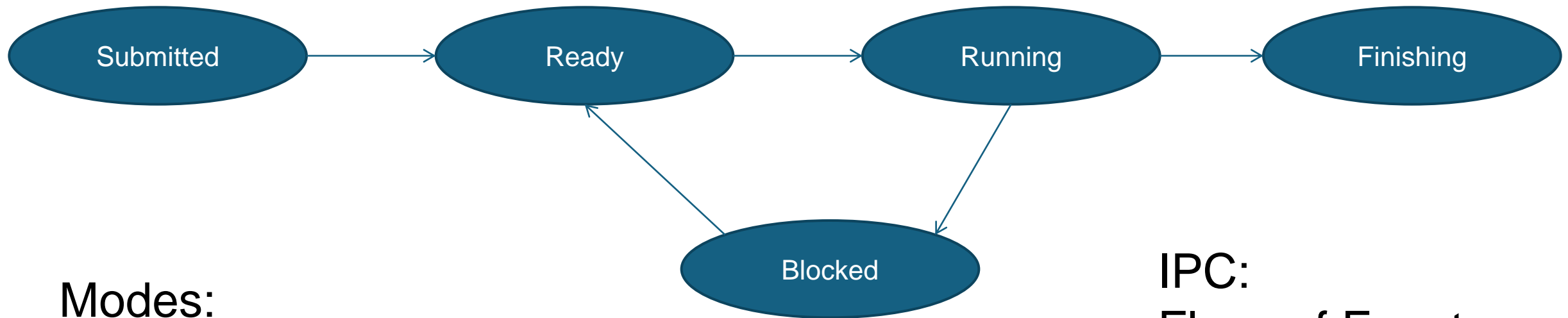


Operating Systems

Dmitry Zaitsev

Lecture 4: Scheduling of processes. Case study: Linux CLI, working with processes.

Process state diagram – the key to scheduling of processes



Modes:
Multiprogramming
Time-slicing
Priorities

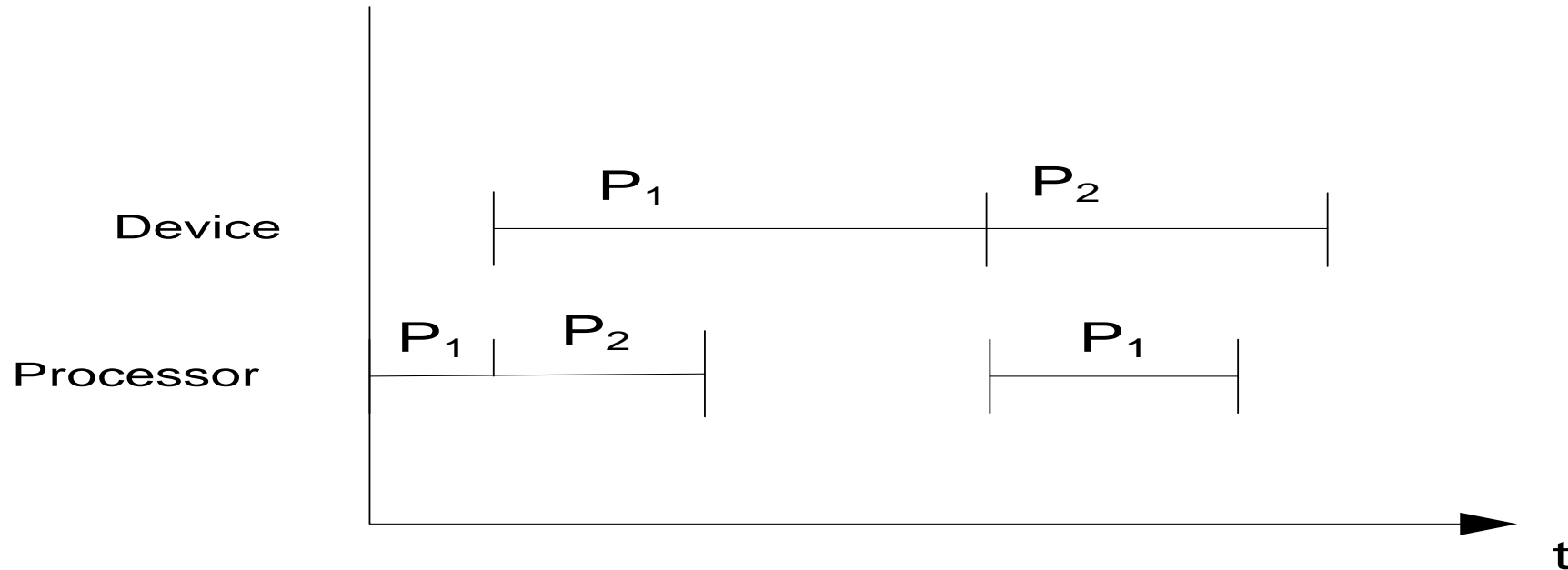
IPC:
Flags of Events
Semaphores
Messages
Shared memory

Process scheduling modes

- Single program mode
- Multiprogramming
- Batch mode and hardware maximal utilization criteria
- Interactive mode and time-slicing
- Priorities of processes
- Absolute and relative priorities
- Static and dynamic priorities
- Optimization of job mixture for batch mode

Multiprogramming

Simultaneous (parallel) work of devices with the goal to increase their utilization

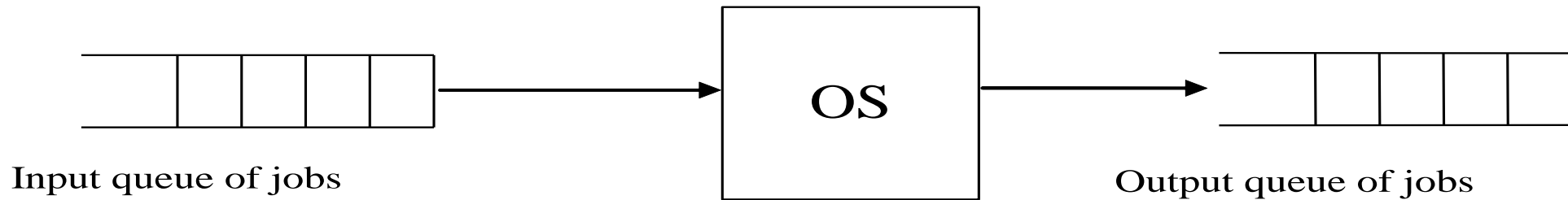


Batch mode

Criterion – maximal utilization of devices

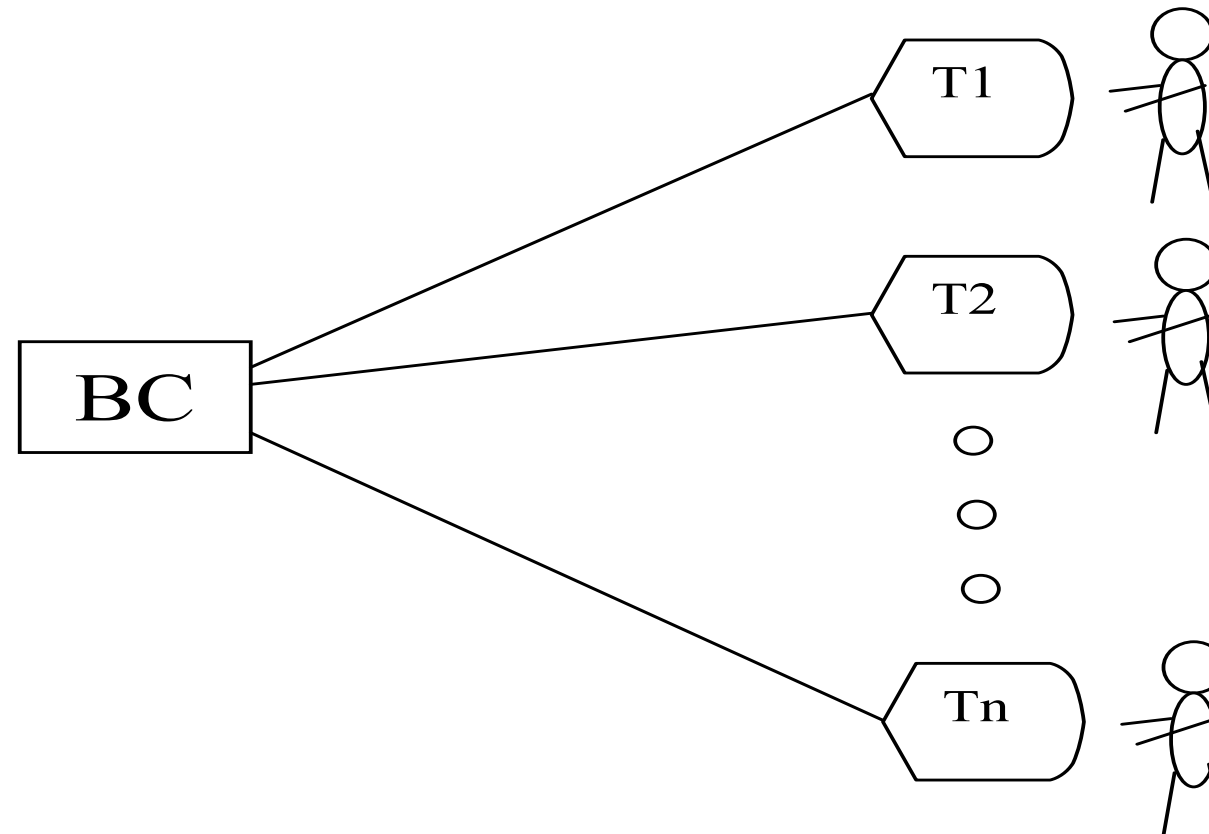
Mixture of jobs

Off-line mode



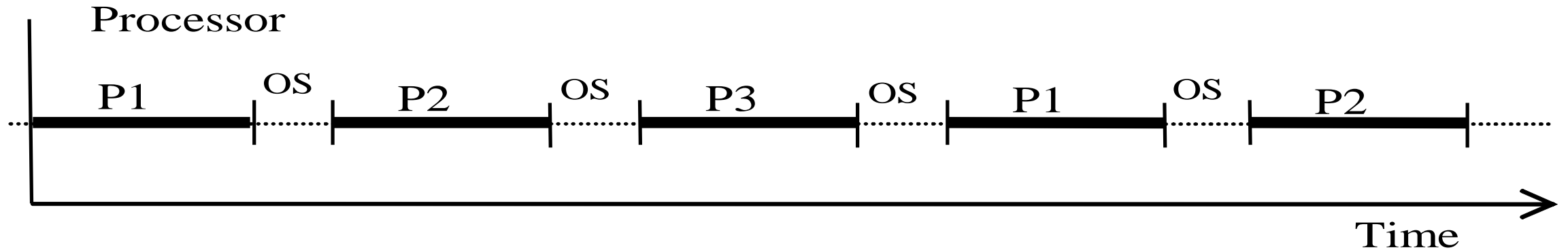
Interactive mode

Criterion – minimal (agreeable) reply time

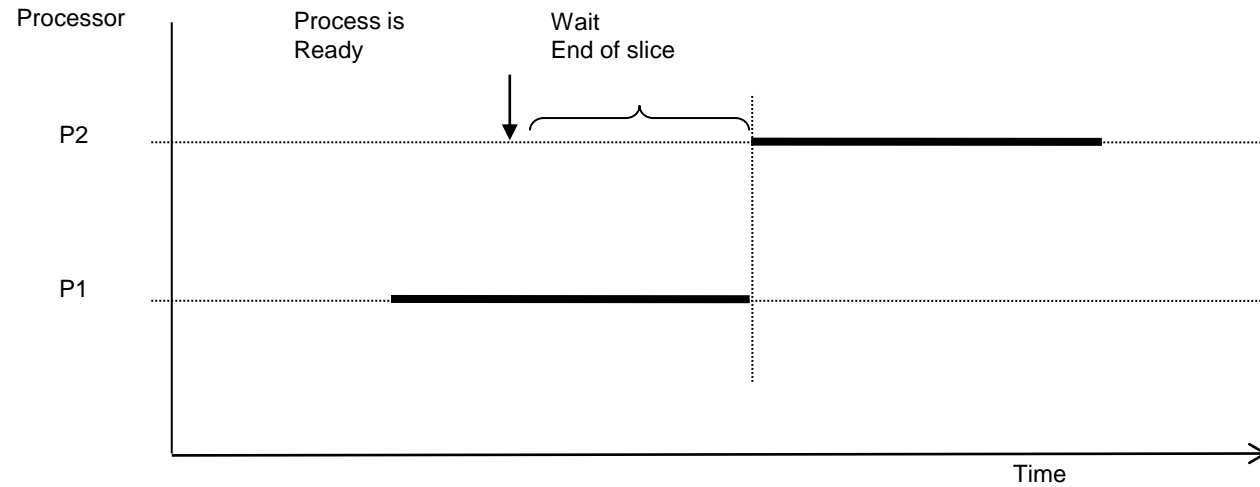


Time slicing

A process occupies processor not longer than a slice of time
Switching of processes via timer interrupt

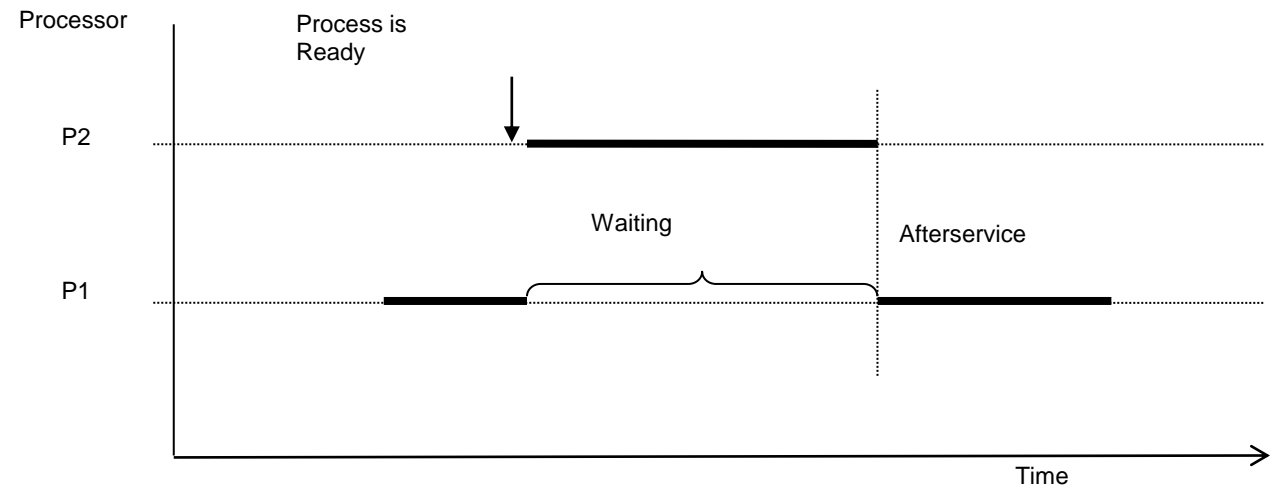


Priorities of processes



Relative

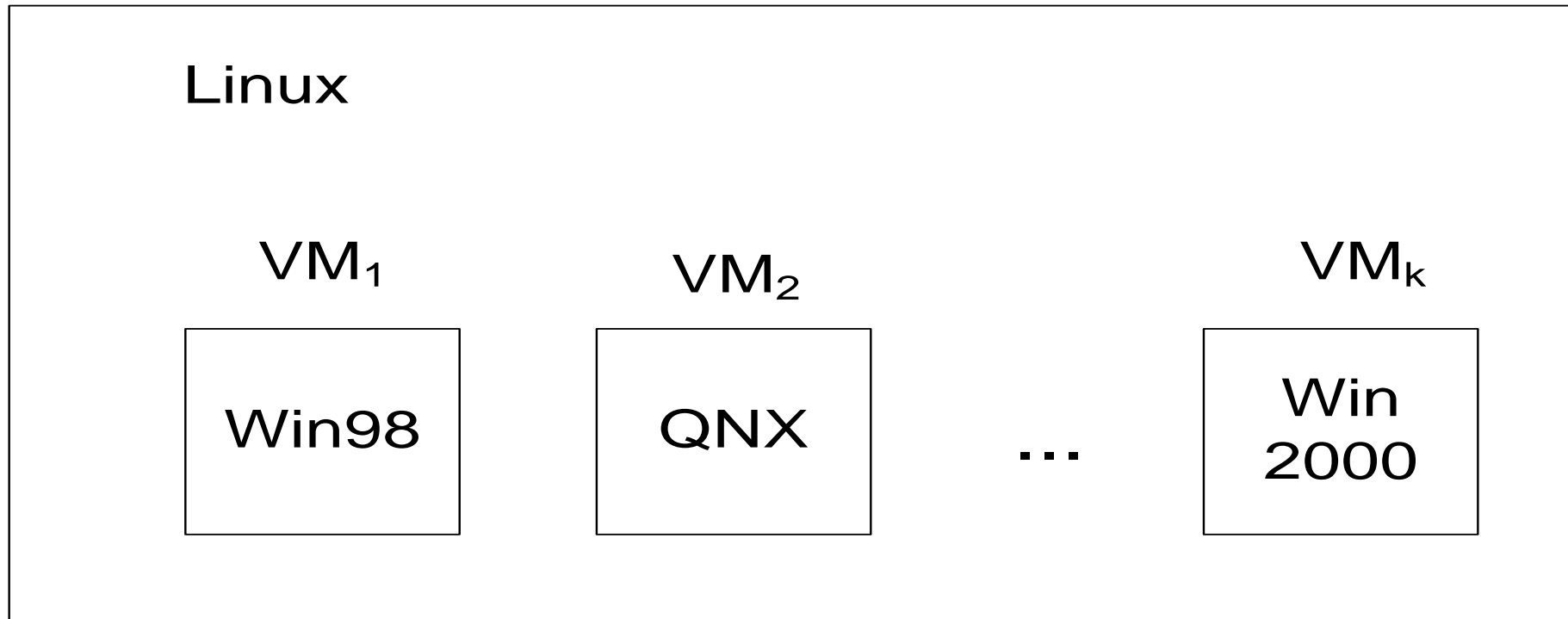
Absolute



Virtual machines

Emulate hardware

Install OS on emulated hardware

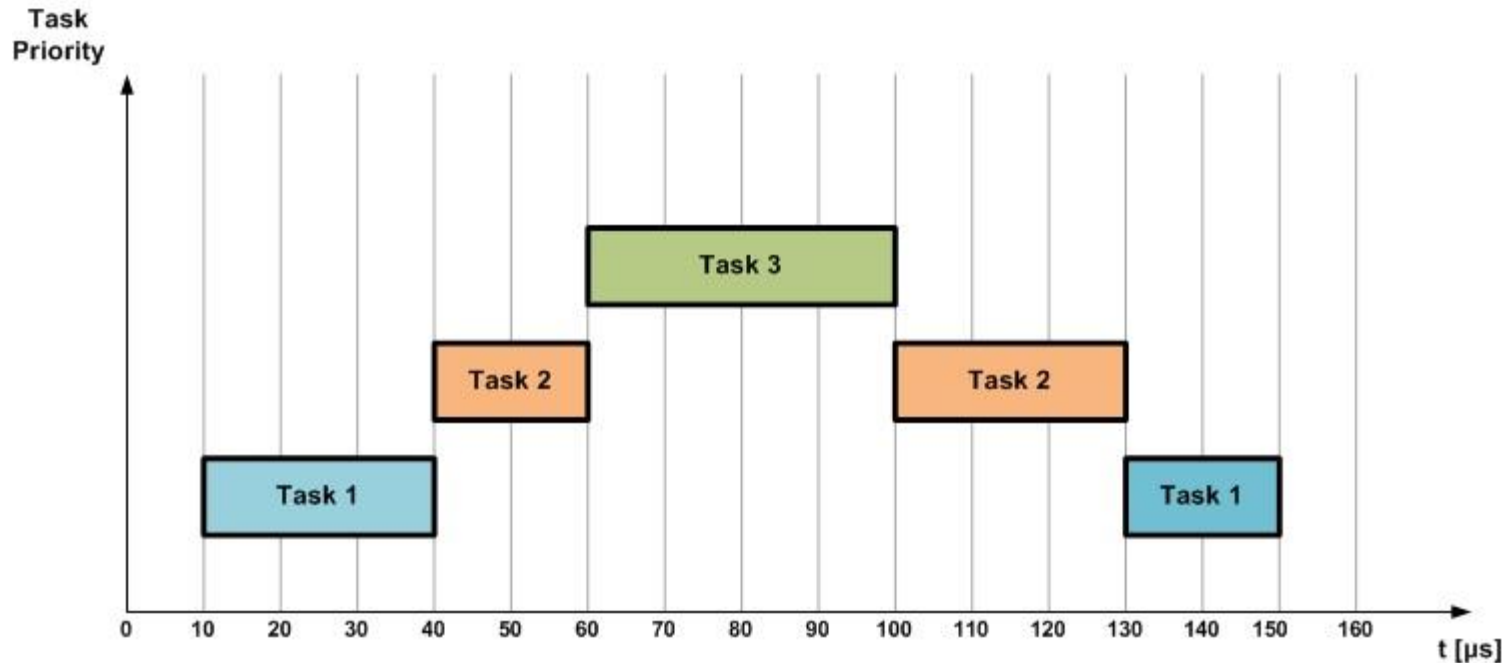
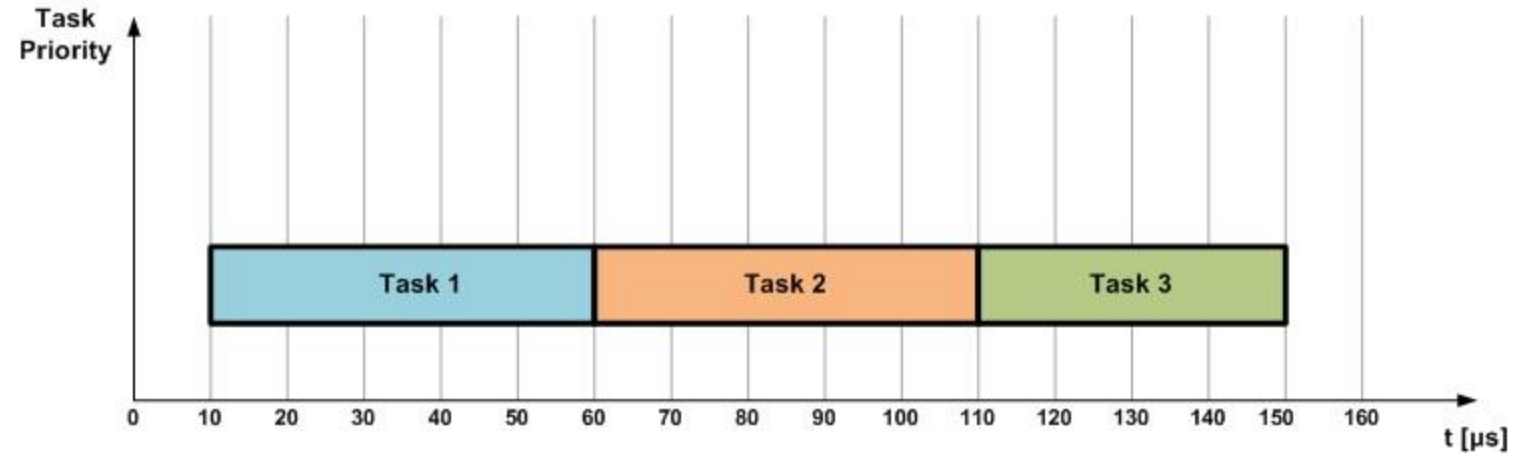


Scheduling of processes

- **Scheduling** – making decisions on which process should be executed at any point in time based on a predefined algorithm.
- Non-preemptive Scheduling (Co-Operative Scheduling) – start a process and then wait for the process to return control (finish)
- Preemptive Scheduling – allows the interruption of a currently running process by a higher priority process with later after-service
- Scheduling Algorithms: First Come, First Served (FCFS); Shortest Job First (SJF); Priority Scheduling; Round-Robin Scheduling (Time-slicing)
- Preemptive scheduling option: Shortest Job First (SJF); Priority Scheduling; Round-Robin Scheduling

Scheduling: Co-operative vs Preemptive

Co-operative
scheduling



Preemptive
scheduling

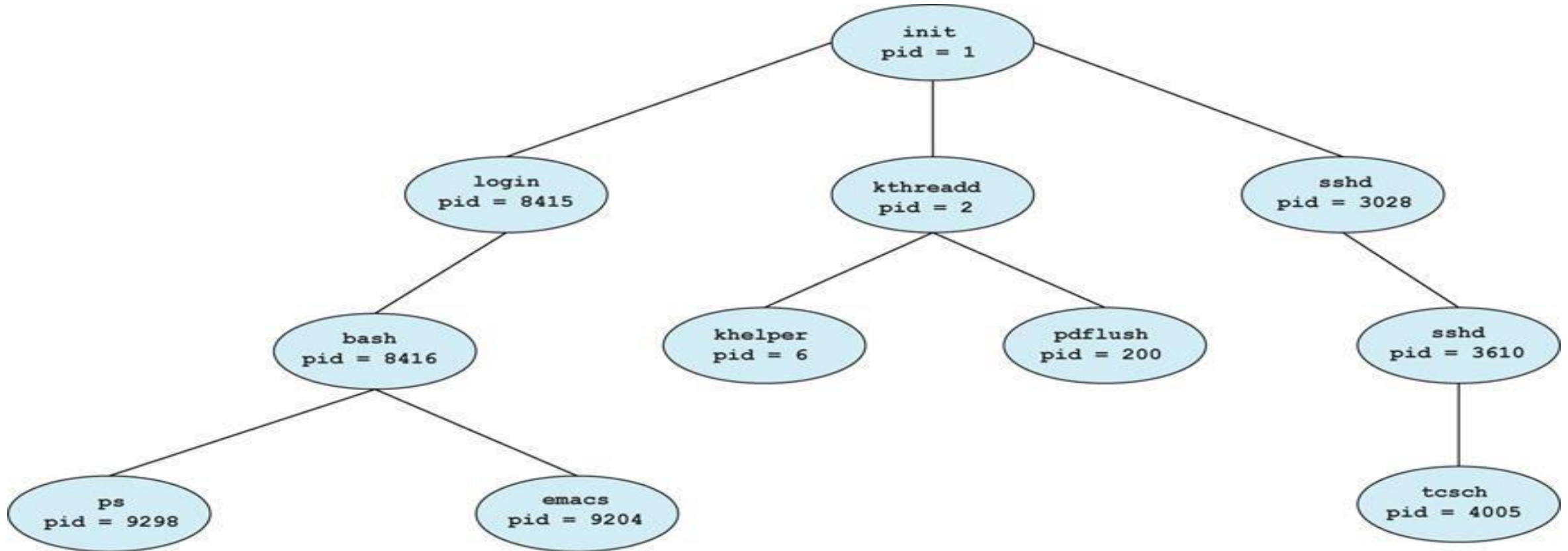
Inter-process communication

- Pipes – a unidirectional communication facility (FIFO)
- Semaphores – an integer number (amount of available resources) and a queue of waiting processes
- Messages – a post box for information exchange
- Common segments of memory – a segment of memory attached to virtual memory of a few processes
- Sockets – a post box for global information exchange (TCP/IP)

Complex system implementation

- Utilize mass-parallel facilities of modern architecture with clusters of nodes, multi-core processors, and GPUs/NPUs
- Implement a system as a set of communicating processes and threads within a process
- Use Inter-Process Communication (IPC) facilities of OS for information exchange between processes
- Higher reliability – certain independence of processes
- Scalability – run on a single computer or on a cluster (communicating via TCP/IP sockets)
- Further specialization with OpenMP, MPI, OpenCL/CUDA, etc.

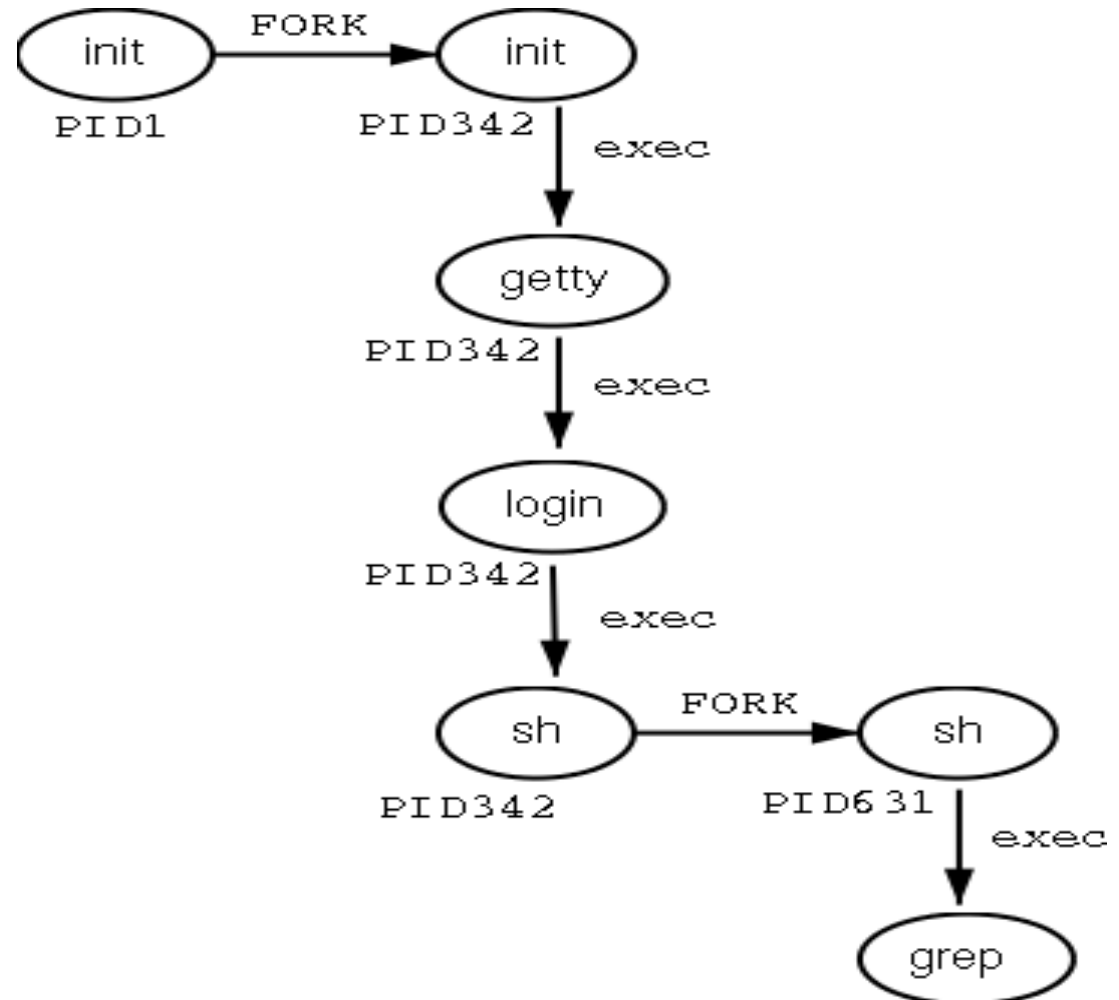
Processes tree in Linux



Fork & Exec

Fork – duplicate (clone)
the process

Exec – run
Executable file



Attributes of a process

- Flags: 1, 4
- State: D,R,S, T, X, Z
- User identifier (UID)
- Process identifier (PID)
- Parent process identifier (PPID)
- Priority (PRI, NI)
- Terminal (TTY)
- Command (CMD)


```

4 S      0    1705      1  0  80    0 - 13284 poll_s ?      00:00:00 udiskd
0 S    1000    1732    1173  0  80    0 -   7383 poll_s ?      00:00:00 gvfs-gphoto2-v
0 S    1000    1745    1173  0  80    0 -   7105 poll_s ?      00:00:00 gvfs-mtp-volum
0 S    1000    1752    1173  0  80    0 -   2707 poll_s ?      00:00:00 gconfd-2
0 S    1000    1754    1173  0  80    0 -   9892 poll_s ?      00:00:00 gvfs-afc-volum
0 S    1000    1783    1173  0  80    0 -  16075 poll_s ?      00:00:00 gvfsd-trash
0 S    1000    1790    1173  0  80    0 -   9294 poll_s ?      00:00:00 gvfsd-burn
0 S    1000    1808    1343  0  80    0 -  20914 poll_s ?      00:00:00 telepathy-indi
0 S    1000    1816    1173  0  80    0 -  11147 poll_s ?      00:00:00 mission-contro
0 S    1000    1829    1343  0  80    0 -  15116 poll_s ?      00:00:00 zeitgeist-data
0 S    1000    1834    1173  0  80    0 -  10590 poll_s ?      00:00:00 zeitgeist-daem
0 S    1000    1840    1173  0  80    0 -  14015 poll_s ?      00:00:00 zeitgeist-fts
0 S    1000    1849    1840  0  80    0 -   1369 unix_s ?      00:00:00 cat
0 S    1000    1882    1343  0  80    0 -  18059 poll_s ?      00:00:00 update-notifie
0 S    1000    1905    1343  0  80    0 -  16293 poll_s ?      00:00:00 deja-dup-monit
1 S      0    2006      2  0  80    0 -      0 worker ?      00:00:00 kworker/u2:0
1 S      0    2047      2  0  80    0 -      0 worker ?      00:00:00 kworker/u2:2
4 S      0    2049      1  0  80    0 -   1114 wait   tty1      00:00:00 login
1 S      0    2124      2  0  80    0 -      0 worker ?      00:00:00 kworker/0:0
4 S    1000    2125    2049  0  80    0 -   2048 wait   tty1      00:00:00 bash
1 S      0    2175      2  0  80    0 -      0 worker ?      00:00:00 kworker/u2:1
0 R    1000    2176    2125  0  80    0 -   1554 -     tty1      00:00:00 ps

```

```
daze@daze-fast:~$
```

```
daze@daze-fast:~$
```

```
daze@daze-fast:~$
```

```
daze@daze-fast:~$ ps -l
```

```

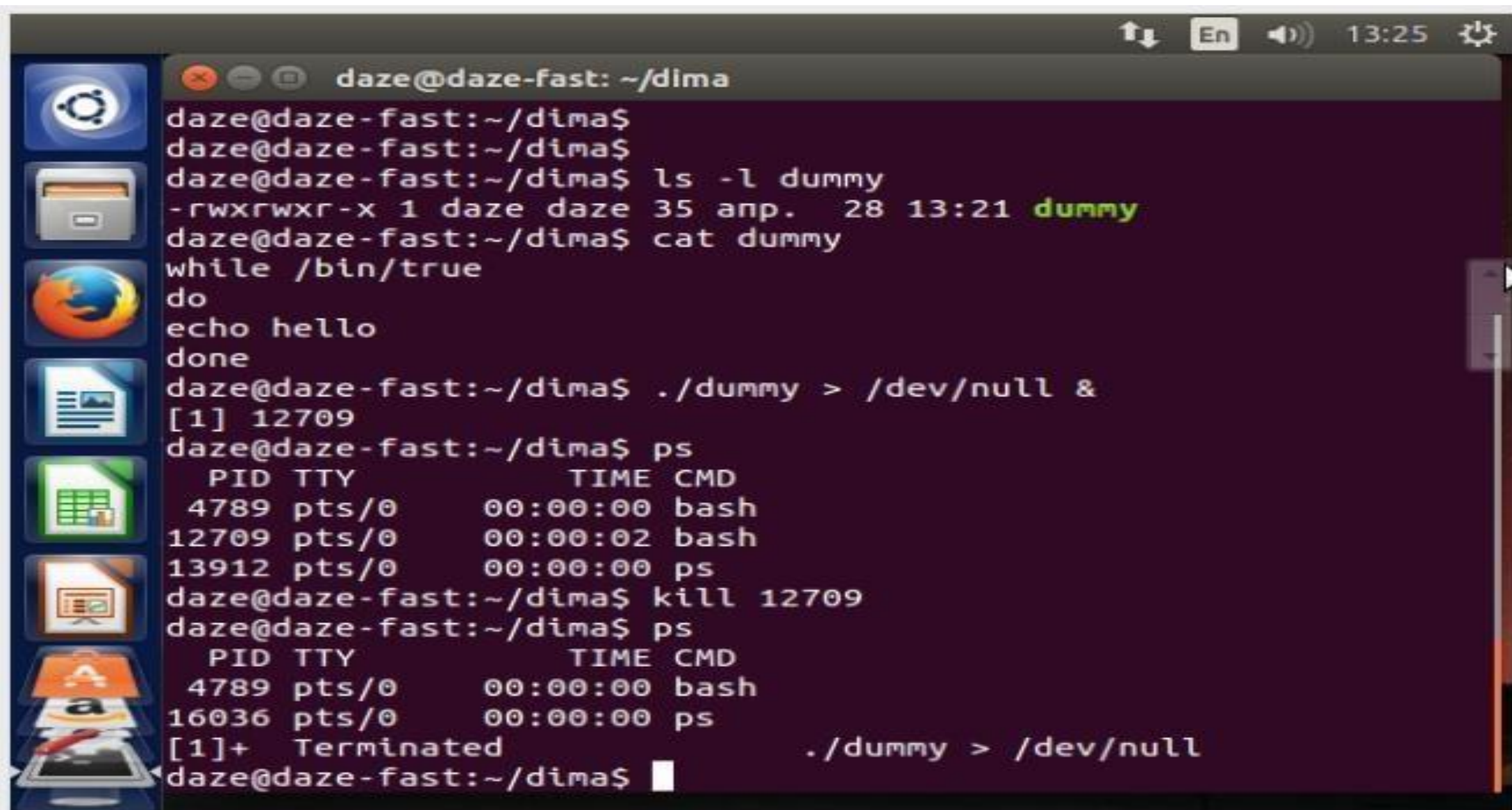
F S      UID      PID  PPID   C  PRI   NI  ADDR  SZ  WCHAN    TTY          TIME CMD
4 S    1000    2125    2049  0  80    0  -   2048 wait   tty1      00:00:00 bash
0 R    1000    2177    2125  0  80    0  -   1554 -     tty1      00:00:00 ps

```

```
daze@daze-fast:~$
```

Basic commands

- file-name – start a process
 - file-name & – start a process in background
 - ps – processes list
 - kill [signal] PID – send a signal
-
- ps -e all processes long
 - ps -l output
 - kill -9 PID compulsory termination



A terminal window titled "daze@daze-fast: ~/dima" is shown. The window has a dark purple background and a light gray title bar. On the left side, there is a vertical dock with icons for a terminal, a file manager, a web browser, a document, a spreadsheet, a presentation, and a folder. The terminal content shows the user "daze" at the prompt "daze@daze-fast: ~/dima\$". The user runs "ls -l dummy", which outputs a file named "dummy" with permissions "-rwxrwxr-x", owner "daze", group "daze", size "35", and timestamp "anp. 28 13:21". The user then runs "cat dummy", which outputs "while /bin/true". The user then runs "do", "echo hello", and "done". The user then runs "./dummy > /dev/null &", which outputs "[1] 12709". The user then runs "ps", which outputs a table of processes. The user then runs "kill 12709", which outputs "daze@daze-fast: ~/dima\$ ps". The user then runs "ps", which outputs a table of processes. The user then runs "[1]+ Terminated", which outputs "daze@daze-fast: ~/dima\$". The user then runs "./dummy > /dev/null", which outputs "daze@daze-fast: ~/dima\$".

```
daze@daze-fast: ~/dima
daze@daze-fast:~/dima$
daze@daze-fast:~/dima$ ls -l dummy
-rwxrwxr-x 1 daze daze 35 anp. 28 13:21 dummy
daze@daze-fast:~/dima$ cat dummy
while /bin/true
do
echo hello
done
daze@daze-fast:~/dima$ ./dummy > /dev/null &
[1] 12709
daze@daze-fast:~/dima$ ps
  PID TTY          TIME CMD
 4789 pts/0        00:00:00 bash
12709 pts/0        00:00:02 bash
13912 pts/0        00:00:00 ps
daze@daze-fast:~/dima$ kill 12709
daze@daze-fast:~/dima$ ps
  PID TTY          TIME CMD
 4789 pts/0        00:00:00 bash
16036 pts/0        00:00:00 ps
[1]+  Terminated                  ./dummy > /dev/null
daze@daze-fast:~/dima$
```

Priorities

- nice [*Option*]... [*Command* [*Arg*]...]
- sudo nice -n 10 echo Hello
- renice Priority -p PID
- renice 16 -p 13245

No Hang Up

- nohup command arguments
- Output: \$HOME/nohup.out
- nohup ./dummy > dummy-work &
- exit
- login
- ps

```

daze@daze-fast:~/dima$
daze@daze-fast:~/dima$
daze@daze-fast:~/dima$
daze@daze-fast:~/dima$
daze@daze-fast:~/dima$
daze@daze-fast:~/dima$
daze@daze-fast:~/dima$
daze@daze-fast:~/dima$
daze@daze-fast:~/dima$ ps -lU daze
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY  TIME CMD
1 S  1000  1459    1   0  69 -11 - 25357 poll_s ?    00:00:00 pulseaudio
0 R  1000  2975    1  21  80  0 - 568 - ?    00:00:53 sh
0 R  1000  5969  6461   0  80  0 - 1613 - tty1 00:00:00 ps
4 S  1000  6461  4433   0  80  0 - 2048 wait  tty1 00:00:00 bash
daze@daze-fast:~/dima$
daze@daze-fast:~/dima$ ls -l dummy-work
-rw-rw-r-- 1 daze daze 409110 anp. 28 15:46 dummy-work
daze@daze-fast:~/dima$
daze@daze-fast:~/dima$ ps -lU daze
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY  TIME CMD
1 S  1000  1459    1   0  69 -11 - 25357 poll_s ?    00:00:00 pulseaudio
0 R  1000  2975    1  21  80  0 - 568 - ?    00:00:56 sh
4 S  1000  6461  4433   0  80  0 - 2048 wait  tty1 00:00:00 bash
0 R  1000  9844  6461   0  80  0 - 1613 - tty1 00:00:00 ps
daze@daze-fast:~/dima$
daze@daze-fast:~/dima$ ls -l dummy-work
-rw-rw-r-- 1 daze daze 437646 anp. 28 15:46 dummy-work
daze@daze-fast:~/dima$

```

Start on a schedule

- `at -f file TIME`
 - `at -f file -t time-arg`
 - `atq`
 - `batch`
-
- `at -f ./eee 16:00`

Background and foreground jobs

- A process that connects to the terminal is called a foreground job
- A process that disconnects from the terminal and cannot communicate with the user is called a background job.
- Commands: *fg*, *bg*

