

Advanced Programming in the UNIX Environment

Week 13, Segment 4: Process Priorities

**Department of Computer Science
Stevens Institute of Technology**

Jan Schaumann

`jschauma@stevens.edu`

`https://stevens.netmeister.org/631/`

Process Priorities

All processes (including those in a jail) compete for the same resources: CPU cycles, memory, disk space, ...

Recall `getrlimit(2)/setrlimit(2)` from Week 06, Segment 5.

```
$ ulimit -a
time          (-t seconds   ) unlimited
file          (-f blocks    ) unlimited
data          (-d kbytes    ) 262144
stack         (-s kbytes    ) 4096
coredump      (-c blocks    ) unlimited
memory        (-m kbytes    ) 992196
locked memory (-l kbytes    ) 330732
thread        (-r threads   ) 1024
process       (-p processes  ) 1024
nofiles       (-n descriptors) 1024
vmemory       (-v kbytes    ) unlimited
sbsize        (-b bytes     ) unlimited
$
```

Process Priorities

All processes (including those in a jail) compete for the same resources: CPU cycles, memory, disk space, ...

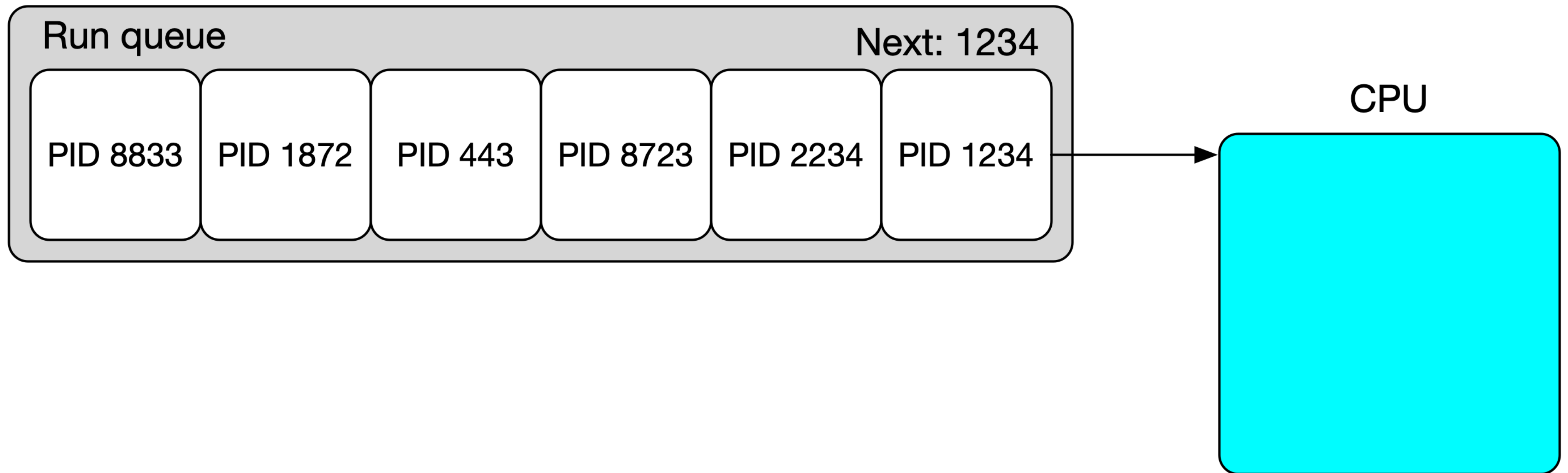
Recall `getrlimit(2)/setrlimit(2)` from Week 06, Segment 5.

```
$ ulimit -a
```

time	(-t seconds) unlimited
file	(-f blocks) unlimited
data	(-d kbytes) 262144
stack	(-s kbytes) 4096
coredump	(-c blocks) unlimited
memory	(-m kbytes) 992196
locked memory	(-l kbytes) 330732
thread	(-r threads) 1024
process	(-p processes) 1024
nofiles	(-n descriptors)	1024
vmemory	(-v kbytes) unlimited
sbsize	(-b bytes) unlimited

```
$
```

Priority Scheduling



Priority Scheduling

Run queue

Next: 2234

PID 8833

PID 1872

PID 443

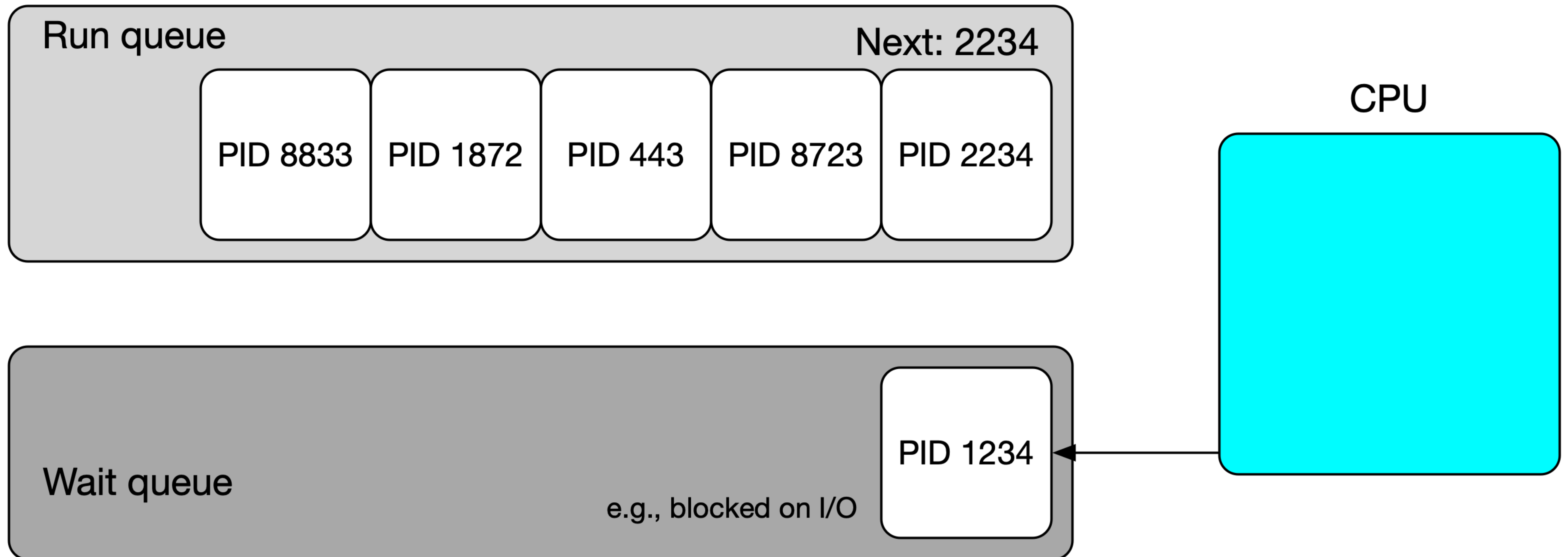
PID 8723

PID 2234

CPU

PID 1234

Priority Scheduling



Priority Scheduling

Run queue

Next: 8723

PID 8833

PID 1872

PID 443

PID 8723

Wait queue

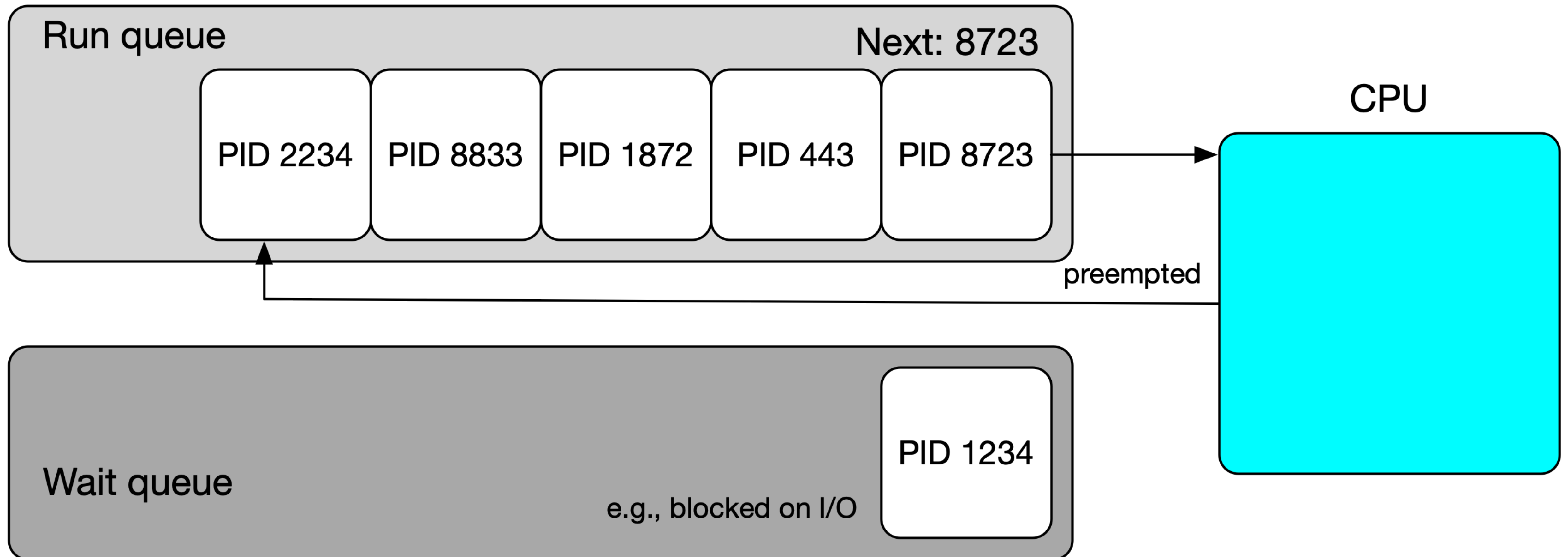
e.g., blocked on I/O

PID 1234

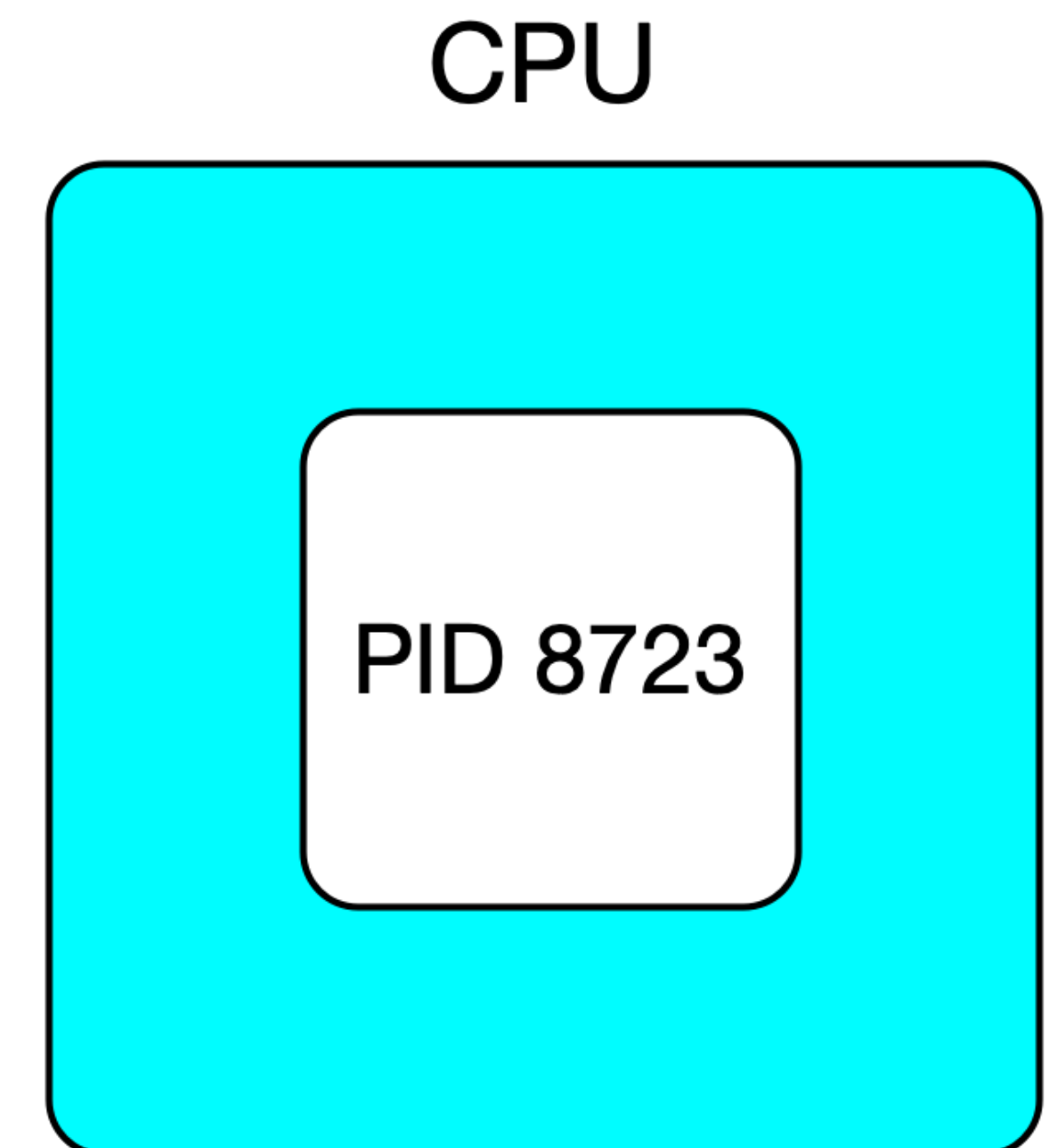
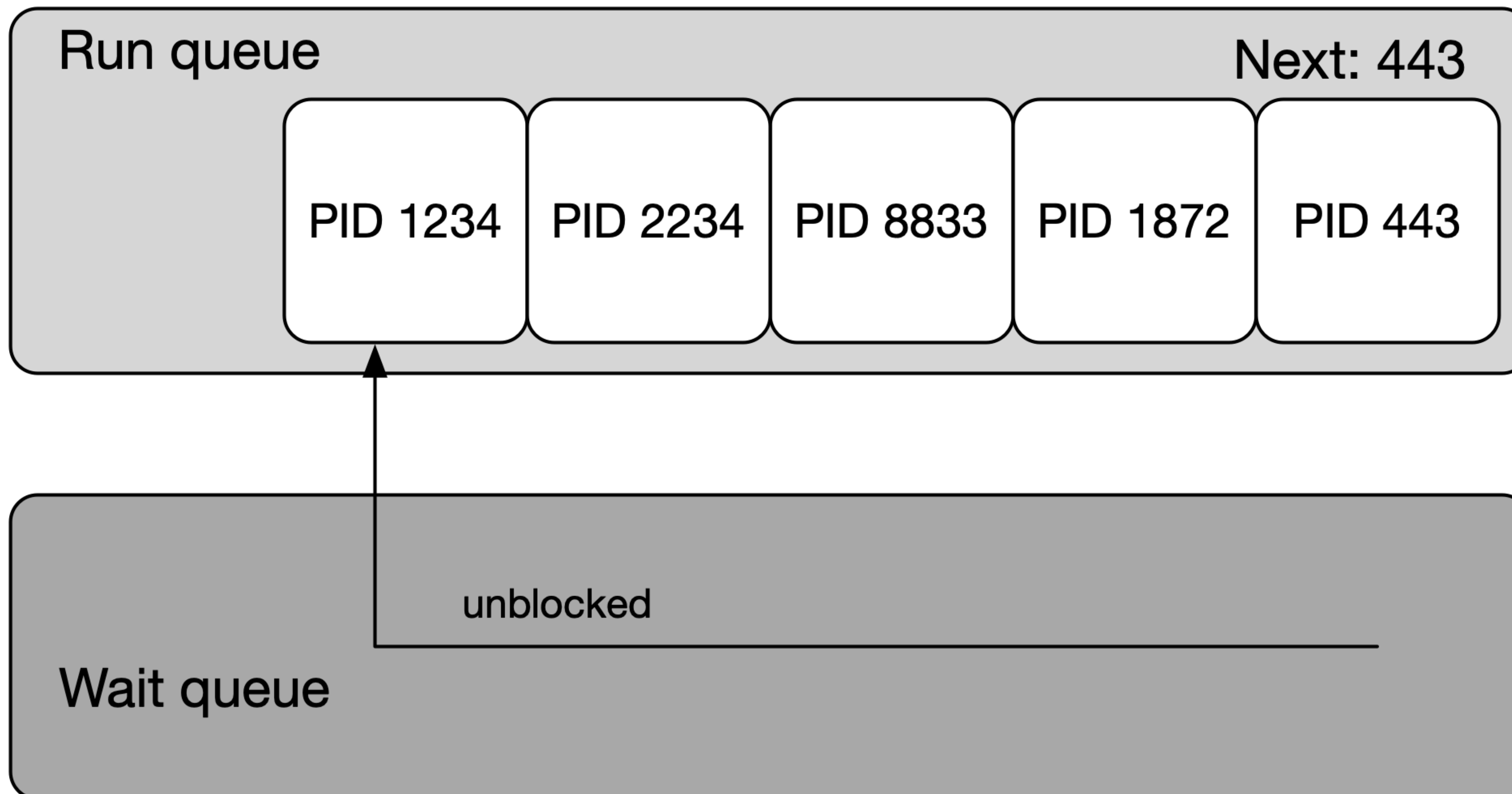
CPU

PID 2234

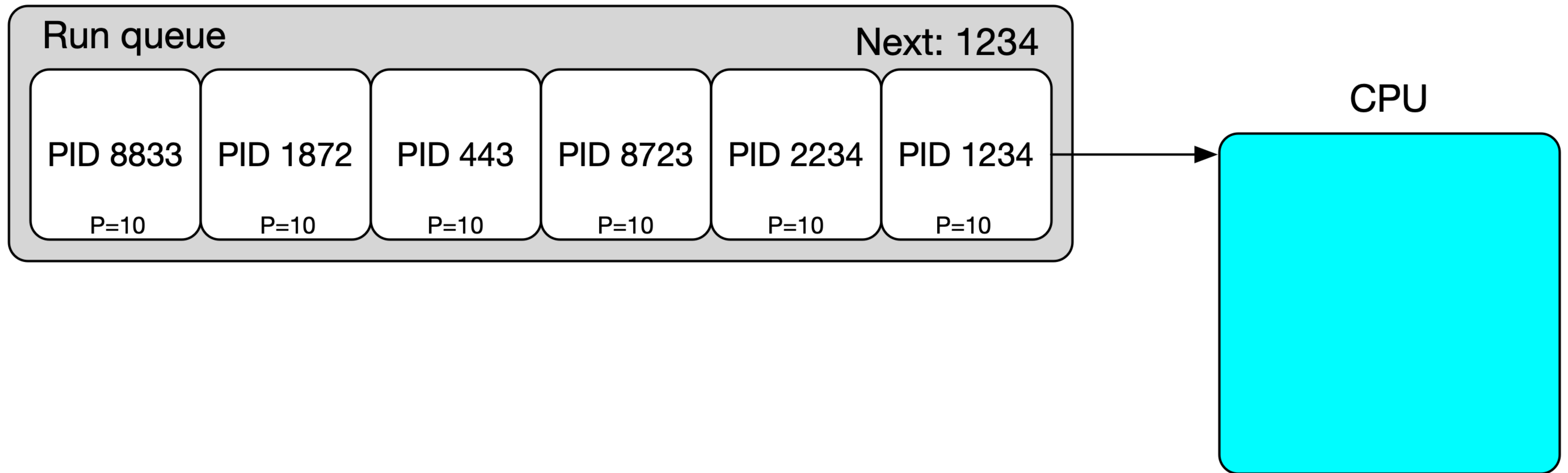
Priority Scheduling



Priority Scheduling



Priority Scheduling



Priority Scheduling

Run queue

Next: 8723

PID 8833

P=11

PID 1872

P=10

PID 443

P=7

PID 8723

P=15

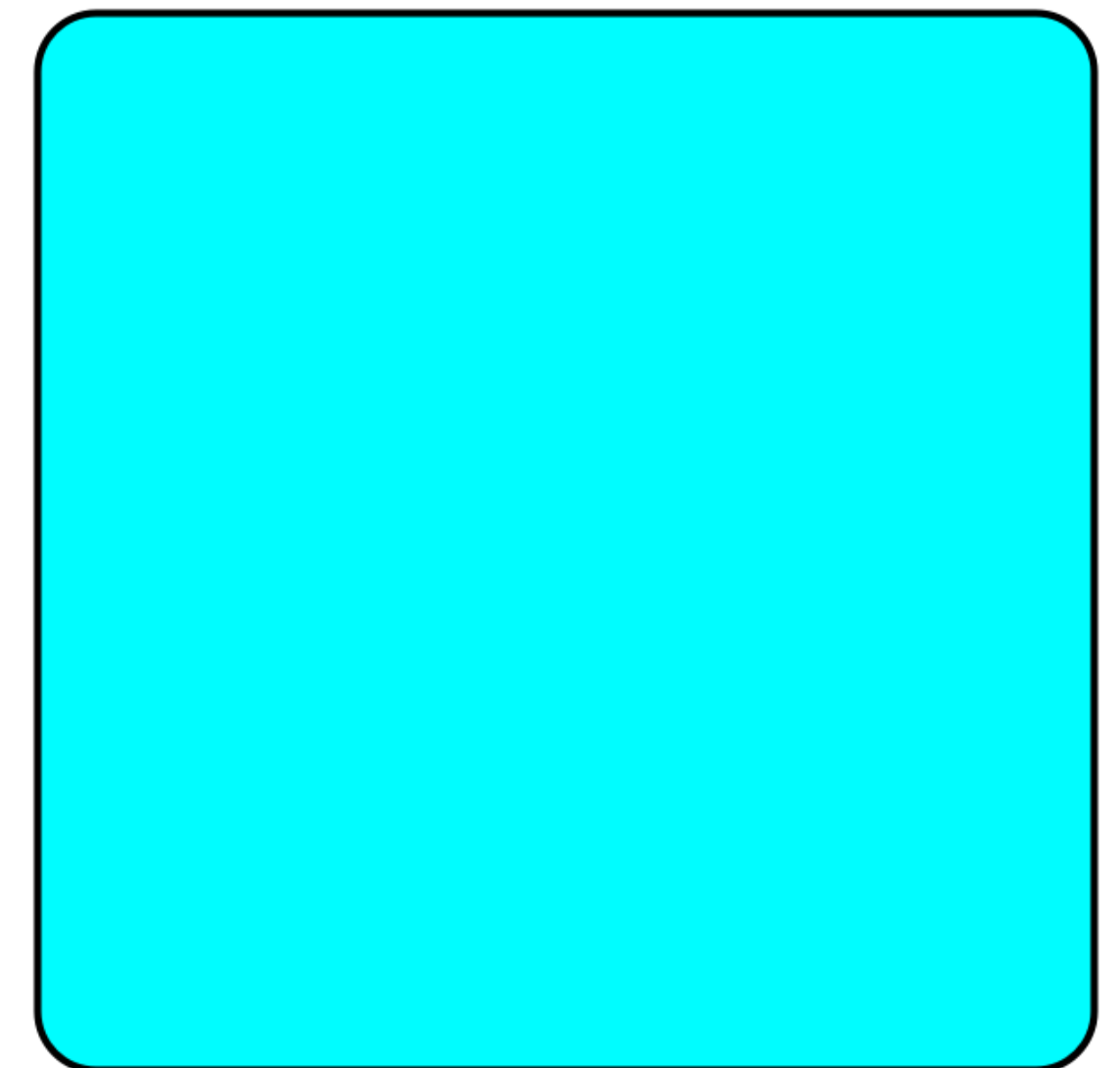
PID 2234

P=5

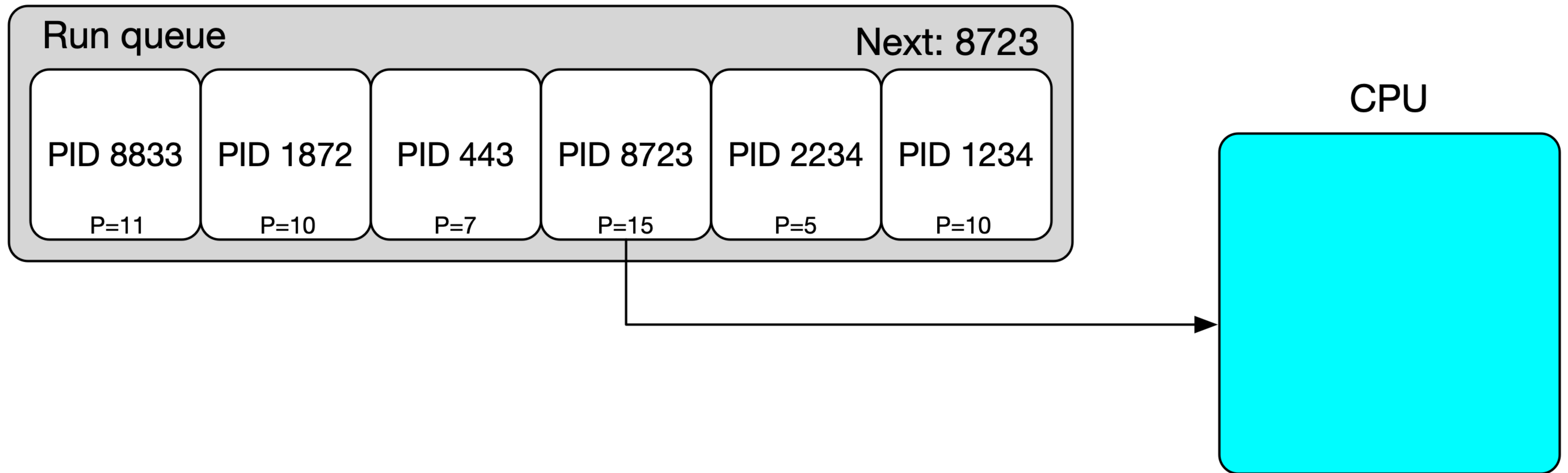
PID 1234

P=10

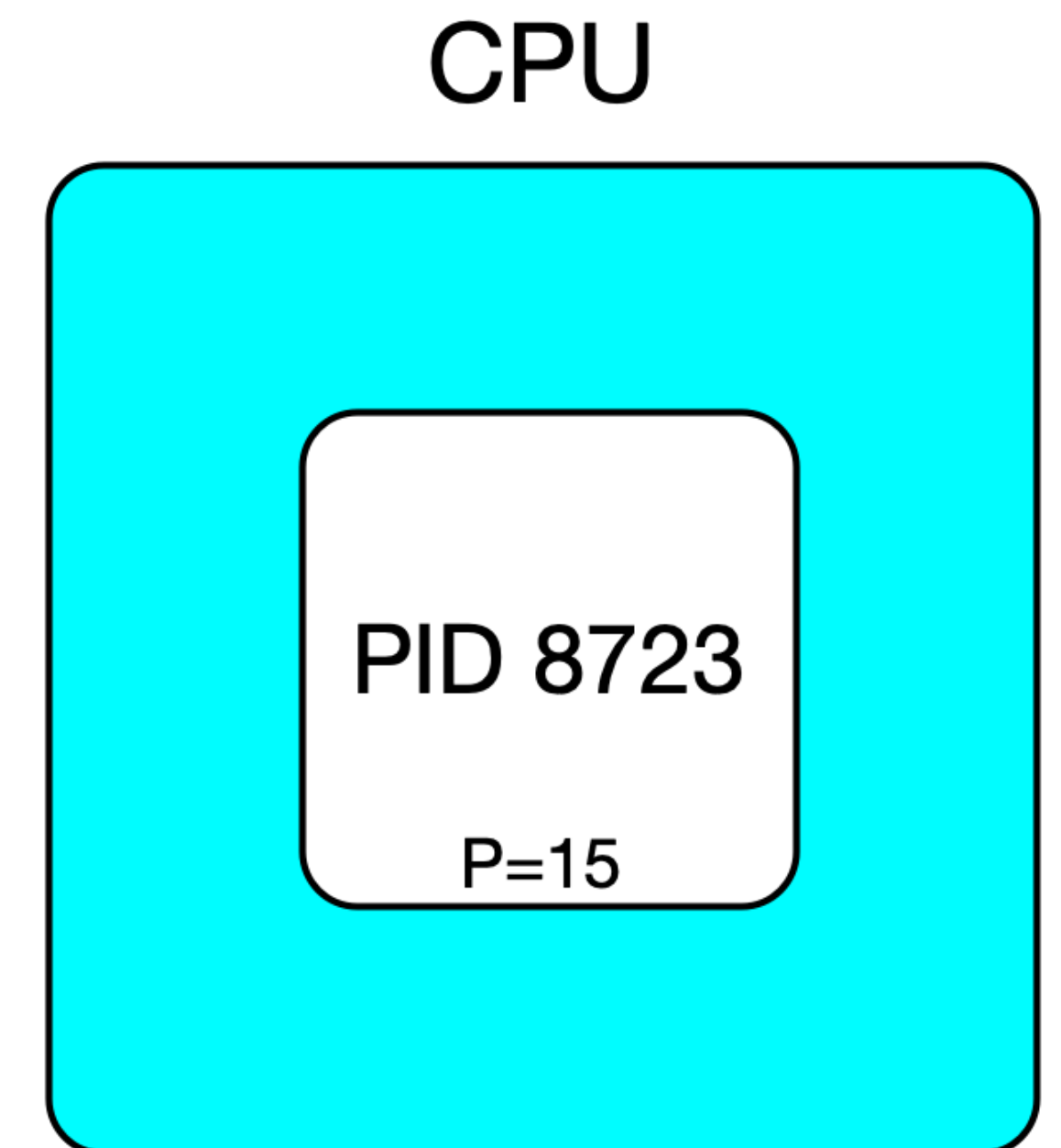
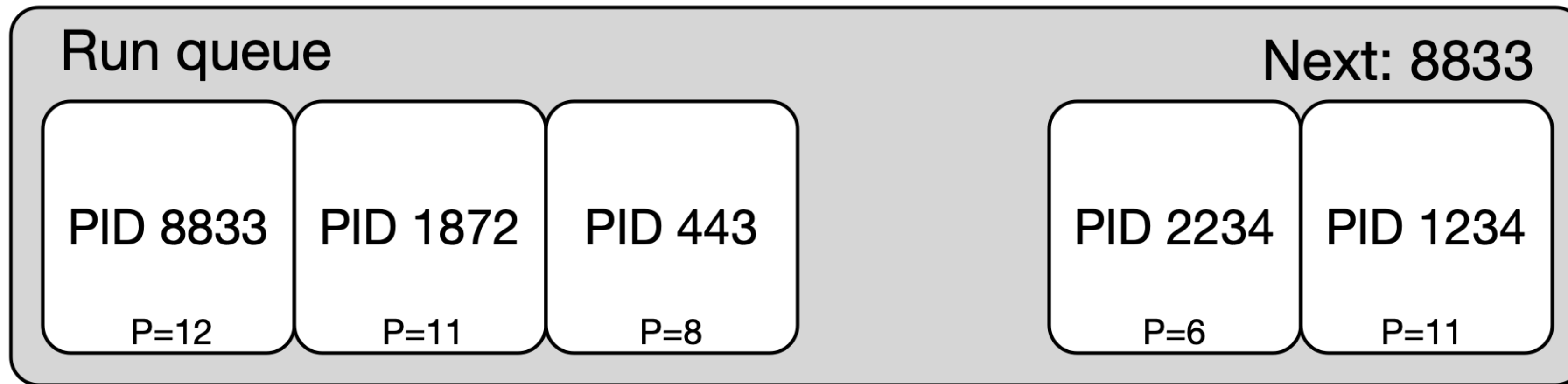
CPU



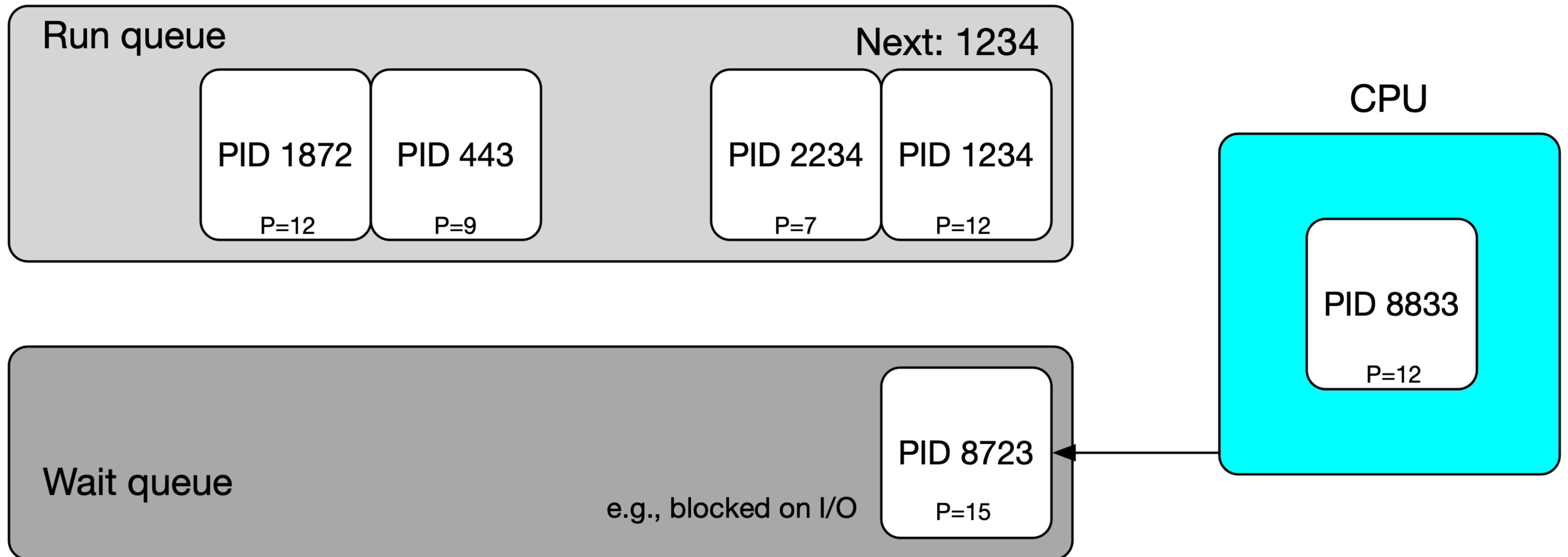
Priority Scheduling



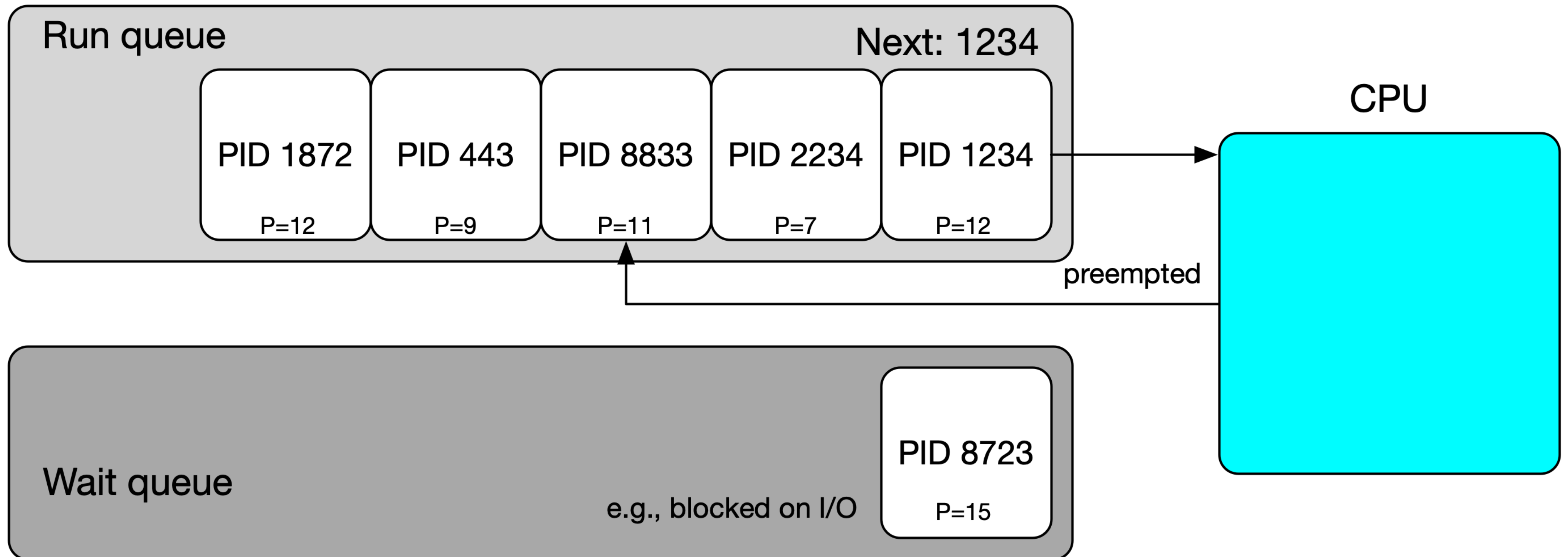
Priority Scheduling



Priority Scheduling



Priority Scheduling



Priority Scheduling

Run queue

Next: 1872

PID 1872

P=13

PID 443

P=10

PID 8833

P=12

PID 2234

P=8

Wait queue

e.g., blocked on I/O

PID 8723

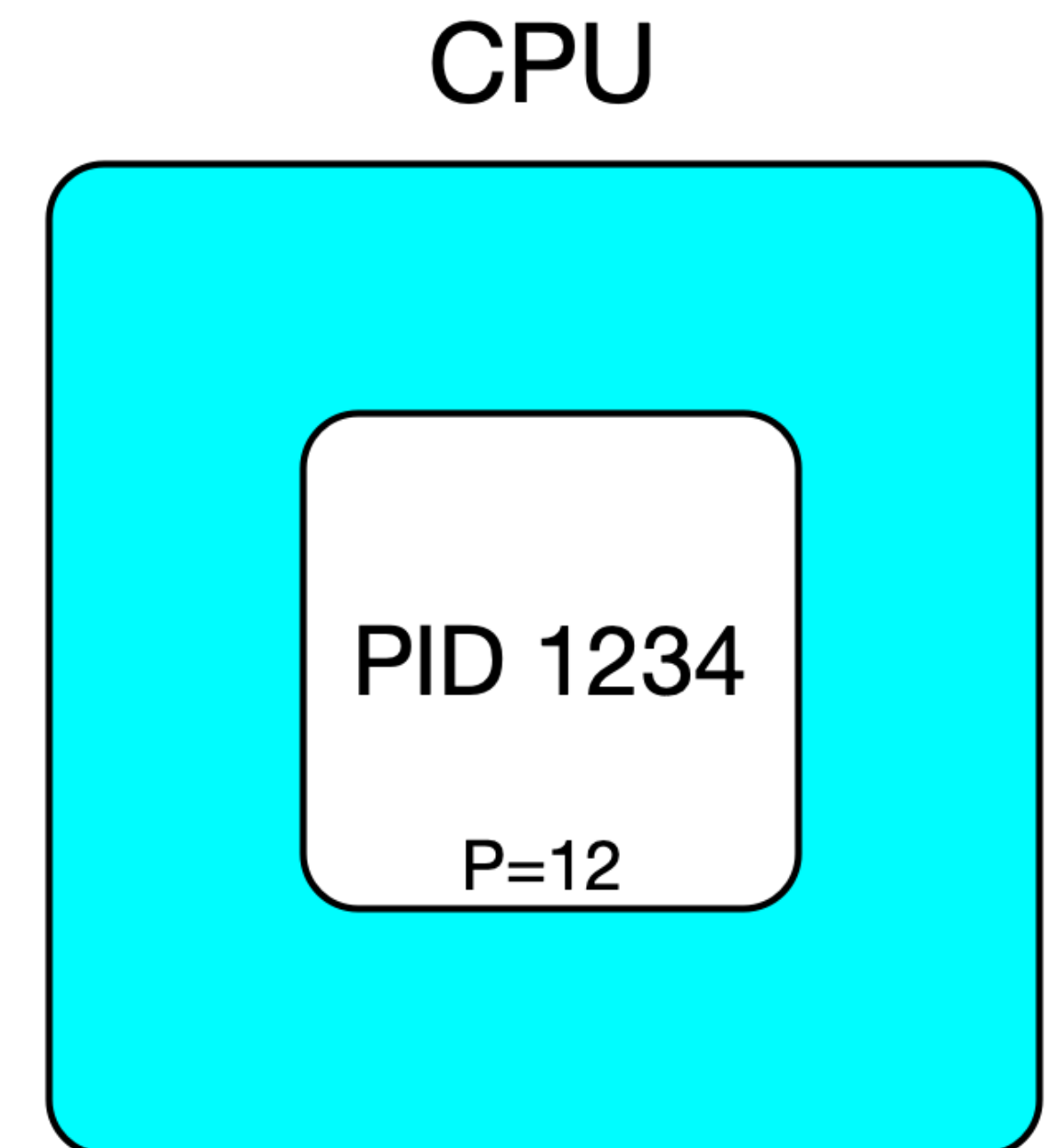
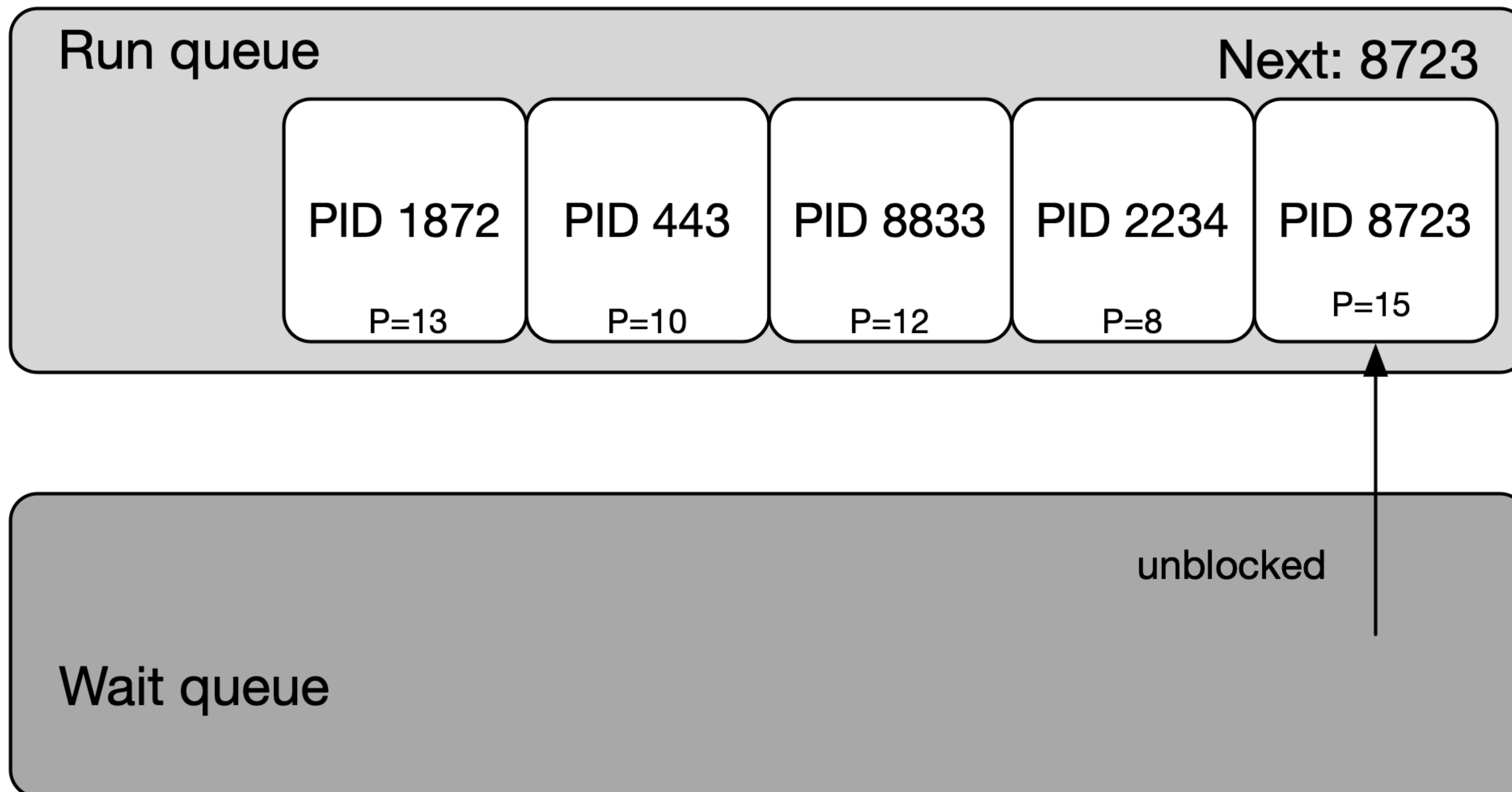
P=15

CPU

PID 1234

P=12

Priority Scheduling



Priority Scheduling

Run queue

Next: 1872

PID 1234

P=11

PID 1872

P=14

PID 443

P=11

PID 8833

P=13

PID 2234

P=9

Wait queue

CPU

PID 8723

P=15

getpriority(2)/setpriority(2)

```
#include <sys/resource.h>
```

```
int getpriority(int which, id_t who);
```

```
int setpriority(int which, id_t who, int prio);
```

Returns: 0 on success, -1 on error

- default priority is 0
- *which* is one of PRIO_PROCESS, PRIO_PGRP, PRIO_USER; *who* is PID, PGID, or UID.
- *prio* is a value $-20 \leq \text{prio} \leq 20$
- only the superuser may lower values
- getpriority(2) may return -1; need to inspect errno

```
My priority still is: 5
jschauma@apue$ sudo ./a.out 5
My current priority is: 0
My new priority is: 5
My priority is: 0
jschauma@apue$ nice -n 10 ./a.out 5
My current priority is: 10
a.out: setpriority: Permission denied
jschauma@apue$ nice -n 10 ./a.out 15
My current priority is: 10
My new priority is: 15
Unable to setpriority(): Permission denied
My priority still is: 15
jschauma@apue$ nice -n -5 ./a.out 5
nice: setpriority: Permission denied
My current priority is: 0
My new priority is: 5
Unable to setpriority(): Permission denied
My priority still is: 5
jschauma@apue$ sudo nice -n -5 ./a.out 5
My current priority is: -5
My new priority is: 5
My priority is: -5
jschauma@apue$
```

Process Priorities

- processes can voluntarily self-restrict their resource utilization (revisit Week 06, Segment 5)
- CPU usage priority can be adjusted using `setpriority(2)`
- use `nice(1)` or `renice(8)` to adjust the niceness of your process or process group (revisit Week 07, Segment 2)
- once you're nice, you cannot go back
- priority does not influence CPU placement