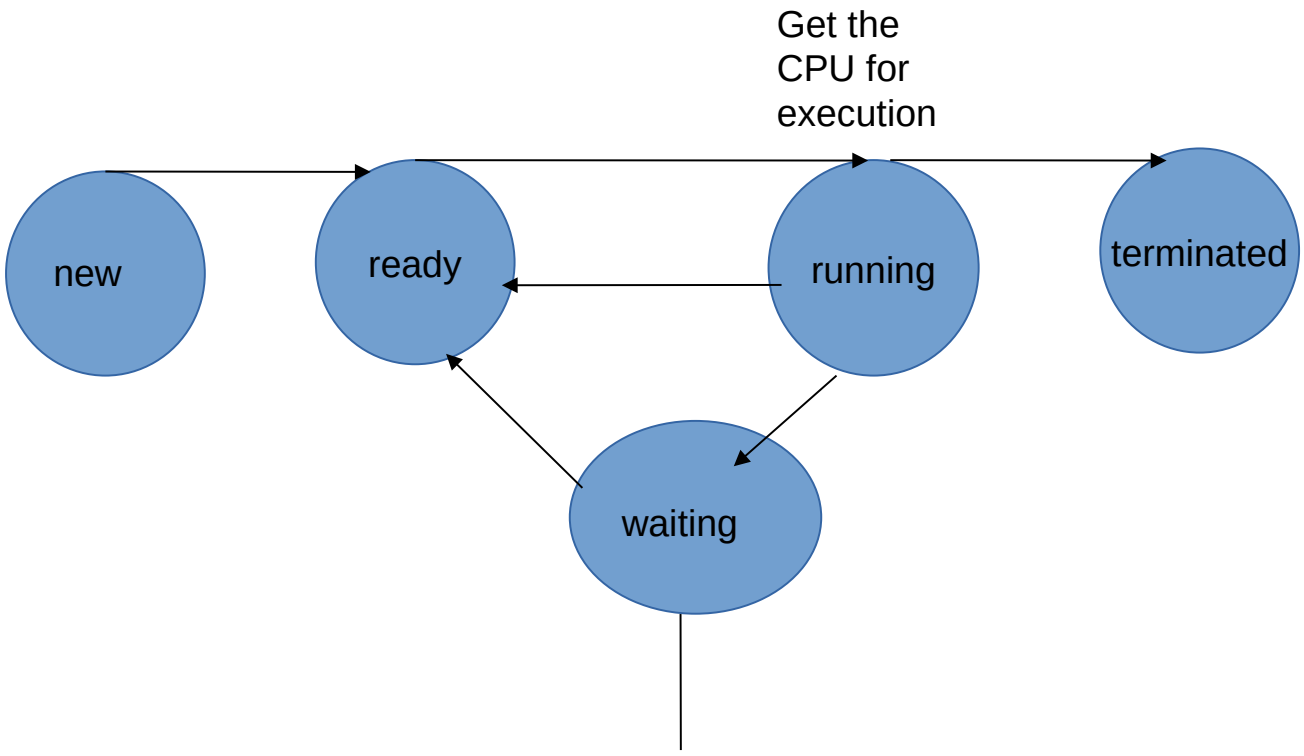
Pre-emptive and Non Pre-emptive scheduling

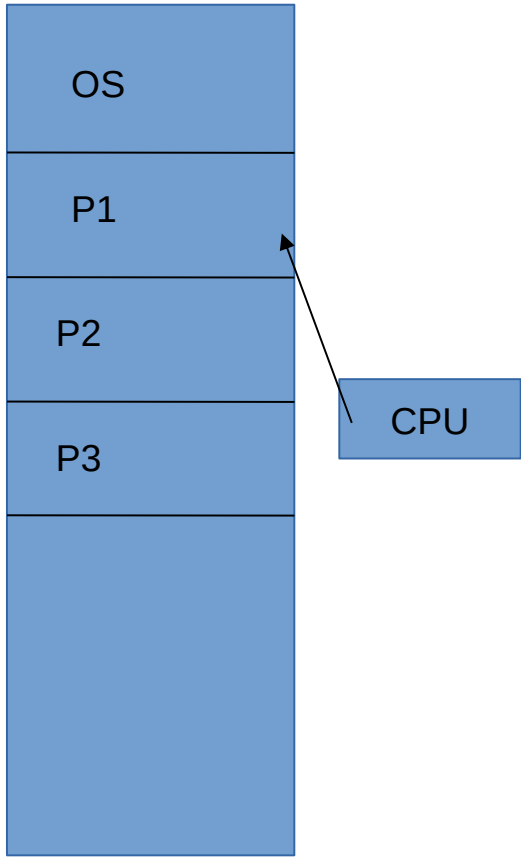
Pre-emptive scheduling

CPU scheduling decision may take place under the following circumstances:

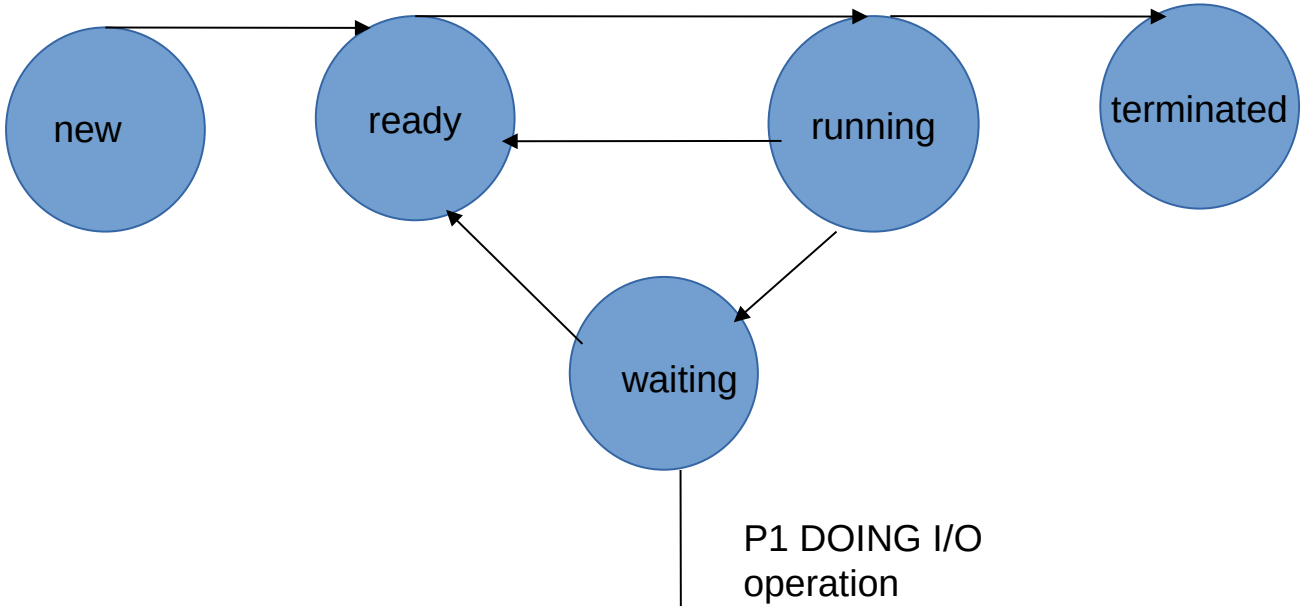
1. when a process switches from the running state to the waiting state.
2. when a process switches from the running state to the ready state.
3. when a process switches from the waiting state to the ready state.
4. when a process terminate.

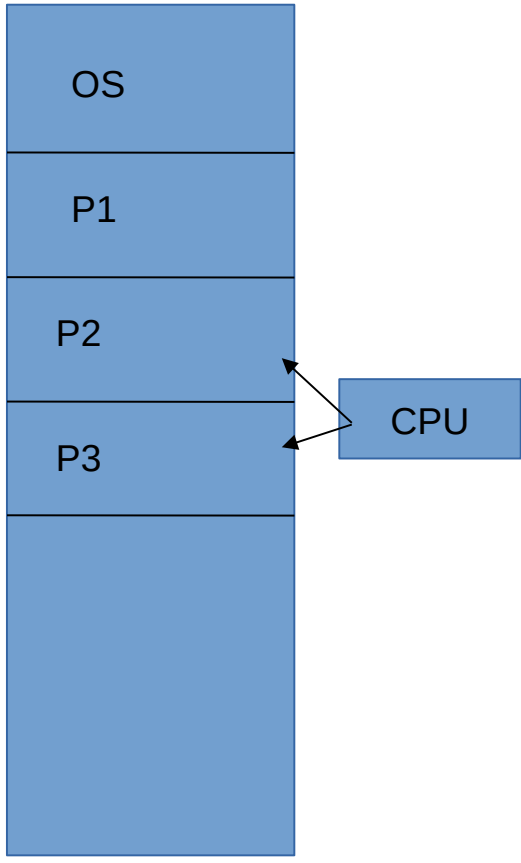
when a process switches from the running state to the waiting state.



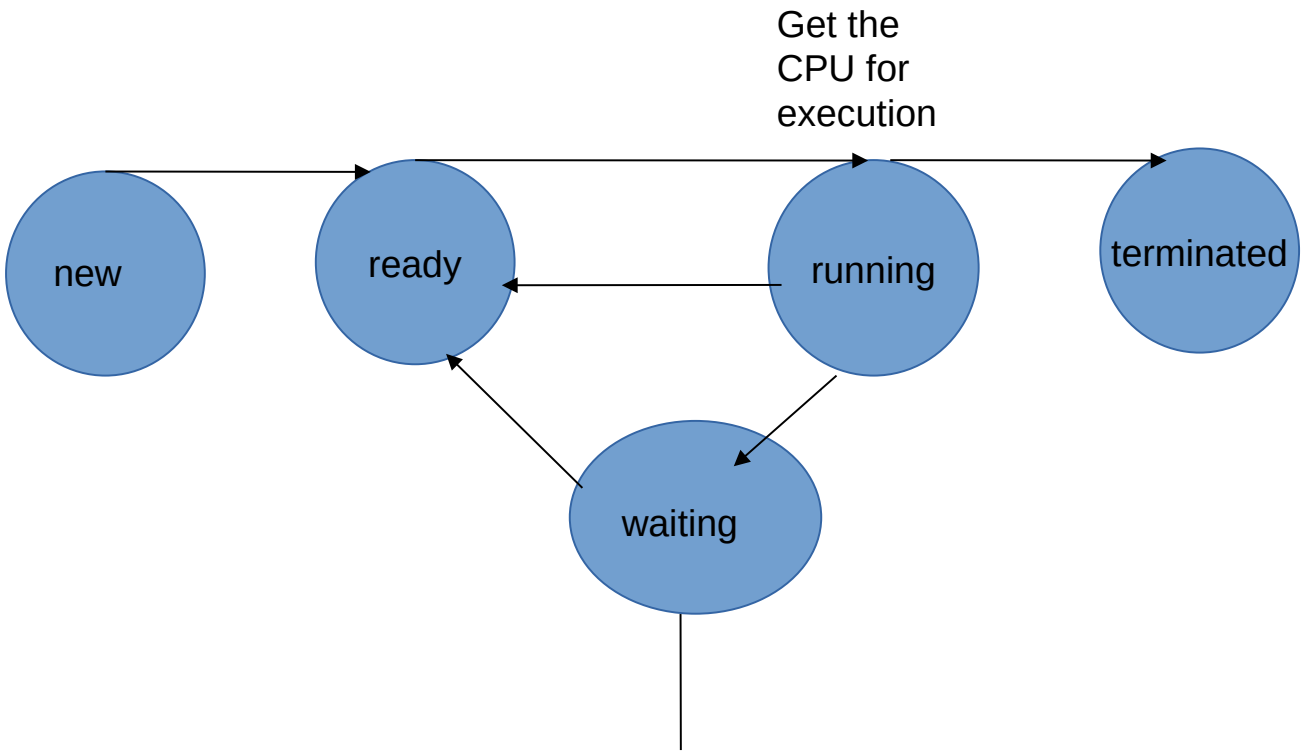


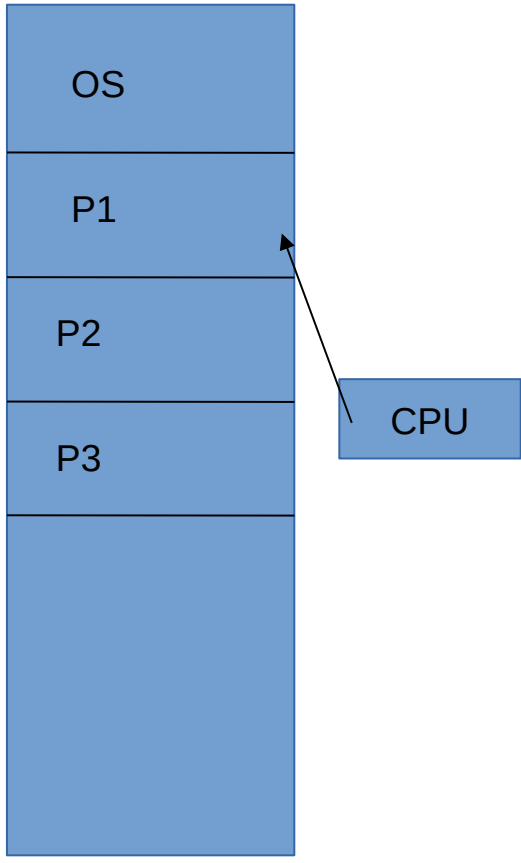
when a process switches from the running state to the waiting state.



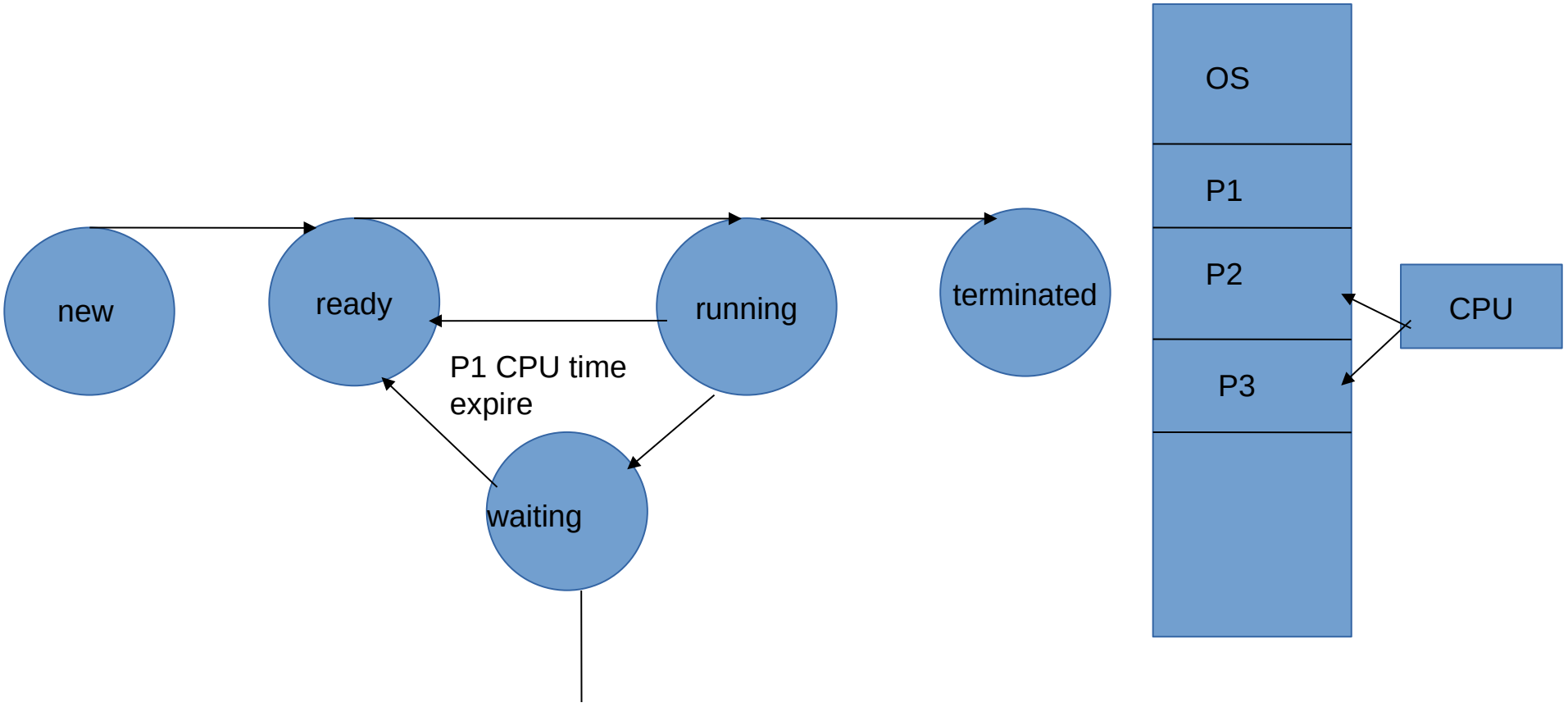


when a process switches from the running state to the ready state.





when a process switches from the running state to the ready state.



Pre-emptive and Non Pre-emptive scheduling

Non **Pre-emptive scheduling**

CPU scheduling decision may take place under the following circumstances:

1. when a process switches from the running state to the waiting state.

2. when a process terminate.

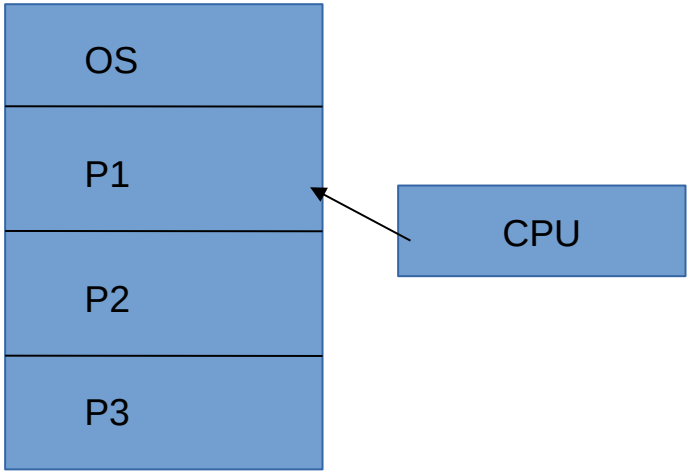
Disadvantage of Pre-emptive scheduling

1. Data inconsistency may arise. E.g. When two processes share data , one may be in the midst of updating the shared data when it is pre-empted and the second process is run. The second process may try to read the data which are currently in an inconsistent state.

2. During the processing of a system call, the kernel may be busy updating some kernel data structure. If it is pre-empted before the complete updating is done , they are no longer in a consistent state.

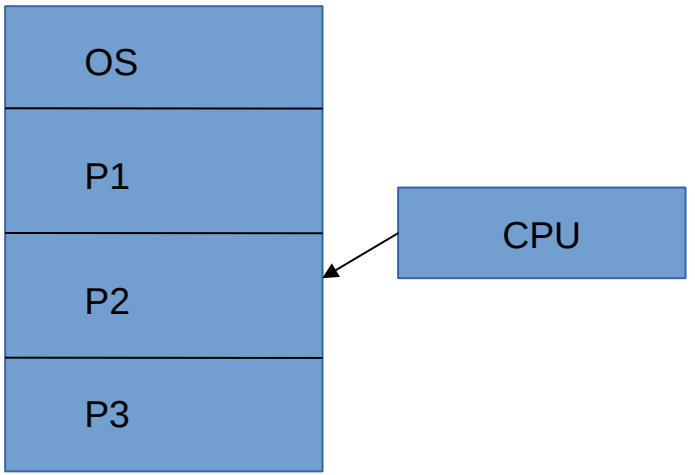
Dispatcher

Dispatcher is a module that gives control of the CPU to the process selected by the short-term scheduler.



Dispatcher

Dispatcher is a module that gives control of the CPU to the process selected by the short-term scheduler.



Short term scheduler selects p2

Dispatcher functions:

1. switching context.

2. switching to user mode.

3. jumping to the proper location in the user program to restart the program.

The time it takes for the dispatcher to stop one process and then start another running is known as dispatcher latency.

Scheduling criteria

1. CPU utilization. 0%-100%
2. Throughput : number of processes completed per unit time.
3. turnaround time: the interval from the time of submission to the time of completion for a process.
4. waiting time: sum of the periods spent waiting in the ready queue.
5. response time: from the submission of a request until the first response is produced.

CPU scheduling Algorithm

First come first served(FCFS)

Non-preemptive

The process that request the CPU first is allocated the CPU first.

It is implemented easily with a FIFO queue.

The average waiting time under FCFS is quite long

Consider the following situation with arrival time for each process is 0

Process	cpu time
P1	24
P2	3
P3	3

Average waiting time

Waiting time for P1=0

Waiting time for p2=24

Waiting time for p3= 27

Average waiting time= (0+24+27)/3=17

Consider the following situation with arrival time for each process is 0

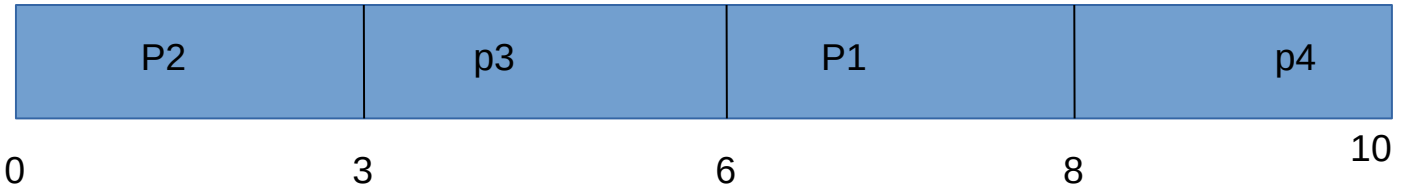
Process	cpu time
P2	3
P3	3
P1	24

What is the average waiting time?

Consider the following situation

Process	cpu time	arrival time
P2	3	0
P3	3	1
P1	2	2
P4	5	2

What is the average waiting time?



Convey effect

Consider a situation where FCFS is used and there is a CPU bound process and many I/O bound processes.

The CPU bound process will get the CPU and hold it.

During this time all the other processes will finish their I/O operation and move into the ready queue.

Now the I/O devices are idle .

When the Processes finish their CPU operation they come back to the I/O queue to performed I/O operation.

Now the CPU sit idle.

This is called convey effect.

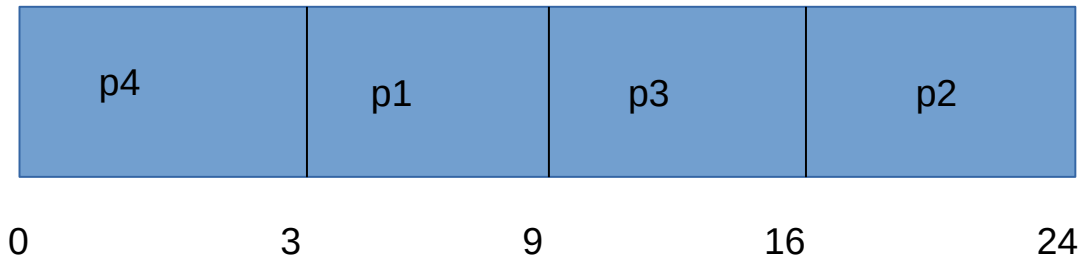
Shortage job first scheduling(SJF)

This algorithm associates with each process the length of the process next CPU time.

When the CPU is available all the processes in the ready queue are examined and the CPU is given to the process that has the smallest next CPU time.

Consider the processes with arrival time 0.

Process	CPU time
P1	6
P2	8
P3	7
P4	3



Average waiting time

Waiting time for P1=3

Waiting time for p2=16

Waiting time for p3=9

Waiting time for p4=0

Average waiting time=(3+16+9+0)/4=7

What is the average waiting time for the above problem using FCFS?????

Advantage & Disadvantage

Adv: It gives the the minimum average waiting time.

Dis adv:

It is difficult to get the length of the next CPU time.

Pre-emptive SJF Algorithm

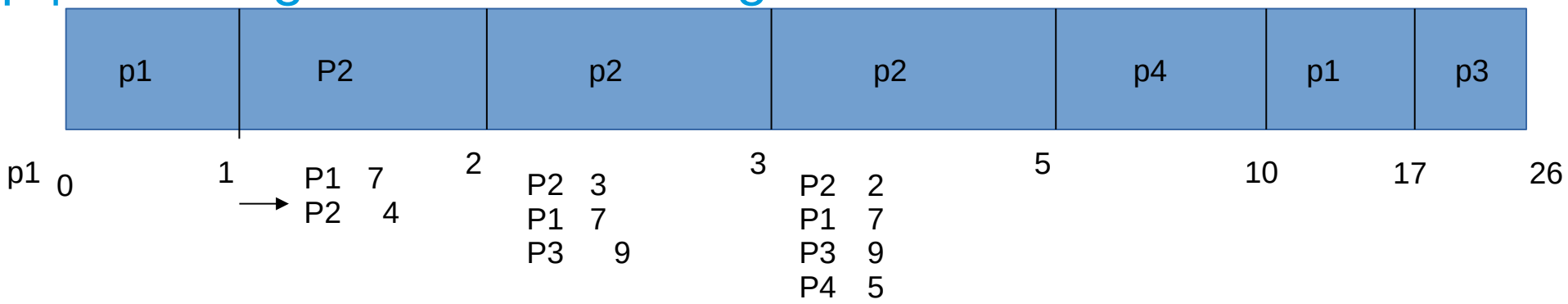
When a new process arrives at the ready queue while a previous process is executing, the new process may have a shorter CPU time than what is left of the currently executing process.

A pre-emptive SJF will pre-empt the currently executing process and the CPU will be given to the new process.

But a non pre-emptive SJF algorithm will allow the currently running process to finish its CPU time.

Consider the following situation

Process	arrival time	CPU time
P1	0	8
P2	1	4
P3	2	9
P4	3	5



Average waiting time?

Apply Non pre emptive SJF in the same data and find out average waiting time