Lecture 3: Process Management – Part 2

Barnaby Martin

`barnaby.d.martin@dur.ac.uk`

First Come, First Served (FCFS)

Process	Priority	Arrival Time	Burst Time
A	–	3	7
B	–	6	3
C	–	0	5
D	–	5	4
E	–	4	1

C			A	E	D	B													
C	C	C	C	C	A	A	A	A	A	A	A	E	D	D	D	D	B	B	B
			A	A	E	E	E	E	E	E	E	D	B	B	B	B			
				E	D	D	D	D	D	D	D	B							
						B	B	B	B	B	B								

First Come, First Served (FCFS)

- The average waiting time may or may not be lengthy!
- A simple algorithm to implement.
- May result in a ‘convoy’ effect, for example short processes waiting on a long process to finish.

Shortest-Job-First (SJF)

- Compares each process waiting to determine the length of its next CPU burst.
- Use these lengths to schedule the process with the shortest time.
  - If two processes have the same length then FCFS.
- Non pre-emptive.
  - Once the CPU is given to the process it cannot be pre-empted until the process has completed its CPU burst.

Shortest-Job-First (SJF)

Process	Priority	Arrival Time	Burst Time
A	–	3	7
B	–	6	3
C	–	0	5
D	–	5	4
E	–	4	1

C			A	E	D	B													
C	C	C	C	C	E	B	B	B	D	D	D	D	A	A	A	A	A	A	A
			A	A	A	A	A	A	A	A	A	A							
				E	D	D	D	D											

Shortest-Job-First (SJF)

- The average waiting time?
- A simple algorithm to implement?
- Long processes waiting on a short process to finish. Worse?

Round Robin (RR)

Each process gets a small unit of CPU time: a time quantum or time slice usually $q = 10$ to 100 milliseconds.

After this time has elapsed, the process is pre-empted 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 $\frac{1}{n}$ of the CPU time in chunks of at most q time units at once.

Round Robin (RR)

Process	Priority	Arrival Time	Burst Time
A	1	3	7
B	3	6	3
C	5	0	5
D	2	5	4
E	4	4	1

C			A	E	D	B													
C	C	C	C	A	A	E	C	D	D	A	A	B	B	D	D	A	A	B	A
			A	E	E	C	D	A	A	B	B	D	D	A	A	B	B	A	
				C	C	D	A	B	B	D	D	A	A	B	B				
					D	A	B												
						B													

Round Robin (RR)

Round Robin (with priority) operates with a data structure that is a queue using a set time-slice. At the time-step that a process arrives it is already placed at the back of the queue for consideration to be running. A process runs either until it finishes or its time-slice has elapsed. When a process has not finished but is pre-empted due to the elapsing of its time-slice, then it is sent to the back of the queue. Where processes both compete for the back of the queue, the one with the higher priority goes first. The next process to be chosen by the scheduler is always from the front of the queue.

Round Robin (RR)

At each time-step where after a process is finished or pre-empted the order to proceed is:

1. incoming processes and pre-empted process (if exists) compete for back of queue (which could also be front of queue if queue is just one process).
2. process is selected from front of queue.

Shortest Remaining Time First (SRTF)

If a new process arrives in the ready queue with a CPU burst length less than the remaining time of executing process then pre-empt the running process and run the process with the shorter CPU burst length.

It can be viewed as a pre-emptive version of SJF.

Shortest Remaining Time First (SRTF)

Process	Priority	Arrival Time	Burst Time
A	–	3	7
B	–	6	3
C	–	0	5
D	–	5	4
E	–	4	1

C			A	E	D	B													
C	C	C	C	C	E	B	B	B	D	D	D	D	A	A	A	A	A	A	A
			A	A	A	A	A	A	A	A	A	A							
				E	D	D	D	D											

Shortest Remaining Time First (SRTF)

Process	Priority	Arrival Time	Burst Time
A	–	3	1
B	–	6	3
C	–	0	5
D	–	5	4
E	–	4	7

C			A	E	D	B													
C	C	C	A	C	C	B	B	B	D	D	D	D	E	E	E	E	E	E	E
			C	E	E	E	E	E	E	E	E	E							
					D	D	D	D											

Shortest Remaining Time First (SRTF)

- The average waiting time?
- A simple algorithm to implement?
- Short processes vs long process?

Priority Scheduling

In **Priority Scheduling**, the algorithm associates a priority number (integer) for each process.

The CPU is allocated by the dispatcher to the process with the highest priority (**smallest integer = highest priority**).

Problem: Starvation (also known as indefinite blocking).The low priority processes may never execute.

Solution: Ageing, as time progresses increase the priority of the process.

Priority based CPU scheduling can be either pre-emptive or non pre-emptive.

Round Robin (with priority)

With **priority in Round Robin**, we can choose to implement either

- a priority queue, or
- a normal queue where the priority is used only when competing for entry onto the queue.

We will choose the **latter**!

Multilevel queue scheduling

Multilevel queue scheduling is where processes are partitioned into different queues, e.g. foreground v. background processes.

The approach is based on the different queues having different (performance) requirements e.g. response time.

Problem: provides differentiation but not flexible, since a process remains in the same queue regardless of adapting requirements.
Feedback queues An adaption of multilevel queue scheduling, however the process may proceed from one queue to the next. This is the most common approach to scheduling but also difficult to implement (or more precisely: get right).

Multilevel Queue

The **ready** queue is partitioned into separate queues:

- *e.g. foreground (interactive) and background (batch)*

Each queue has its own scheduling algorithm:

- *e.g. foreground is RR and background is FCFS*

Scheduling must be undertaken between the queues:

- Fixed priority scheduling,
  e.g. serve all from foreground then from background
- Time slice: each queue gets a certain amount of CPU time which it can schedule amongst its processes,
  e.g. 80% to foreground in RR, 20% to background in FCFS

Multilevel Feedback Queue

A process can **move** between the various queues: **priority ageing** may also be used.

A multilevel feedback queue scheduler may use the following parameters:

- The number of queues.
- The scheduling algorithms for each queue.
- The method used to determine when to upgrade a process.
- The method used to determine when to demote a process.
- The method used to determine which queue a process will enter.

Round Robin example on board, time slice 2

Process	Priority	Arrival Time	Burst Time
A	5	0	3
B	4	5	7
C	3	6	1
D	2	2	5
E	1	7	4

A		D			B	C	E												
A	A	D	D	A	D	D	B	B	C	E	E	D	B	B	E	E	B	B	B
		A	A	D	B	B	C	C	E	D	D	B	E	E	B	B			
						C	E	E	D	B	B	E							
							D	D	B										

Round Robin example on board, time slice 3

Process	Priority	Arrival Time	Burst Time
A	5	0	8
B	4	2	4
C	2	4	2
D	3	6	5
E	1	9	1

A		B		C		D			E										
A	A	A	B	B	B	A	A	A	C	C	D	D	D	B	E	A	A	D	D
		B	A	A	A	C	C	C	D	D	B	B	B	E	A	D	D		
				C	C	D	D	D	B	B	E	E	E	A	D				
						B	B	B	E	E	A	A	A	D					
									A	A									