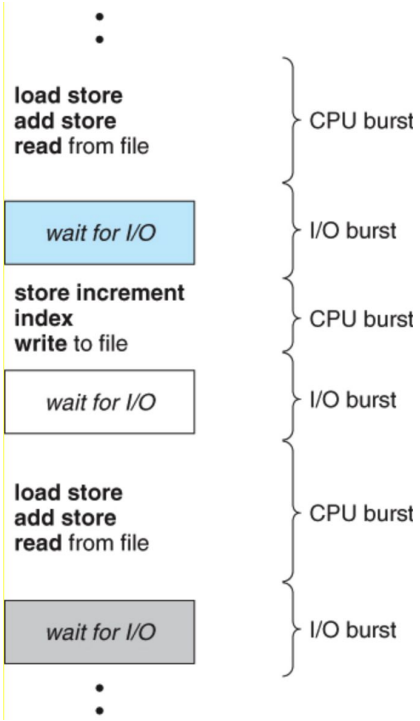
Process Scheduling

Md. Asifur Rahman
Lecturer
Department of CSE, RUET

CPU / IO Burst Cycle

Almost all processes alternate between two states in a continuing cycle :

- A CPU burst of performing calculations, and
- An I/O burst, waiting for data transfer in or out of the system.



CPU bound process and I/O bound process

- CPU Bound means the rate at which process progresses is limited by the speed of the CPU. A task that performs calculations on a small set of numbers, for example multiplying small matrices, is likely to be CPU bound.
  - A CPU-bound program might have a few long CPU bursts.
- I/O Bound means the rate at which a process progresses is limited by the speed of the I/O subsystem. A task that processes data from disk, for example, counting the number of lines in a file is likely to be I/O bound.
  - An I/O-bound program typically has many short CPU bursts.

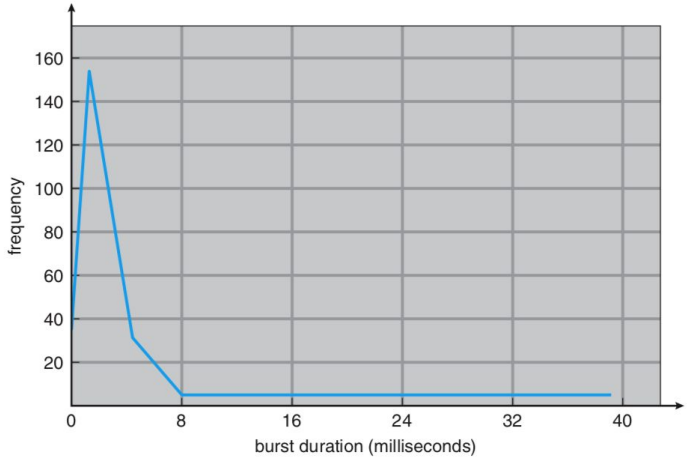


Figure Histogram of CPU-burst durations.

Schedulers

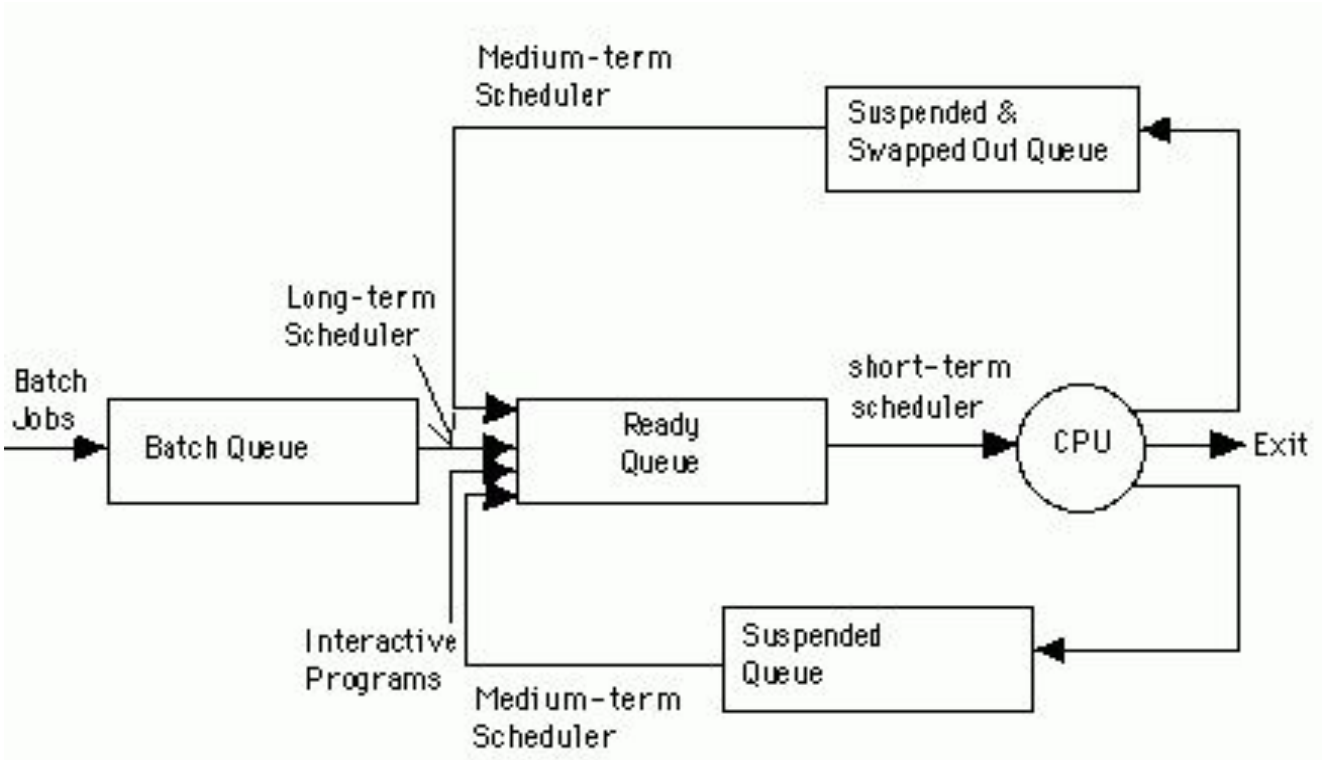
- **Long term scheduler:** This scheduler selects process from secondary memory pool and load them into the ready queue of Main Memory (MM).
  - This invoke very infrequently because process selection takes time.
  - Controls the number of processes in main memory.
  - It controls the degree of multiprogramming (Number of processes on MM).
  - It select a good mix of **I/O bound** processes and **CPU bound** processes to maximize both the I/O devices utilization and the CPU utilization.

- **Short term scheduler:** The job of this scheduler is to select process from ready queue for CPU allocation when current process releases the CPU.
  - This scheduler is invoked very frequently.
  - Short-term scheduler increases the system performance.

Schedulers

- **Medium term scheduler:** This scheduler is a part of swapping. So, we can call process swapping scheduler. It does swapping from main memory to disk or vice versa. It works for suspending and resuming the process.
    - Swap in/out partially executed processes to temporarily free up main memory by reducing the degree of multiprogramming (i.e., the number of processes in main memory).
    - We need this scheduler to improve the mix of I/O bound processes and CPU bound processes in main memory or to react to an increase in dynamic memory requests

Schedulers


Medium-term
Scheduler
Suspended &
Swapped Out Queue
Long-term
Scheduler
Batch
Jobs
Batch Queue
Ready
Queue
short-term
scheduler
CPU
Exit
Interactive
Programs
Suspended
Queue
Medium-term
Scheduler

Schedulers

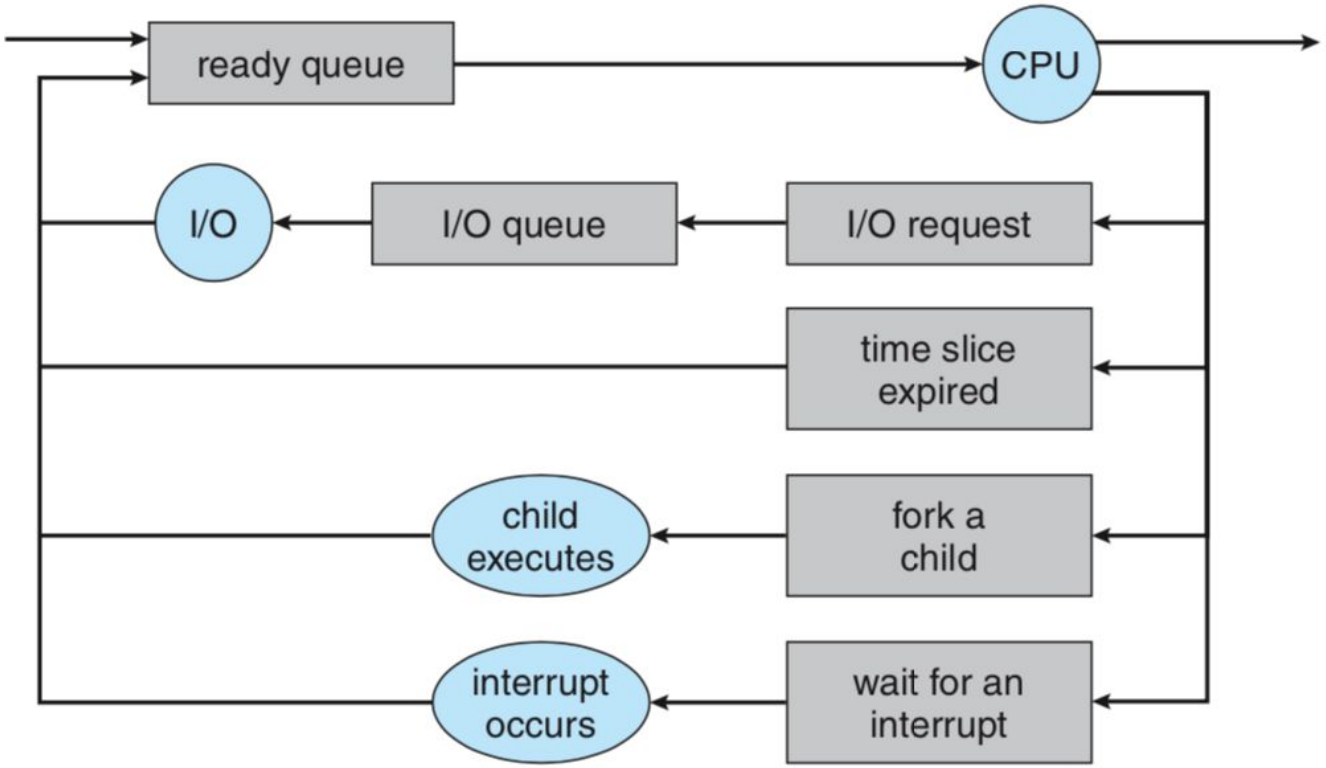


Figure 3.6 Queueing-diagram representation of process scheduling.

Preemptive & Non-preemptive scheduling

CPU scheduling decisions take place under one of four conditions:

- When a process switches from the running state to the waiting state, such as for an I/O request or invocation of the wait( ) system call.
- When a process switches from the running state to the ready state, for example in response to an interrupt.
- When a process switches from the waiting state to the ready state, say at completion of I/O or a return from wait( ).
- When a process terminates.

❏ For conditions 1 and 4 there is no choice - A new process must be selected.

❏ For conditions 2 and 3 there is a choice - To either continue running the current process, or select a different one.

If scheduling takes place only under conditions 1 and 4, the system is said to be **non-preemptive**, or **cooperative**. Under these conditions, once a process starts running it keeps running, until it either voluntarily blocks or until it finishes. Otherwise the system is said to be **preemptive**.

Dispatcher

- After a process has been selected by the scheduler.
- **Dispatcher** takes place. Dispatcher is a kernel program than gives control of the CPU to the selected (by scheduler) process by performing the following tasks:
    - Switching context.
    - Switching to user mode.
    - Jumping to the proper location in the newly loaded program.
    -

The dispatcher needs to be as fast as possible, as it is run on every context switch. The time consumed by the dispatcher is known as **dispatch latency.**

Terminology

1. CPU Utilization:
2. Throughput:
3. **Turnaround time**: Completion time - Arrival time
4. **Waiting time:** Turnaround time - Burst time
5. **Response time:** Time to CPU Allocation - Arrival Time

Scheduling algorithm

1. First Come First Served Scheduling
2. Shortest Job First Scheduling
3. Shortest Remaining Time Scheduling
4. Round Robin Scheduling
5. Priority Scheduling
6. Multilevel Feedback Queue Scheduling

First Come First Served Scheduling (FCFS)

Process ID	Arrival Time	Burst Time
P0	0	3
P1	2	3
P2	8	4
P3	9	4

First Come First Served Scheduling (FCFS)

Process ID	Arrival Time (A)	Burst Time (B)	Completion Time (C)	Turn Around Time (T = C-A)	Waiting Time (W=T-B)	Response Time
P0	0	3	3	3	0	0
P1	2	3	6	4	1	1
P2	8	4	12	4	0	0
P3	9	4	16	7	3	3

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
p0	p0	p0	p1	p1	p1			p2	p2	p2	p2	p3	p3	p3	p3	p0

First Come First Served Scheduling (FCFS)

Process ID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time	Response Time
P0	0	3	3	3	0	0
P1	2	3	6	4	1	1
P2	8	4	12	4	0	0
P3	9	4	16	7	3	3

Average turn-around time = (3+4+4+7) / 4 = 18/4 = 4.5
Average waiting time = (0+1+0+3) / 4 = 4/4 = 1
Average response time = (0+1+0+3) / 4 = 1

Shortest Job First Scheduling (SJF)

Process ID	Arrival Time	Burst Time
P0	0	3
P1	2	4
P2	2	2
P3	4	3

Shortest Job First Scheduling (SJF)

Process ID	Arrival Time (A)	Burst Time (B)	Completion Time (C)	Turn Around Time (T = C-A)	Waiting Time (W=T-B)	Response Time
P0	0	3	3	3	0	0
P1	2	4	12	10	6	6
P2	2	2	5	3	1	1
P3	4	3	8	4	1	1

0	1	2	3	4	5	6	7	8	9	10	11
p0	p0	p0	p2	p2	p3	p3	p3	p1	p1	p1	p1

Shortest Remaining Time Scheduling

Process ID	Arrival Time	Burst Time
P0	0	3
P1	2	4
P2	3	2
P3	4	3

Round Robin Scheduling

Process ID	Arrival Time	Burst Time
P0	0	3
P1	2	4
P2	3	2
P3	4	3

Time Slice / Time Qunat = 2

1. Priority Scheduling (Book -Abraham Silberschatz)
2. Multilevel Feedback Queue Scheduling (Book -Abraham Silberschatz)

Advantages and Disadvantages of the Scheduling algorithms

- Self Study From Book (Abraham Silberschatz)

References

[1] https://tutorialwing.com/process-scheduling-and-process-schedular/

[2] Modern Operating System – by Tanenbaum

[3] Operating System Concepts - by Abraham Silberschatz