Operating Systems

CS2006

Dr. Rana Asif Rehman

Lecture 8

CPU Scheduling-II

22nd February 2023

1 CS-2006 Operating Systems

What’s in today’s lecture

 Basic Concepts

 Scheduling Criteria

 Scheduling Algorithms

2 CS-2006 Operating Systems

Scheduling Algorithms

 First come, First serve (FCFS)

 Shortest Job First (SJF)

 Priority Scheduling

 Round-Robin Scheduling

 Multi-level Queue Scheduling

 Multi-level Feed back queue Scheduling

CS-2006 Operating Systems

3

2. Shortest-Job-First (SJF) Scheduling

 Associate with each process the length of its next CPU burst. Use these lengths to schedule the process with the shortest time

 Two schemes:

 Nonpreemptive – once CPU is given to the process it cannot be preempted until the process completes its CPU burst  Preemptive – if a new process arrives with CPU burst length less than the remaining time of current executing process, preempt. This scheme is know as the Shortest-Remaining

Time-First (SRTF)

 SJF is optimal – gives minimum average waiting time for a given set of processes

4 CS-2006 Operating Systems

Process Arrival Time Burst Time P10.0 7

P22.0 4

P34.0 ~~1~~

P45.0 ~~4~~

Example of Non-Preemptive SJF 0 3 ~~16~~ 7

 SJF (non-preemptive)

P1 P3 P2

P4

8 12

 Average waiting time = (0 + 6 + 3 + 7)/4 = 4

5 CS-2006 Operating Systems

SRTF - Shortest Remaining Time  Preemptive version of SJF

 Ready queue ordered on length of time till completion(shortest first)  Arriving jobs inserted at proper position  Shortest job

 Runs to completion (i.e. CPU burst finishes) or

First

 Runs until a job with a shorter remaining time arrives (i.e.placed in the ready queue) 6 CS-2006 Operating Systems

Process Arrival Time Burst Time P10.0 7

P22.0 4

P34.0 ~~1~~

P45.0 ~~4~~

Example of Preemptive SJF (i.e., SRTF)

 Preemptive SJF (i.e., SRTF)

P1 P2 P3

P2 P1

P4

2 4 0 11

5 7

16

 Average waiting time = (9 + 1 + 0 +2)/4 = 3

7 CS-2006 Operating Systems

Shortest-Job-First (SJF) Scheduling Ready queue treated as a priority queue based on smallest CPU time requirement  Arriving jobs inserted at proper position in queue  Shortest job (1st in queue) runs to completion  In general, SJF is often used in long-term scheduling  Advantages: provably optimal w.r.t. average waiting time Disadvantages: Unimplementable at the level of short-term CPUscheduling.Also, starvation is possible!  Can do it approximately: use exponential averaging to predict

length of next CPU burst==> pick shortest predicted burst next! 8 CS-2006 Operating Systems

Determining Length of Next CPU Burst

 Can only estimate the length

 Can be done by using the length of previous CPU bursts, using exponential averaging

th

1. actual lenght of CPU burst t n

n



th

2. predicted value for the next (i.e., 1) CPU burst



n



1

 

n

th

3. predicted value for the CPU burst



n



n

4. , 0 1  

 

5. Define :

(1 ) . n 1 n n   t    

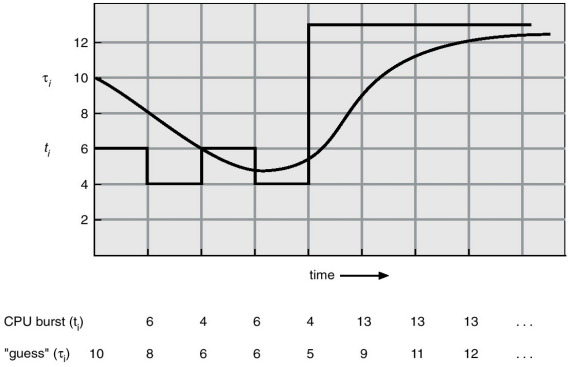
 = 0 implies making no use of recent history ( n+1 n) = 1 implies n+1 = tn(past prediction not used)  = 1/2 implies weighted (older bursts get less and less weight)

9 CS-2006 Operating Systems

Prediction of the Length of the Next CPU Burst

This figure is for

  0.5 and 0 10

10 CS-2006 Operating Systems

 A priority number (integer) is associated with each process  Priority can be internally computed (e.g., may involve time limits, memoryusage) or externally (e.g., user type, funds being paid)  In SJF, priority is simply the predicted next CPU burst time  The CPU is allocated to the process with the highest priority(smallest integer might mean highest priority)  Starvation is a problem, where low priority processes may never

3. Priority Scheduling  Solution: as time progresses, the priority of the long waiting(starved) processes is increased.This is called aging execute

11 CS-2006 Operating Systems

Priority scheduling

 Priority scheduling can be Preemptive or Non Preemptive

 When a process arrives and enters the Ready Queue Its priority is compared with the currently Running Process

 If Higher

 Preemptive Scheduling

 Run the New process

 Non-Preemptive Scheduling

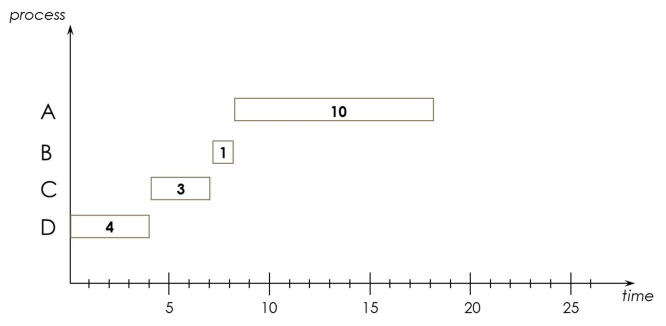
 Continue running the process

CS-2006 Operating Systems

12

Priority Scheduling

process priority service turnaround waitingtime tstime tttime tw

A 4 10 18 8B 3 1 8 7C 2 3 7 4D 1 4 4 0

AVERAGE 9.25 4.75

13 CS-2006 Operating Systems

 RR reduces the penalty that short jobs suffer with FCFSby preempting running jobs periodically  Each process gets a small unit of CPU time (timequantum), usually 10-100 milliseconds

 The CPU blocks the current job when its reserved timequantum (time-slice) is exhausted

4. Round Robin (RR)

 The current job is then put at the end of the ready queue if ithas not yet completed

 If the current job is completed, it will exit the system(terminate)

14 CS-2006 Operating Systems

 If there are n processes in the ready queue and the timequantum is q, then each process gets 1/n of the CPUtime in chunks of at most q time units at once. Noprocess waits more than (n -1)q time units  Performance: the critical issue with the RR policy is thelength of the quantum q

Round Robin (RR)

 q is large: RR will behave like FIFO and hence interactiveprocesses will suffer  q is small: the CPU will be spending more time on context

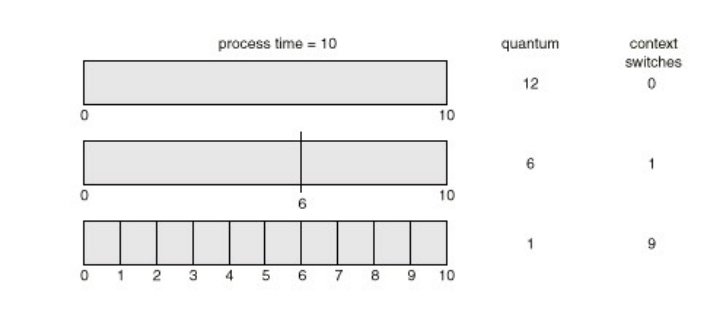
 q must be large with respect to context switch, otherwise overhead is too

switching

high

15 CS-2006 Operating Systems

Time Quantum and Context Switch Time

16 CS-2006 Operating Systems

Process Burst Time

Example of RR with Time Quantum = 20

0 20 37 57 77 97 117 121 134 154 162

 Typically, h~~igher average turnaround than S~~JF but betterresponse

P153

P217

P368

P424

 The Gantt chart is:

P1 P2 P3 P4 P1 P3 P4 P1 P3 P3

17 CS-2006 Operating Systems

1 2 3 4 5 199 200

Round Robin’s Disadvantage • A~~vg completion time~~?

 Good for Varying sized jobs

 But what about same-sized jobs?

 Assume 2 jobs of time =100 each:

CPU A B A B A A BA Btime

• (200 + 200) / 2 = 200

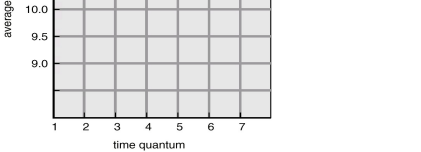
• How does this compare with FCFS for same two jobs? • (100 + 200) / 2 = 150

CS-2006 Operating Systems

18

Turnaround Time Varies With The Time Quantum

 Increasing the time quantum does not necessarily improve the average turnaround time!



19 CS-2006 Operating Systems

5. Multilevel Queue Scheduling

 Ready queue is partitioned into separate queues:  For example, foreground (interactive) and background (batch) Each queue has its own scheduling algorithm:  Foreground – RR

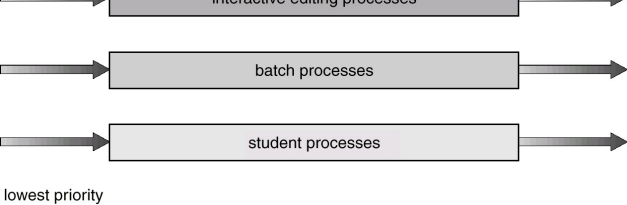
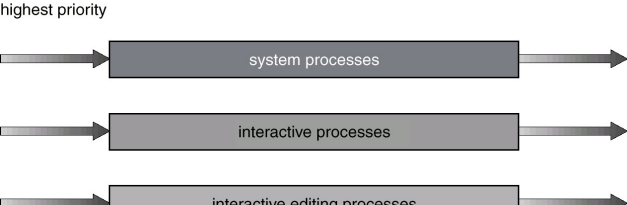
 Background – FCFS

 Scheduling must be done between the queues  Fixed priority scheduling; (i.e., serve all from foreground then from background)

 Possibility of starvation

 Time slice – each queue gets a certain amount of CPU time which it can schedule amongst its processes

 i.e., 80% to foreground in RR and 20% to background in FCFS 20 CS-2006 Operating Systems

Multilevel Queue Scheduling 21 CS-2006 Operating Systems

 A process can move between the various queues; aging can beimplemented this way  Multilevel-feedback-queue scheduler defined by the following number of queues

 scheduling algorithms for each queue

 method used to determine when to upgrade a process

6. Multilevel Feedback Queue

 method used to determine when to demote a process  method used to determine which queue a process will enter when that process needs service

parameters:

22 CS-2006 Operating Systems

Multilevel Feedback Queue Scheduling

 Example

 If a process used too much CPU time, then move it to a lower priority queue

 If a process waits too long in a lower priority queue, then move it to a higher priority queue

CS-2006 Operating Systems

23

 Three queues:

 Q0 – time quantum 8 milliseconds

 Q1 – time quantum 16 milliseconds

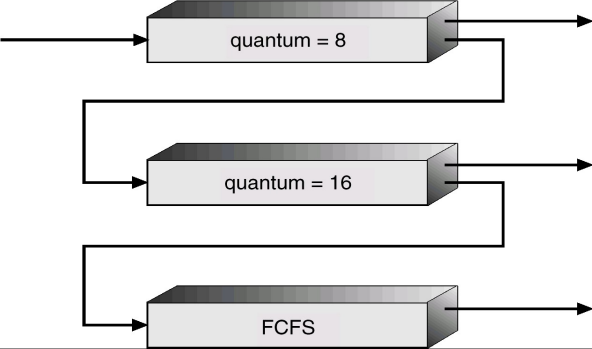
 A new job enters queue Q0 which is served FCFS. When it gainsCPU, the job receives 8 milliseconds. If it does not finish in 8milliseconds, the job is moved to queue Q1

Multilevel Feedback Queue  At Q1, the job is again served FCFS and receives 16 additionalmilliseconds. If it still does not complete, it is preempted andmoved to queue Q2

 Q2 – FCFS

 Scheduling

24 CS-2006 Operating Systems

Multilevel Feedback Queue 25 CS-2006 Operating Systems

• At t=0 there are three CPU bound processes (no I/O) in the system. units for P3. The system has three RR queues with the following time slices: 1 for queue1, 2 for queue 2, and 4 for queue3. RR- 1 Unit of • Gantt Chart

MLFQ Example

• Average Completion Time

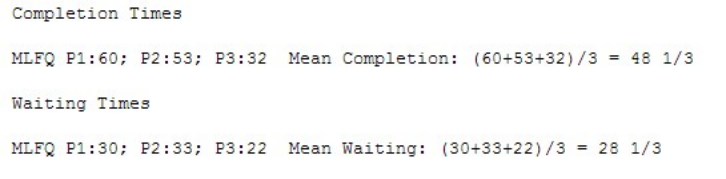
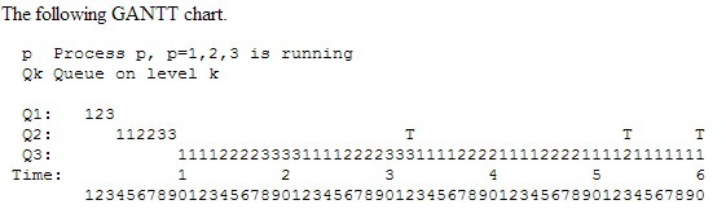
• Average Waiting Time

The CPU burst lengths are: 30 units for P1, 20 units for P2, and 10 time slice

CS-2006 Operating Systems

26

Solution:



CS-2006 Operating Systems

27

Summary of CPU Scheduling Algorithms

 First-Come, First-Served (FCFS) Scheduling  Shortest-Job-First (SJF) Scheduling  Nonpreemptive

 Preemptive or Shortest Remaining Time First (SRTF)  Priority Scheduling

 Preemptive

 Nonpreemptive

 Round Robin (RR)

 Multilevel Queue Scheduling

 Multilevel Feedback Queue

28 CS-2006 Operating Systems

References

 Operating System Concepts (Silberschatz, 9th edition) Chapter 5

29 CS-2006 Operating Systems