CSC227: Operating Systems
Spring 2017

PART 3

CPU SCHEDULING ALGORITHMS

Dr. Soha S. Zaghloul

LAYOUT

- CPU Scheduling Algorithms
  - First-Come, First-Served (FCFS)
  - Shortest-Job First (SJF)
  - Shortest-Remaining Time First (SRTF)

First-Come First-Served (FCFS)

The process that requests the CPU first (ie. Comes to the Ready Queue first), is allocated the CPU first.

FCFS is implemented as follows:

- When a new process arrives to the Ready Queue, it is appended to the tail.
- When the CPU is free, it is allocated to the process at the head of the queue.
- The selected process is removed from the Ready Queue and allocated to the CPU.
- The algorithm continues until there is no process in the Ready Queue.

FCFS has the following properties:

- Non-preemptive.
- The resulting waiting time depends on the order of the processes in the Ready Q.

The main advantage of FCFS lies in its simplicity.

The disadvantages of FCFS may be summarized in the following:

- The resulting waiting time is often long.
- Unsuitable for time-sharing systems since it is not guaranteed that each user gets its share fairly (since it is non-preemptive)
- A convoy effect may occur in case there is a long CPU-burst process with other relatively short CPU-burst ones. This is explained in more details in the next slide.

First-Come First-Served (FCFS) – The Convoy Effect

The convoy effect occurs when there is an unstable mixture of CPU- and I/O- bound processes.

Consider the following example:

- Assume there are 10 I/O-bound processes &1 CPU-bound process in the Ready Q.
- The I/O-bound processes, when assigned the CPU, will leave the CPU quickly.
- The I/O-bound processes will be in the Wait status at the I/O device.
- During this time, the CPU-bound process is assigned to the CPU.
- While the CPU-bound process is running, the I/O-bound processes complete the I/O operation, and are re-appended to the Ready Queue waiting for the CPU.
- Clearly the I/O device is idle during this time, and the competition is high on the CPU.

The convoy effect means an uneven distribution of processes that compete for the CPU from one side, and the I/O device from the other side.

Therefore, at one time it is found that most processes are competing for the I/O device whereas the CPU is idle.

Suddenly, at a subsequent time most processes that were competing for the I/O device complete the I/O operation and are re-appended to the Ready Queue.

Now, most processes compete for the CPU whereas the I/O device is idle.

This phenomenon is known as the convoy effect.

FCFS – EXAMPLES

Given:

Process Number	Burst Time (in ms)
P1	24
P2	3
P3	3

Result:

Waiting Time
0
24
27

Average Waiting Time
= (0 + 24 + 27)/3 = 17 ms

Required:

Calculate the average waiting time using the Gantt Chart.

Solution:

Given:

Process Number	Burst Time (in ms)
P2	3
P3	3
P1	24

Result:

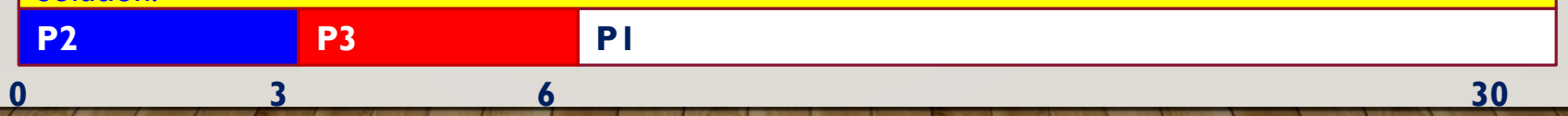
Waiting Time
0
3
6

Average Waiting Time
= (0 + 3 + 6)/3 = 3 ms

Required:

Calculate the average waiting time using the Gantt Chart.

Solution:



SHORTEST-JOB FIRST (SJF)

The CPU is assigned to the process that has the smallest CPU burst.

SJF is implemented as follows:

- When the CPU is idle, the process with the least **next** CPU burst is assigned to the processor.
- If two processes have equal next CPU burst, then the FCFS algorithm is applied.
- Note that in SJB, the order of the processes in the Ready Queue is irrelevant.
- Moreover, the arrival time of the processes to the Ready Queue is also irrelevant.

SJF has the following properties:

- Non-preemptive.
- Often used in long-term schedulers.

The advantages of SJF may be summarized as follows:

- Provides smaller average waiting time as compared to the FCFS.
- In general, it is proved to be optimal.

The disadvantages of SJF may be summarized as follows:

- It is difficult to know the length of the next burst of a process.
- Therefore, it is predicted based on historical data of the same process.

SHORTEST-JOB FIRST (SJF) – EXAMPLE

Given:

Process Number	Next Burst Time (in ms)
P1	6
P2	8
P3	7
P4	3

Result:

Waiting Time
3
16
9
0

Average Waiting Time
= (3 + 16 + 9 + 0)/4
= 7 ms

Required:

Calculate the average waiting time using the Gantt Chart.

Solution:

PREDICTION OF THE NEXT CPU BURST (1)

The value of the next CPU burst is calculated from the following formula:

$$\zeta_{n+1} = \alpha\, t_n + (1 - \alpha)\, \zeta_n$$

- n is the burst number
- ζ_{n+1} is the value of the next burst
- t_n stores the most recent history (ie. most recent burst value)
- ζ_n stores the past history (ie. burst value of the past history)
- α is a constant between 0 and 1. It controls the relative weight of the recent and past history.

The constant α designates the following:

- If $\alpha > 0.5$ ➔ more weight is given to the recent history.
  - This is used when the history is assumed old and irrelevant.
- If $\alpha < 0.5$ ➔ more weight is given to the past history.
  - This is used when the current conditions are assumed to be transient.
- If $\alpha = 0.5$ ➔ equal weight is given to both the recent and past history.
  - This is the most common case.

PREDICTION OF THE NEXT CPU BURST (2)

The previously mentioned formula calculates the current burst in terms of the previous one:

$$\zeta_{n+1} = \alpha\, t_n + (1 - \alpha)\, \zeta_n$$

For example:

- ζ_3 is calculated in terms of ζ_2
- ζ_2 is calculated in terms of ζ_1
- ζ_1 is calculated in terms of ζ_0
- ζ_0 is known as the initial burst. It is given an assumed value.

PREDICTION OF THE NEXT CPU BURST – EXAMPLE

Calculate the 5th burst of a process, given that its initial burst is equal to 5 ms, and the recent history is given a weight of 0.4.

Solution: The formula is $\zeta_{n+1} = \alpha\, t_n + (1 - \alpha)\, \zeta_n$

- ζ_0 = 5 ms *(given)*
- ζ_1 = 0.4 * ζ_0 = 0.4 * 5 = 2 ms
- ζ_2 = 0.4 * ζ_1 + 0.6 ζ_0 = 0.4 * 2 + 0.6 * 5 = 3.8 ms
- ζ_3 = 0.4 * ζ_2 + 0.6 ζ_1 = 0.4 * 3.8 + 0.6 * 2 = 2.72 ms
- ζ_4 = 0.4 * ζ_3 + 0.6 ζ_2 = 0.4 * 2.72 + 0.6 * 3.8 = 3.368 ms
- ζ_5 = 0.4 * ζ_4 + 0.6 ζ_3 = 0.4 * 3.368 + 0.6 * 2.72 = 3 ms

SHORTEST-REMAINING TIME FIRST (SRTF)

SRTF is the preemptive version of the SJF.

SRTF is implemented as follows:

- As soon as a new process *Pi* is appended to the Ready Queue, its next burst time is compared to the remaining time of the currently running process *Pj*.
- If the burst time of *Pi* is smaller than that of *Pj*, then *Pj* is preempted and returned to the Ready Queue.
- The CPU is then assigned to *Pi*.
- Therefore, it is important to record the arrival times of all processes.
- The arrival time of a process is the time at which it is appended to the Ready Q.

SRTF has the following properties:

- Preemptive.
- Often used in long-term schedulers.

The advantage of SRTF is that it is proved to be generally optimal.

The disadvantages of SRTF may be summarized as follows:

- Frequent context switching – due to preemption – causes an overhead that cannot be ignored when evaluating the scheduling algorithm.
- Like SJF, the value of the next CPU burst should be calculated empirically.

SRTF – EXAMPLE

Given:

Process Number	Arrival Time	Next Burst Time (in ms)
P1	0	8
P2	1	4
P3	2	9
P4	3	5

Solution:

Time	Remaining Time				
	P1	P2	P3	P4	Minimum
0	8	NA	NA	NA	P1
1	7	4	NA	NA	P2
2	7	3	9	NA	P2
3	7	2	9	5	P2
4	7	1	9	5	P2
5	7	*	9	5	P4
6	7	*	9	4	P4
7	7	*	9	3	P4
8	7	*	9	2	P4
9	7	*	9	1	P4
10	7	*	9	*	P1

and so on. The final Gantt chart is shown below:

SRTF – EXAMPLE (CONT'D)

To calculate the average waiting time, consider the arrival time and the Gantt chart.

Given:

Process Number	Arrival Time
P1	*0*
P2	*1*
P3	*2*
P4	*3*

The waiting time of each process is calculated as the difference between the starting time (time at which the process starts to run) and the arrival time (as given).

Waiting Time:

P1 = 10 - 1 = 9 ms
P2 = 1 – 1 = 0 ms
P3 = 17 – 2 = 15 ms
P4 = 5 – 3 = 2 ms
Average waiting time= (9 + 0 + 15 + 2)/4 = 6.5 ms

Comparing this result with the non-preemptive algorithm SJF ➔ the SJF gives an average waiting time of 7.75 ms.

If a process is preempted once (or more) then its waiting time is the sum of times the process is waiting in the Ready Queue.