CPU – Scheduling-Part 2

CS3600

Spring 2022

FCFS,SJF, SRT Solution

Process	Arrival time	Burst Time
P1	0	8
P2	1	4
P3	2	9
P4	3	5

Using the formula calculate the turn around time and wait time, Compare with FCFS, SJF,SRT

Time ->

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	TT=CT-AT	Avg TT	AVG WT
Burst Time	**FCFS**																														
8	P1	P1																											8-0=8		
4	P2		P2																										12-1=11		
9	P3			P3																									21-2=19	(8+11+19+23)/4	8.75
5	P4				P4																								26-3=23	=15.25	
	SJF																														
8	P1	P1																											8-0=8		
4	P2		P2																										12-1=11		
9	P3			P3																									26-2=24	(8+11+24+14)/4	7.75
5	P4				P4																								17-3=14	=14.25	
	SRT																														
8	P1	P1																											17-0=17		
4	P2		P2																										4-1=3		
9	P3			P3																									26-2=24	(17+3+24+6)/4=	6
5	P4				P4																								9-3=6	12.5	

- Arrival Time (AT)
- Burst Time (BT)
- Completion Time (CT)

- Turn Around Time (TT) = CT –AT
- Waiting Time (WT) = TT –BT

Performance of the algorithms

- Objective:
  - Maximize the number of processes completed per unit of time.
  - Minimise waiting Time.
- Compare algorithms
  - We take average Turn Around Time (TT) for each Algorithm .
- Another important objective is to guarantee fairness.
- ***Starvation*** is the indefinite postponement of a process while other processes are allowed to proceed. Both SJF and SRT can lead to starvation.

Assumptions and Metrics

- Assumptions
  - All jobs only use the CPU (i.e., they perform no I/O)
  - The run-time of each job is known.

- Performance Metrics
  - Turnaround Time
  - Response Time

Turn Around Time (TT) = CT –AT
Waiting Time (WT) = TT –BT
Response Time (RT) = First Instance of CPU -AT

Improve Response Time

- Time slice for each process

- Design Considerations
  - Amortization
  - Response time

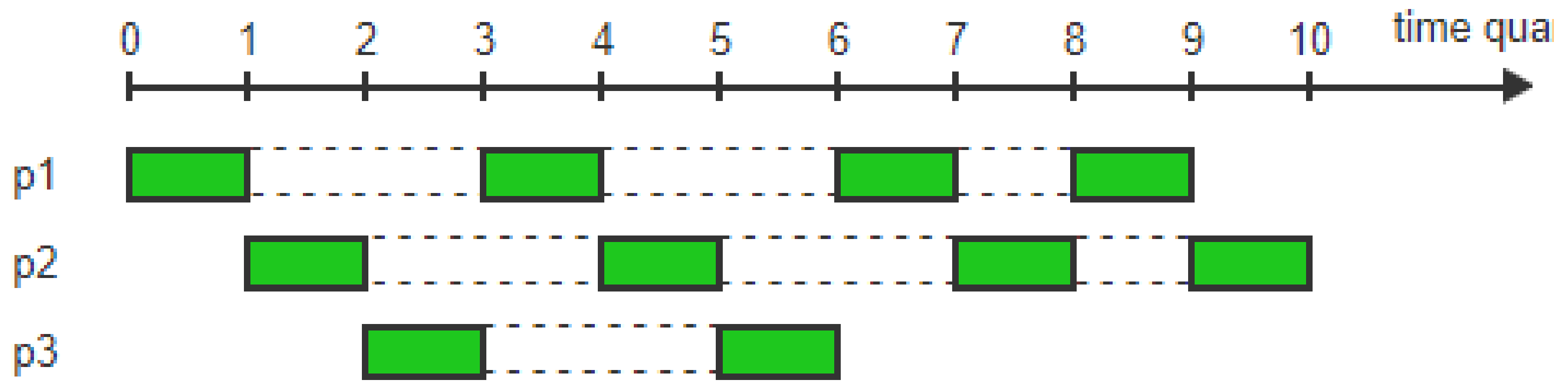
RR scheduling algorithm

- A ***time quantum***, Q, is a small amount of time (typically 10 to 100 milliseconds) during which a process is allowed to use the CPU.

- The ***round-robin (RR) algorithm*** uses a single queue of processes.
  - The priority is determined solely by a process's position within the queue. The process at the head of the queue has the highest priority and is allowed to run for Q time units.
  - When Q ends, the process is moved to the tail of the queue and the next process now at the head of the queue is allowed to run for Q time units.

RR scheduling algorithm

- Three processes p1,p2,p3 with total CPU times of 4, 4, and 2 start executing under RR.

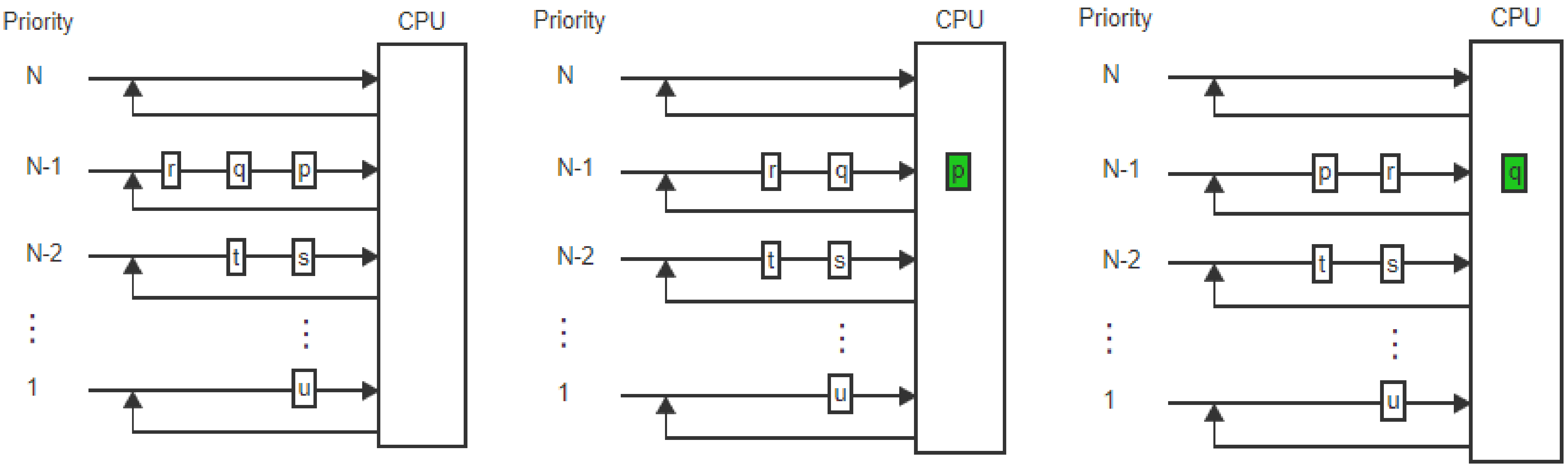


Scheduling of Interactive processes

- Interactive processes must time-share the CPU using pre-emptive scheduling to give each process a chance to make progress in a timely manner.

Multilevel scheduling algorithm

- *Multilevel (ML) scheduling* maintains a separate queue of processes at each priority level.
  - Within each level, processes are scheduled using RR.



Multilevel Feedback (MLF) scheduling algorithm

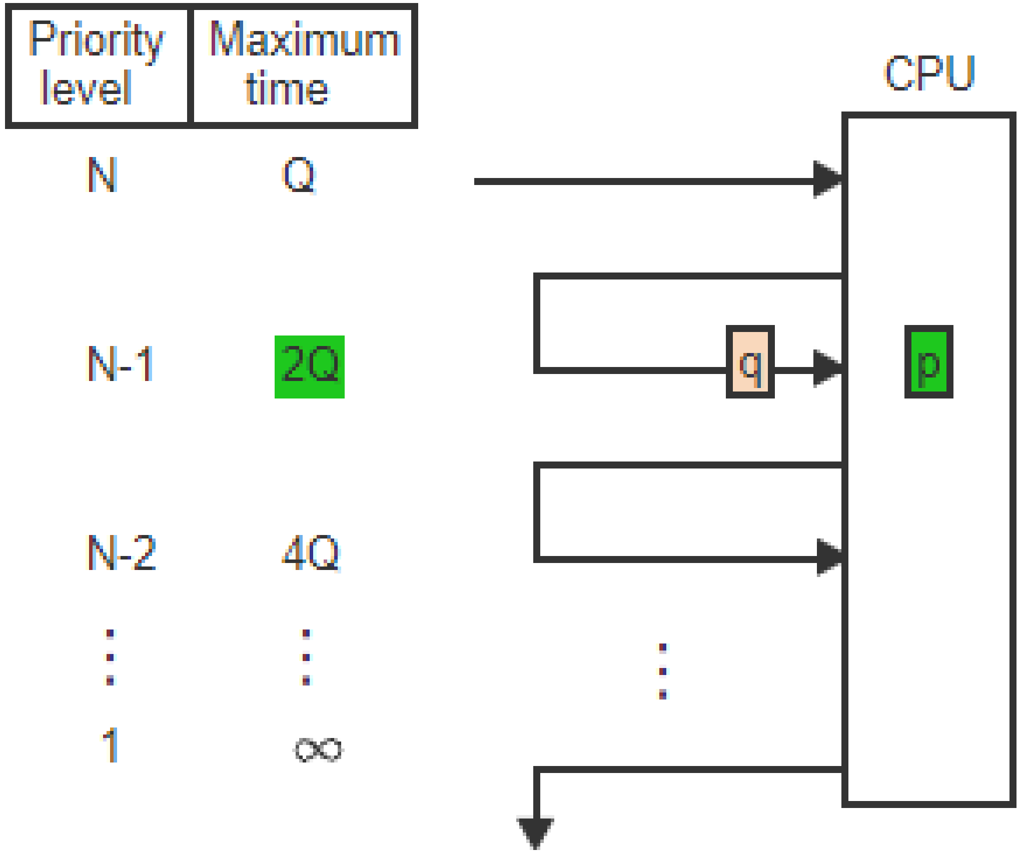
- Multilevel feedback scheduling is similar to ML but addresses the problems of starvation and fairness by:
  - using a different time quantum at each priority level
  - changing the priority of every process dynamically

MLF

- Under ***MLF algorithm*** a newly arriving process enters the highest-priority queue, N, and is allowed to run for Q time units.
- When Q is exceeded, the process is moved to the next lower priority queue, N-1, and is allowed to run for 2Q time units.
- The quantum size is doubled with each decreasing priority level. Thus, at priority level L, the maximum time allowed is $2^{(16-L)}Q$ time units.

MLF

MLF maintains N priority levels. Starting with a quantum Q at level N, the maximum time allowed at each level doubles with each lower level.



Worksheet 3

- Q1)Two processes, p1 and p2 arrive at time 0 and start executing using RR scheduling. (p1 starts before p2) The total CPU time of p1 is 30-time units, and p2 is 50. The quantum is Q = 10. The context switching time is S = 0.Find Turnaround Time(TT) of P1,P2 and Average Turnaround Time(ATT).

- Q2) Two processes, p1 and p2 are executing using RR scheduling. The context switching time is S = 5.
  - Determine the maximum quantum size Q such that the gap between the end of a process pi's quantum and the start of pi's next quantum does not exceed M = 30-time units.
  - Determine the percentage of CPU time wasted on context switching.

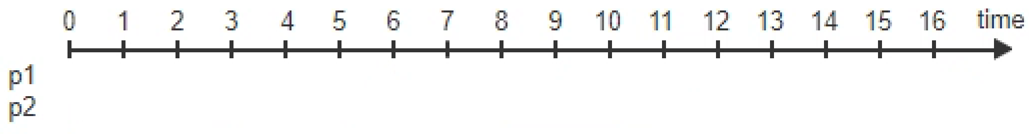
Scheduling of real-time processes

- A ***real-time process*** is characterized by continual input, which must be processed fast enough to generate nearly instantaneous output.
- Each arriving input item is subject to a deadline.
  - Example: Streaming of audio or video, processing and displaying radar data, control of robots or of fly-by-wire aircraft.

- A ***period*** is a time interval (typically in milliseconds or even microseconds) within which each input item must be processed. The end of each period is the implicit deadline for processing the current item.

RM scheduling algorithm

- The ***rate monotonic (RM)*** algorithm schedules processes according to the period. The shorter the period, the higher the priority.



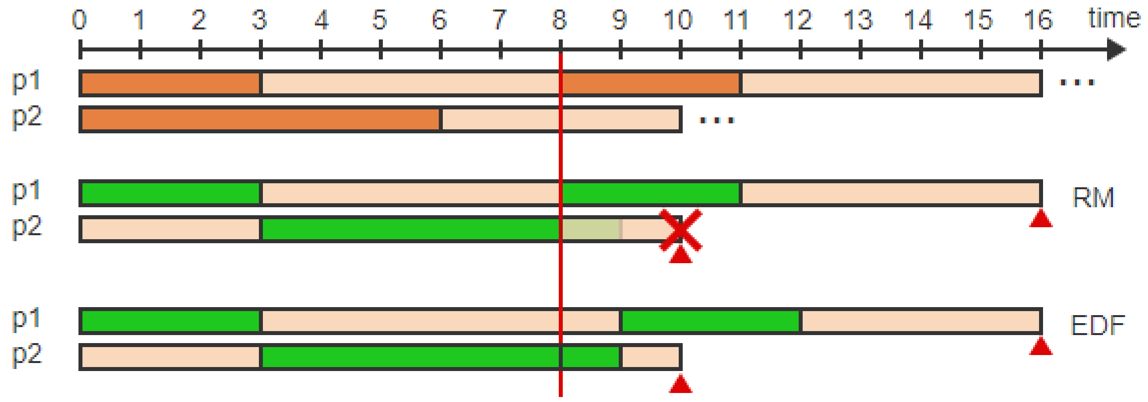
Process	Period	Tot. CPU
p1	4	1
p2	5	3

Two periodic processes. p1 and p2, with the given periods and total CPU times, are scheduled using RM.

EDF Scheduling algorithm

- The ***earliest deadline first (EDF)*** algorithm schedules processes according to the shortest remaining time until the deadline. The shorter the remaining time, the higher the priority.

Process	Period	Tot. CPU
p1	8	3
p2	10	6



Performance of the algorithms

- A schedule is ***feasible*** if the deadlines of all processes can be met.
- The fraction of CPU time used by a process i, is $\frac{Ti}{Di}$ where Ti is the total CPU time and Di is the period of process i.
- ***CPU utilization (U)*** is the sum of the individual fractions of CPU times used by each process.

- If U = 1 then the CPU is utilized 100%. A feasible schedule exists as long as U ≤ 1. No schedule is feasible if U > 1, since the total CPU time of all processes would exceed 100%.

- EDF is an optimal algorithm in that a feasible schedule is always guaranteed if U ≤ 1.
- RM U is less than approximately 0.7 (empirical evidence)

Worksheet Q3

- Three periodic processes with the following characteristics are to be scheduled: (D is the period and T is the total CPU time).
- Determine if a feasible schedule exists.

Determine how many more processes, each with T = 3 and D = 20, can run concurrently under EDF.

	D	T
p1	20	5
p2	100	10
p3	120	42

Worksheet 03 - Q4

- Do it as a team, only single document has to be submitted from 1 team in Canvas with team members names .

  - Submit the solution in Worksheet 03_classwork in Canvas

To-do (02/17/22)

- Weekly Quiz
- Complete Worksheet 03
- HW3 based on scheduling algorithms