Department of Computer Science & Engineering, IIT Guwahati

CS 343 - Operating Systems: Quiz #1 (28.08.2022)

Each question carries 6 marks. 5x6=30 marks, Duration 90 minutes.

Write necessary steps used for solving. Answers without justification will get 0 marks.

1. In a system there are two types of processes: type A processes and type B processes. All processes of type A execute the same code, and all processes of type B execute the same code. The code for each process type is shown below. Here, X and Y are general semaphores. X is initialized to 2, and Y is initialized to 0. Suppose three processes of type A and two processes of type B are brought into execution simultaneously. Once a process starts, assume that there is no context switching before its completion.

Process A	Process B
Wait (X); Signal (Y);	Wait (Y); Wait (Y); Signal (X); Signal (Y);

Consider the following two independent order of execution of the process instances. Is it possible for processes to finish in the order given. If so, show an execution sequence showing the values of semaphores X and Y in every step of each process instance. If not, explain why?

(a) Case-1: Order of process AABAB

(b) Case-2: Order of process AABBA

Solution:

(a) Execution order AABAB is possible. [1 mark for Yes/No + 2 marks for illustration]
The following is an execution sequence showing that AABAB is possible:

A_1	A_2	B_1	A_3	B_2	X	Y
					2	0
Wait(X)					1	0
Signal(Y)					1	1
	Wait(X)				0	1
	Signal(Y)				0	2
		Wait(Y)			0	1
		Wait(Y)			0	0
		Signal(X)			1	0
		Signal(Y)			1	1
			Wait(X)		0	1
			Signal(Y)		0	2
				Wait(Y)	0	1
				Wait(Y)	0	0
				Signal(X)	1	0
				Signal(Y)	1	1

(b) Execution order AABBA is NOT possible. [1 mark for Yes/No + 2 marks for illustration]
The following is an execution sequence showing that AABBA is not possible:

A_1	A_2	B_1	B_2	A_3	X	Y
					2	0
Wait(X)					1	0
Signal(Y)					1	1
	Wait(X)				0	1
	Signal(Y)				0	2
		Wait(Y)			0	1
		Wait(Y)			0	0
		Signal(X)			1	0
		Signal(Y)			1	1
			Wait(Y)		1	0
			Wait(Y)		1	*
			Signal(X)			
			Signal(Y)			

The sequence AABBA is impossible. From the above execution sequence, after **A_1**, **A2** and **B_1** semaphores X and Y have values 1 and 1, respectively. While executing **B_2**, after first Wait (Y), semaphores X and Y have values 1 and 0, respectively. Hence the second Wait(Y) get blocked on semaphore Y. So **B_2** and A_3 cannot complete. As a result, the order of AABBA is impossible.

2. Consider a self-driving car whose automated driving mechanism is implemented by two processes defined as follows.
Control () – [CPU burst time = 60 seconds] Analyse output by LookUp () and adjust the speed of the car based on front road conditions.
LookUp () – [CPU burst time = 15 seconds] Capture of high resolution images of the front road terrain and store the captured images for the Control () to analyse further and take action.
The Control () and LookUp () are triggered in every 100 seconds and 15 seconds, respectively. Here triggering means a new instance of the respective process is created and is in ready state. Assume the first triggering of Control () and LookUp () occurs at T=0 and T=20, respectively. The design permits only one process to be in the running state based on pre-emptive shortest job first scheduling. The main objective of the system is to make the car move in a safe way by adjusting the speed as per front road conditions.
(a) Comment on the efficiency of the system in meeting its objectives.
(b) Keeping both CPU burst time of the processes and CPU scheduling algorithm unmodified, suggest your recommendations to improve the efficiency of the system

Solution:

C : Control(), L : LookUp().

C	L	L	L	L
20	35	50	65	80

(a) This is NOT an efficient system in meeting its objectives. Because as per the Gantt Chart, using preemptive shortest remaining time first scheduling algorithm, L will preempt C at T=35 seconds and thereafter as an when one L completes the new instance of L will execute. C will never get a change to run. As a result the Control() function is not getting invoked thereby missing the defined objectives. [3 marks]

(b) To improve the efficiency the C and L should run in interleaved fashion. For each completed L, the C should act upon the output of L. This can be done by adjusting the periodicity of the triggering time, such that C and L are interleaved. Since the burst time of C is 60 seconds and L is 15 seconds, there should be only one C and L triggering in a window of 75 seconds. Triggering after a large time interval will increase waiting time. [3 marks]

Example: C at 0, 75, 150, 225.. and L at 60, 135, 210, 285…

C	L	C	L	C	L
0	60	75	135	150	225

3. Consider a ready queue which gets a new process at regular intervals as per the following pattern. New process of CPU burst 5 cycles is added in every clock cycle X (X is a multiple of 10) and a new process of CPU burst 7 cycles is added in every clock cycle Y (Y is a multiple of 11). First process enters ready Q at clock cycle 10 and moves to running state at the same cycle itself. FCFS scheduling is used. Neglect overhead of context switching. What is the average waiting time for first 24 processes that enters the ready queue?

Solution:

Let Ai and Bi be the process having the burst time 5 and 7.

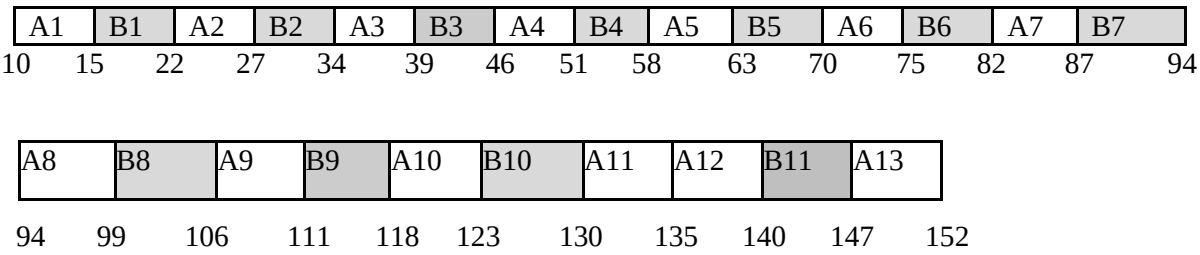
The first 24 processes that arrives in the ready Q consists of 13 A processes and 11 B processes.

At the T= 110 we can choose between A or B process since the arrival time of A and B will be same. Accordingly, any one of the following cases is valid.

Arrival of Ai = 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130 [13 processes] [2 marks]

Arrival of Bi = 11, 22, 33, 44, 55, 66, 77, 88, 99,110,121 [11 processes]

Case 1: If B10 is chosen before A11 [2 marks]



Total Waiting Time of Ai= 0+2+4+6+8+10+12+14+16+18+20+15+17=142

Total Waiting Time of Bi=4+5+6+7+8+9+10+11+12+13+19=104

Average Waiting Time=(142+104)/24=10.25 [2 marks]

Case 2 : If A11 is chosen before B10 [2 marks]

Total Waiting Time of Ai= 0+2+4+6+8+10+12+14+16+18+13+15+17=135

Total Waiting Time of Bi=4+5+6+7+8+9+10+11+12+18+19=104

Average Waiting Time=(135+109)/24=10.167 [2 marks]

4. Consider an operating system that uses dynamic priority based pre-emptive scheduling. There are 4 user process A, B, C and D with arrival time T= 0, 10, 20 and 30, respectively. The CPU burst of these processes are 15, 20, 10 and 10 cycles, respectively and they all have a priority value 2 at arrival time. The process is eligible for considering for running state only if its priority value reaches 0. The priority value of a process will be decremented by 1, if it is in waiting state for 5 consecutive cycles. Priority values of a waiting state process will not go to negative and will stay at 0 till it is promoted to running state. Similarly, a process that is in running state for 5 consecutive cycles will get its priority value incremented by 1 thereby moving to waiting state again. If any tie occurs for picking a process for running state then the winner is that process with least balance CPU burst and still if a tie exists then the winner is chosen based on FCFS. When no user process is eligible to be in the running state, OS process called docker will be running.
   (a) At time T=40, what is the balance CPU burst and priority value of each process that is not yet completed?
   (b) What are the turnaround times for processes A and C?
   (c) What is the fraction of OS docker process in running state in a window of 0 to T cycles, where T is the time by which all the four user process have completed its CPU burst?

Solution:

Priority Variation Chart with time & Gantt Chart [3 marks]

Time	Process A [Burst/Priority]	Process B [Burst/Priority]	Process C [Burst/Priority]	Process D [Burst/Priority]
0	15 / 2			
5	15 / 1			
10	15 / 0 *	20 / 2		
15	10 / 1	20 / 1		
20	10 / 0 *	20 / 0	10 / 2	
25	5 / 1	20 / 0 *	10 / 1	
30	5 / 0 *	15 / 1	10 / 0	10 / 2
35	0 / completes	15 / 0	10 / 0 *	10 / 1
40		15 / 0	5 / 1	10 / 0 *
45		15 / 0	5 / 0 *	5 / 1
50		15 / 0	0 / completes	5 / 0 *
55		15 / 0 *		0 / completes
60		10 / 1		
65		10 / 0 *		
70		5 / 1		
75		5 / 0 *		
80		0 / completes		

DO= Docker

Gantt Chart:

DO	DO	PA	DO	PA	PB	PA	PC	PD	PC	PD	PB	DO	PB	DO	PB

0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80

(a) [1.5 marks]

Remaining Process	PB	PC	PD
Priority	0	1	0
Balance Burst time	15	5	10

(b) Turnaround Time of (Process A) = time of completion – arrival = 35-0 =35 [1 mark]
Turnaround Time of (Process C) = time of completion – arrival = 50-20 =30

(c) T = 80, Docker Time = 25, Fraction = 25/80 = 5/16 = 0.3125 [0.5 mark]

5. Consider 5 identical process P1, P2,..P5 that have to be scheduled on a processor. These processes can be context switched at any arbitrary time and in any order. The code of each process Pi, i=1,2...5, is given below. P and Q are binary semaphores initialized to 1 and 0, respectively. X is shared variable initialized to 0. Assume the operation specified on each line in the following code is atomic and context switching can happen only after completion of the current line of instruction. How many unique output patterns can get printed (based on possible context switch order) once all the process complete their execution? List all the patterns. Explain the logic used in finding the answer.

```
Process Pi {
    Wait (P);
    X=X +1;
    Print “B”;
    If (X==5) {
            Print “C”;
            Signal (Q);
    }
    Signal (P);
    Wait (Q);
    Print “A”;
    Signal (Q);
    }
```

Solution:

Number of unique pattern possible: 1 [1 mark]
Pattern is: BBBBBCAAAAA [2 marks]
No process will complete Wait (Q) until some process makes Signal(Q) inside the If condition. This can only happen when X=5 (X=X+1 has to executed 5 times). Hence each process will print B once. Hence B will be printed 5 times in total. The fifth process will have If condition true and Signal (Q) is done. Then each process will Print A once.
Hence BBBBBCAAAAA is the only possible pattern [3 marks]