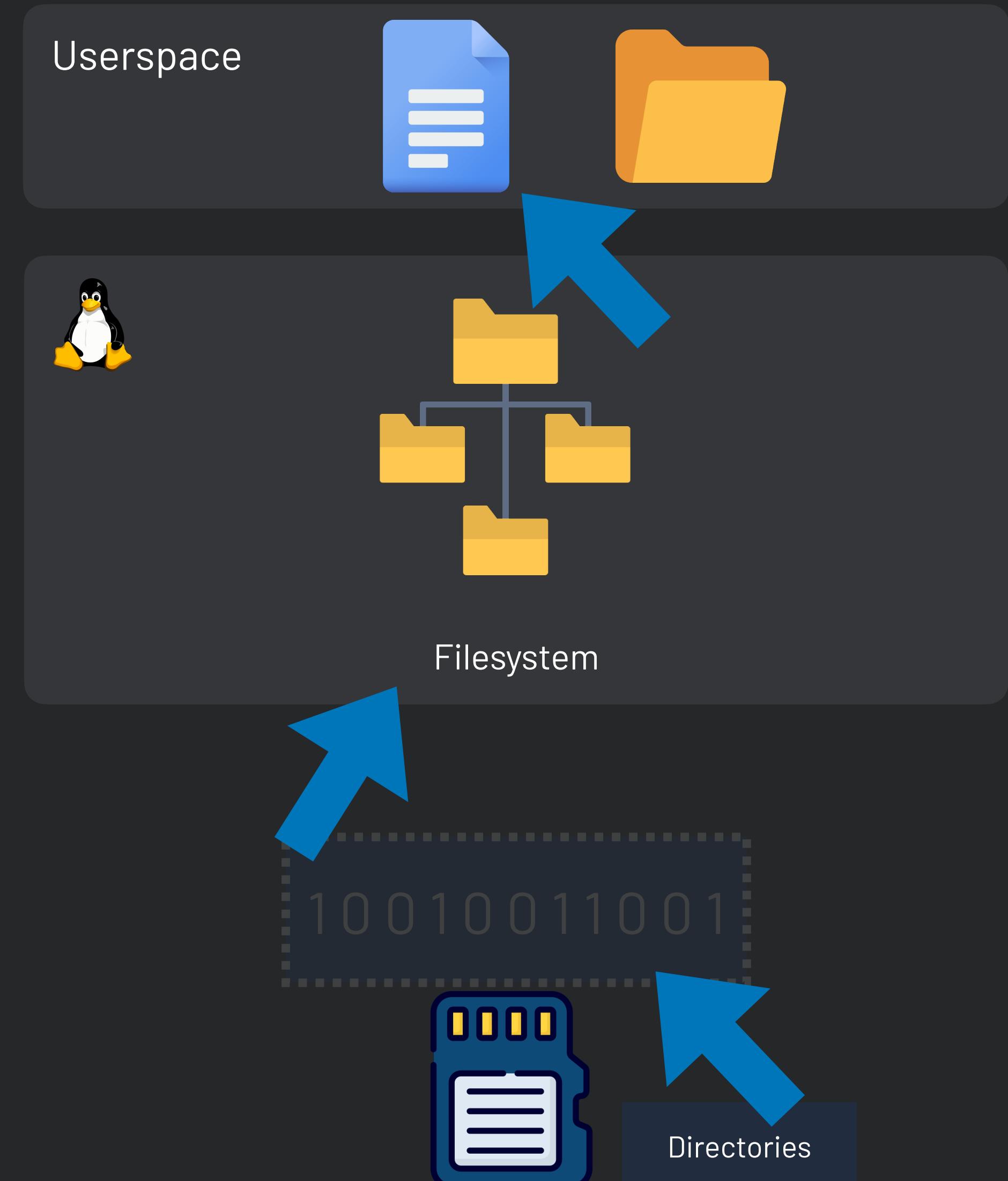
Stack
Functions

Commands

Interaction
Hardware

2

Adding new storage runtime



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

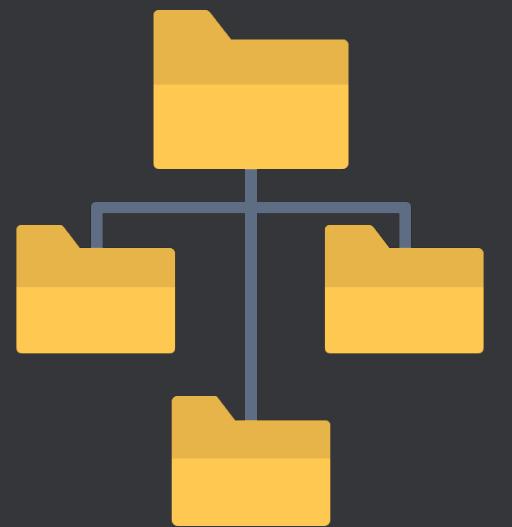
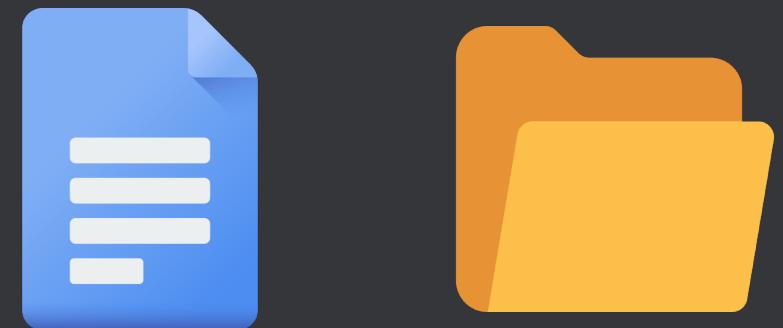
In session exercises

Stack
Functions

Commands

Interaction
Hardware

Userspace



Filesystem

lsblk # get all storages connected to linux machine

make file system type: ext4
sudo mkfs.ext4 /dev/sdb1

start the partition manipulator.
sudo fdisk /dev/sdX



Directories

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

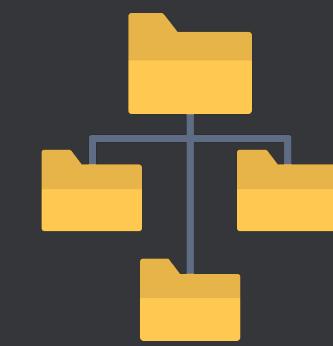
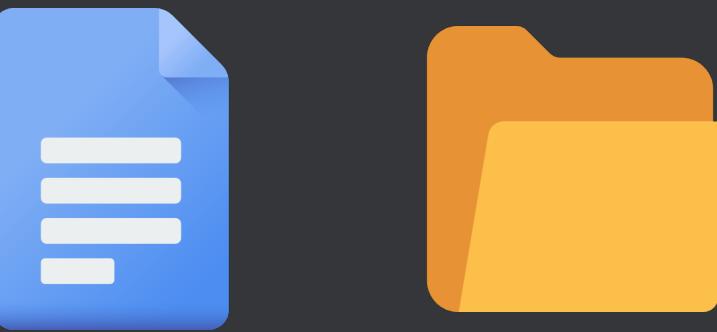
In session exercises

Stack
Functions

Commands

Interaction
Hardware

Userspace



Filesystem

```
● ● ●  
# Show all mounted filesystems:  
mount  
  
# Mount a device to a directory.  
mount -t filesystem_type path/to/device_file path/to/target_directory
```



```
lsblk # get all storages connected to linux machine
```



```
# make file system type: ext4  
sudo mkfs.ext4 /dev/sdb1
```



```
# start the partition manipulator.  
sudo fdisk /dev/sdX
```

10010011001



Directories

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

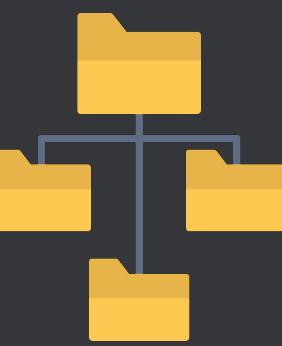
```
# Based on object.  
# 1. File.  
# 2. directory.  
# 3. path
```



Userspace



```
# Show all mounted filesystems:  
mount  
  
# Mount a device to a directory.  
mount -t filesystem_type path/to/device_file path/to/target_directory
```



Filesystem

```
lsblk # get all storages connected to linux machine
```

```
# make file system type: ext4  
sudo mkfs.ext4 /dev/sdb1
```

```
# start the partition manipulator.  
sudo fdisk /dev/sdX
```

10010011001



Directories

Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

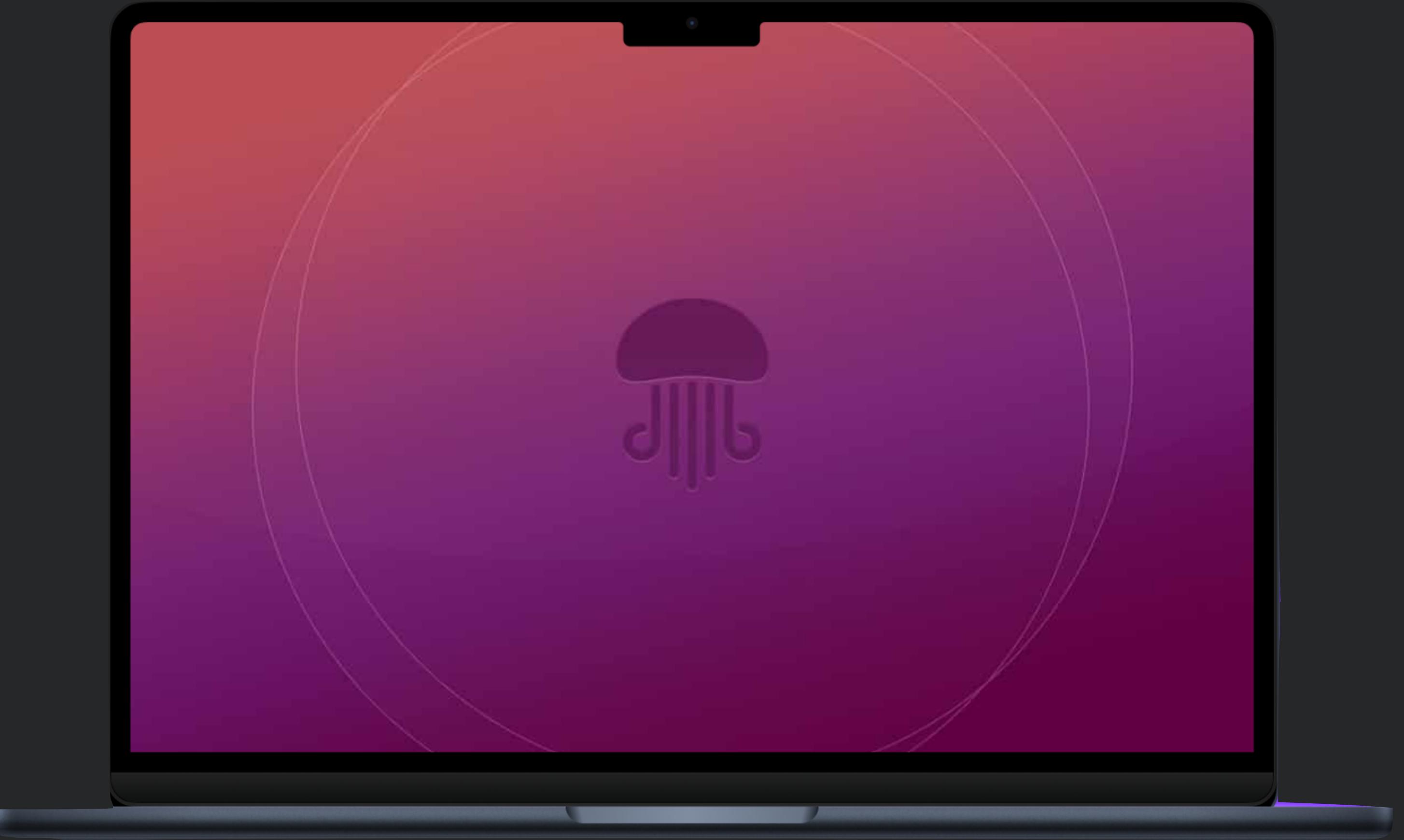
In session exercises

Adding new storage runtime

Stack
Functions

Commands

Interaction
Hardware



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

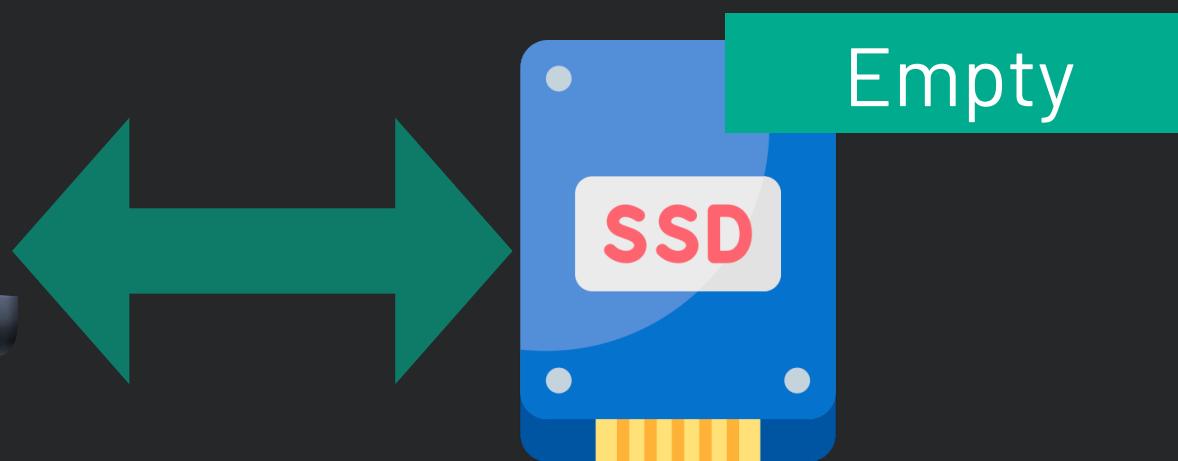
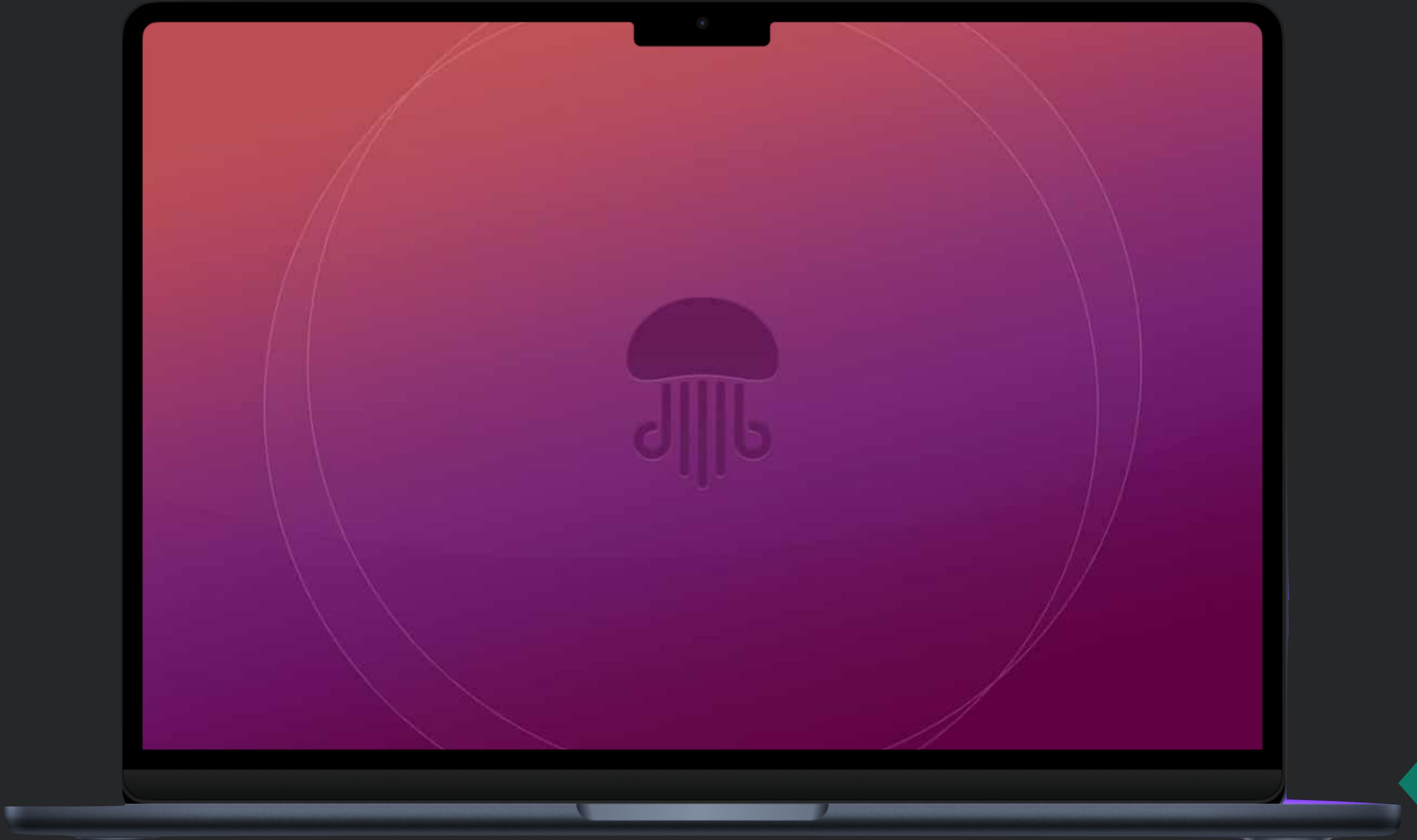
In session exercises

Adding new storage runtime

Stack
Functions

Commands

Interaction
Hardware



Introduction to Linux
Stacks

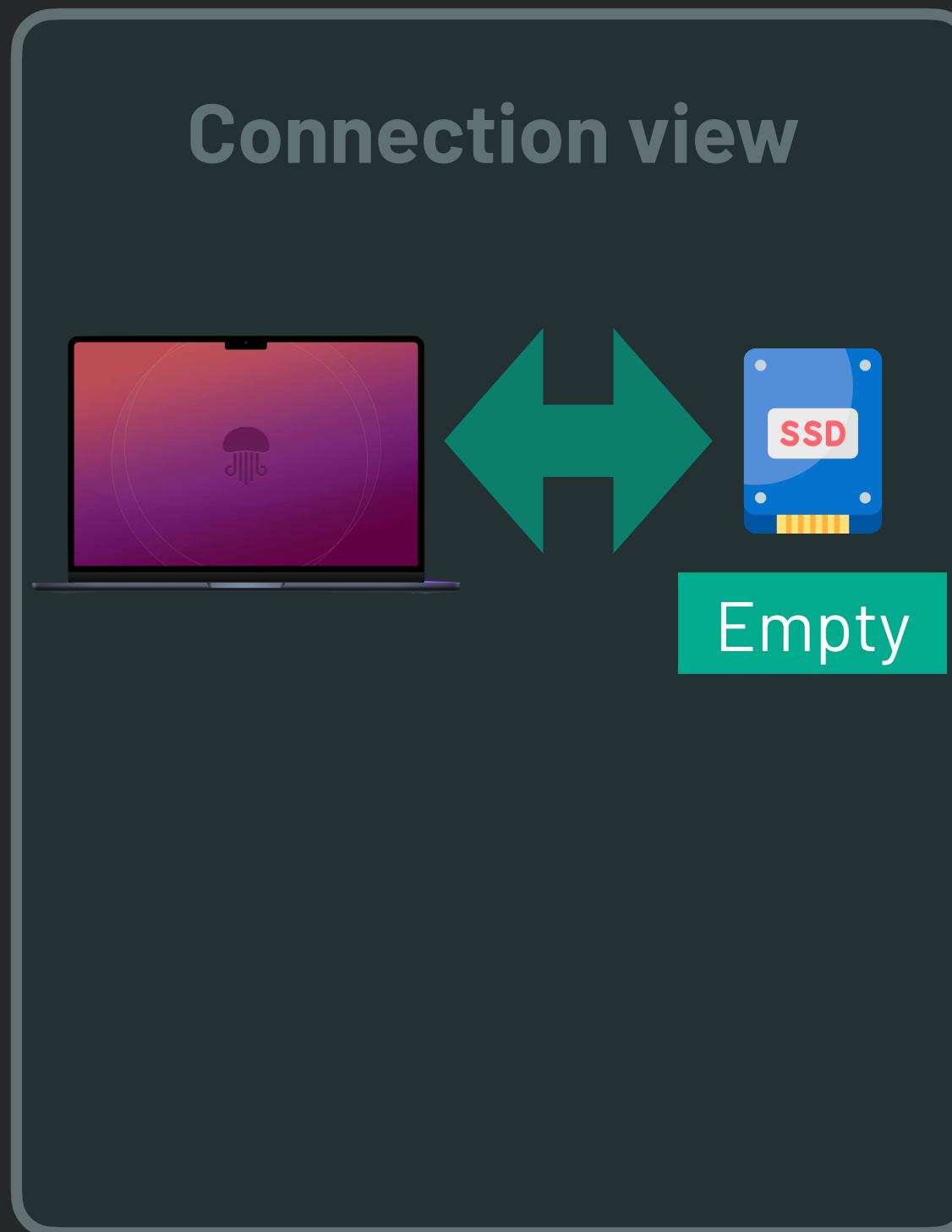
File system Stack

In session exercises

Process Management
Stack

In session exercises

Adding new storage runtime



Connection view

Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

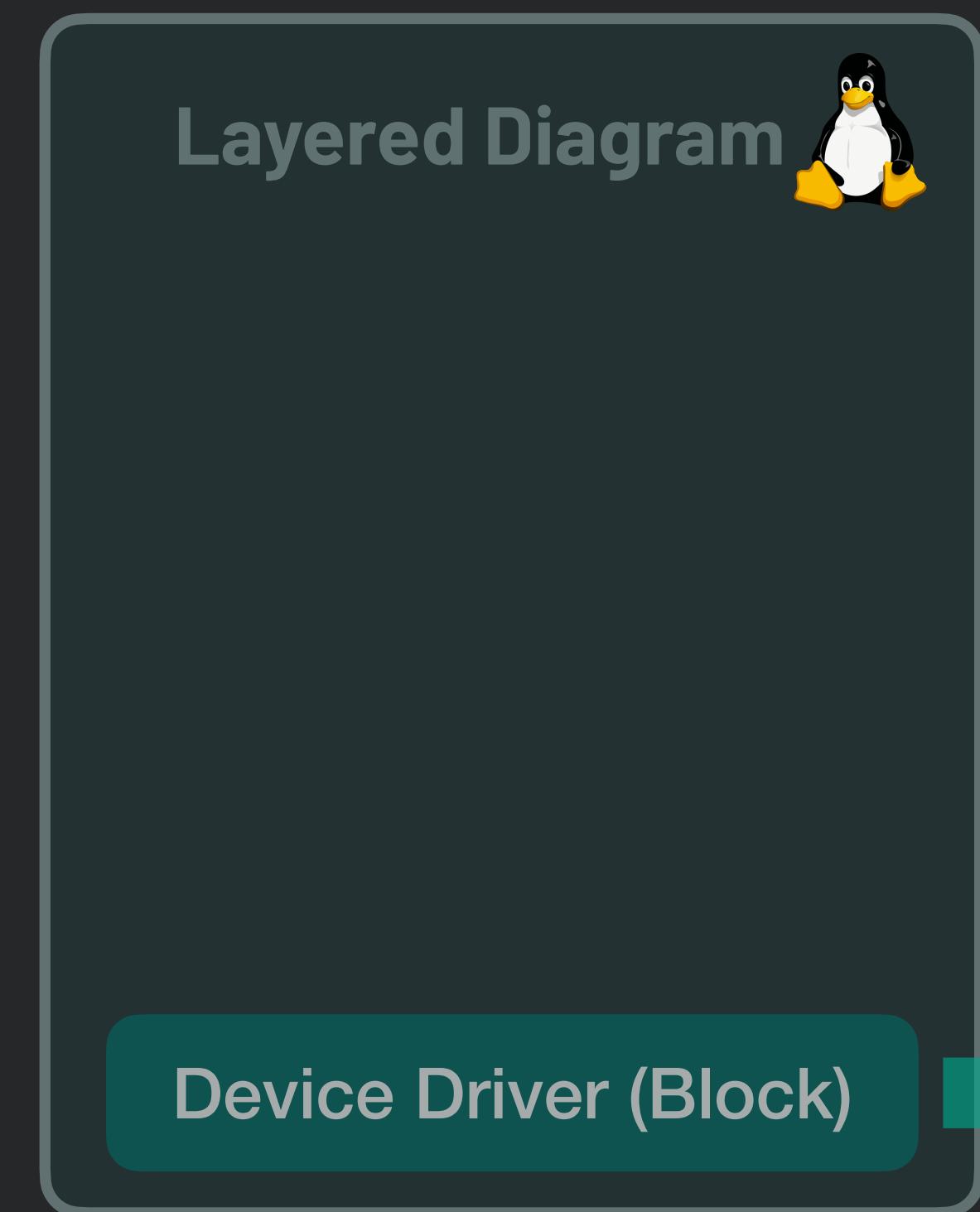
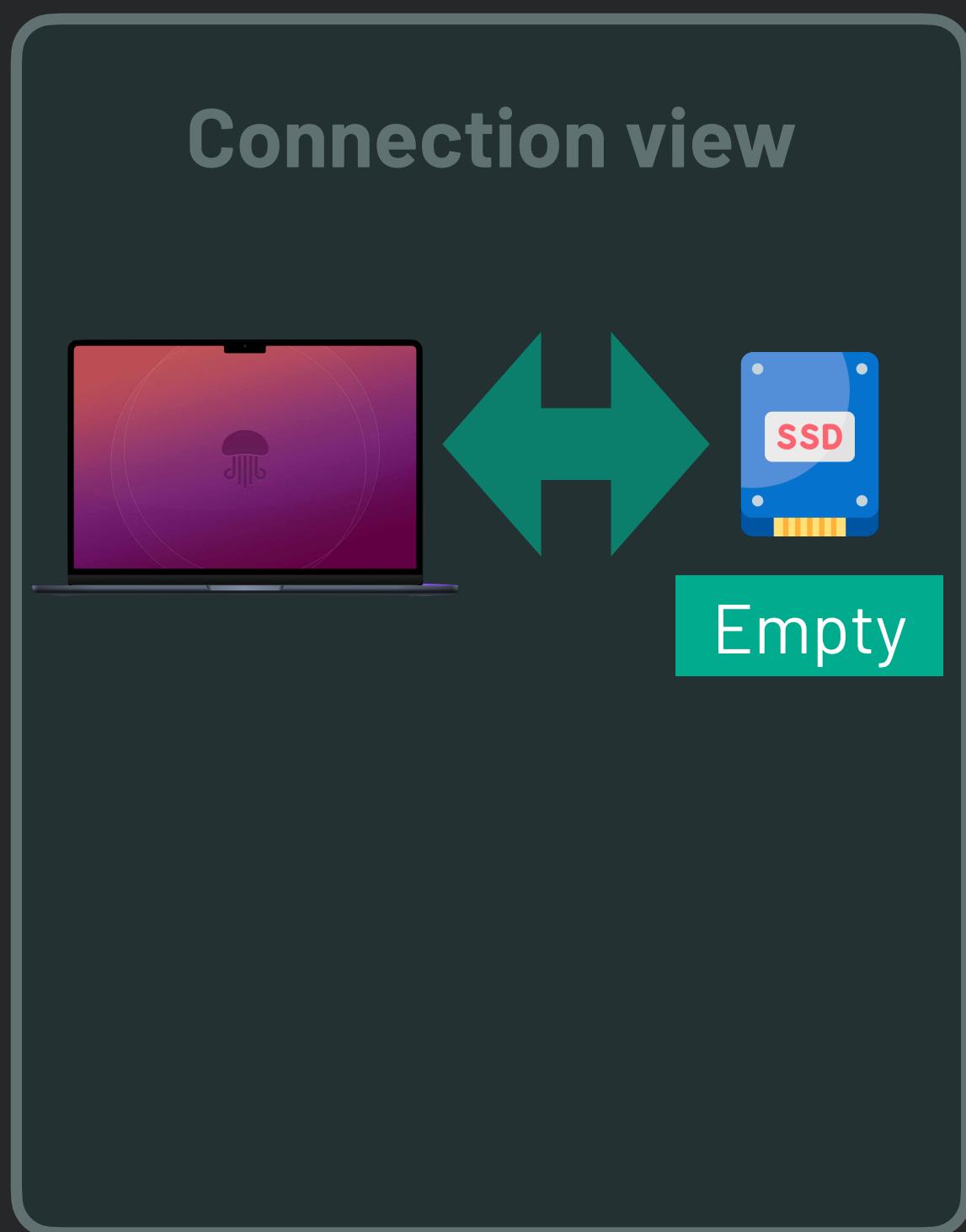
File system Stack

In session exercises

Process Management
Stack

In session exercises

Adding new storage runtime



```
lsblk # get all storages connected to linux machine
```

Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

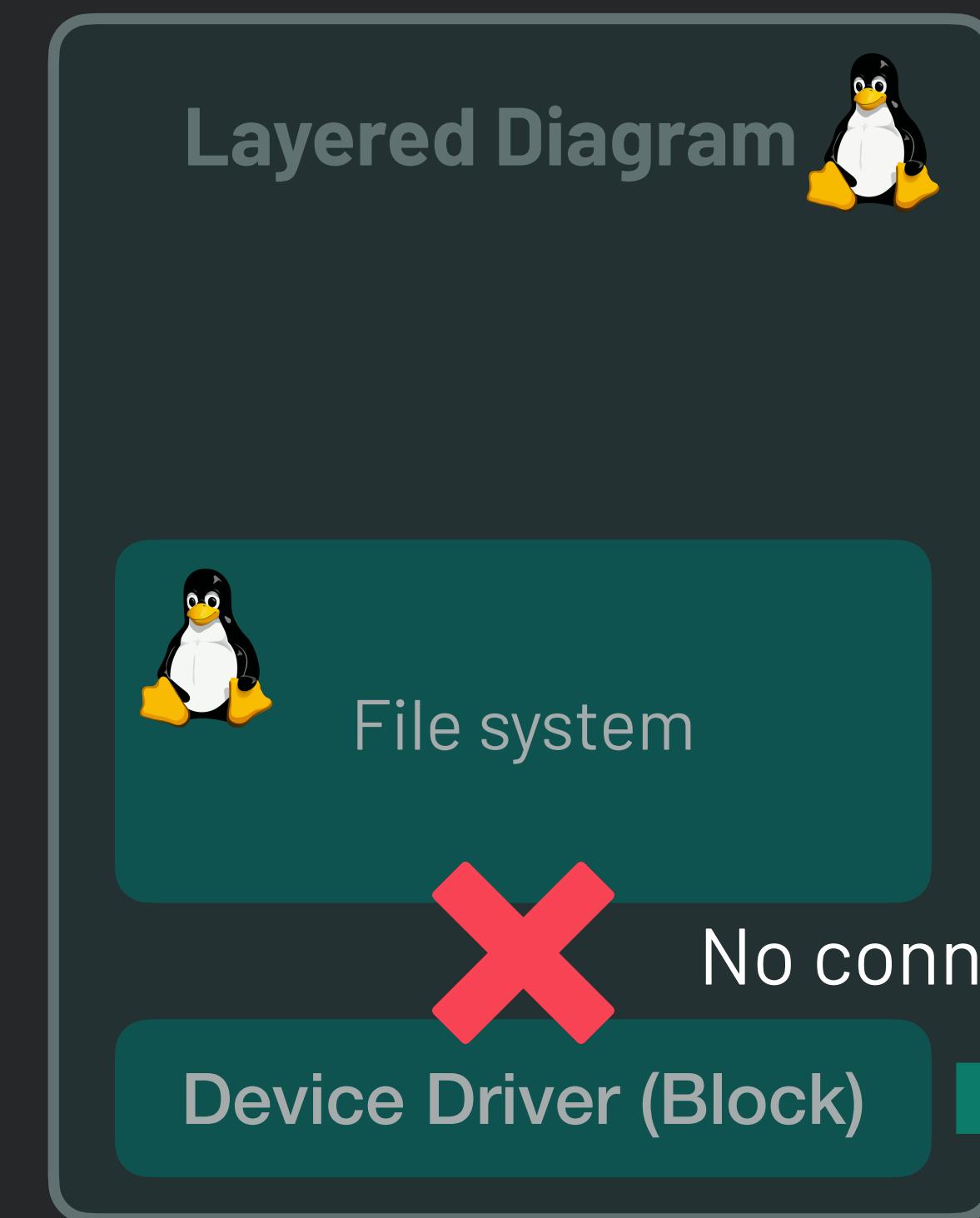
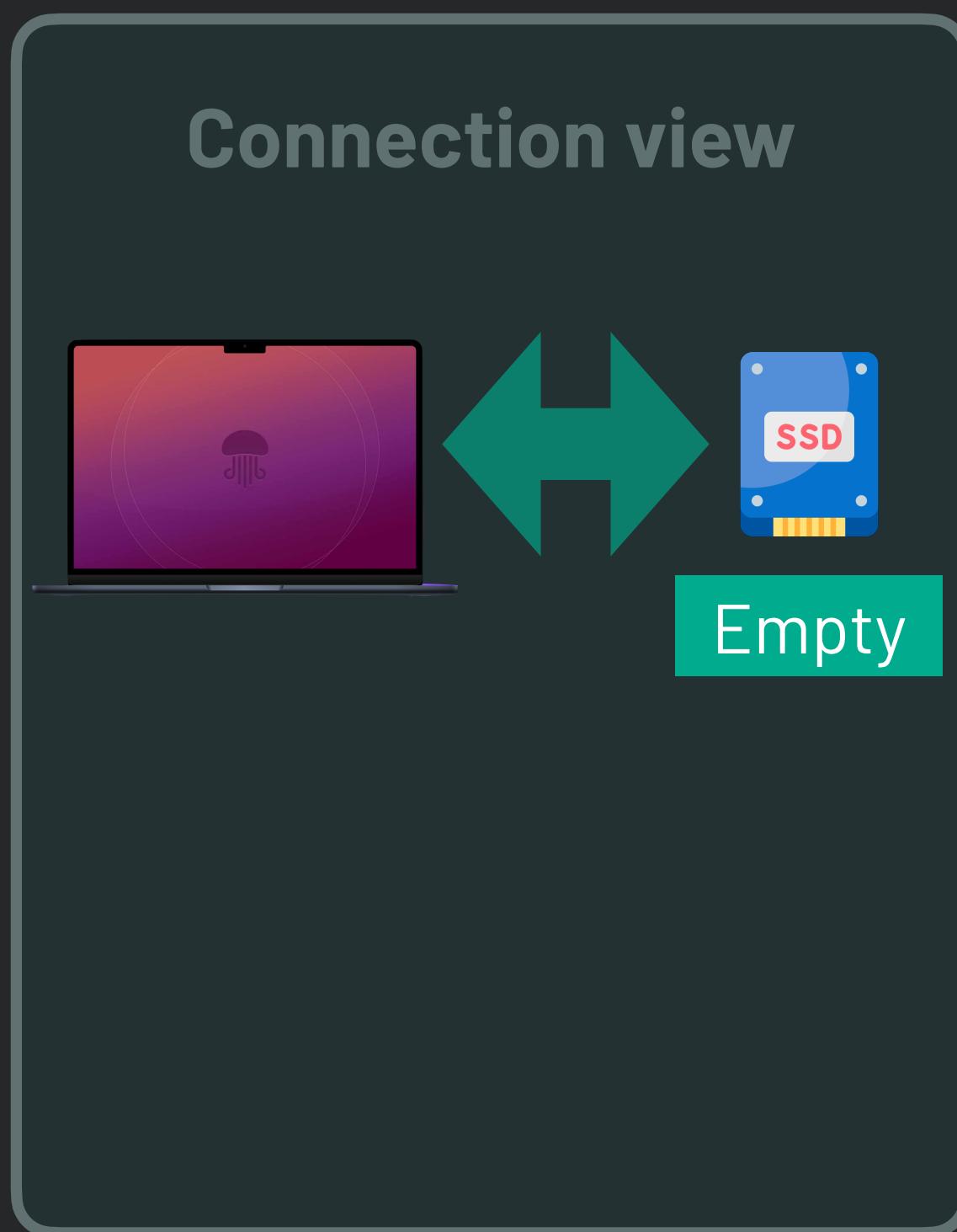
File system Stack

In session exercises

Process Management
Stack

In session exercises

Adding new storage runtime



lsblk # get all storages connected to linux machine

Empty

Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

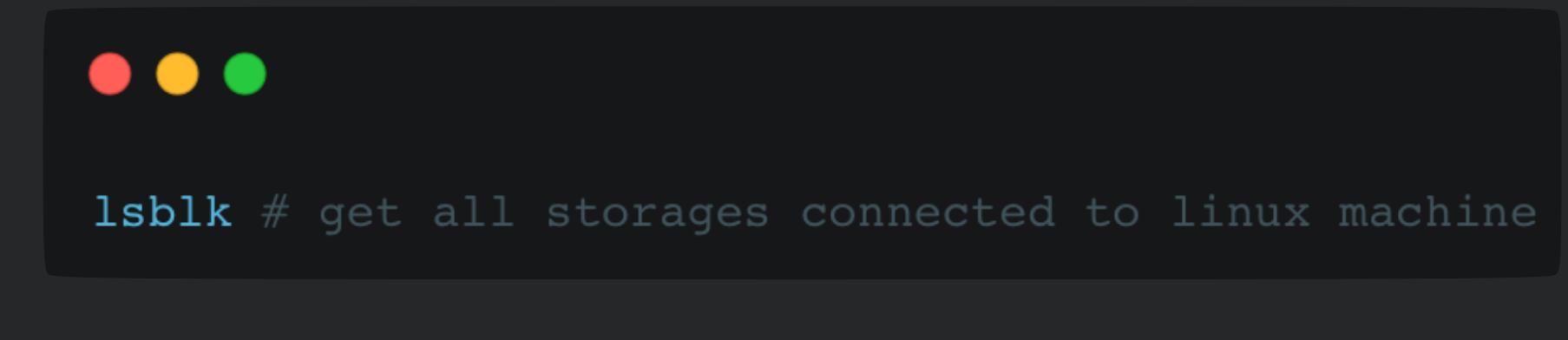
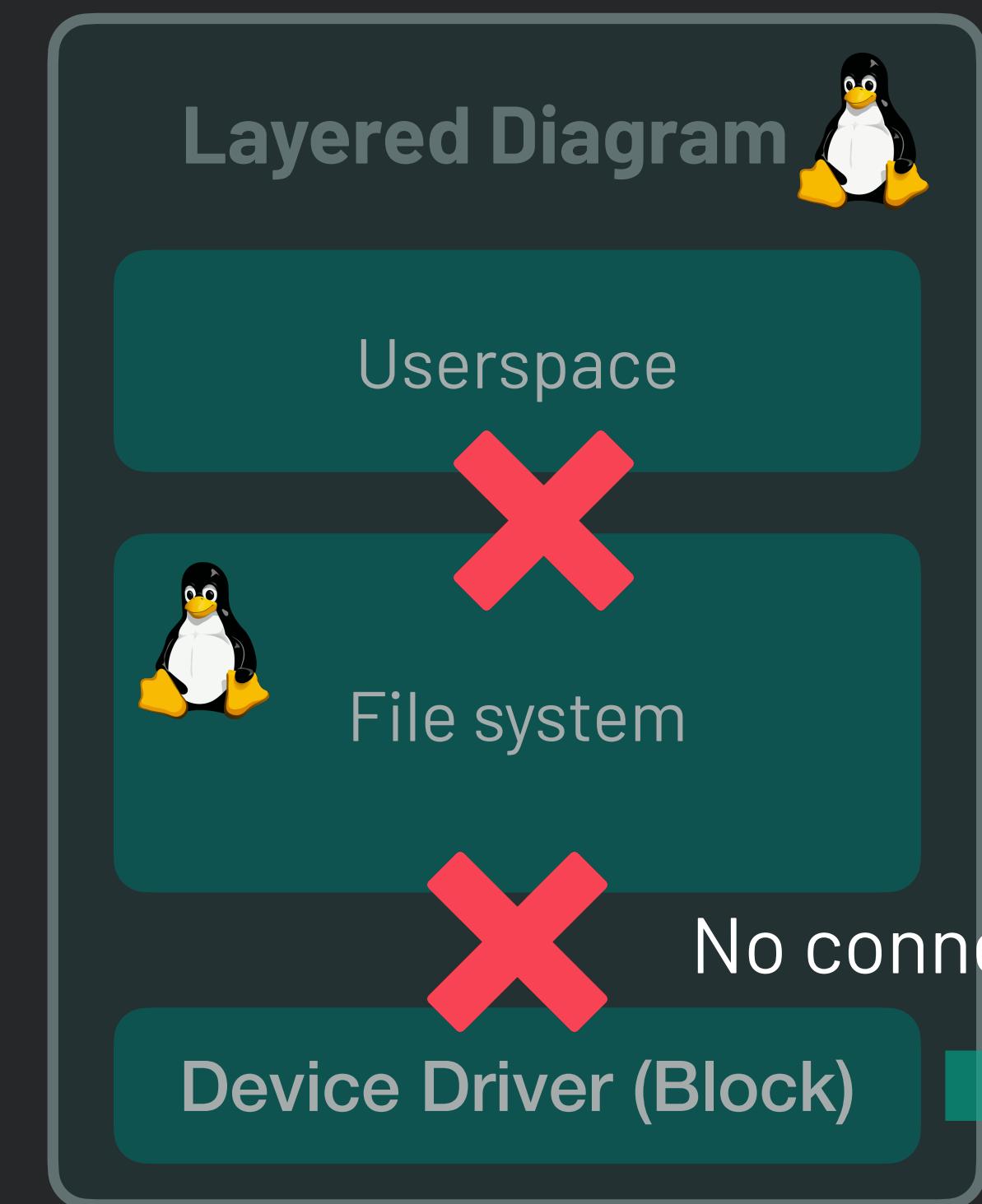
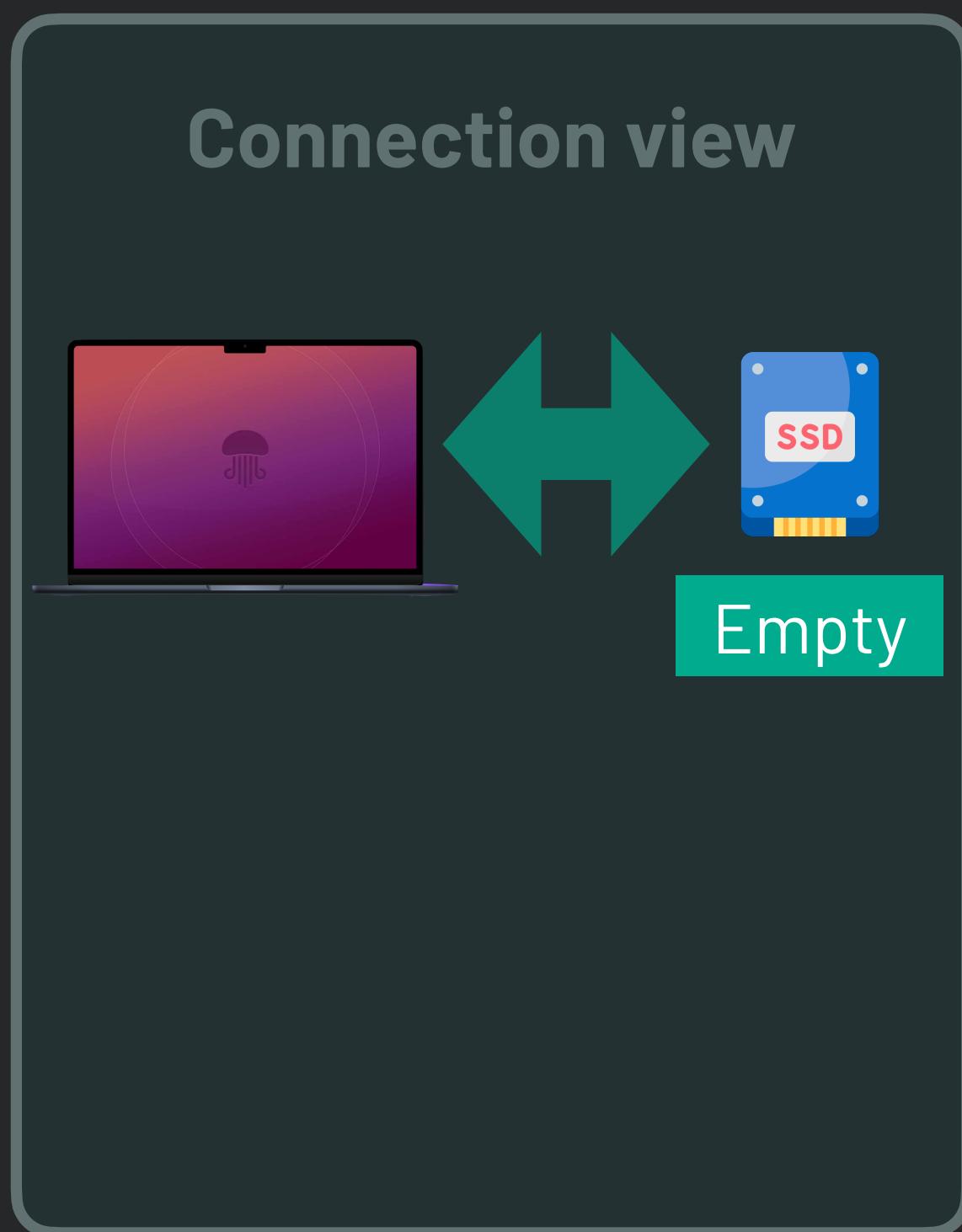
File system Stack

In session exercises

Process Management
Stack

In session exercises

Adding new storage runtime



Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

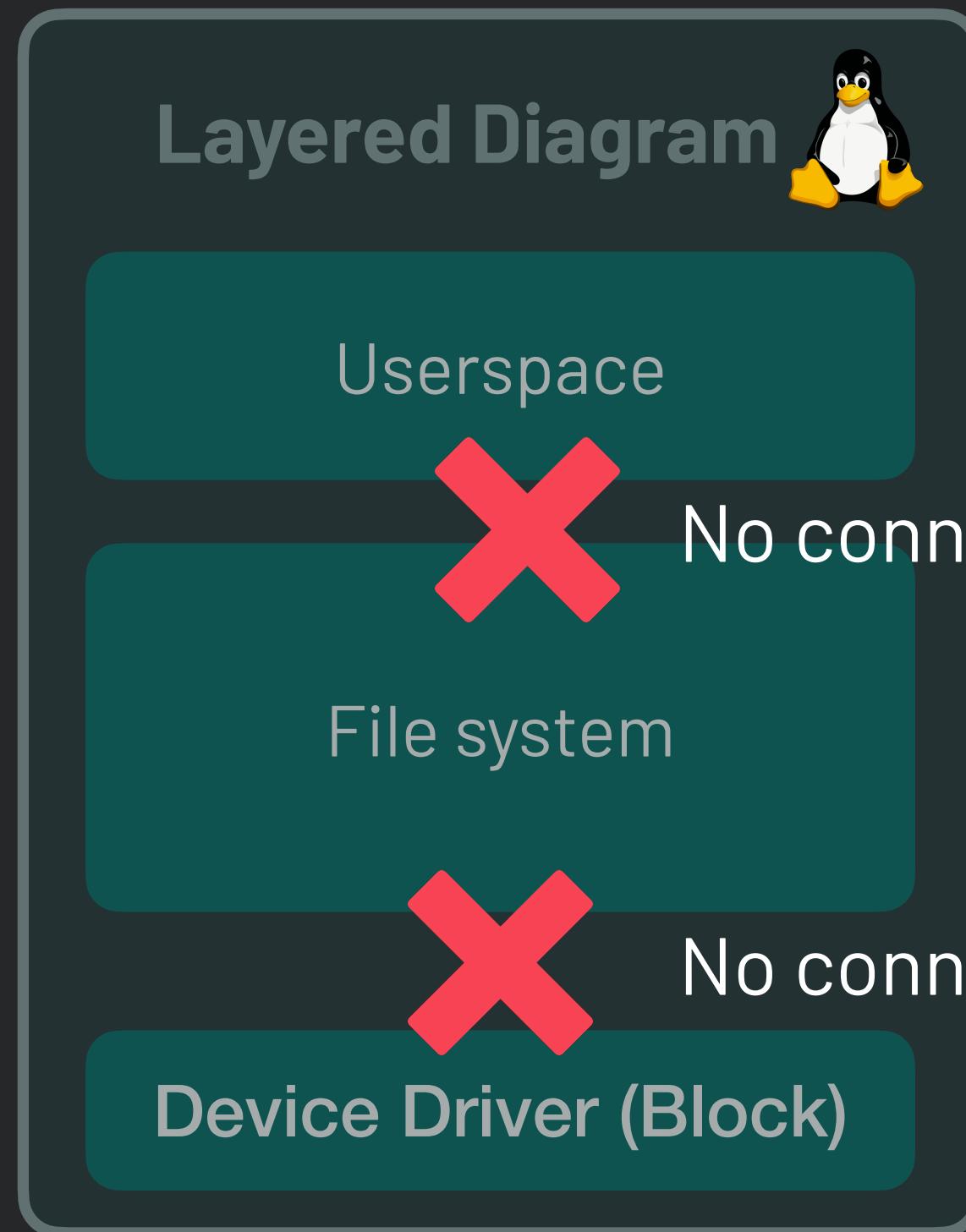
File system Stack

In session exercises

Process Management
Stack

In session exercises

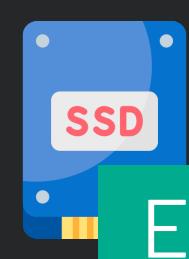
Adding new storage runtime



Stack
Functions

Commands

Interaction
Hardware



Empty



Preparation Storage device to be understandable by filesystem stack

Introduction to Linux
Stacks

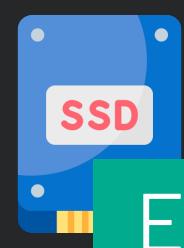
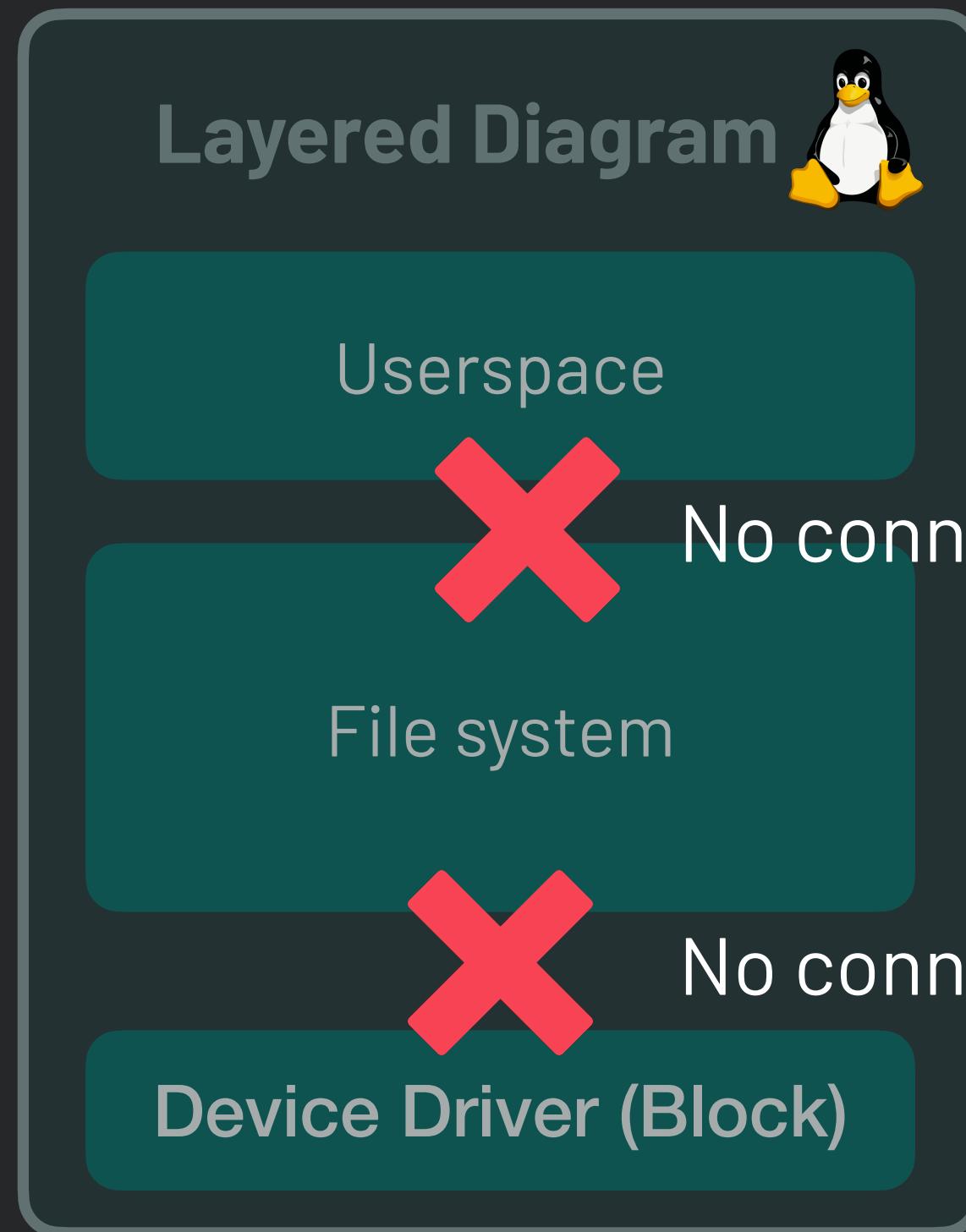
File system Stack

In session exercises

Process Management
Stack

In session exercises

Adding new storage runtime



Empty



Create partition



```
# start the partition manipulator.  
sudo fdisk /dev/sdX
```

Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

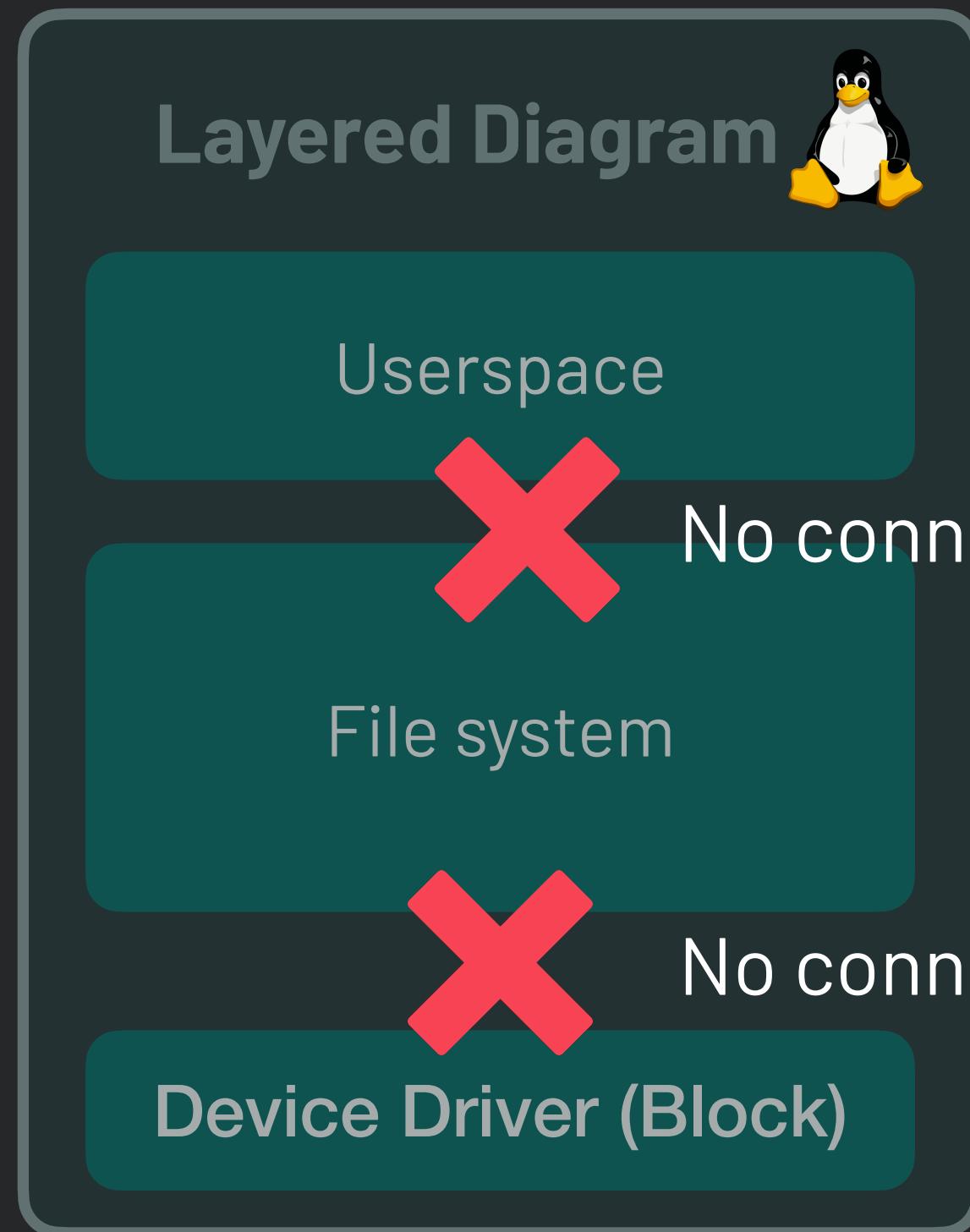
File system Stack

In session exercises

Process Management
Stack

In session exercises

Adding new storage runtime



Stack
Functions

Commands

Interaction
Hardware



partition

Introduction to Linux
Stacks

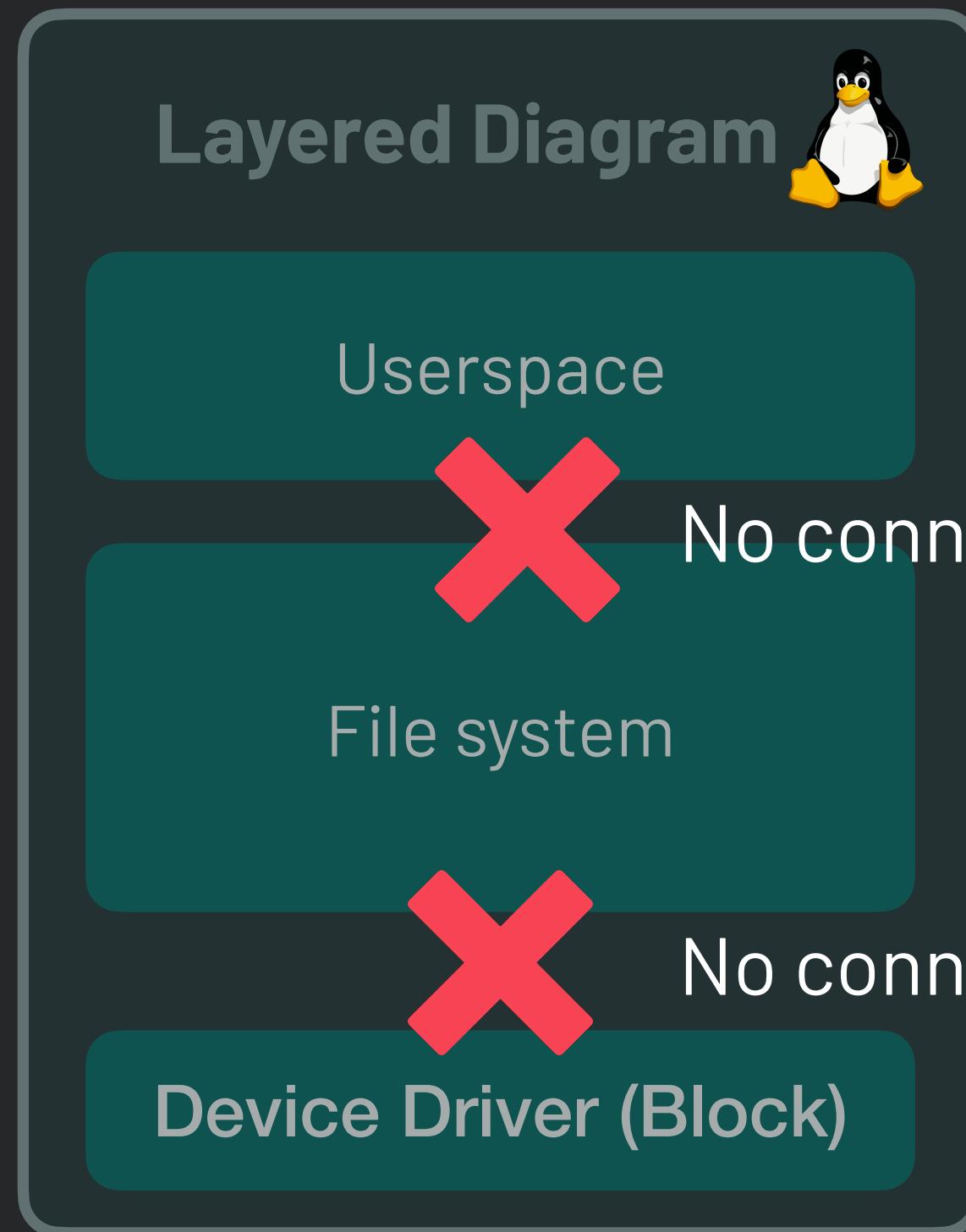
File system Stack

In session exercises

Process Management
Stack

In session exercises

Adding new storage runtime



Understand filesystem protocols / types :

1. Ext3
 2. Ext4
 3. FAT
-

Stack
Functions

Commands

Interaction
Hardware



partition

Introduction to Linux
Stacks

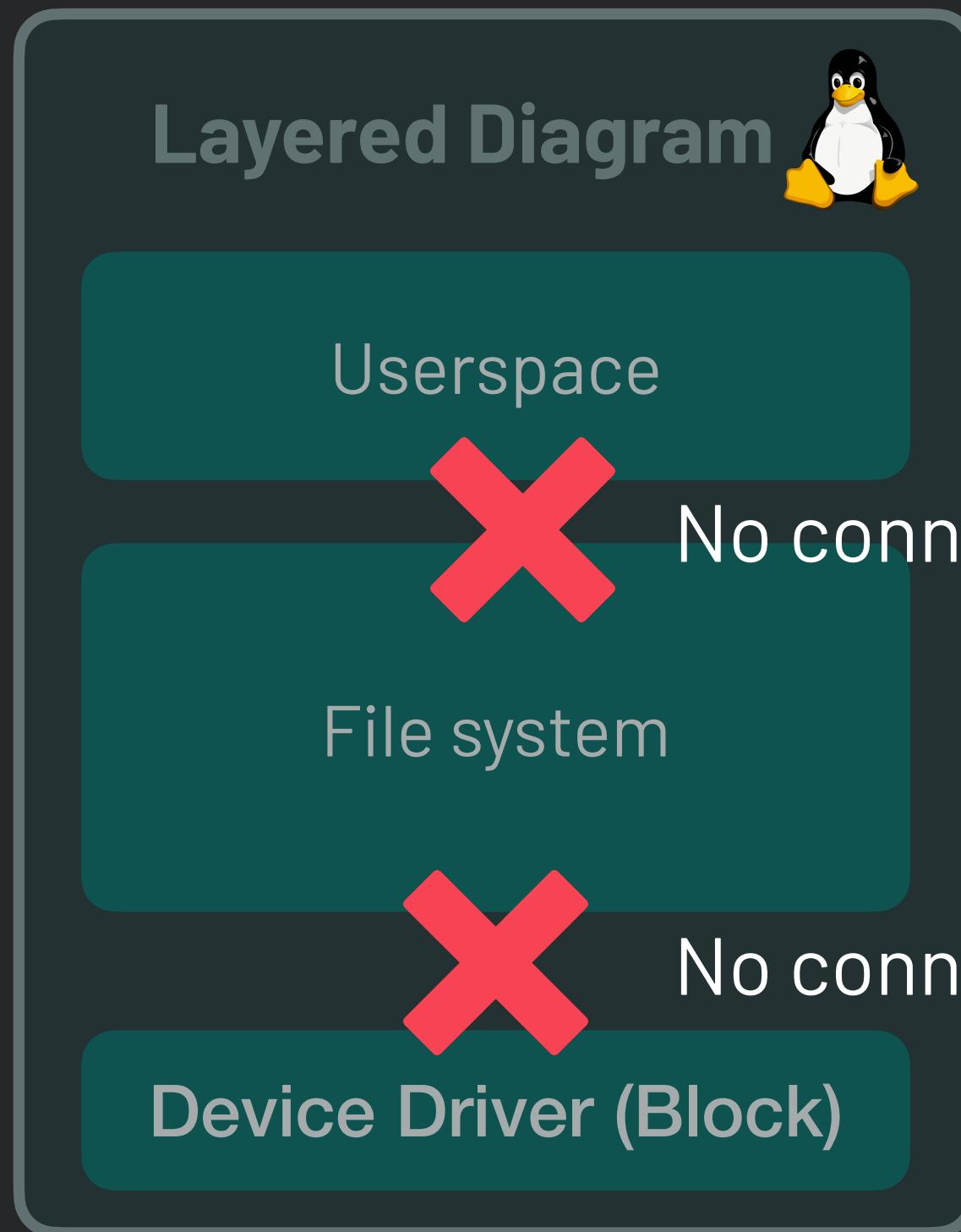
File system Stack

In session exercises

Process Management
Stack

In session exercises

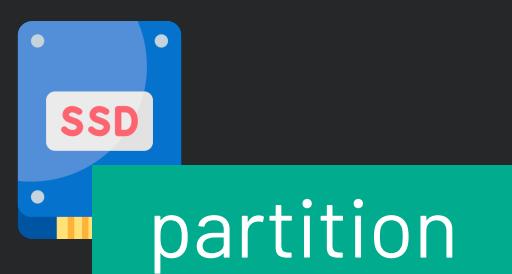
Adding new storage runtime



Understand filesystem protocols / types :

1. Ext3
2. Ext4
3. FAT

.....



Add filesystem



```
# make file system type: ext4  
sudo mkfs.ext4 /dev/sdb1
```

Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

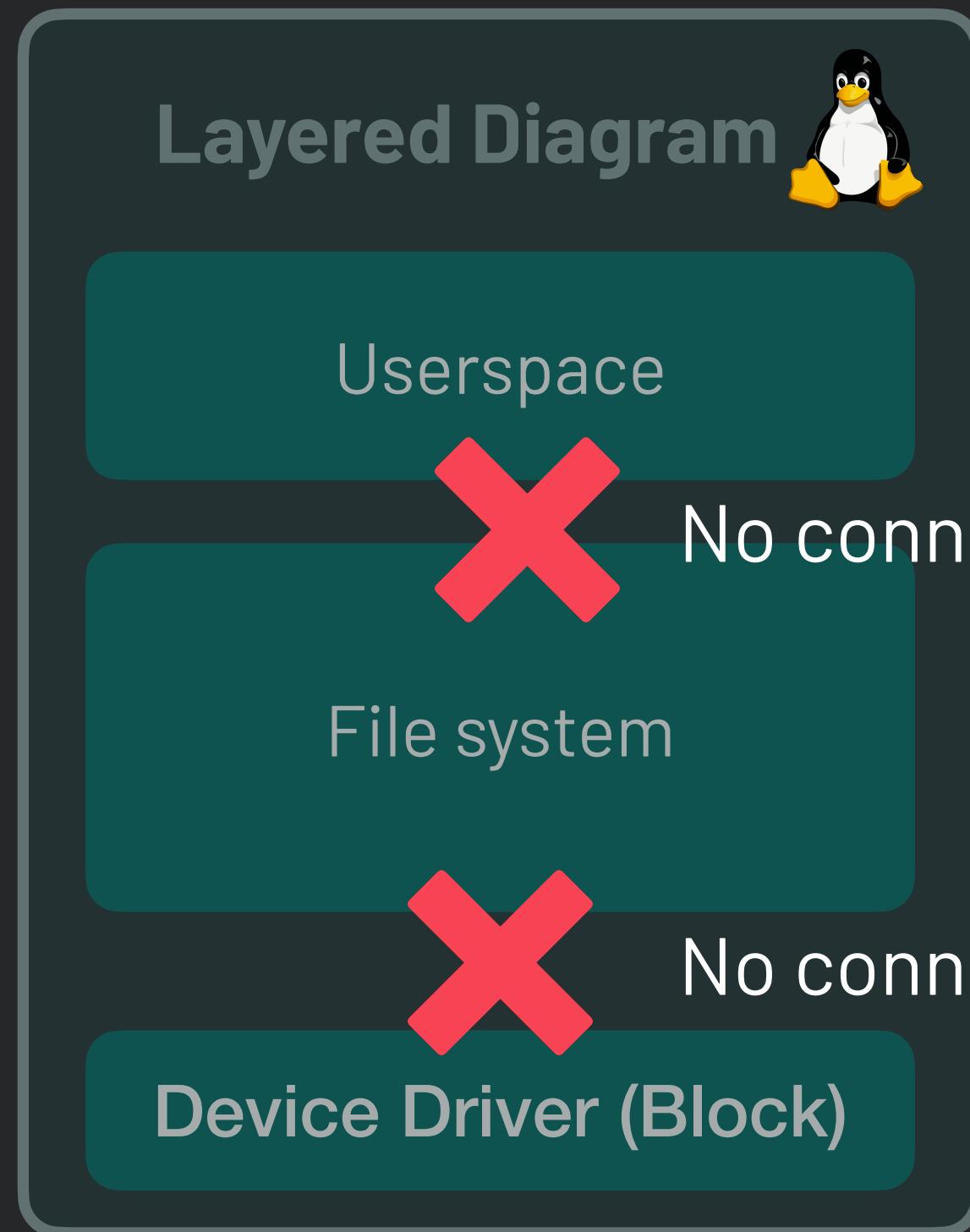
File system Stack

In session exercises

Process Management
Stack

In session exercises

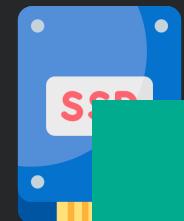
Adding new storage runtime



Stack
Functions

Commands

Interaction
Hardware



Has a
filesystem

Introduction to Linux
Stacks

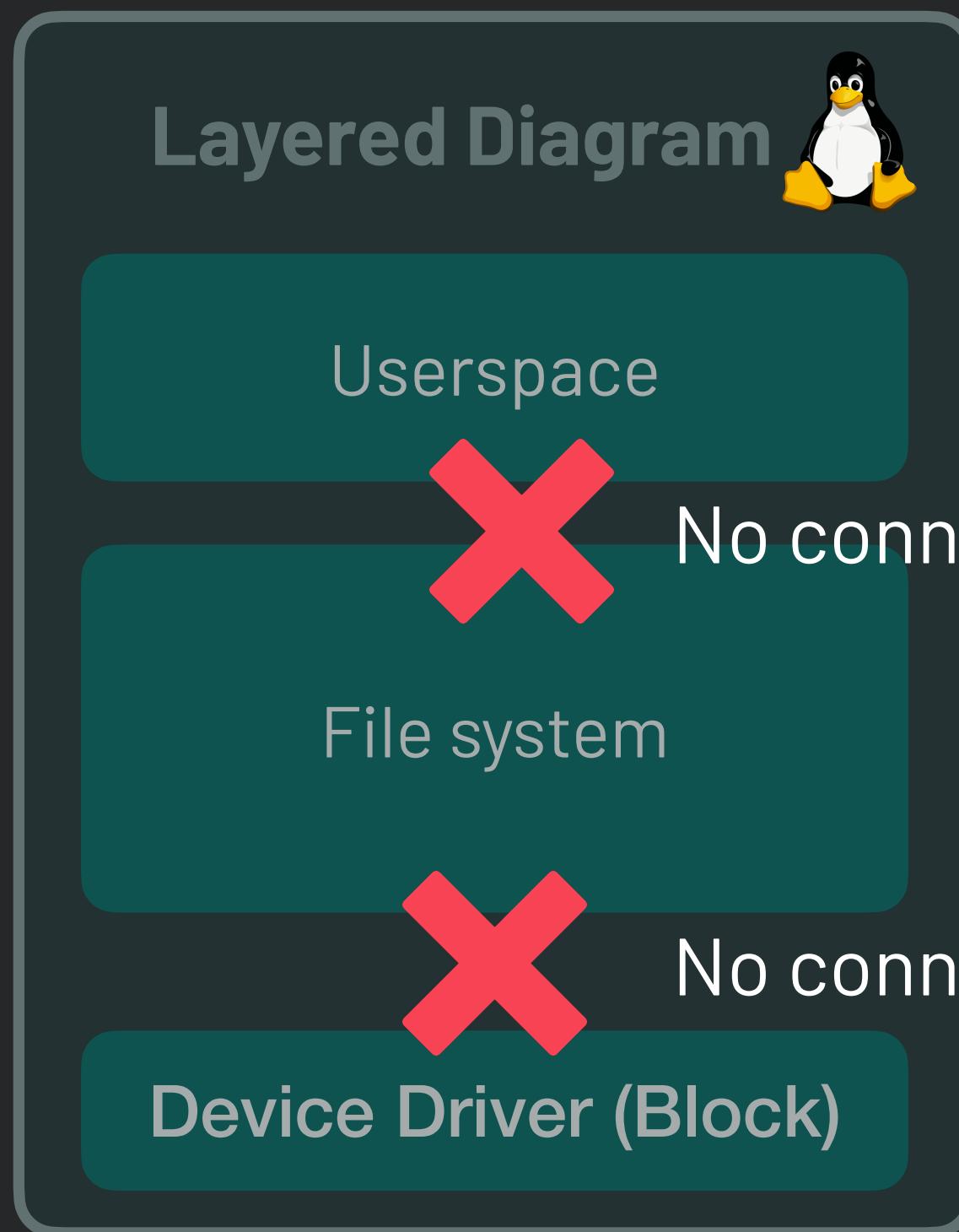
File system Stack

In session exercises

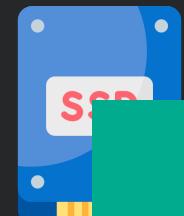
Process Management
Stack

In session exercises

Adding new storage runtime



```
# Show all mounted filesystems:  
mount  
  
# Mount a device to a directory.  
mount -t filesystem_type path/to/device_file path/to/target_directory
```



Has a
filesystem

Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

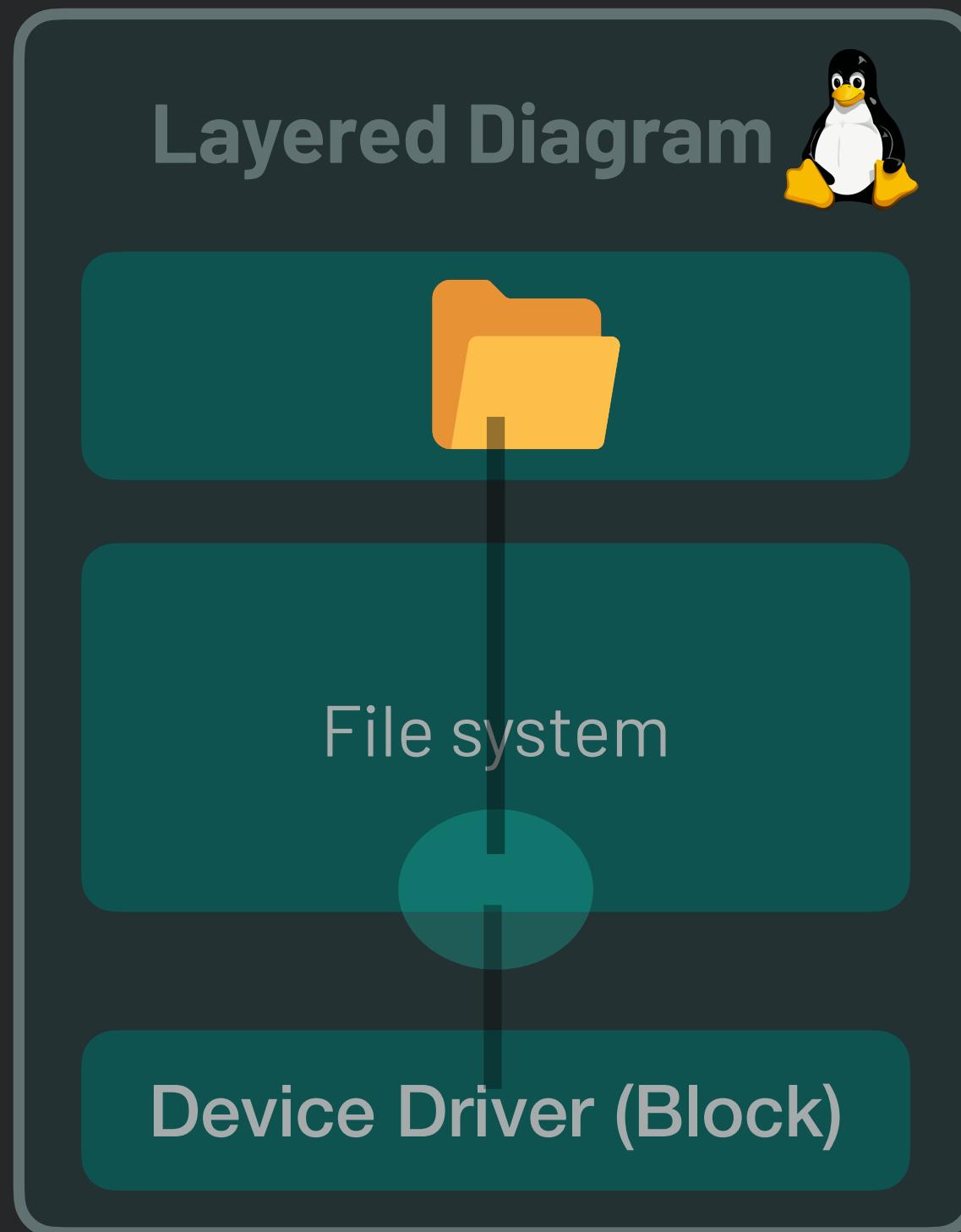
File system Stack

In session exercises

Process Management
Stack

In session exercises

Adding new storage runtime



● ● ●

```
# Show all mounted filesystems:  
mount
```

```
# Mount a device to a directory.  
mount -t filesystem_type path/to/device_file path/to/target_directory
```



Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

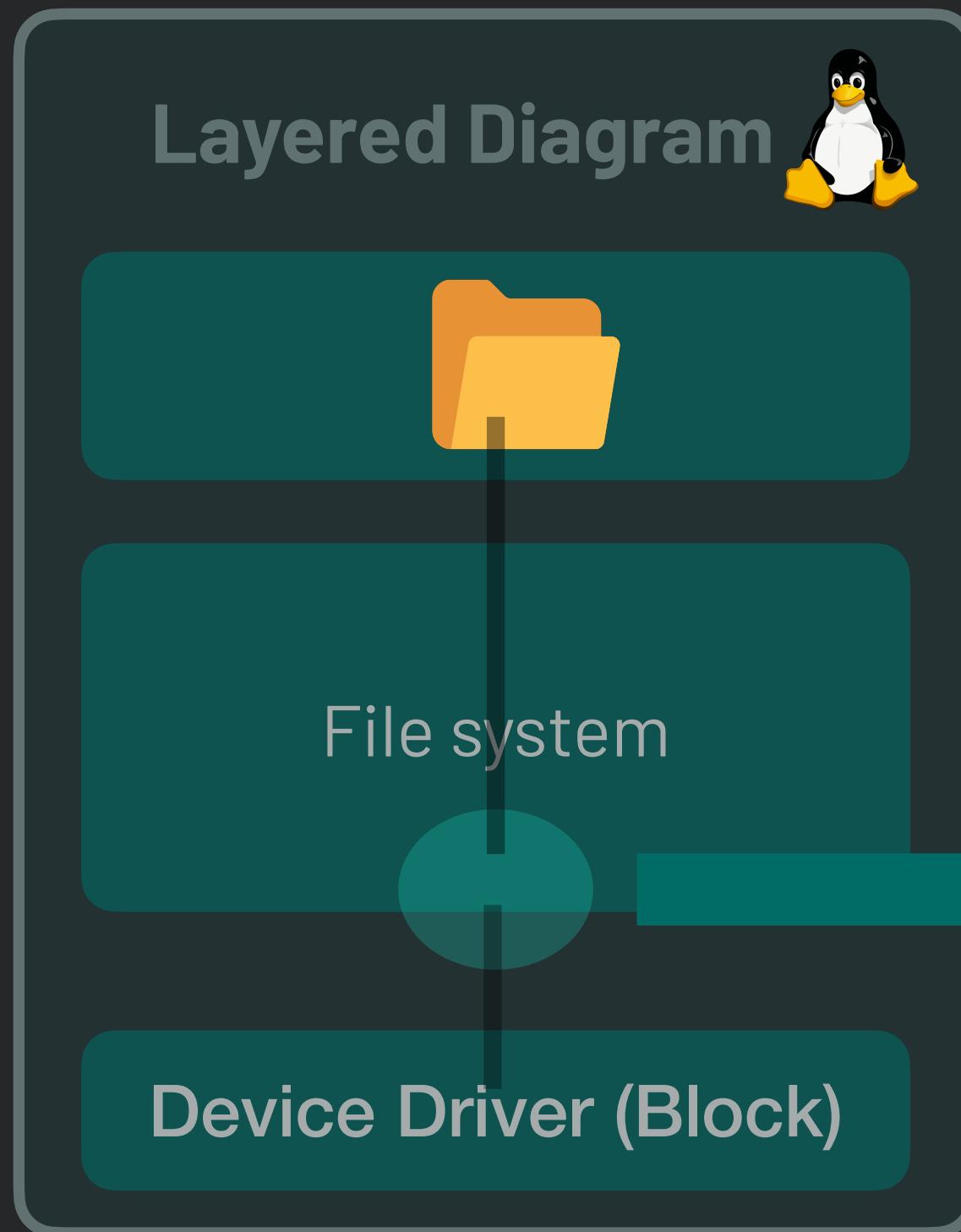
File system Stack

In session exercises

Process Management
Stack

In session exercises

Adding new storage runtime



Stack
Functions

Commands

Interaction
Hardware



Connected

Introduction to Linux
Stacks

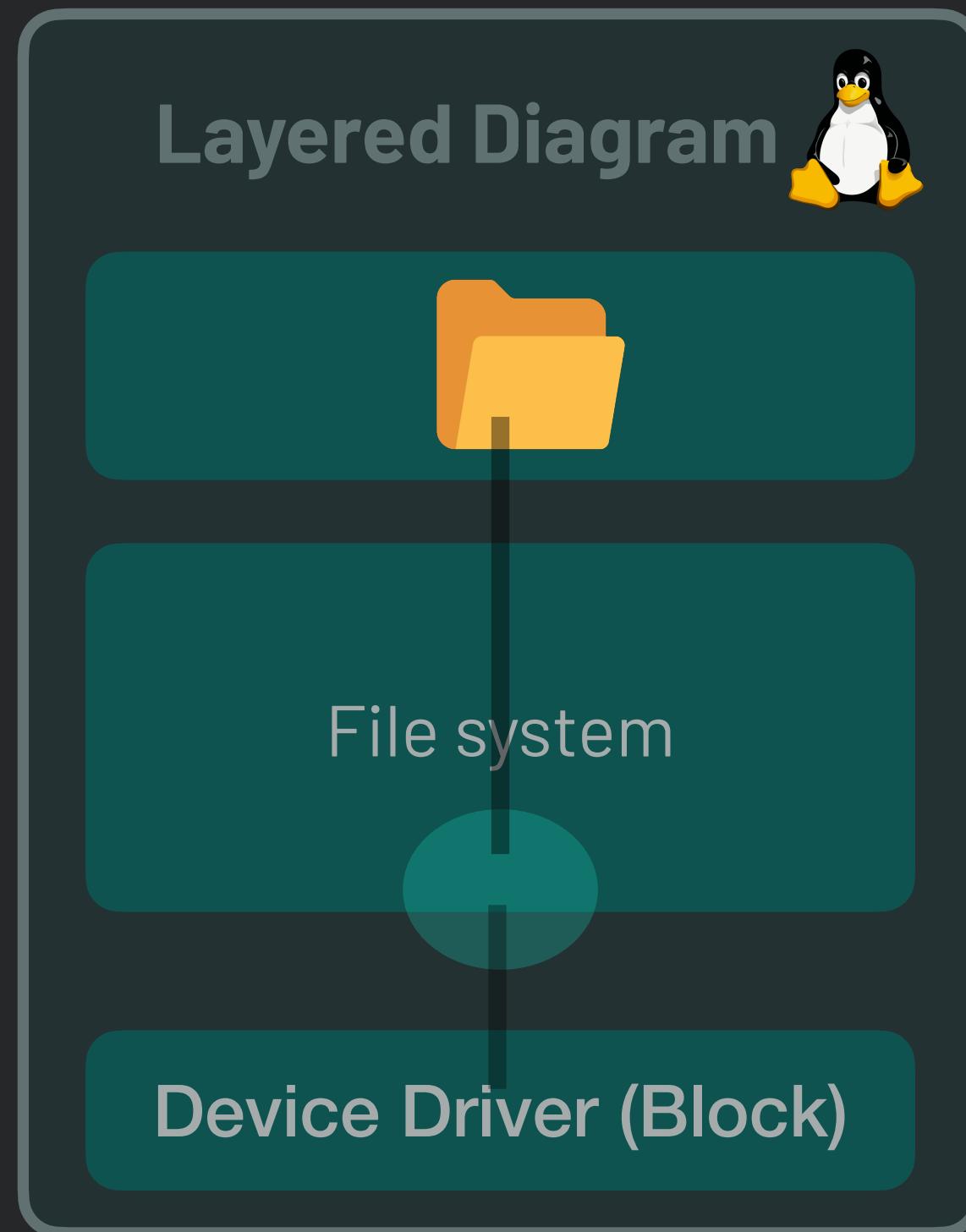
File system Stack

In session exercises

Process Management
Stack

In session exercises

Adding new storage runtime



Stack
Functions

Commands

Interaction
Hardware



Connected

Introduction to Linux
Stacks

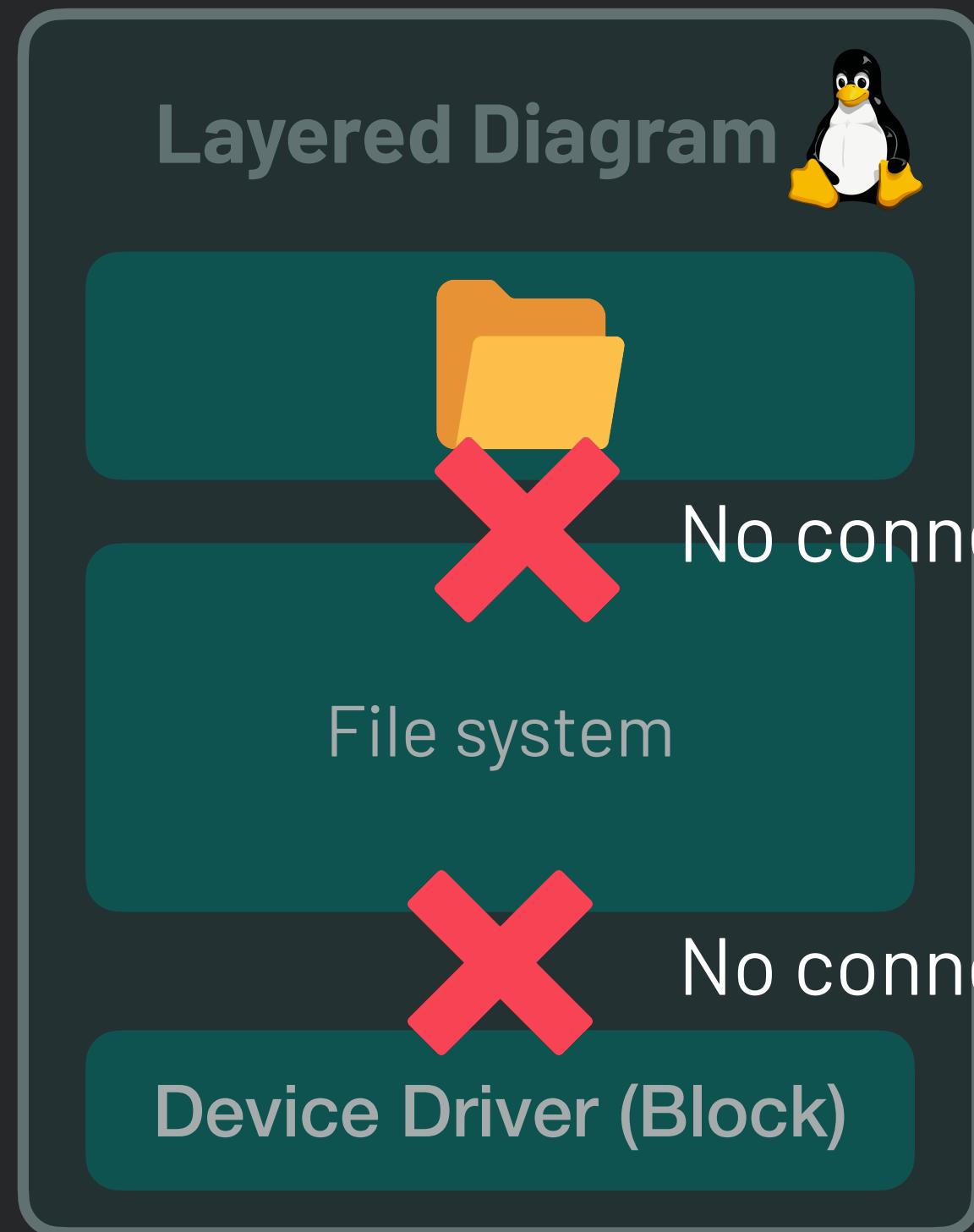
File system Stack

In session exercises

Process Management
Stack

In session exercises

Removing new storage runtime

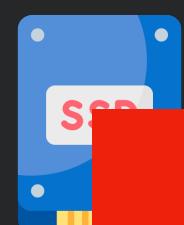


```
# Unmount a filesystem, by passing the path to the source it is mounted from:  
umount path/to/device_file  
  
#Unmount a filesystem, by passing the path to the target  
#where it is mounted:  
umount path/to/mounted_directory
```

Stack
Functions

Commands

Interaction
Hardware



Not
Connected

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Stack
Functions

Commands

Interaction
Hardware

Layered Diagram



By default

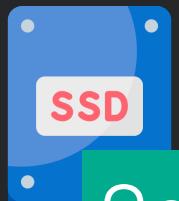
At least we have one mounted file system called "**Root filesystem**"



File system



Device Driver (Block)



Connected

Introduction to Linux
Stacks

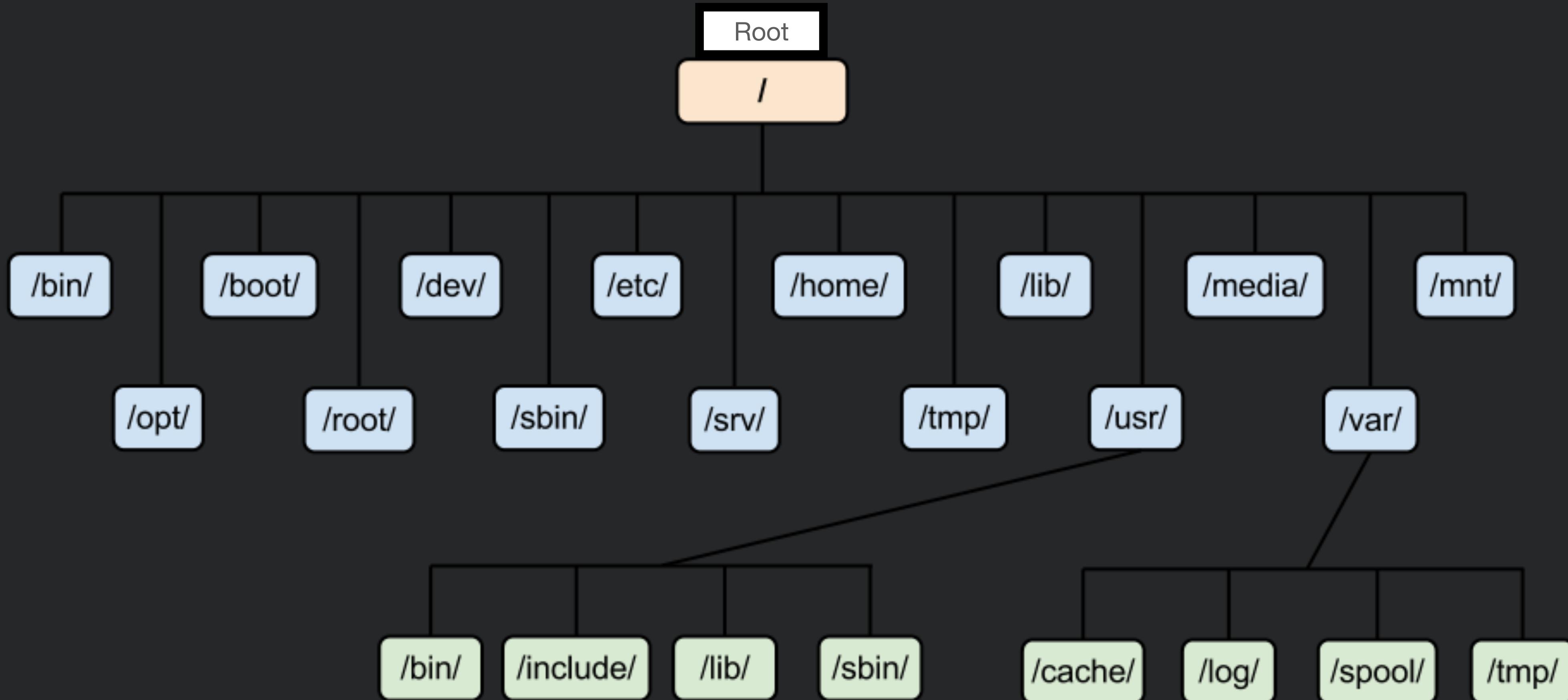
File system Stack

In session exercises

Process Management
Stack

In session exercises

"Root filesystem"



Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

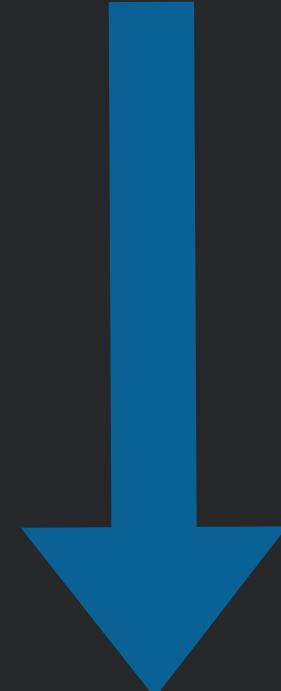
File system Stack

In session exercises

Process Management
Stack

In session exercises

"Root filesystem"



"Mounted at booting time" ★

Path

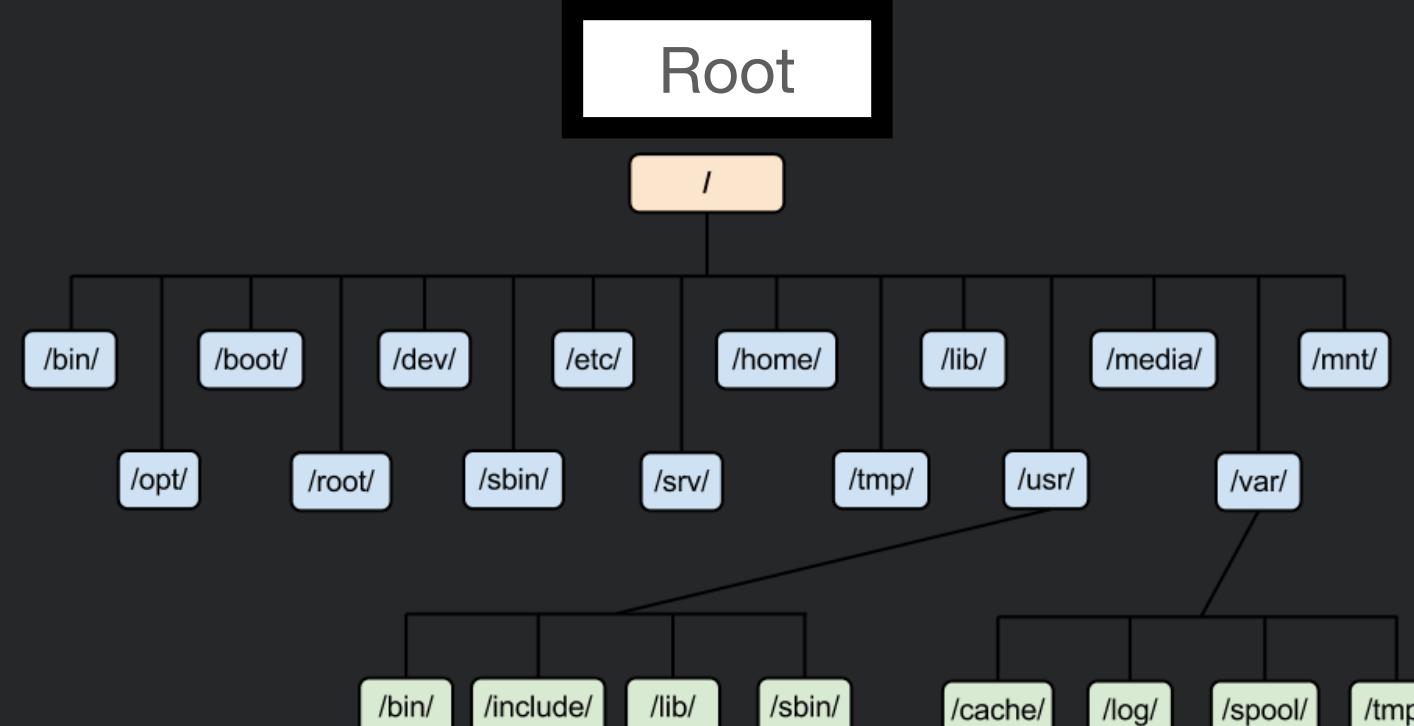
Directory

File

Stack
Functions

Commands

Interaction
Hardware



Introduction to Linux
Stacks

File system Stack

In session exercises

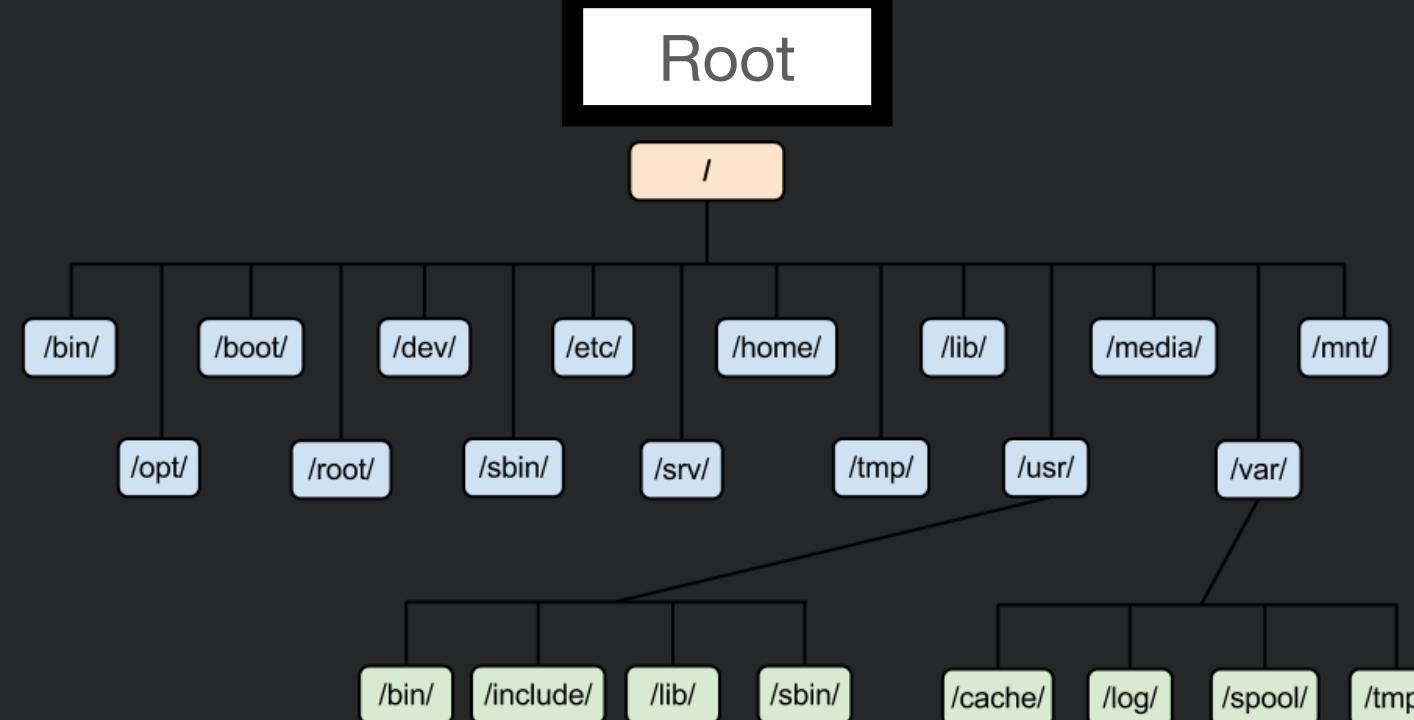
Process Management
Stack

In session exercises

Stack
Functions

Path

```
# go to another directory.  
# 1. relative.  
# 2. absolute.  
cd <path>  
  
# print working directory  
pwd
```



Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

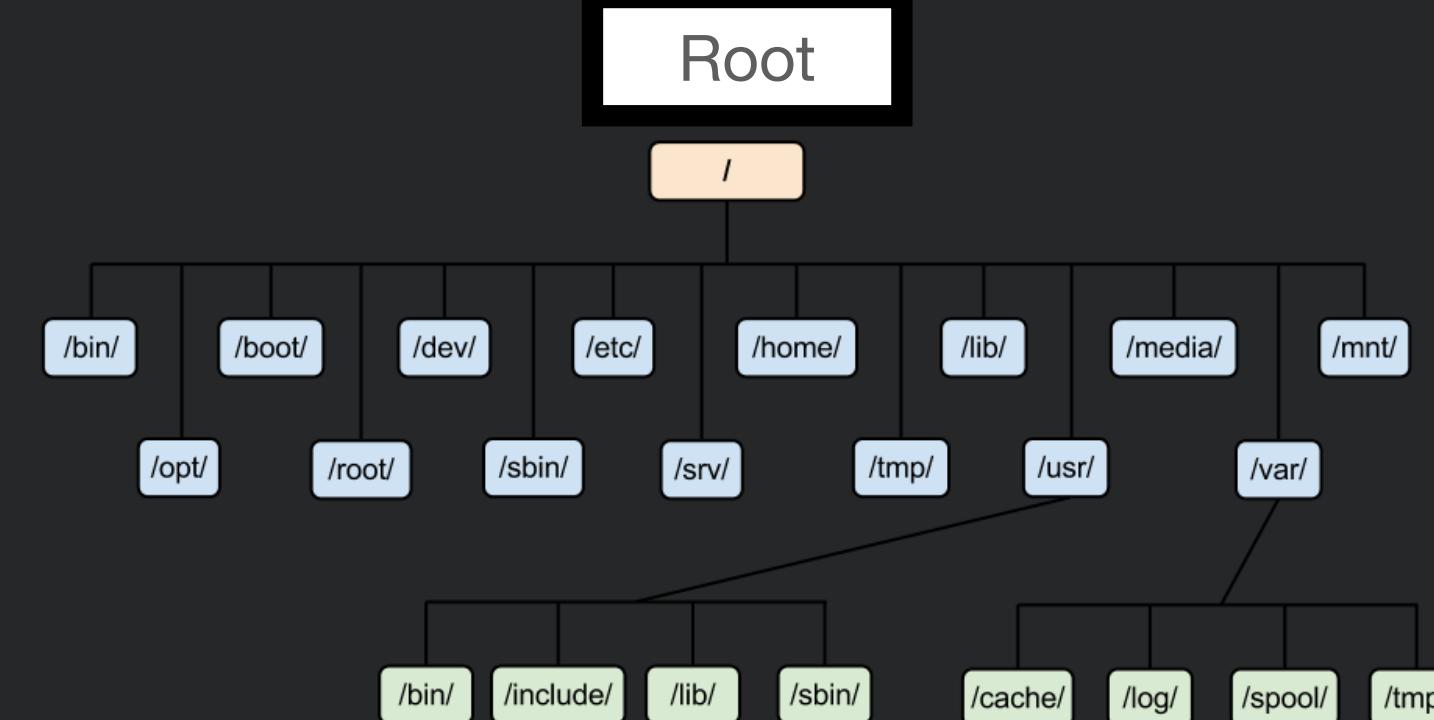
In session exercises

Stack
Functions

Path

```
# go to another directory.  
# 1. relative.  
# 2. absolute.  
cd <path>
```

```
# print working directory  
pwd
```



Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

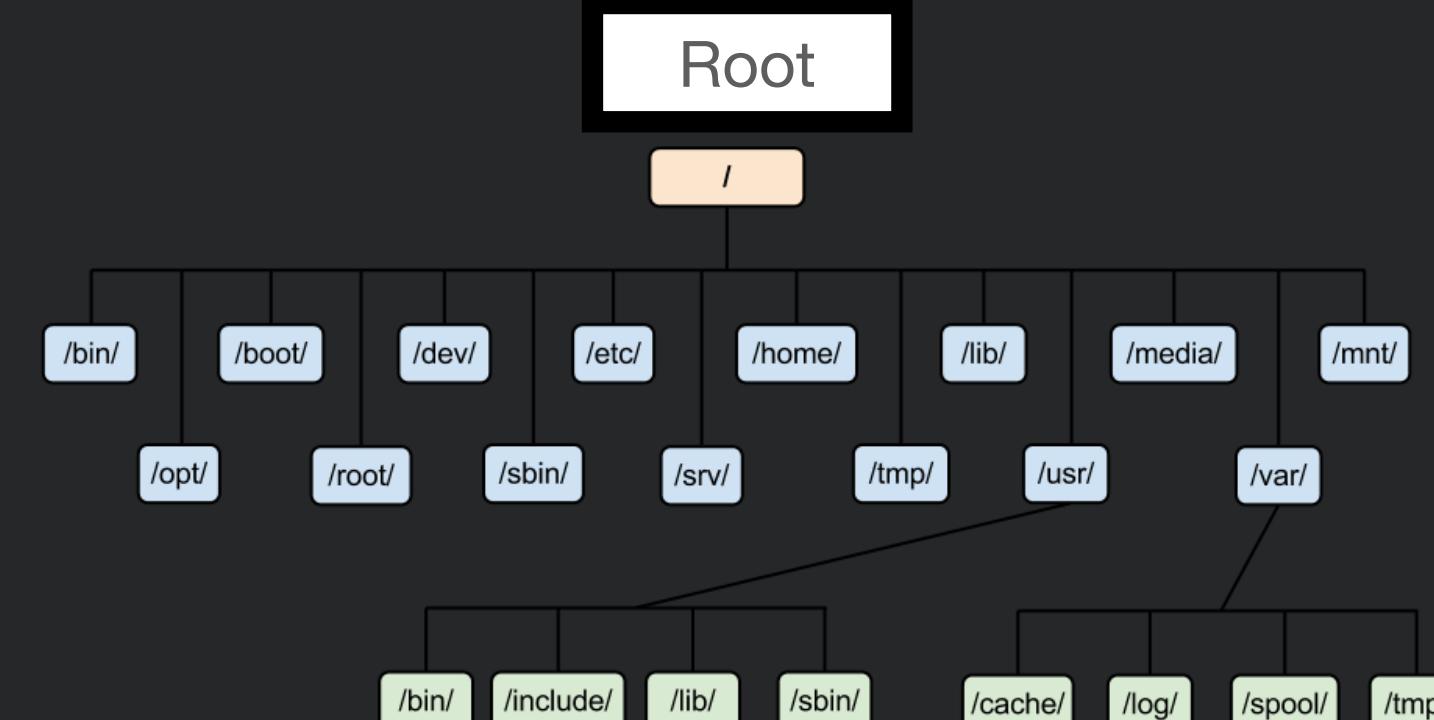
In session exercises

Stack
Functions

Path

```
# go to another directory.  
# 1. relative.  
# 2. absolute.  
cd <path>
```

```
# print working directory  
pwd
```



Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Directory

```
# make directory
mkdir

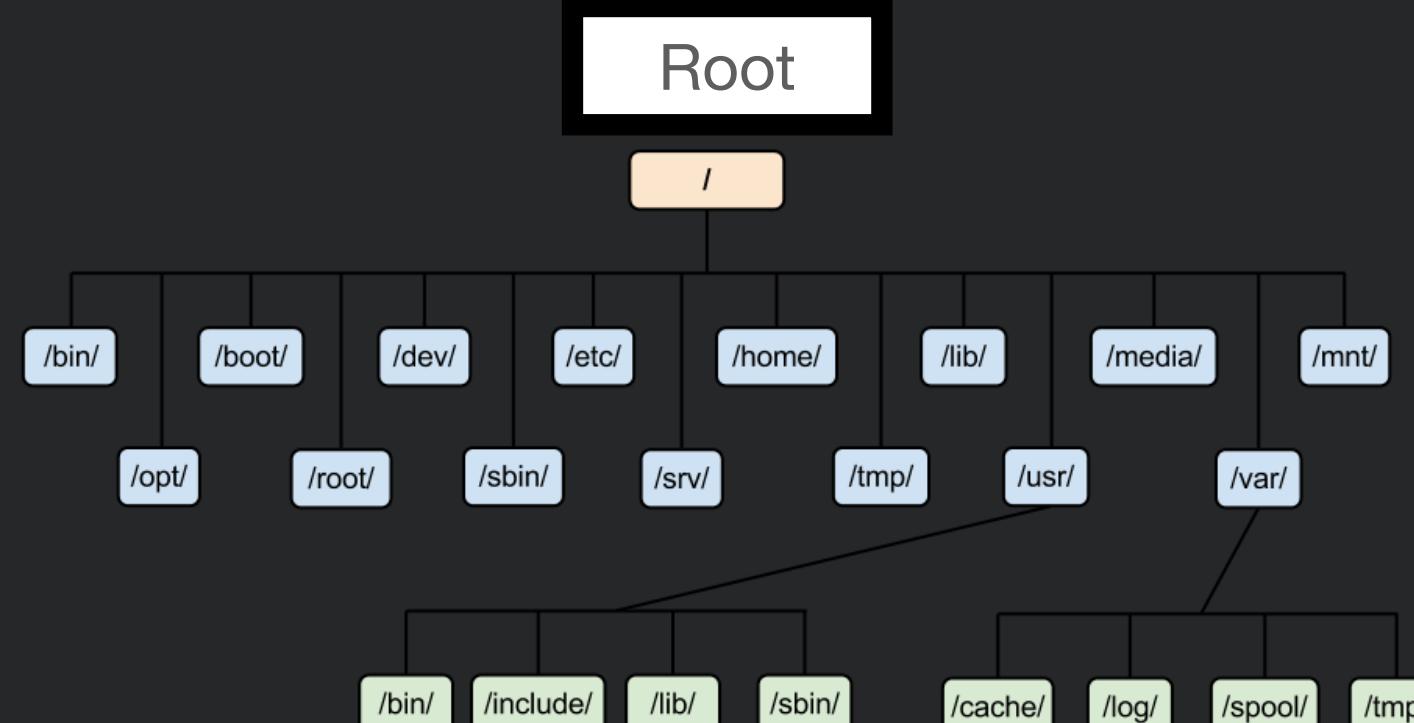
# remove directory
rm -r

# move directory
mv

# copy directory
cp

# list directory content
ls

# get directory size.
du -h --max-depth=1 <path>
```



Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Directory

```
# make directory
mkdir

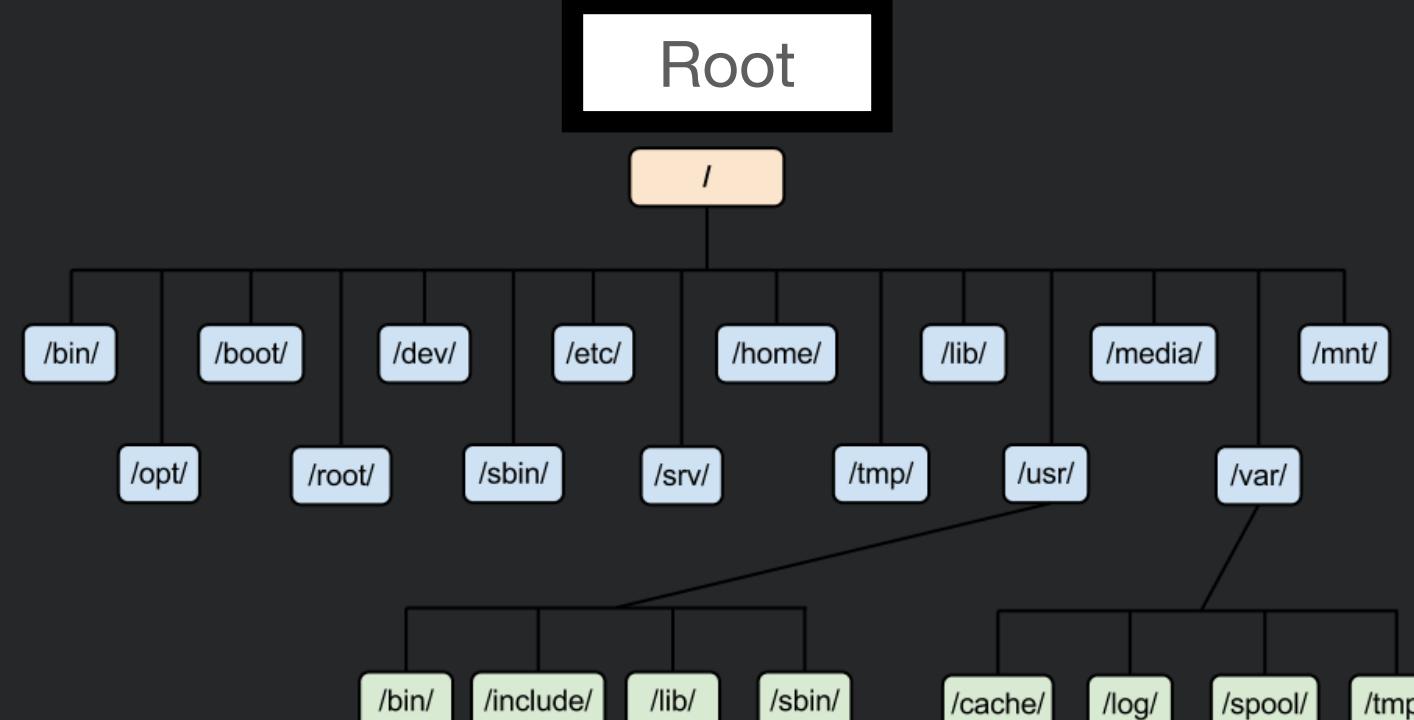
# remove directory
rm -r

# move directory
mv

# copy directory
cp

# list directory content
ls

# get directory size.
du -h --max-depth=1 <path>
```



Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Directory

```
# make directory
mkdir

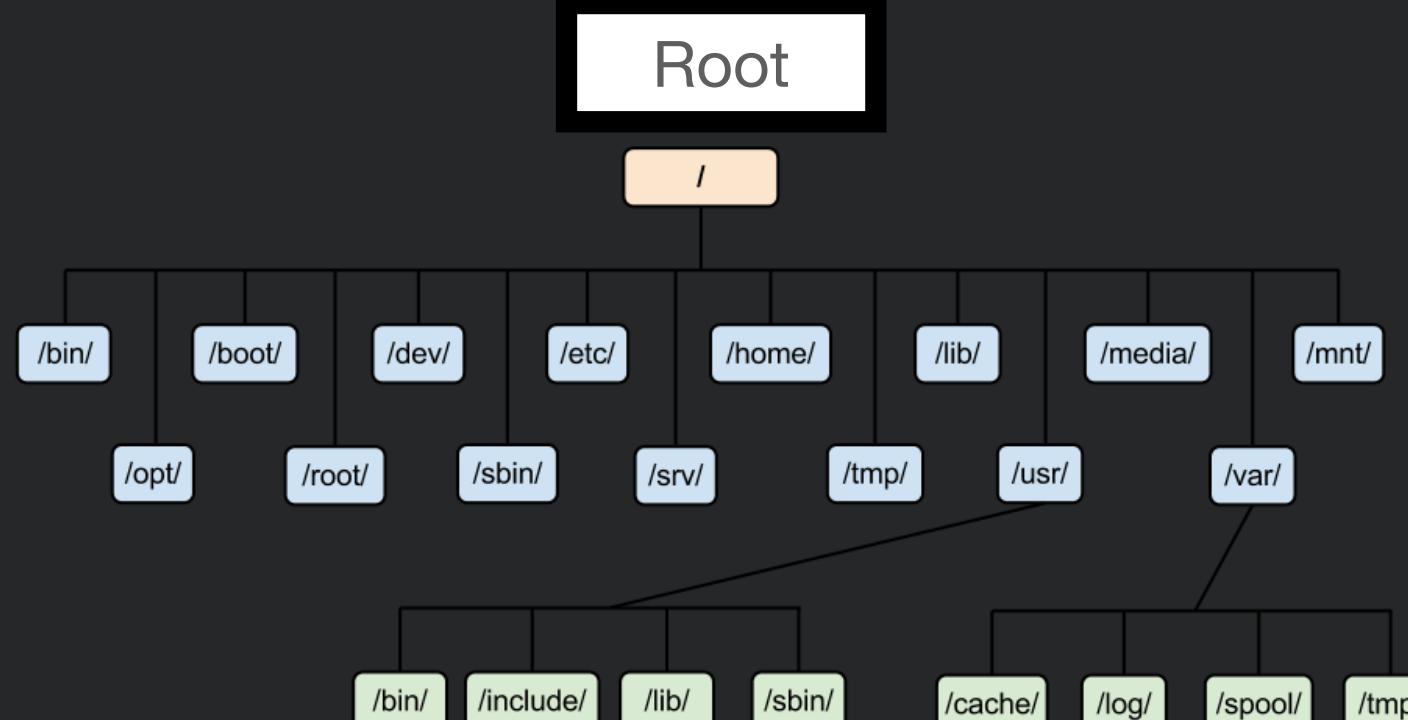
# remove directory
rm -r

# move directory
mv

# copy directory
cp

# list directory content
ls

# get directory size.
du -h --max-depth=1 <path>
```



Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Directory

```
# make directory  
mkdir
```

```
# remove directory  
rm -r
```

```
# move directory  
mv
```

```
# copy directory  
cp
```

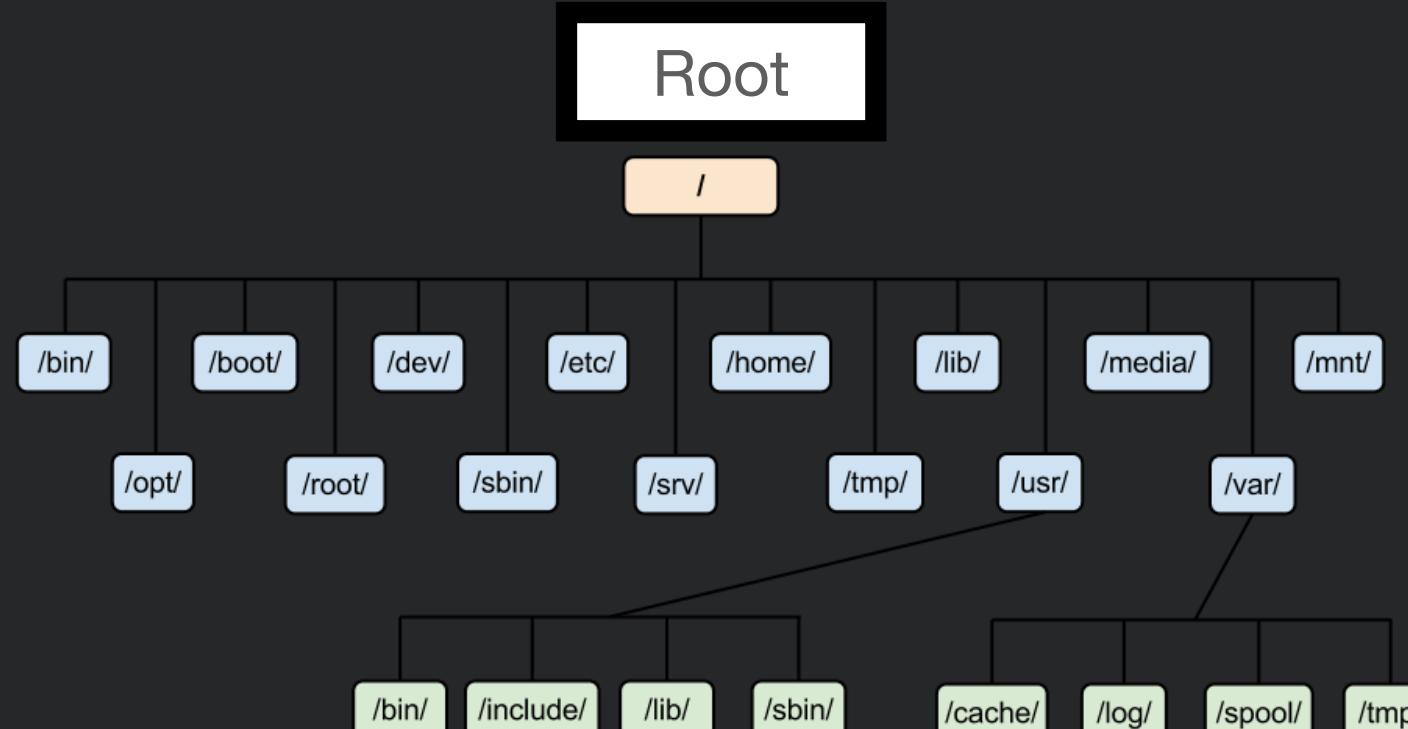
```
# list directory content  
ls
```

```
# get directory size.  
du -h --max-depth=1 <path>
```

Stack
Functions

Commands

Interaction
Hardware



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Directory

```
# make directory  
mkdir
```

```
# remove directory  
rm -r
```

```
# move directory  
mv
```

```
# copy directory  
cp
```

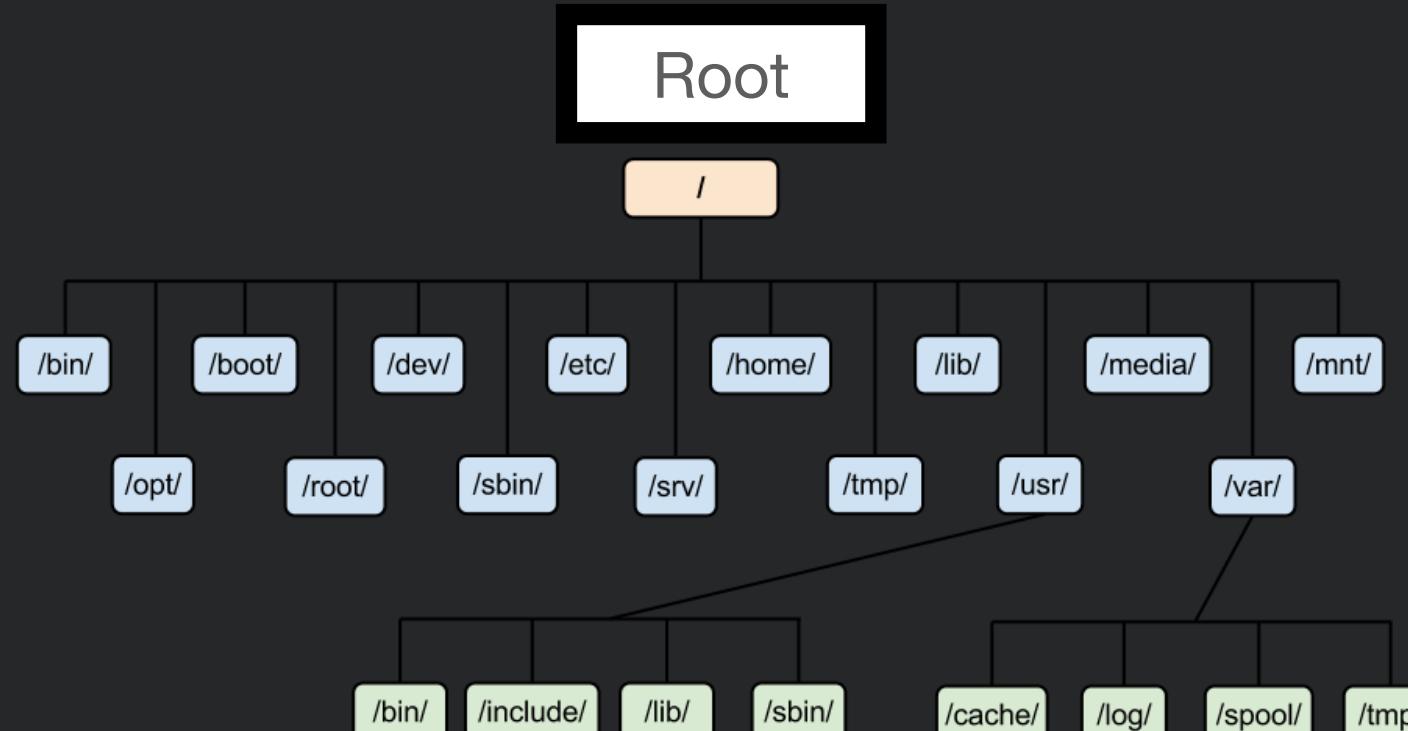
```
# list directory content  
ls
```

```
# get directory size.  
du -h --max-depth=1 <path>
```

Stack
Functions

Commands

Interaction
Hardware



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Directory

```
# make directory  
mkdir
```

```
# remove directory  
rm -r
```

```
# move directory  
mv
```

```
# copy directory  
cp
```

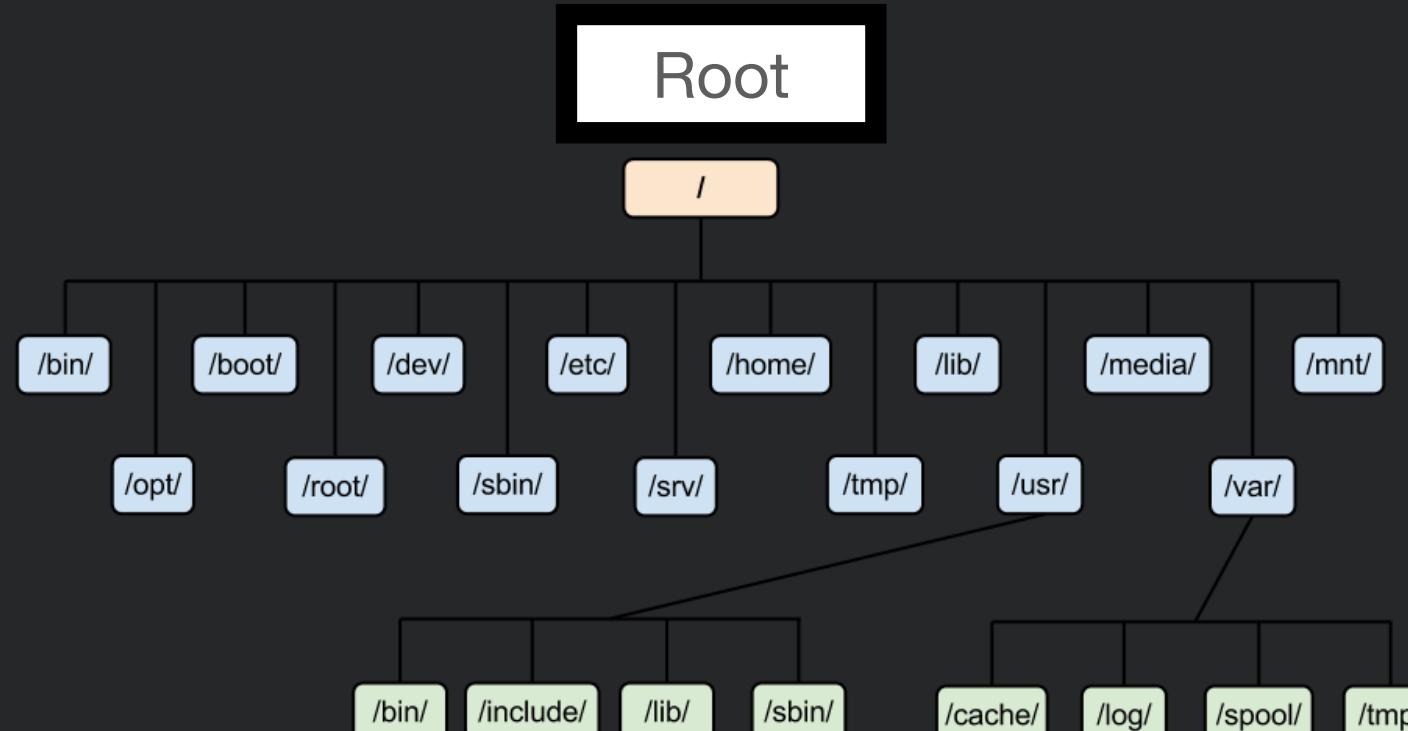
```
# list directory content  
ls
```

```
# get directory size.  
du -h --max-depth=1 <path>
```

Stack
Functions

Commands

Interaction
Hardware



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Directory

```
# make directory  
mkdir
```

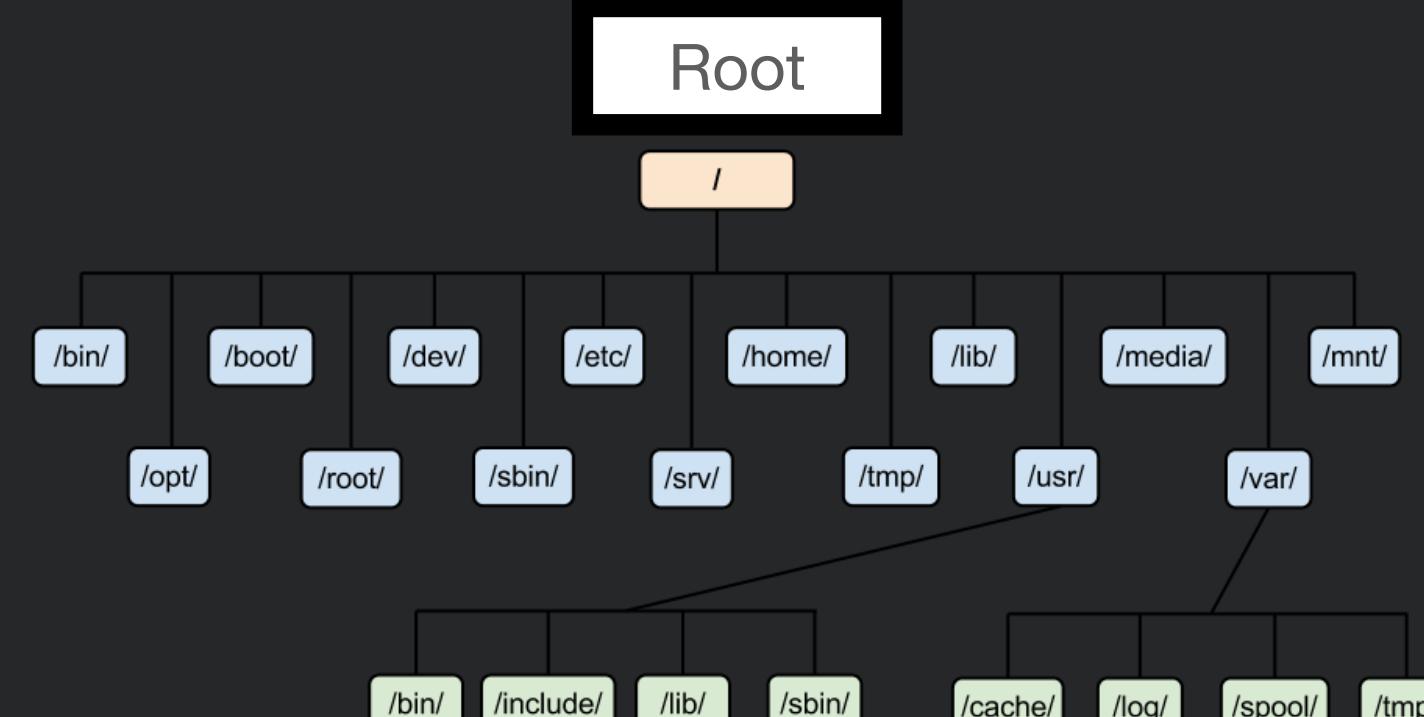
```
# remove directory  
rm -r
```

```
# move directory  
mv
```

```
# copy directory  
cp
```

```
# list directory content  
ls
```

```
# get directory size.  
du -h --max-depth=1 <path>
```



Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

File

```
# create file
touch

# write file
vi
gedit
nano

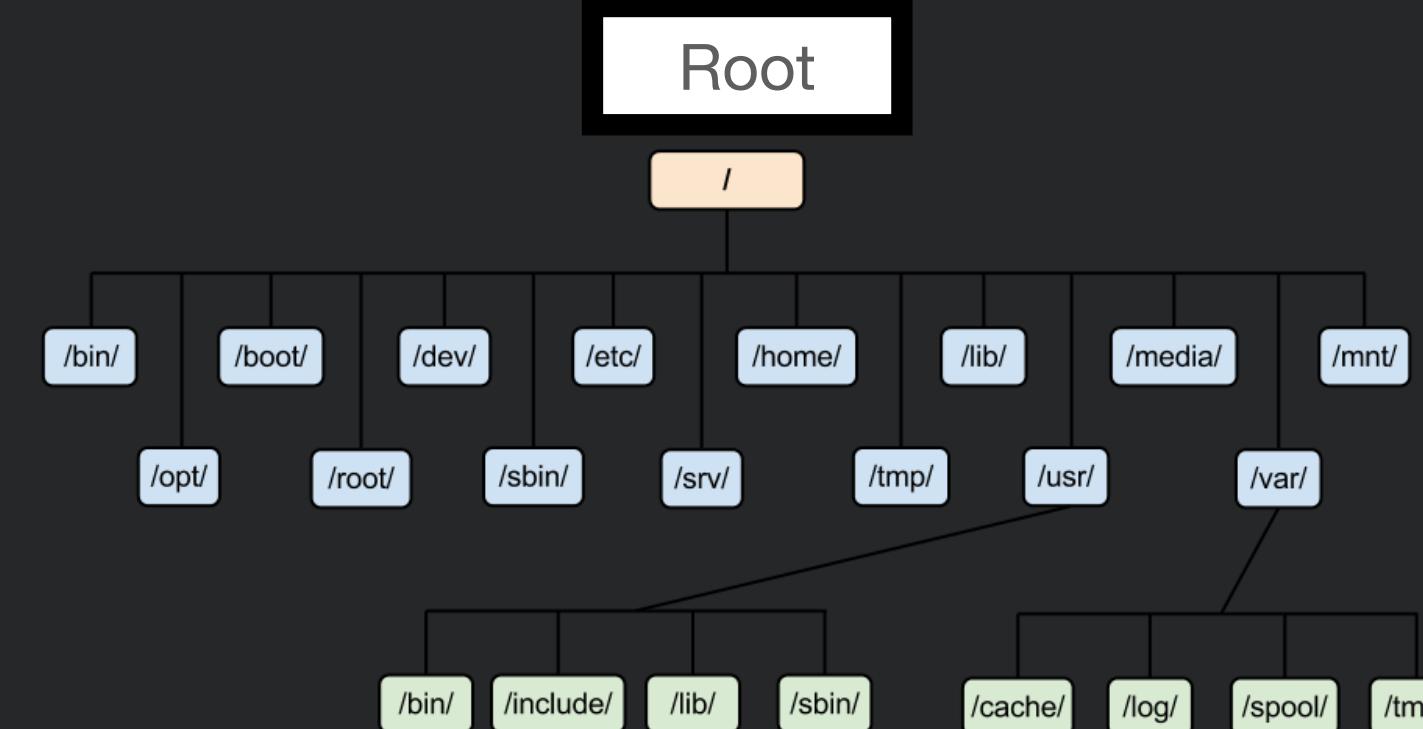
# Read directory
vi
cat
nano
gedit

# get information about file
file

# remove file
rm

# move file
mv

# copy file
cp
```



Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

File

```
# create file
touch

# write file
vi
gedit
nano

# Read directory
vi
cat
nano
gedit

# get information about file
file

# remove file
rm

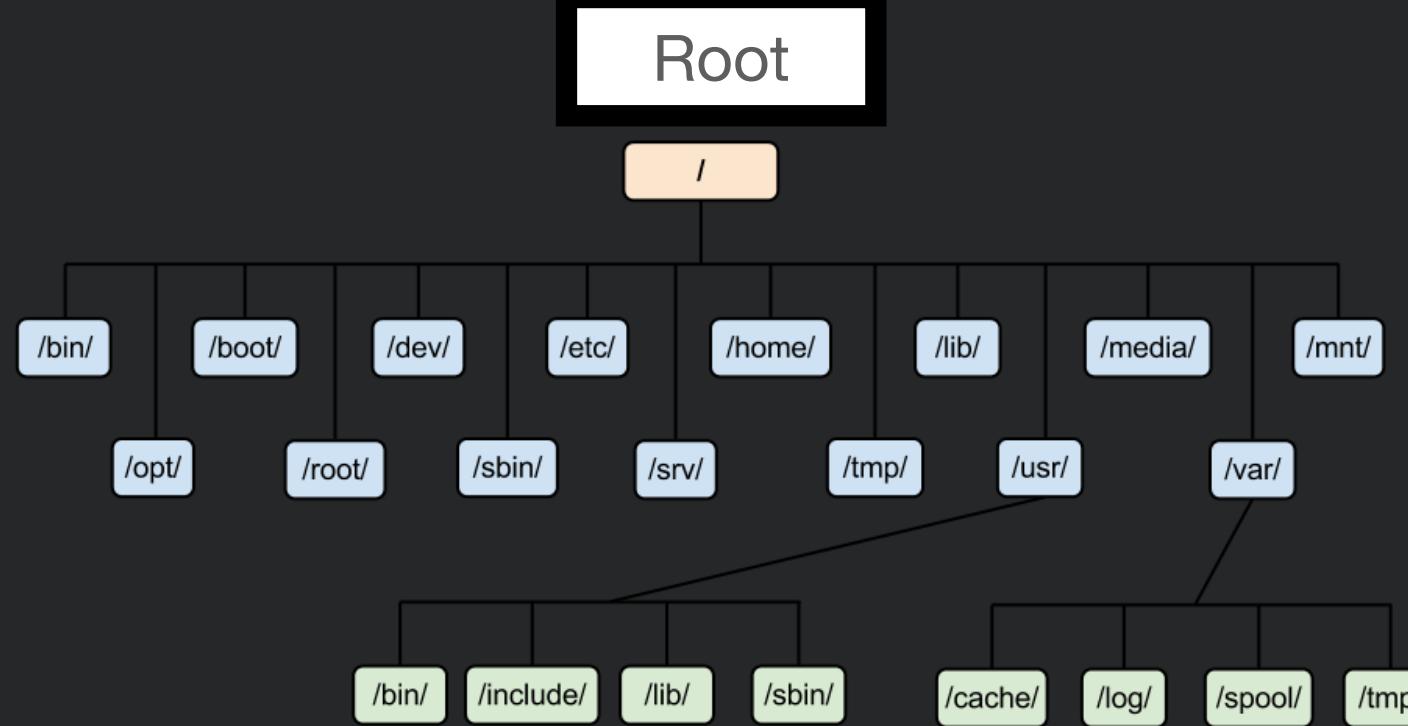
# move file
mv

# copy file
cp
```

Stack
Functions

Commands

Interaction
Hardware



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

File

```
# create file
touch

# write file
vi
gedit
nano

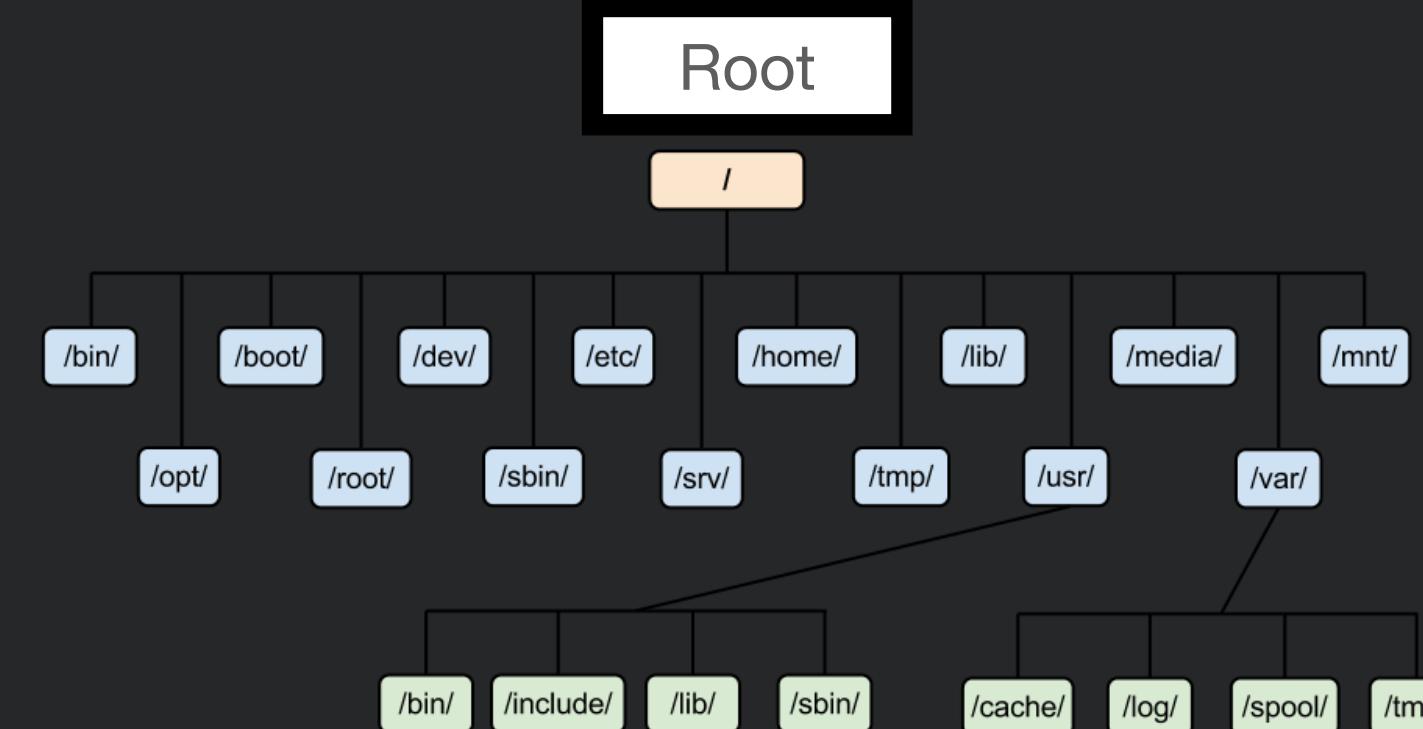
# Read directory
vi
cat
nano
gedit

# get information about file
file

# remove file
rm

# move file
mv

# copy file
cp
```



Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

File

```
# create file  
touch
```

```
# write file  
vi  
gedit  
nano
```

```
# Read directory  
vi  
cat  
nano  
gedit
```

```
# get information about file  
file
```

```
# remove file  
rm
```

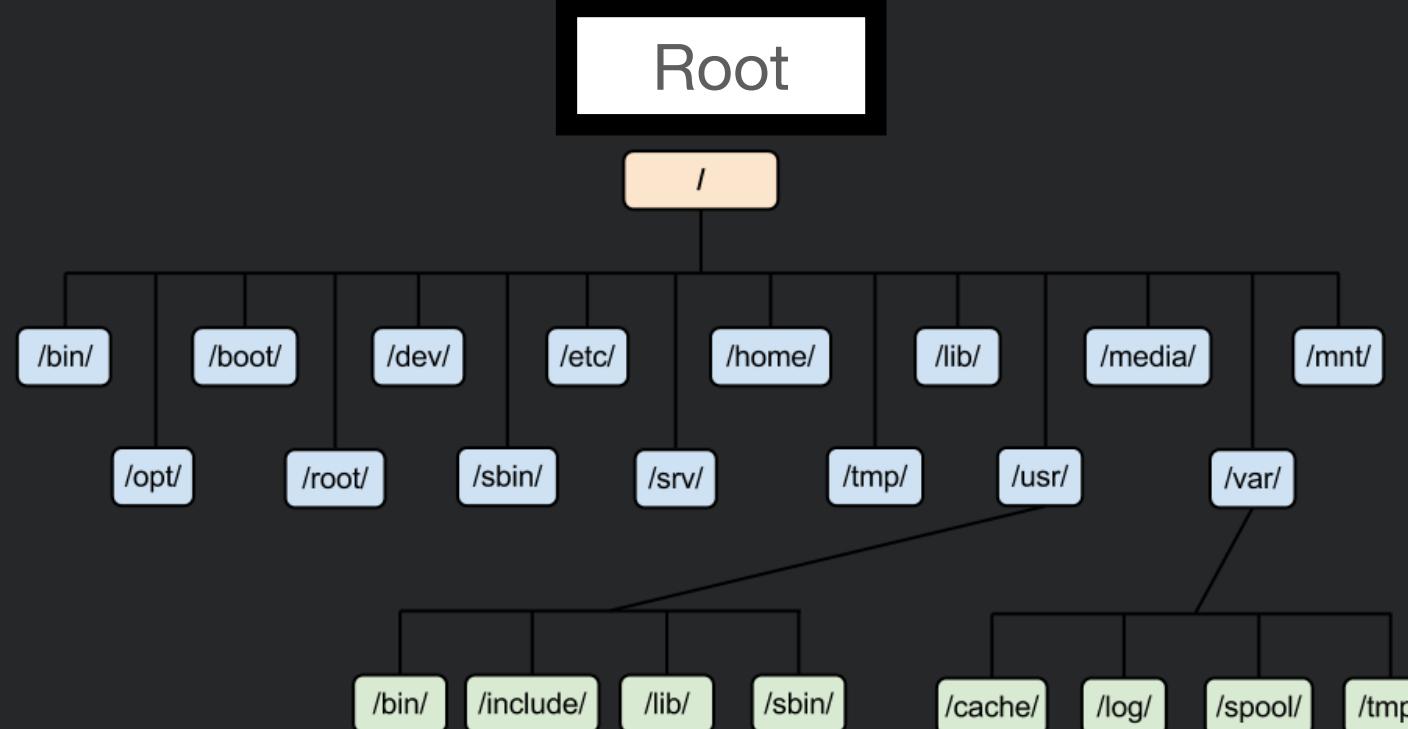
```
# move file  
mv
```

```
# copy file  
cp
```

Stack
Functions

Commands

Interaction
Hardware



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

File

```
# create file
touch

# write file
vi
gedit
nano

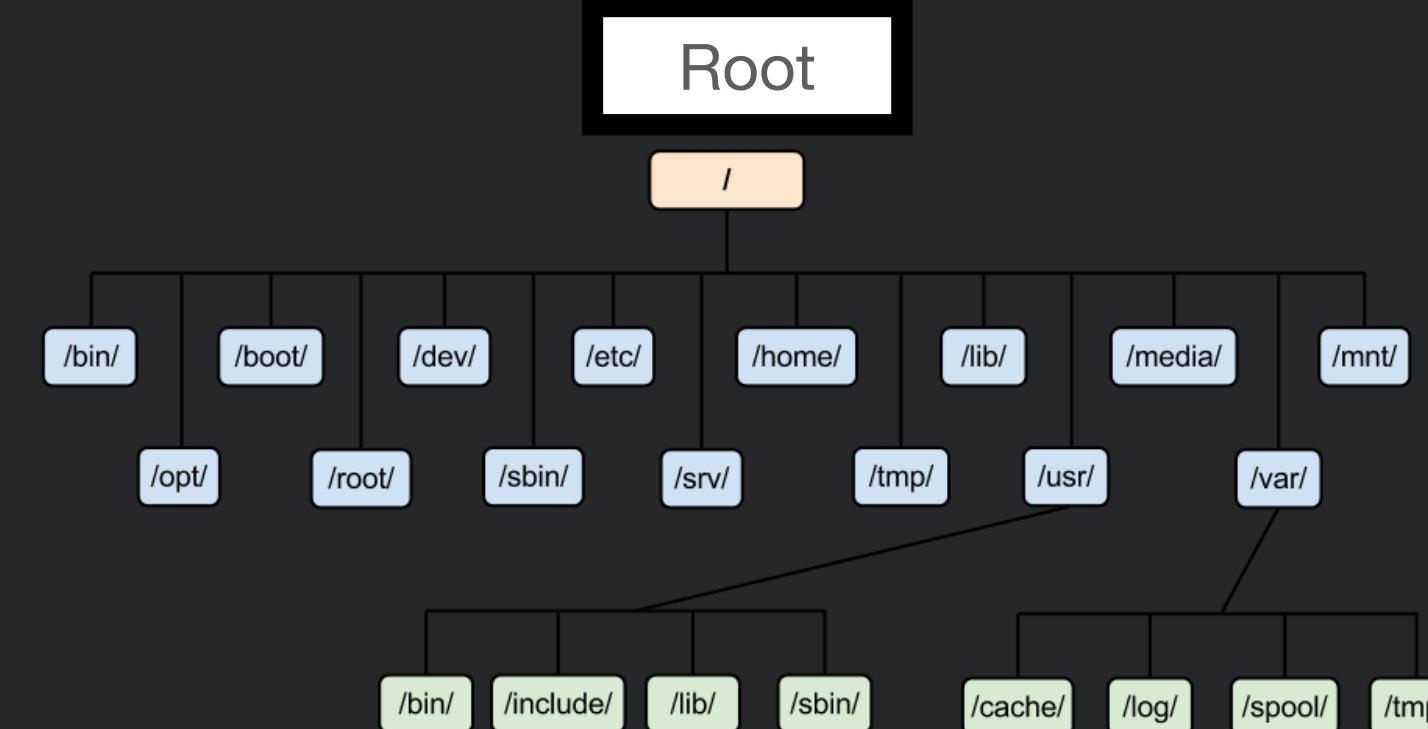
# Read directory
vi
cat
nano
gedit

# get information about file
file

# remove file
rm

# move file
mv

# copy file
cp
```



Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

File

```
# create file
touch

# write file
vi
gedit
nano

# Read directory
vi
cat
nano
gedit

# get information about file
file
```

```
# remove file
rm
```

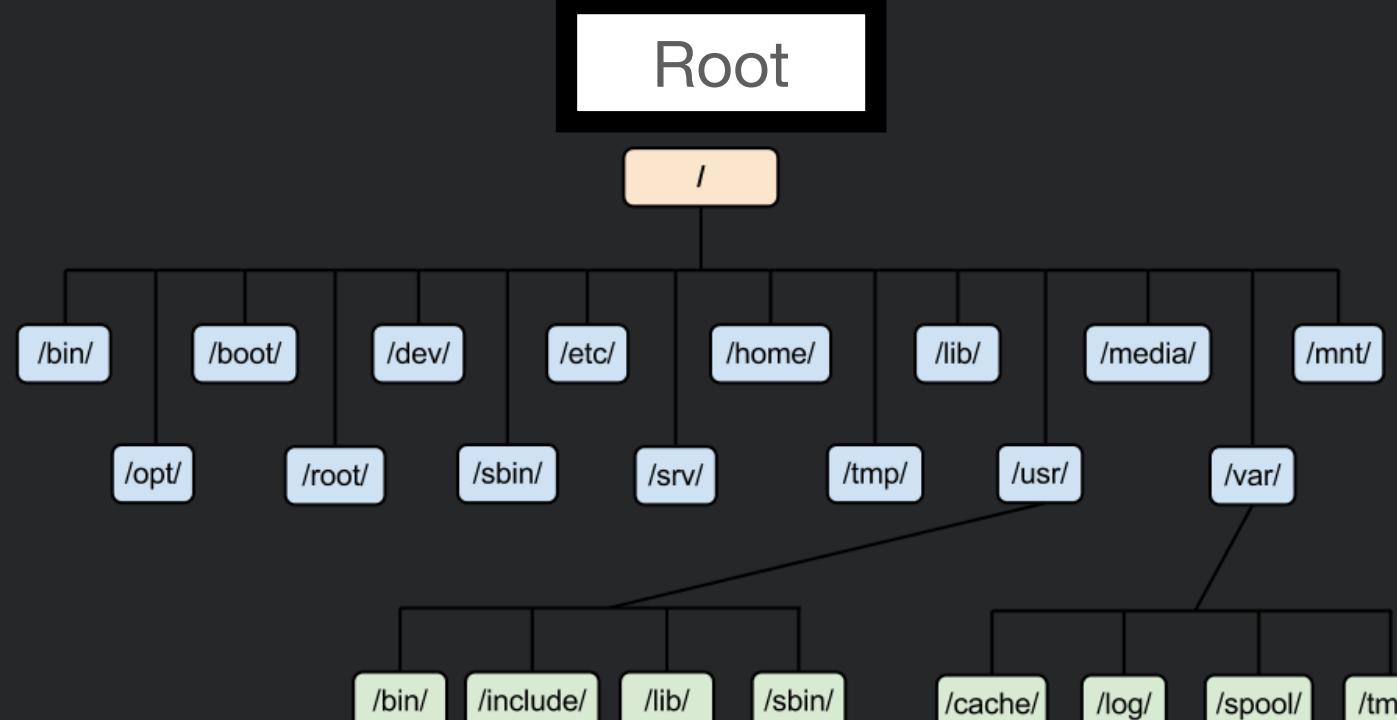
```
# move file
mv
```

```
# copy file
cp
```

Stack
Functions

Commands

Interaction
Hardware



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

File

```
# create file
touch

# write file
vi
gedit
nano

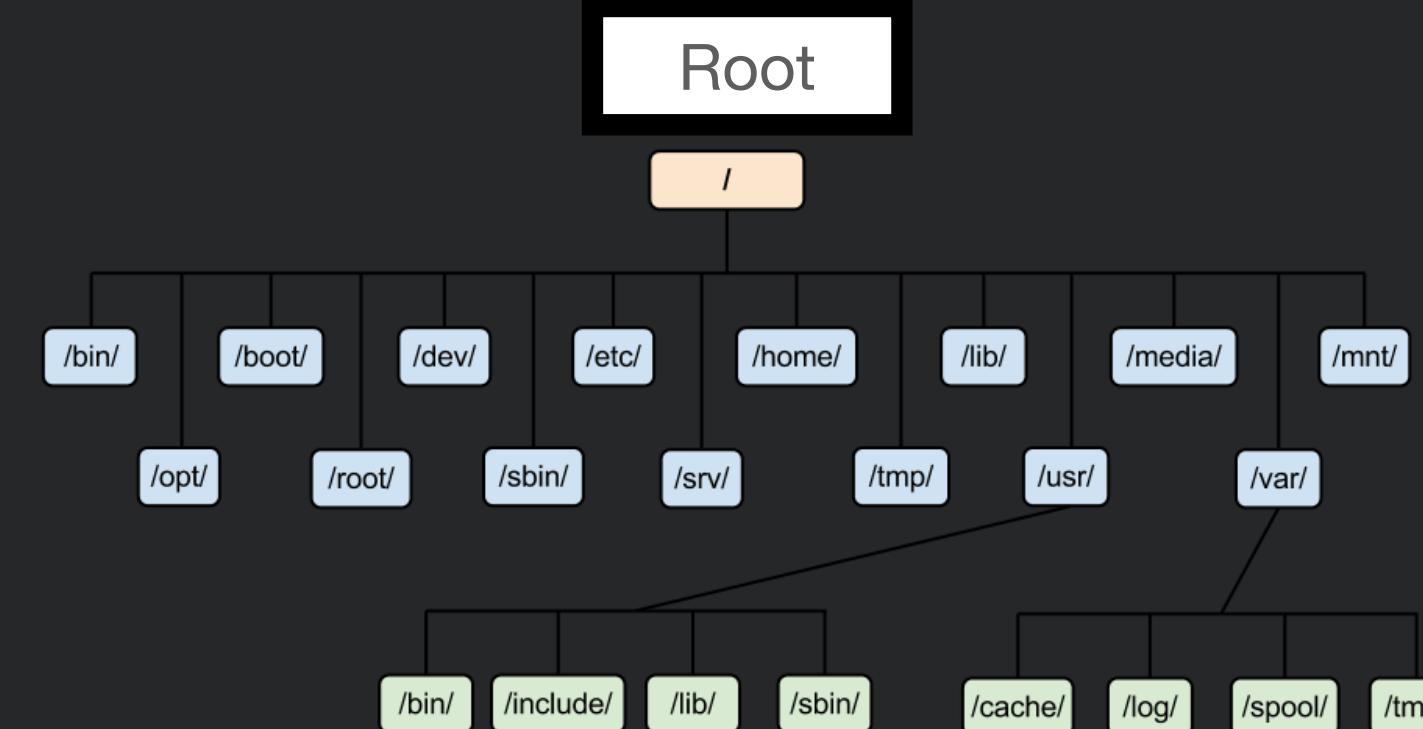
# Read directory
vi
cat
nano
gedit

# get information about file
file

# remove file
rm

# move file
mv

# copy file
cp
```



Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

File

```
# create file
touch

# write file
vi
gedit
nano

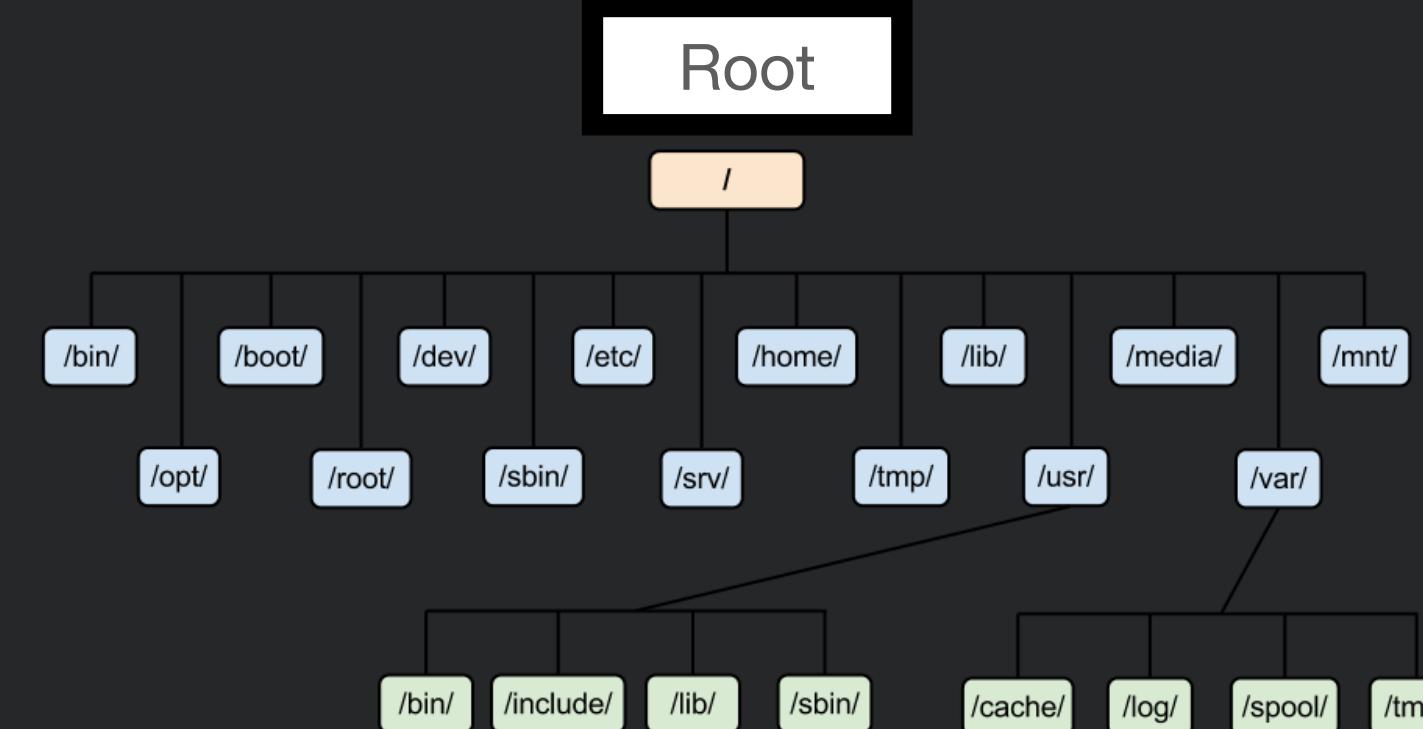
# Read directory
vi
cat
nano
gedit

# get information about file
file

# remove file
rm

# move file
mv

# copy file
cp
```



Stack
Functions

Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Stack
Functions

Commands

Interaction
Hardware

Storage / Block Device

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises



Task (A) Requirements:

```
# Make at least two partitions on your SD-CARD using gparted.  
# Create for each partitions filesystem ( first one ext4 &  
second one ext2 ).  
# Mount two partitions on your root filesystem.  
# Add some files inside each one.  
# reboot your machine.  
# check if mounting points still exists, it should not.  
# Make the ext4 persistance by adding /etc/fstab file —→  
(search how you can do that).  
# reboot your system.  
# Check if the ext4 is mounted.
```

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises



Exercise 1: Basic Navigation

Use `ls` to list all files and directories **in** the current directory.

Use `cd` to navigate to a specific directory.

Use `pwd` to print the current working directory.

Exercise 2: File and Directory Operations

Create a directory named "practice" **in** the current directory using `mkdir`.

Create an empty file named "file.txt" within the "practice" directory using `touch`.

Copy "file.txt" to a new file "file_backup.txt" using `cp`.

Move "file_backup.txt" to another directory using `mv`.

Rename "file.txt" to "new_file.txt" using `mv`.

Delete the "new_file.txt" using `rm`.

Exercise 3: File Viewing and Editing

Create a text file using `echo` or a text editor like `nano`.

View the contents of the file using `cat`. View the contents of the file using `less`. Edit the file using `nano` or another text editor. Redirect the output of a command (e.g., `ls`) to a file using `>`.

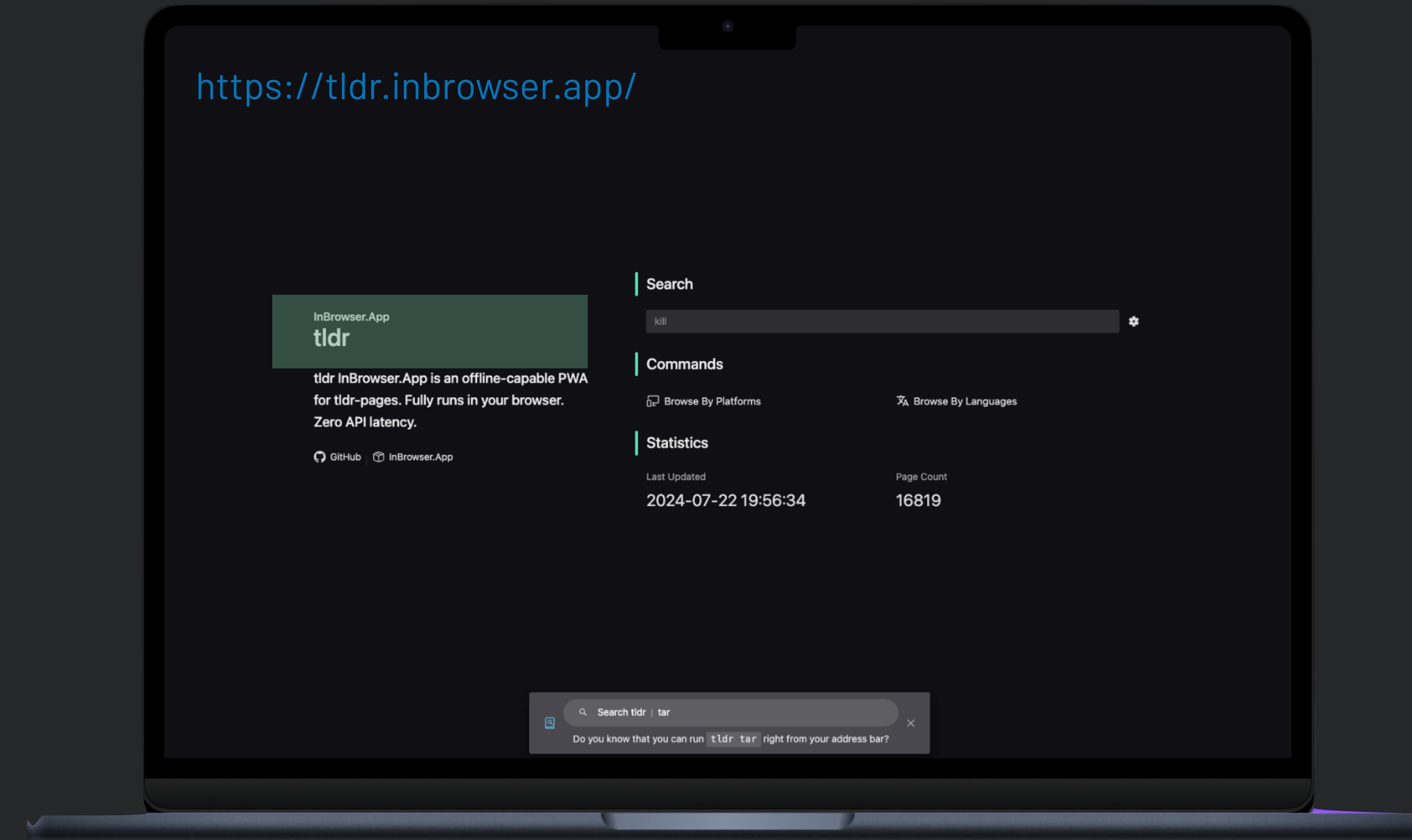
Introduction to Linux
Stacks

File system Stack

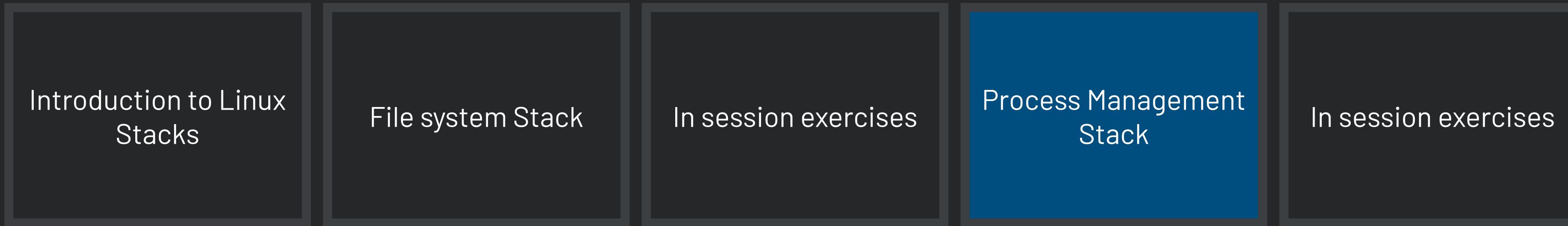
In session exercises

Process Management
Stack

In session exercises



New Link



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Stack
Definition

Commands

Interaction
Hardware

Process = Program in RAM

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Stack
Definition

Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Process = Program in RAM



main.cpp

Stack
Definition

Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

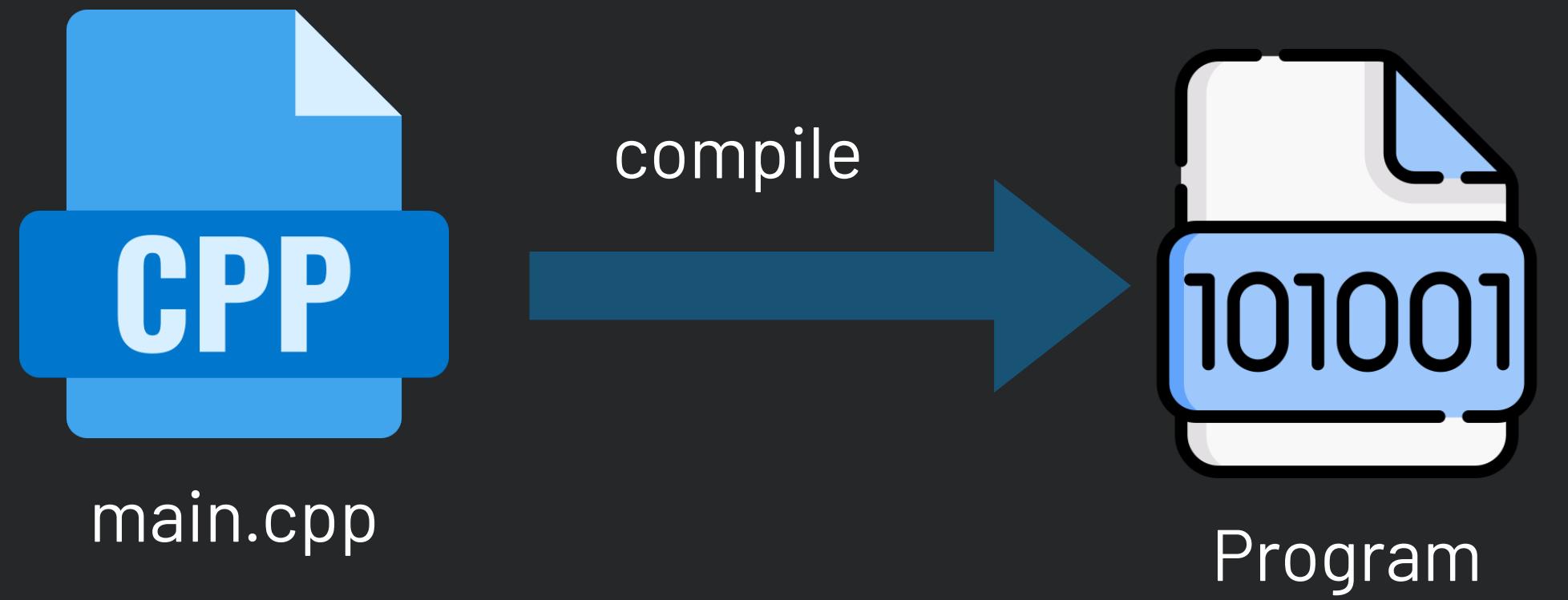
In session exercises

Stack
Definition

Commands

Interaction
Hardware

Process = Program in RAM



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Stack
Definition

Commands

Interaction
Hardware

Process = Program in RAM



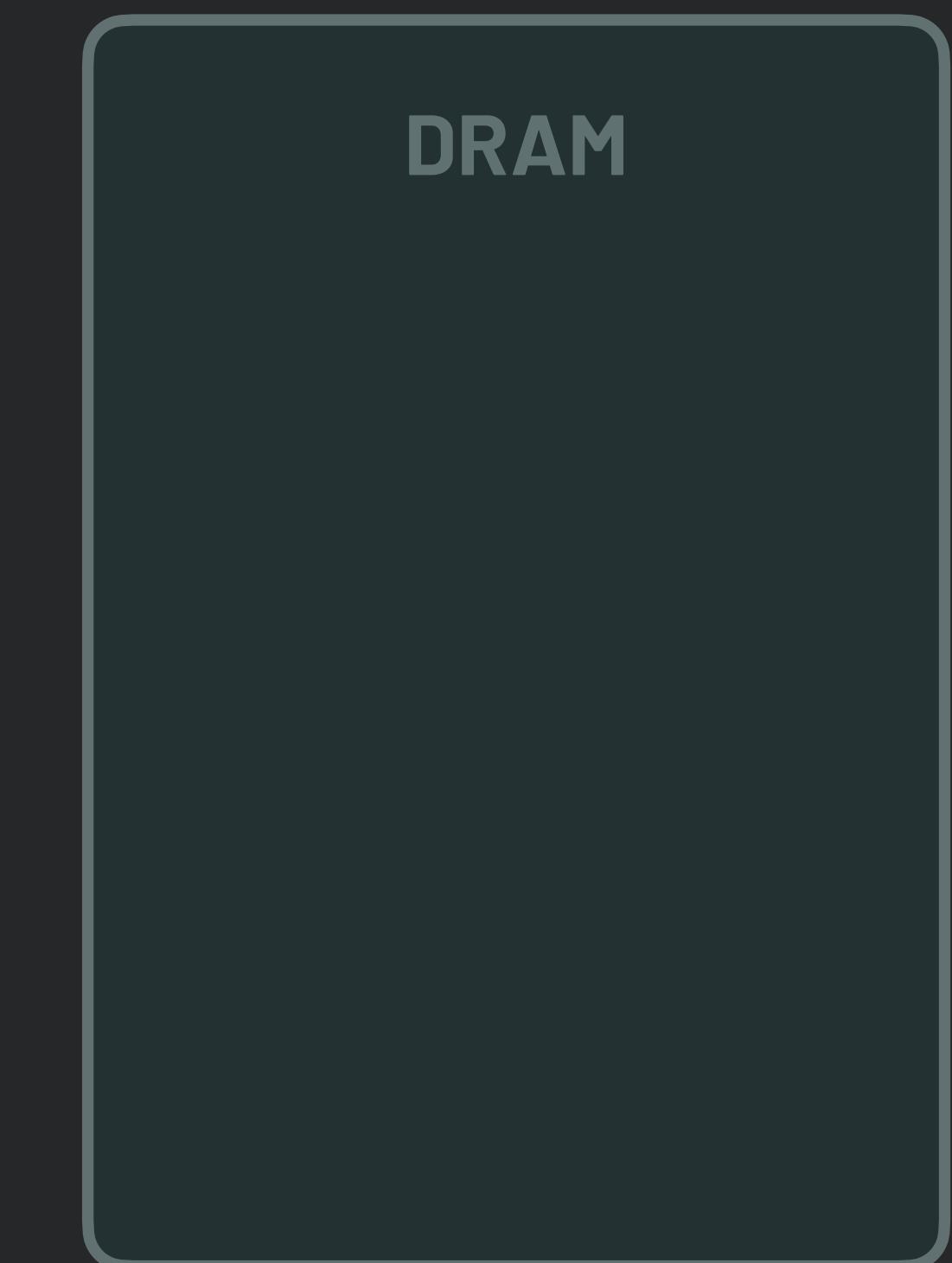
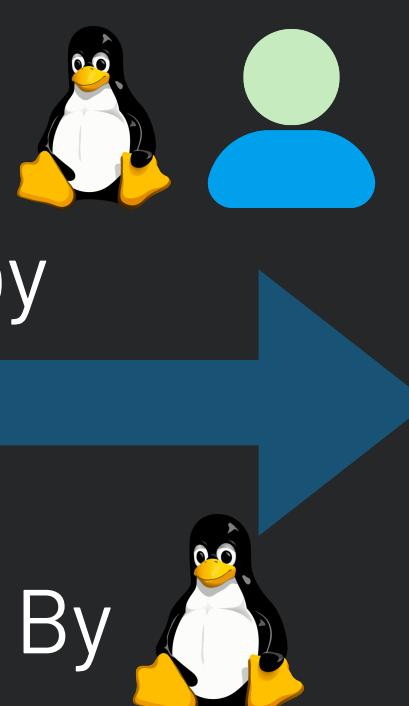
compile



Program

Execute by

Add RAM By



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

Process = Program in RAM

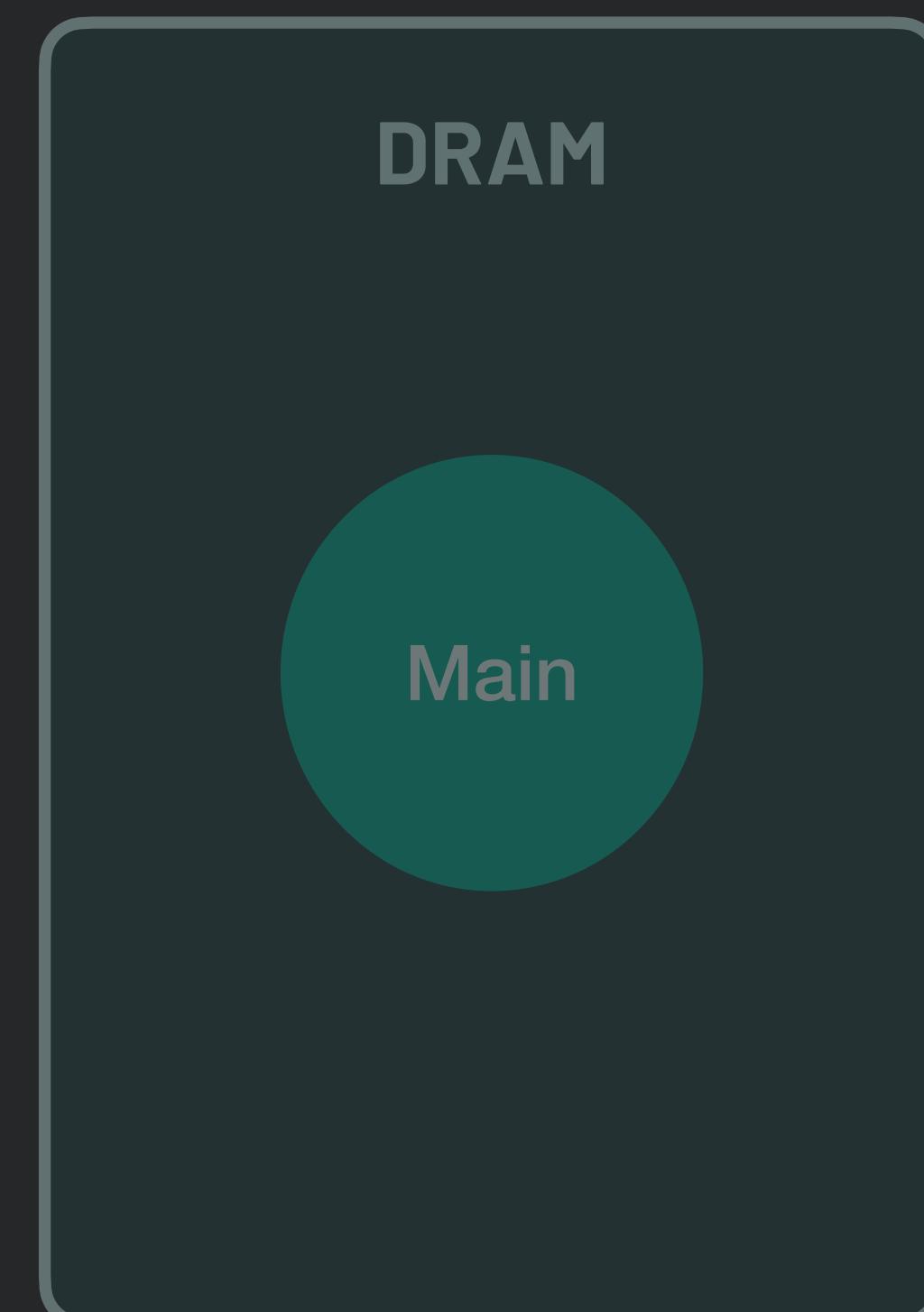
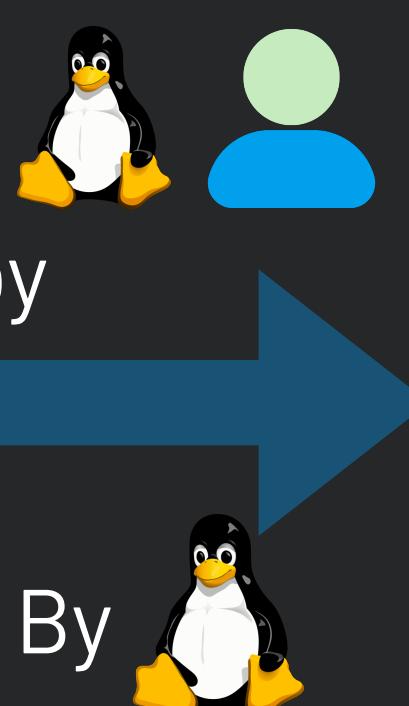


compile



Execute by

Add RAM By



Stack
Definition

Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

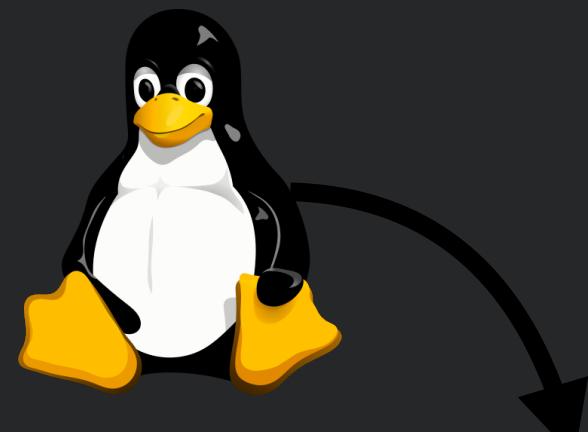
Process Management
Stack

In session exercises

Process properties

DRAM

Main



- Has priority
- State
- Has a portion from RAM (STACK/HEAP)
- PID

Stack
Definition

Commands

Interaction
Hardware

Introduction to Linux
Stacks

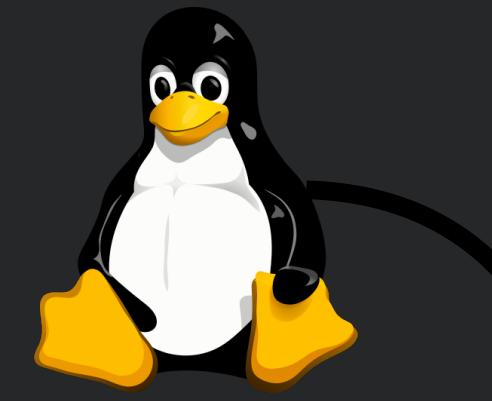
File system Stack

In session exercises

Process Management
Stack

In session exercises

Process properties



- Has priority
- State
- Has a portion from RAM (STACK/HEAP)
- PID

DRAM

Main

Stack
Definition

Commands

Interaction
Hardware

Introduction to Linux
Stacks

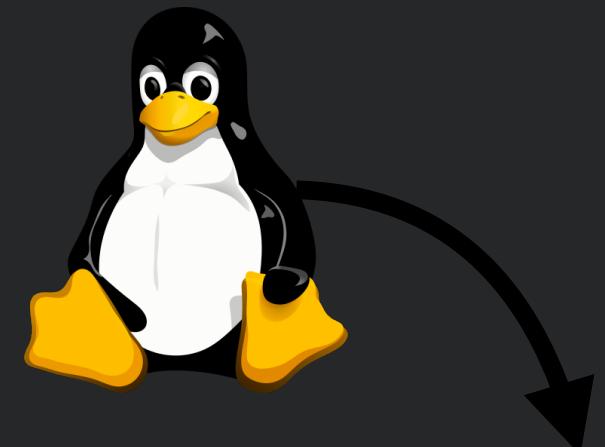
File system Stack

In session exercises

Process Management
Stack

In session exercises

Process properties



- Has priority
- State
- Has a portion from RAM (STACK/HEAP)
- PID

DRAM

Main

Stack
Definition

Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises



Run a process

mode 1:

 # foreground

./program_name

 # background

./program_name &

Kill a process.

kill pid

Assign a priority to specific process

at runtime

renice

before running a process

nice

get relation between processes.

pstree

monitoring processes.

top

ps

Stack
Definition

Commands

Interaction
Hardware

DRAM

Init



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

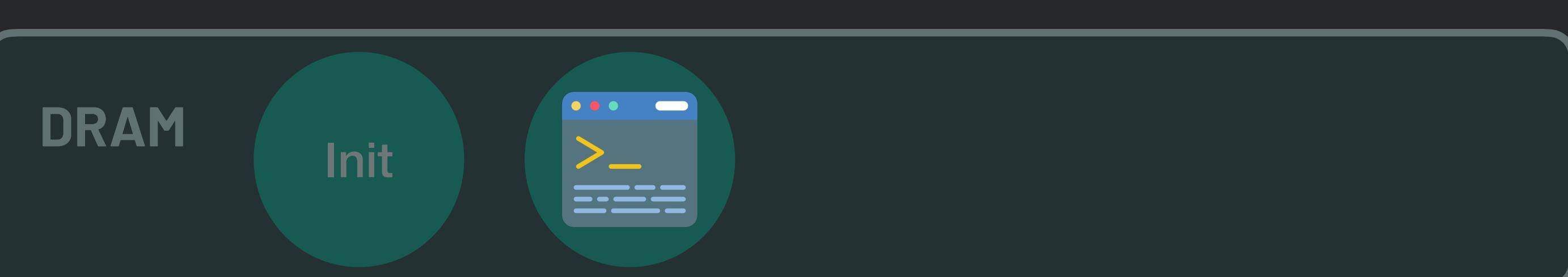
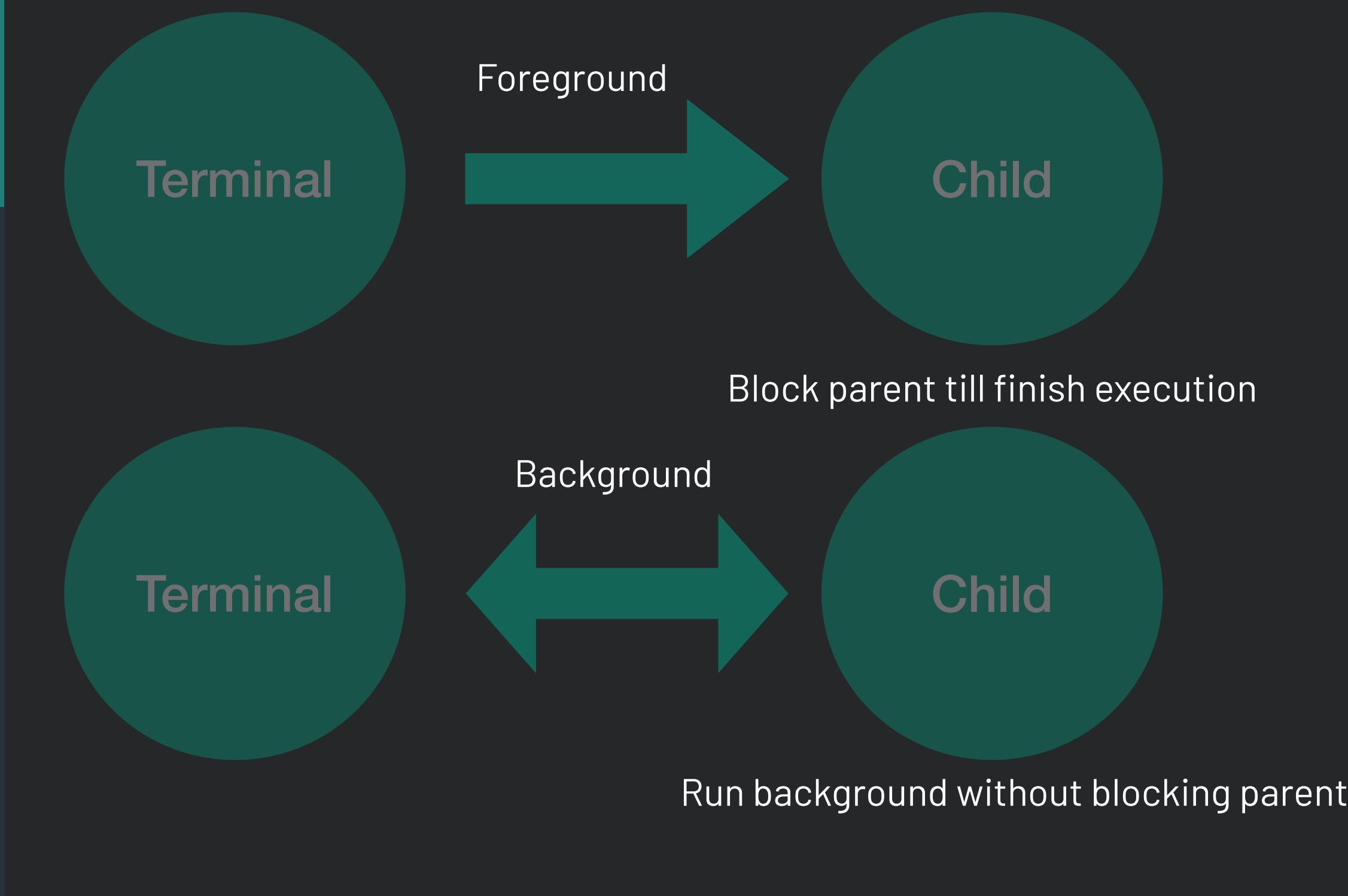
In session exercises

Stack
Definition

Commands

Interaction
Hardware

```
● ● ●  
  
# Run a process  
# mode 1:  
# foreground  
. /program_name  
# background  
. /program_name &  
  
# Kill a process.  
kill pid  
  
  
# Assign a priority to specific process  
# at runtime  
renice  
  
# before running a process  
nice  
  
# get relation between processes.  
pstree  
  
# monitoring processes.  
top  
ps
```



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises



```
# Run a process  
# mode 1:  
# foreground  
../program_name  
# background  
../program_name &
```

```
# Kill a process.  
kill pid
```

```
# Assign a priority to specific process  
# at runtime  
renice  
  
# before running a process  
nice  
  
# get relation between processes.  
pstree  
  
# monitoring processes.  
top  
ps
```

Stack
Definition

Commands

Interaction
Hardware

DRAM

Init



Pid
Child

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises



```
# Run a process
# mode 1:
# foreground
./program_name
# background
./program_name &
```

```
# Kill a process.
kill pid
```

```
# Assign a priority to specific process
# at runtime
renice
```

```
# before running a process
nice
```

```
# get relation between processes.
pstree
```

```
# monitoring processes.
top
ps
```

Stack
Definition

Commands

Interaction
Hardware

DRAM

Init



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises



Run a process

mode 1:

foreground

./program_name

background

./program_name &

Kill a process.

kill pid

Assign a priority to specific process

at runtime

renice

before running a process

nice

get relation between processes.

pstree

monitoring processes.

top

ps



Stack
Definition

Commands

Interaction
Hardware

DRAM

Init



Introduction to Linux
Stacks

File system Stack

In session exercises

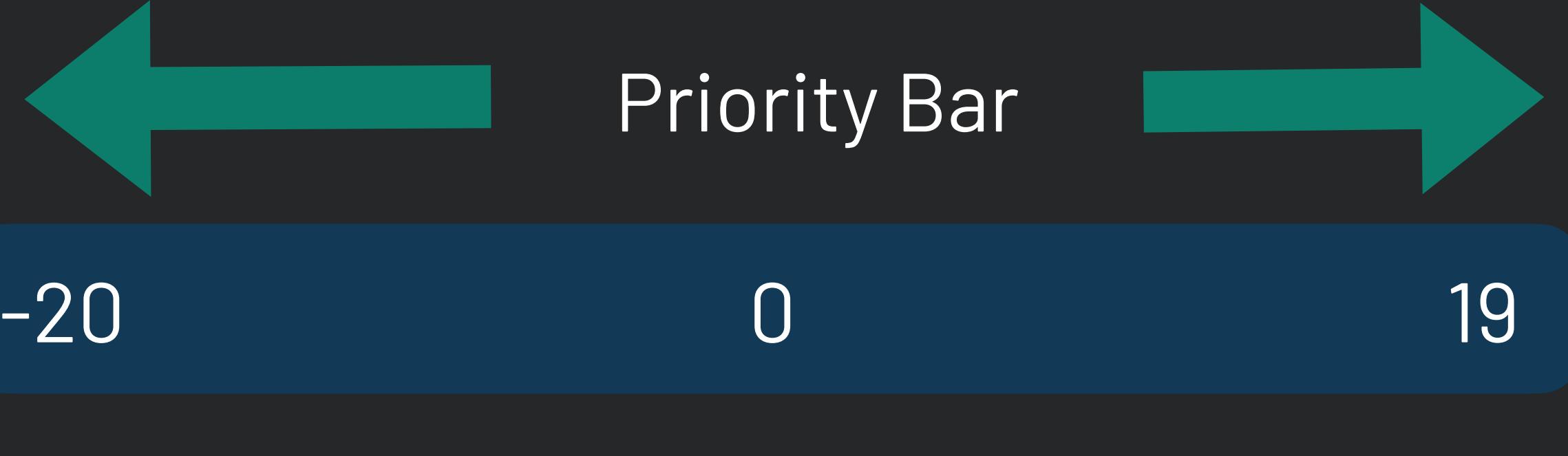
Process Management
Stack

In session exercises



```
# Run a process  
# mode 1:  
# foreground  
../program_name  
# background  
../program_name &  
  
# Kill a process.  
kill pid
```

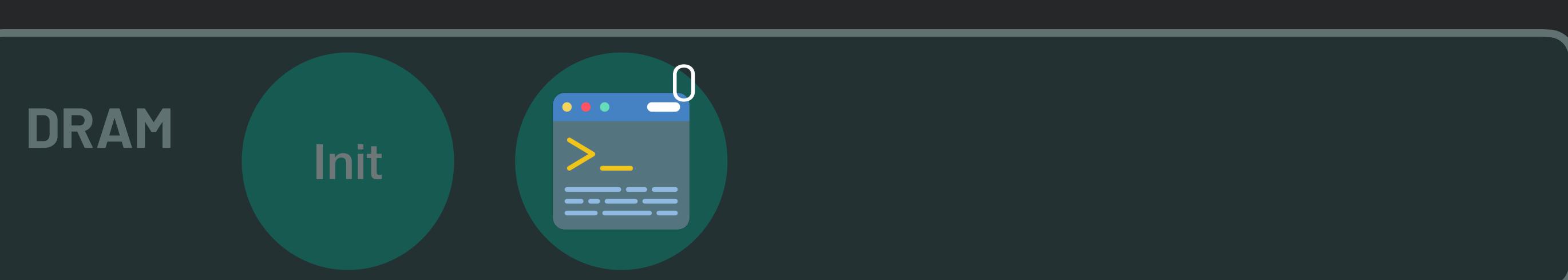
```
# Assign a priority to specific process  
# at runtime  
renice  
  
# before running a process  
nice  
  
# get relation between processes.  
pstree  
  
# monitoring processes.  
top  
ps
```



Stack
Definition

Commands

Interaction
Hardware



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises



Run a process

mode 1:

foreground

./program_name

background

./program_name &

Kill a process.

kill pid

Assign a priority to specific process

at runtime

renice

before running a process

nice

get relation between processes.

pstree

monitoring processes.

top

ps



Stack
Definition

Commands

Interaction
Hardware

DRAM

Init

-20
P1



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises



```
# Run a process
# mode 1:
# foreground
./program_name
# background
./program_name &

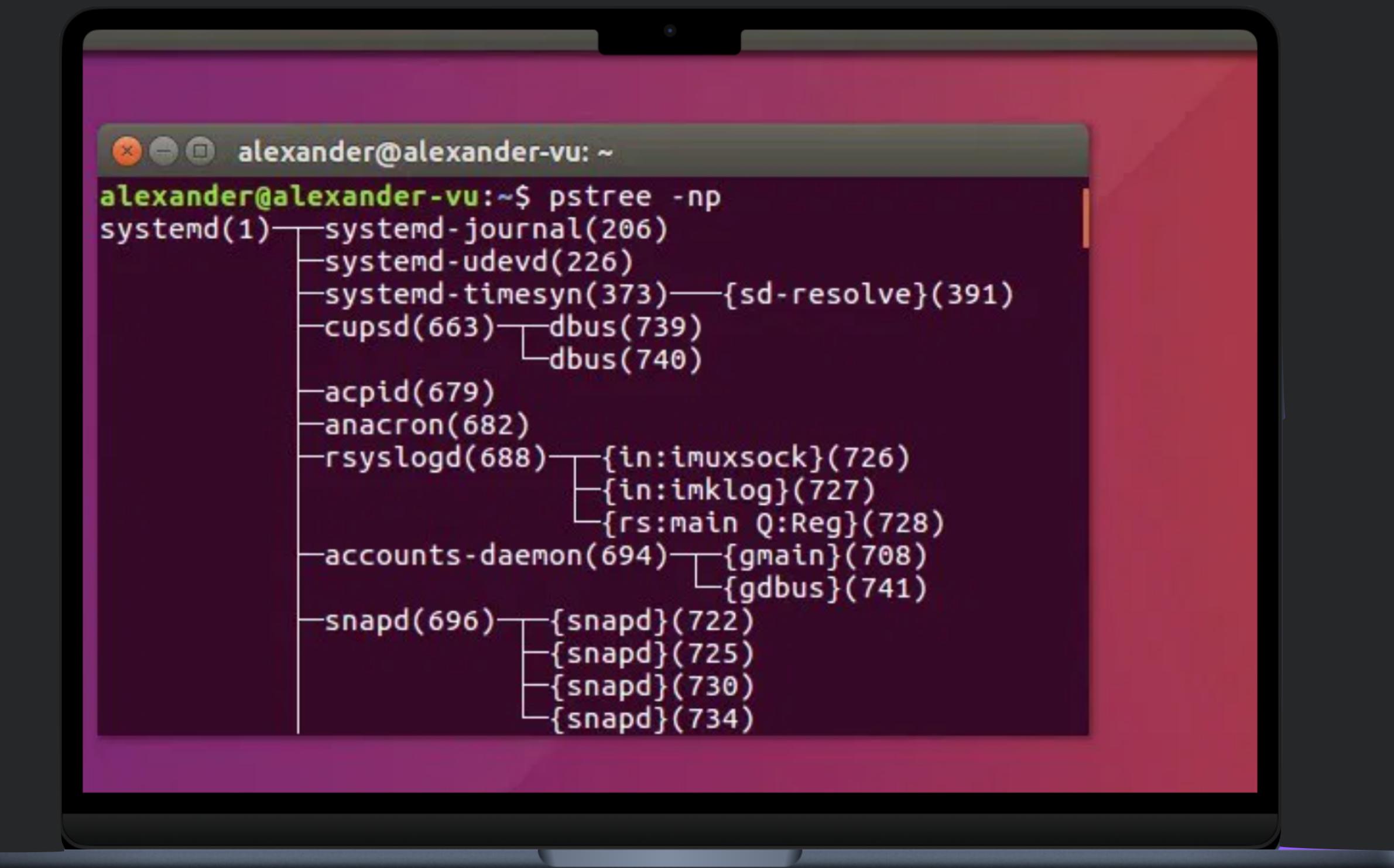
# Kill a process.
kill pid

# Assign a priority to specific process
# at runtime
renice

# before running a process
nice

# get relation between processes.
pstree

# monitoring processes.
top
ps
```



Commands

Interaction
Hardware

DRAM

Init

-20
P1



Introduction to Linux Stacks

File system Stack

In session exercises

Process Management Stack

In session exercises

Stack Definition

Commands

Interaction Hardware

```
# Run a process  
# mode 1:  
# foreground  
.program_name  
# background  
.program_name &  
  
# Kill a process.  
kill pid  
  
# Assign a priority to specific process  
# at runtime  
renice  
  
# before running a process  
nice  
  
# get relation between processes.  
pstree  
  
# monitoring processes.  
top  
ps
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+ COMMAND
1	root	20	0	20752	11240	8308	S	0.0	0.6	0:03.34 systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00 kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00 rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00 rcu_par_gp
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00 kworker/0:0H-events_highpri
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00 mm_percpu_wq
9	root	20	0	0	0	0	S	0.0	0.0	0:00.17 ksoftirqd/0
10	root	20	0	0	0	0	I	0.0	0.0	0:00.35 rcu_sched
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.02 migration/0
12	root	-51	0	0	0	0	S	0.0	0.0	0:00.00 idle_inject/0
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00 cpuhp/0
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00 kdevtmpfs
16	root	0	-20	0	0	0	I	0.0	0.0	0:00.00 netns
17	root	20	0	0	0	0	S	0.0	0.0	0:00.00 rcu_tasks_kthre
18	root	20	0	0	0	0	S	0.0	0.0	0:00.00 kauditd
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00 khungtaskd
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00 oam_reaper
21	root	0	-20	0	0	0	I	0.0	0.0	0:00.00 writeback
22	root	20	0	0	0	0	S	0.0	0.0	0:00.00 ksoftirqd/1

DRAM

Init

-20
P1



Introduction to Linux Stacks

File system Stack

In session exercises

Process Management Stack

In session exercises

Stack Definition

Commands

Interaction Hardware

```
# Run a process  
# mode 1:  
# foreground  
.program_name  
# background  
.program_name &  
  
# Kill a process.  
kill pid  
  
# Assign a priority to specific process  
# at runtime  
renice  
  
# before running a process  
nice  
  
# get relation between processes.  
pstree  
  
# monitoring processes.  
top  
ps
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+ COMMAND
1	root	20	0	20752	11240	8308	S	0.0	0.6	0:03.34 systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00 kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00 rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00 rcu_par_gp
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00 kworker/0:0H-events_highpri
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00 mm_percpu_wq
9	root	20	0	0	0	0	S	0.0	0.0	0:00.17 ksoftirqd/0
10	root	20	0	0	0	0	I	0.0	0.0	0:00.35 rcu_sched
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.02 migration/0
12	root	-51	0	0	0	0	S	0.0	0.0	0:00.00 idle_inject/0
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00 cpuhp/0
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00 kdevtmpfs
16	root	0	-20	0	0	0	I	0.0	0.0	0:00.00 netns
17	root	20	0	0	0	0	S	0.0	0.0	0:00.00 rcu_tasks_kthre
18	root	20	0	0	0	0	S	0.0	0.0	0:00.00 kauditd
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00 khungtaskd
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00 oam_reaper
21	root	0	-20	0	0	0	I	0.0	0.0	0:00.00 writeback
22	root	20	0	0	0	0	S	0.0	0.0	0:00.00 ksoftirqd/1

DRAM

Init

-20
P1



Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises

CPU

Stack
Definition

Commands

Interaction
Hardware

Introduction to Linux
Stacks

File system Stack

In session exercises

Process Management
Stack

In session exercises



Assignment one for kernel module

Check how many cores do you have using top command.

Create number of cores + 2 processes dd if=/dev/zero
of=/dev/null run in background.

Change priority for them:

-20, -10, 0, .. , 19

Monitor them using top command, did you notice any change ?

Kill them all using killall command.