Mac OS X CPU Scheduling

Supervisor: Prof. Dr. Hasan H. Balik

Student Name: Husam A. Mahmod Ateer

Student Number: **163110446**

Outline

- ✓*Operating System Mac OS X*
- ✓*Cpu scheduling*
- ✓*Criterias to check when considering the "best" scheduling algorithm*
- ✓*Scheduling Algorithms*
- ✓*CPU Scheduling Algorithm in Mac OS X*

Operating System Mac OS X

- The world’s most advanced desktop Operating System.
- Best Graphical User Interface.
- Gives a substantial free update to their operating systems.

Operating System Mac OS X

- Designed to be easy to use.
- engineered to take full advantage of the technologies built into every computer we make.
- Comes with powerful apps
- apps work great with iPhone, iPad, and iPod touch, too. 9

Operating System Mac OS X

- Most Popular Mac OS Version Mac OS X Snow Leopard Version 10.6 Initial Release August 28, 2009 10.

- Mac OS X Snow Leopard Mac OS X Snow Leopard is the seventh major release of Mac OS X, Apple's desktop and server OS for Macintosh computers.

CPU Scheduling

- CPU scheduling is a process which allows one process to use the CPU while the execution of another process is on hold(in waiting state).

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

Criterias to check when considering the "best" scheduling algorithm

- **CPU utilization:** CPU would be working most of the time(Ideally 100% of the time).
- **Throughput:** Total number of processes completed per unit time.
- **Turnaround time:** Amount of time taken to execute a particular process.
- **Waiting time:** The sum of the periods spent waiting in the ready queue.

Criterias to check when considering the "best" scheduling algorithm

- **Load average:** Average number of processes residing in the ready queue.
- **Response time:** Amount of time it takes from when a request was submitted until the first response is produced.
- In general ***CPU utilization*** and ***Throughput*** are ***maximized*** and other factors are reduced for proper optimization.

Scheduling Algorithms

- First Come First Serve(FCFS) Scheduling
- Shortest-Job-First(SJF) Scheduling
- Priority Scheduling
- Round Robin(RR) Scheduling.
- Multilevel feedback queue scheduling.

- **Mac OS X supports both the multilevel feedback queue scheduling and round-robin (RR) scheduling algorithm.**

CPU Scheduling Algorithm in Mac OS X

- Round Robin(RR) Scheduling is the preemptive process scheduling algorithm and used exactly in **Mac OS X** .

- Each process is provided a fix time to execute, it is called a quantum.

- Context switching is used to save states of preempted processes.

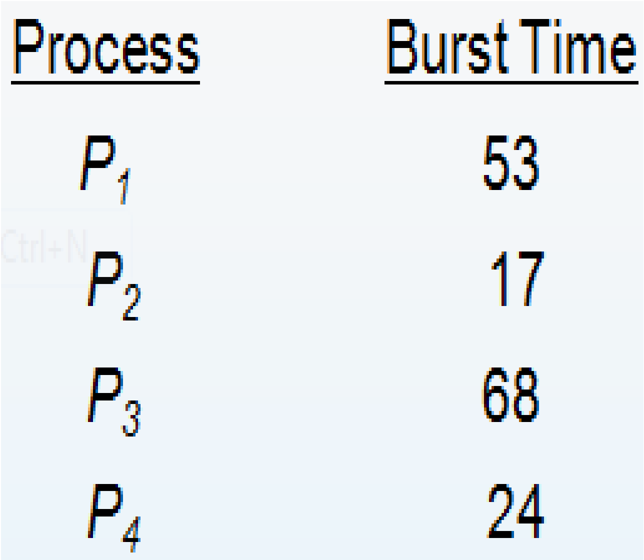
Round Robin(RR) Scheduling

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

- If there are *n* processes in the ready queue and the time quantum is *q*, then each process gets 1/*n* of the CPU time in chunks of at most *q* time units at once. No process waits more than (*n*-1)*q* time units.

Round Robin(RR) Scheduling Example in Mac OS X

- ✓ Process is name of processes thats wait the CPU.
- ✓ Burst Time is the amount of time required by the process from CPU.
- ✓ The unit of time can be anything like nano-second, second, minute etc whatever.



Process	Burst Time
P_1	53
P_2	17
P_3	68
P_4	24

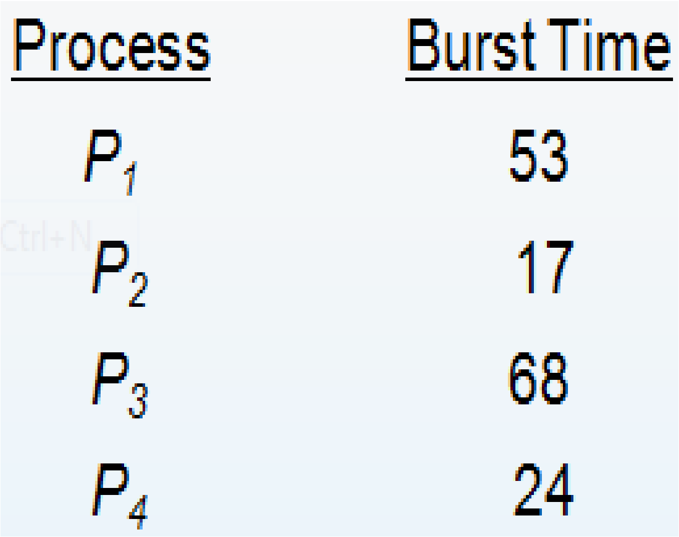
Simple Round Robin(RR) Scheduling Example in Mac OS X

- Consider the Beside list as the ready queue for the CPU.

A. Quantum time= 20,

B. Processes= P1, P2, P3, P4.

C. Burst Time P1=53, P2= 17, P3=68, P4= 24.



Process	Burst Time
P_1	53
P_2	17
P_3	68
P_4	24

Round Robin(RR) Scheduling Example in Mac OS X

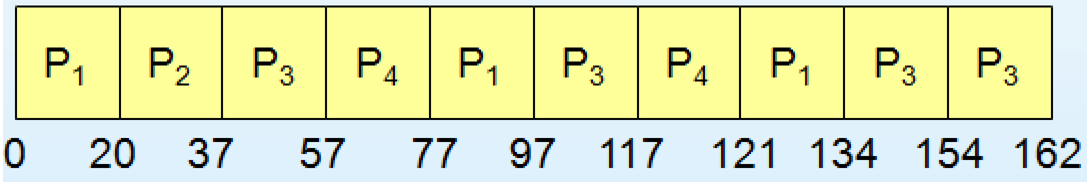
- Total waiting time = (Waiting Time of Process-1)+ (Waiting Time of Process-2)+ (Waiting Time of Process-3)+ (Waiting Time of Process-3)

- Total waiting time= ([(0 – 0) + (77 - 20) + (121 – 97)] + (20 – 0) + [(37 – 0) + (97 - 57) + (134 – 117)] + [(57 – 0) + (117 – 77)])

  = (0 + 57 + 24) + 20 + (37 + 40 + 17) + (57 + 40))

  = (81 + 20 + 94 + 97)

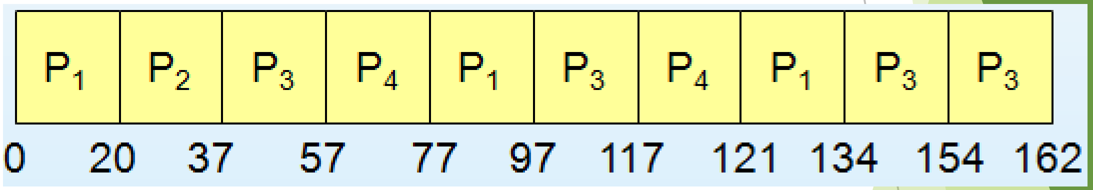
  = 292 = 73



Round Robin(RR) Scheduling Example in Mac OS X

- Average waiting time is = (Total waiting time / Number of Processes)s

- Average waiting time = ([(0 – 0) + (77 - 20) + (121 – 97)] + (20 – 0) + [(37 – 0) + (97 - 57) + (134 – 117)] + [(57 – 0) + (117 – 77)]) / 4
= (0 + 57 + 24) + 20 + (37 + 40 + 17) + (57 + 40)) / 4
= (81 + 20 + 94 + 97)/4
= 292 / 4 = 73



Advantages and Disadvantage of Round-Robin

- ❑ **Advantages**
- ✓ There is fairness since every process gets equal share of CPU.
- ✓ The newly created process is added to end of ready queue.
- ✓ A round-robin scheduler generally employs time-sharing, giving each job a time slot or quantum.

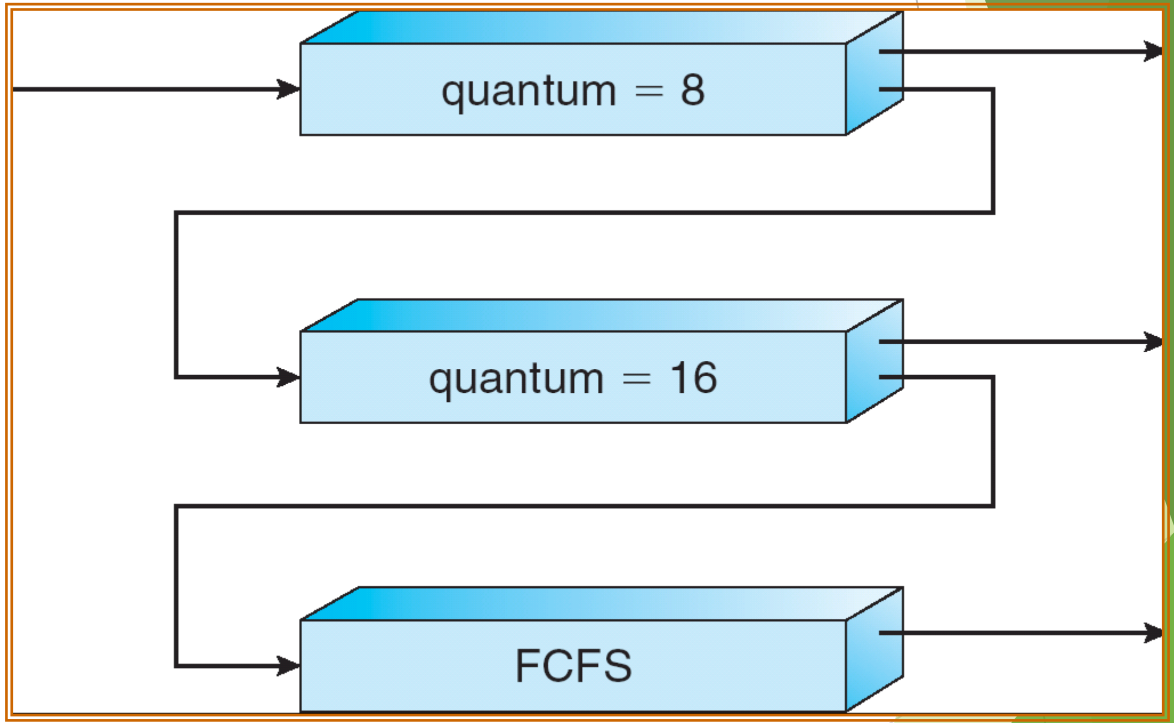
- ❑ **Disadvantage**
- ✓ Poor average waiting time when jobs have similar lengths.
- ✓ Performance depends on length of time slice.

Multilevel Feedback Queue

- A process can move between the various queues.
- Multilevel-feedback-queue scheduler defined by the following parameters:
- number of queues
- scheduling algorithms for each queue
- method used to determine when to upgrade 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

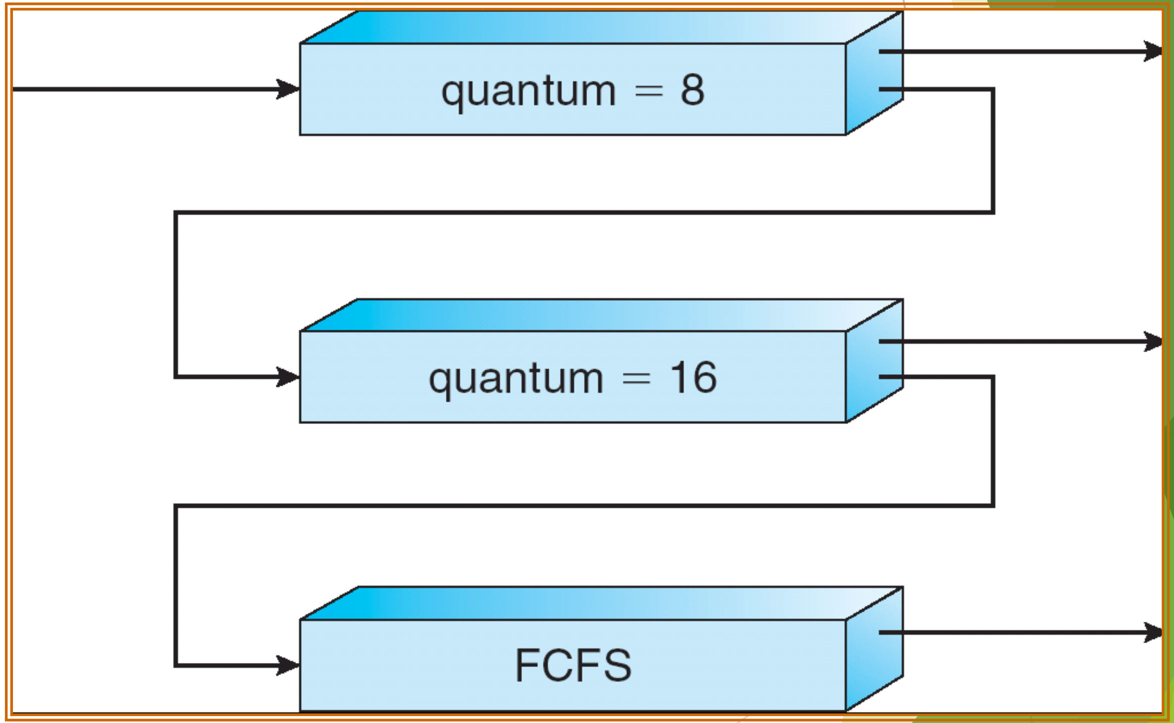
- Three queues:
- Q_0 – RR with time quantum 8 milliseconds
- Q_1 – RR time quantum 16 milliseconds
- Q_2 – FCFS



Example of Multilevel Feedback Queue

- A new job enters queue Q_0 which is served FCFS. When it gains CPU, job receives 8 milliseconds. If it does not finish in 8 milliseconds, job is moved to queue Q_1.

- At Q_1 job is again served FCFS and receives 16 additional milliseconds. If it still does not complete, it is preempted and moved to queue Q_2.



What is the advantage and disadvantage of multilevel queue scheduling?

- The ***advantage*** of multilevel queue scheduling is that it covers all disadvantage of all others scheduling.

- The main ***disadvantage*** of multilevel queue scheduling is that it is very difficult to understand and it.

Thanks for Attention