MS SERVER 2016

CPU SCHEDULING

NAME : KHADIJA IBRAHIM ALMUJRAB

ID NUMBER: 163104461

Supervisor : Prof .Dr. Hasan H. Balik

Outline

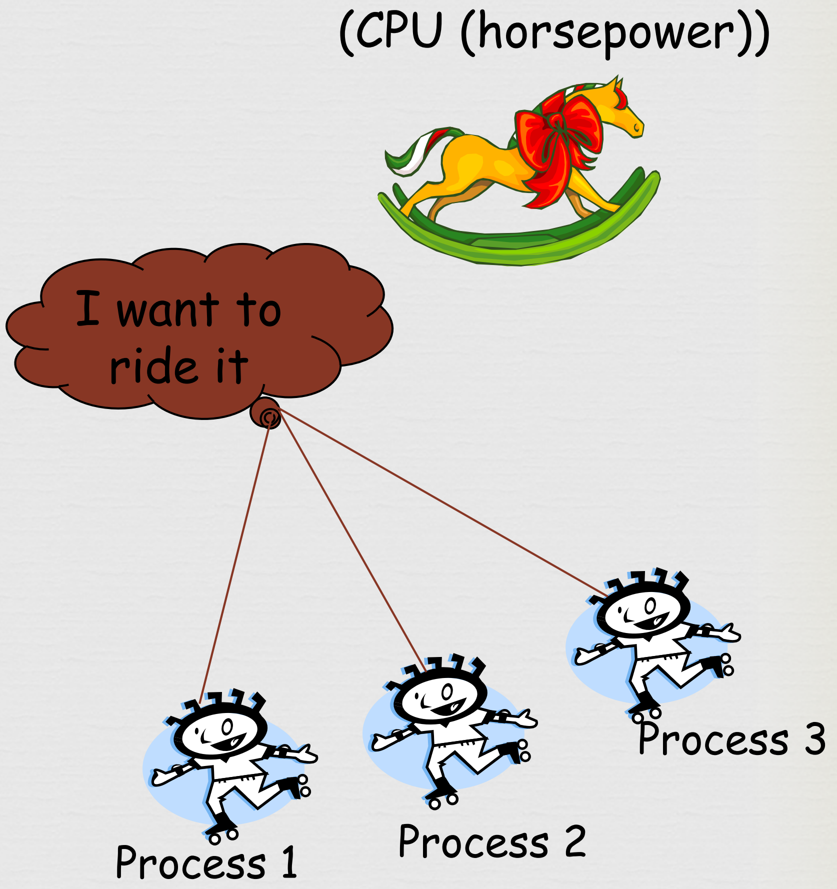
- ❖ Introduction
- ❖ CPU-I/O Burst Cycle
- ❖ Dispatcher VS Scheduler
- ❖ Preemptive and Non preemptive
- ❖ Scheduling Criteria
- ❖ Scheduling Algorithms
  - First Come, First Served (FCFS)
  - Round Robin (RR)
  - Multilevel Feedback Queue Scheduling
- ❖ Advantages and Disadvantages
- ❖ Conclusion
- ❖ References

INTRODUCTION

CPU scheduling : it is determining which processes run when there are multiple runnable processes .

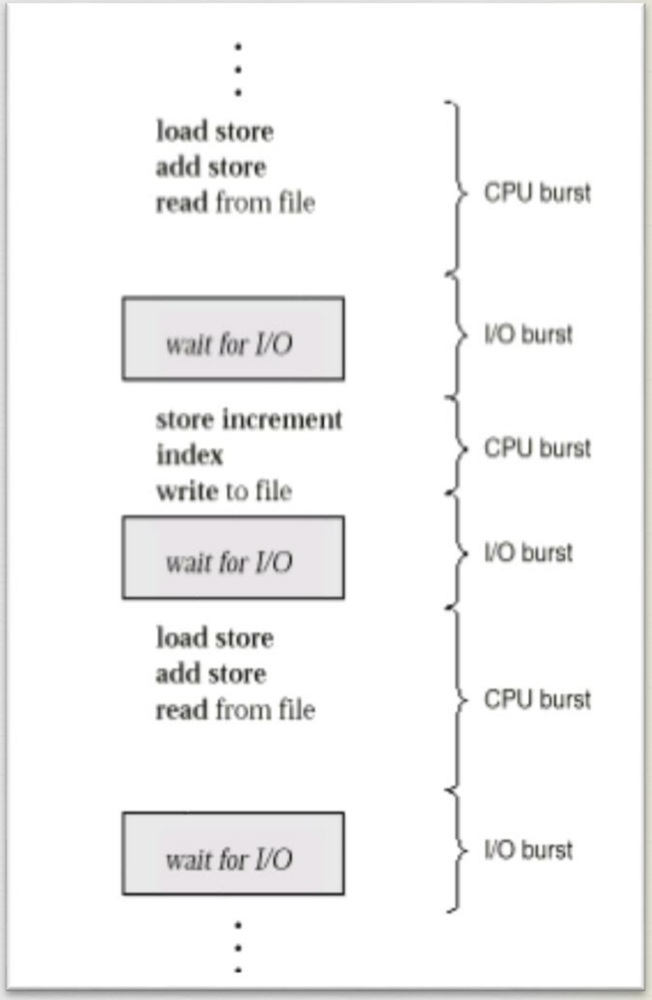
The aim of CPU scheduling is to make the system efficient, fast and fair.

It is important Because it can have a big effect on resource utilization and the overall performance of the system .

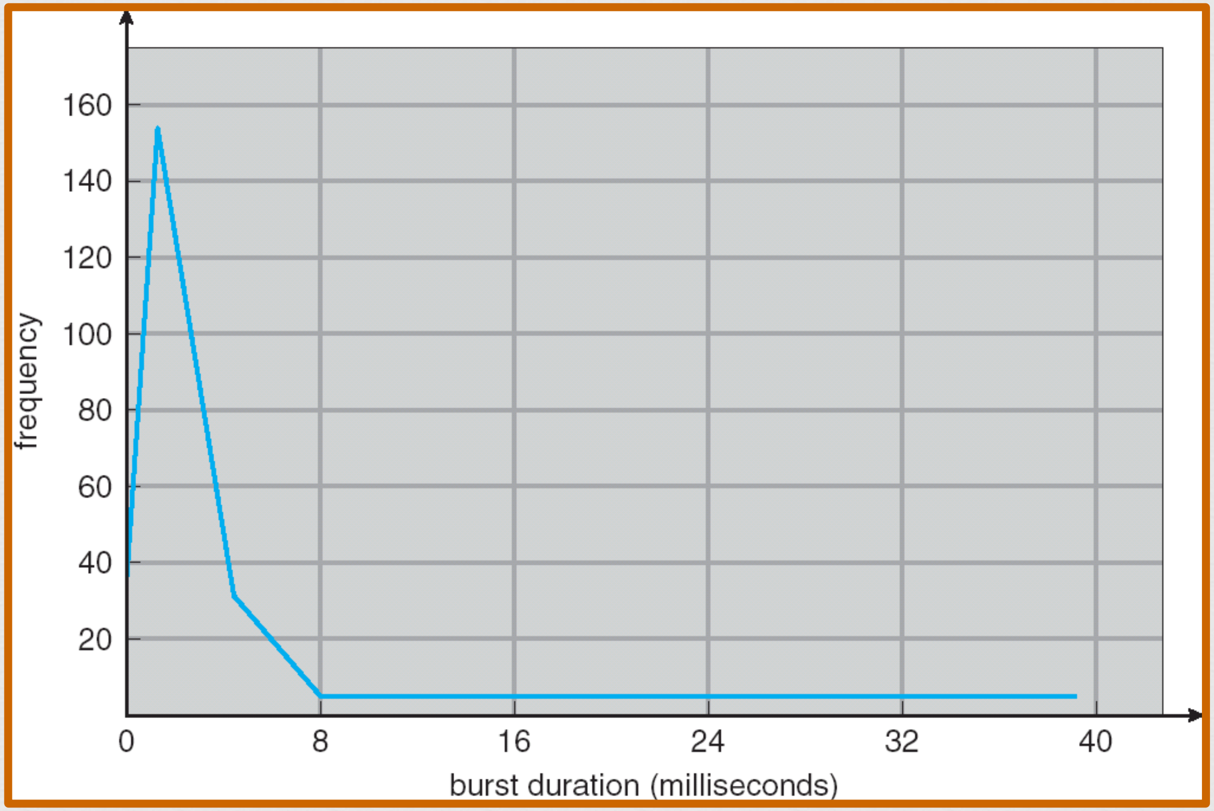


CPU-I/O BURST CYCLE

- Process execution consists of a cycle of CPU execution and I/O wait .
- Process execution start with a CPU burst .
- Each process repeatedly goes through cycles that alternate CPU execution (a CPU burst) and I/O wait .
- process execution will terminate in CPU .
- Usually, in a process execution, there are a large number of short CPU burst and a small number of long CPU burst .



HISTOGRAM OF CPU-BURST TIMES

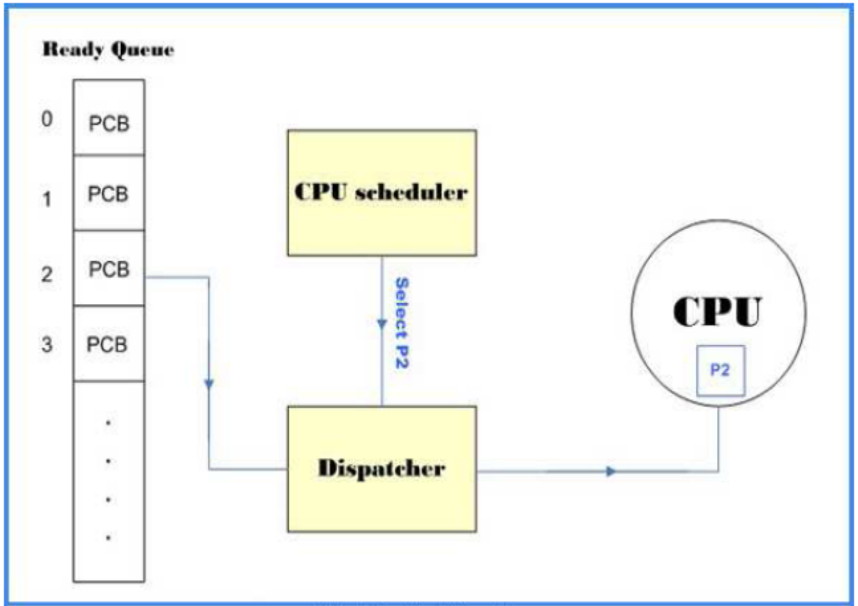


It is characterized by a large number of short CPU bursts and a small number of long CPU bursts. An I/O-bound program typically has many short CPU bursts ; a CPU-bound program might have a few long CPU bursts .

DISPATCHER VS SCHEDULER

- **Dispatcher :** It is module gives control of the CPU to the process selected by the short-term scheduler; this involves:
  - switching context
  - switching to user mode
  - jumping to the proper location in the user program to restart that program .
  - The time it takes for the dispatcher to stop one process and start another process is called **dispatch latency**
- **CPU Scheduler :** Selects from among the processes in memory that are ready to execute, and allocates the CPU to one of them .

– There are three type of CPU scheduler :
  - Short-term scheduler
  - Medium-term scheduler
  - Long-term scheduler



Scheduler Vs. Dispatcher

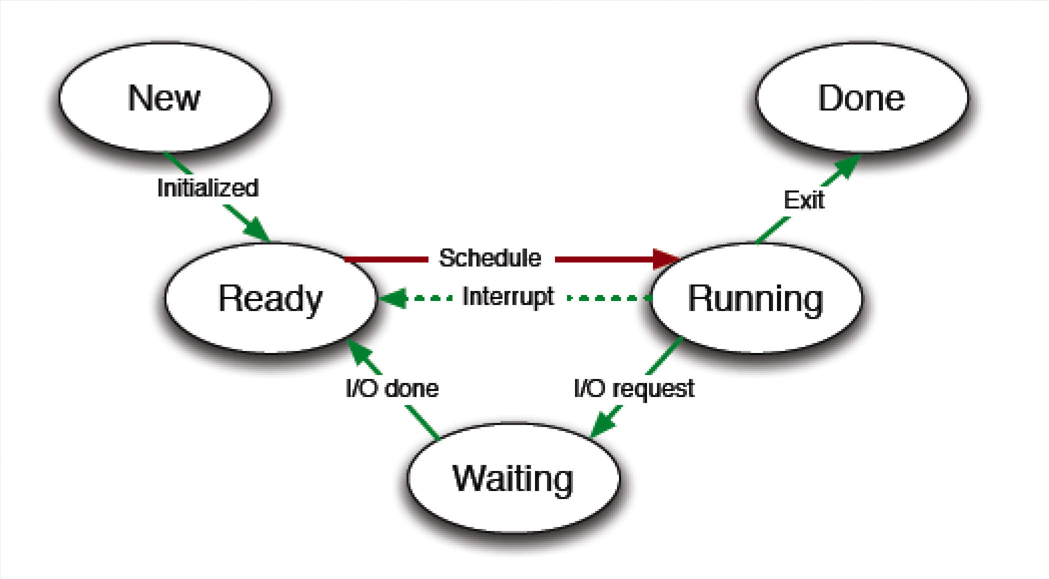
Preemptive Scheduling

Once Processor starts to execute a process it must finish it before executing the other.

Non Preemptive Scheduling

An interrupt causes currently running process to give up the CPU and be replaced by another process .

- CPU scheduling decisions may take place when a process:
  1. Switches from running to waiting state
  2. Switches from running to ready state
  3. Switches from waiting to ready
  4. Terminates
- Scheduling under 1 and 4 is **non preemptive**
- All other scheduling is **preemptive**



SCHEDULING CRITERIA

- Criteria for comparing CPU scheduling algorithms may include the following :

  - **CPU utilization** – keep the CPU as busy as possible .
  - **Throughput** – number of processes that are completed per time unit .
  - **Response time** – amount of time it takes from when a request was submitted until the first response occur .
  - **Waiting time** –the amount of time a process has spent waiting in the ready queue .
  - **Turnaround time** – amount of time to execute a particular process from the time of submission to the time of completion

OPTIMIZATION CRITERIA

- It is desirable to
  - Maximize CPU utilization
  - Maximize throughput
  - Minimize response time
  - Minimize waiting time
  - Minimize turnaround time

- In other cases, it is more important to optimize the minimum or maximum values rather than the average

CPU SCHEDULING ALGORITHMS

FIRST-COME, FIRST-SERVED (FCFS)

- With FCFS the process that requests the CPU first is allocated the CPU first
- Suppose that the processes arrive in the order: P1 , P2 , P3

Process	Burst Time
P_1	12
P_2	3
P_3	3

- The Gantt Chart for the schedule is:

- Waiting time for $P_1 = 0$; $P_2 = 12$; $P_3 = 15$
- Average waiting time: $(0 + 12 + 15)/3 = 9$
- Average turnaround time: $(12 + 15 + 18)/3 = 15$

ROUND ROBIN (RR)

- Each process gets a small unit of CPU time (a *time quantum*), usually 10-100 milliseconds. After this time has elapsed, the process is preempted and added to the end of the ready queue.
- .

Example of RR with Time Quantum = 20

Process	Burst Time
P_1	53
P_2	17
P_3	68
P_4	24

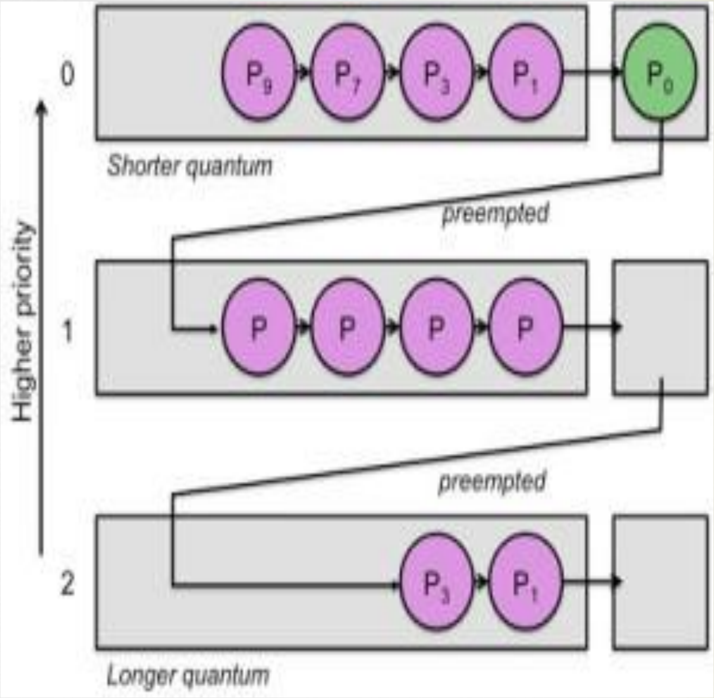
- The Gantt chart is:

P_1	P_2	P_3	P_4	P_1	P_3	P_4	P_1	P_3	P_3

0 20 37 57 77 97 117 121 134 154 162

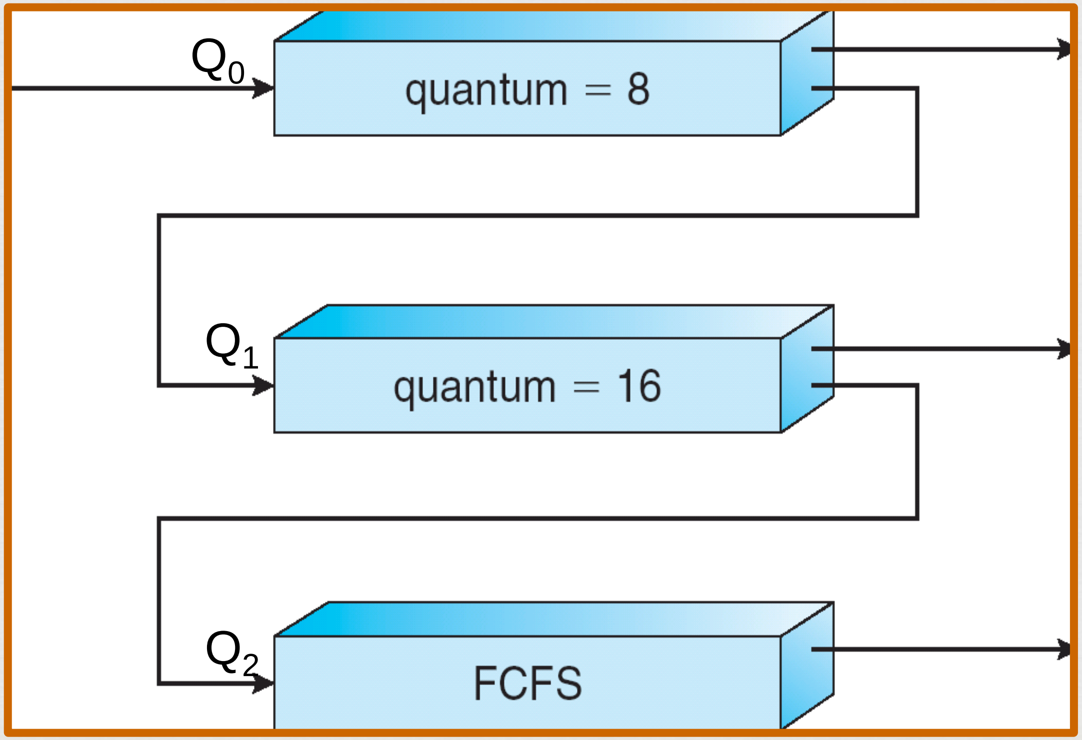
MULTILEVEL FEEDBACK QUEUE

- Partitions the ready queue into several separate queues .
- A process can move between the various queues; aging can be implemented this way .
- It is defined by the following parameters:
  - Number of queues.
  - Scheduling algorithms for each queue
  - Method used to determine when to promote a process .
  - Method used to determine when to demote a process.
  - Method used to determine which queue a process will enter when that process needs service.



EXAMPLE OF MULTILEVEL FEEDBACK QUEUE

- A new job enters queue Q_0 *(RR)* and is placed at the end. When it gains the CPU, the process receives 8 milliseconds. If it does not finish in 8 milliseconds, the process is moved to the end of queue Q_1.
- A Q_1 (RR) process receives 16 milliseconds. If it still does not complete, it is preempted and moved to queue Q_2 (FCFS).



ADVANTAGES AND DISADVANTAGES

Algorithms	Advantages	Disadvantages
FCFS	– Simple – Fair – Easy to understand and implement .	– Waiting time depends on arrival order . – short processes stuck waiting for long process to complete
RR	– Fair (Each process gets a fair chance to run on the CPU) . – Low average wait time, when burst times vary . –Faster response time.	– Increased context switching . – High average wait time, when burst times have equal lengths.
MLFQ	– process that waits too long in a lower priority queue may be moved to a higher priority queue.	– Moving the process around queue produce more CPU overhead.

CONCLUSION

- ❖ Scheduling: selecting a waiting process from the ready queue and allocating the CPU to it .
- ❖ Purpose of Scheduling
  - Make maximum use of CPU time .
  - Make maximum use of resources such as input-output devices .
  - Avoid 'deadlock
- ❖ FCFS scheduling Run Until Done.
- ❖ RR scheduling:
  - Give each queue a small amount of CPU time when it executes, and cycle between all ready queue .
  - Better for short jobs, but poor when processes are the same length .
- ❖ Multi-Level Feedback Scheduling :
  - Multiple queues of different priorities
  - Automatic promotion/demotion of process priority to approximate SJF/SRTF

References

https://en.wikipedia.org/wiki/Scheduling_(computing)#Windows

https://www.ukessays.com/essays/computer-science/task-scheduling-based-on-multilevel-queue-scheduling-computer-science-essay.php

http://www.personal.kent.edu/~rmuhamma/OpSystems/os.html

http://www.studytonight.com/operating-system/cpu-scheduling

http://www.williamstallings.com/Extras/OS-Notes/h6.html

http://sciencehq.com/computing-technology/1353.html

Thanks a lot for everyone