Aviad Hahami
302188347

Exercise 5 – Scheduling

Part 1 (36 points)

The following processes are arriving at the *ready queue* in the following order:

	Arrival Time	Burst Time
P1	0	2
P2	1	6
P3	2	5
P4	3	3
P5	4	5

(Times are in milliseconds)

Assume context switch time is negligible. Fill in the following table with the times for each algorithm:

Algorithm	Average Waiting Time	Average Turnaround Time	Average Response Time*
FCFS	5.8	10.2	5.8
SJF**	4.8	9	4.8
PSJF**	4.6	8.8	4
RR* (q=2)**	9.2	13.4	2

* We define response time as the time from arrival to the first time the process was given some CPU time

** When calculating SJF/PSJF, try to minimize number of context switches when comparing two processes with the same (remaining) burst time

*** When calculating round robin, maintain a FIFO queue of processes waiting for CPU time. Add arriving processes to the end of this queue

Aviad Hahami
302188347

Calculation tables for Part 1

FCFS

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1	X	X																			
P2			X	X	X	X	X	X													
P3									X	X	X	X	X								
P4														X	X	X					
P5																	X	X	X	X	X

Total waiting time: 0+1+6+10+12=29 Average: 5.8
Total turnaround time: 2+7+11+13+18=51 Average: 10.2
Total response time: 0+1+6+10+12=29 Average: 5.8

SJF

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1	X	X																			
P2																X	X	X	X	X	X
P3			X	X	X	X	X														
P4								X	X	X											
P5											X	X	X	X	X						

Total waiting time: 0+14+0+4+6=24 Average: 4.8
Total turnaround time: 2+20+5+7+11=45 Average: 9
Total response time: 0+14+0+4+6=24 Average: 4.8

PSJF

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1	X	X																			
P2																X	X	X	X	X	X
P3			X				X	X	X	X											
P4				X	X	X															
P5											X	X	X	X	X						

Total waiting time: 0+14+3+0+6 = 23 Average: 4.6
Total turnaround time: 2+20+8+3+11 = 44 Average: 8.8
Total response time: 0+14+0+0+6 = 20 Average: 4

RR (q=2)

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1	X	X																			
P2			X	X							X	X						X	X		
P3					X	X							X	X						X	
P4							X	X							X						
P5									X	X						X	X				X

Total waiting time: 0+12+13+9+12=46 Average: 9.2
Total turnaround time: 2+18+18+12+17=67 Average: 13.4
Total response time: 0+1+2+3+4=10 Average: 2

Aviad Hahami
302188347

IDC Herzliya 20/05/2015

Part 2 (20 points)

For each of the following statements, answer whether it is true or false and **explain!**

A. RR is a preemptive scheduling algorithm

True / ~~False~~

Round Robin is defined as a preemptive scheduling algorithm. If a process didn't complete his job during his time he will be paused and moved to end of queue

B. PSJF will always yield better average waiting times than SJF

~~True~~/ **False**

Consider the following example:

Task	Arrival time	Burst time
P1	0	2
P2	1	3
P3	2	4
P4	3	5
P5	4	6

In this case both PSJF and SJF will yield the same Avg. waiting time (6ms)

C. Starvation may occur in SJF

True / ~~False~~

Since SJF prioritizes processes by their estimated burst time, there might be a process with maximal burst time that will be prioritized last regardless his arrival time.

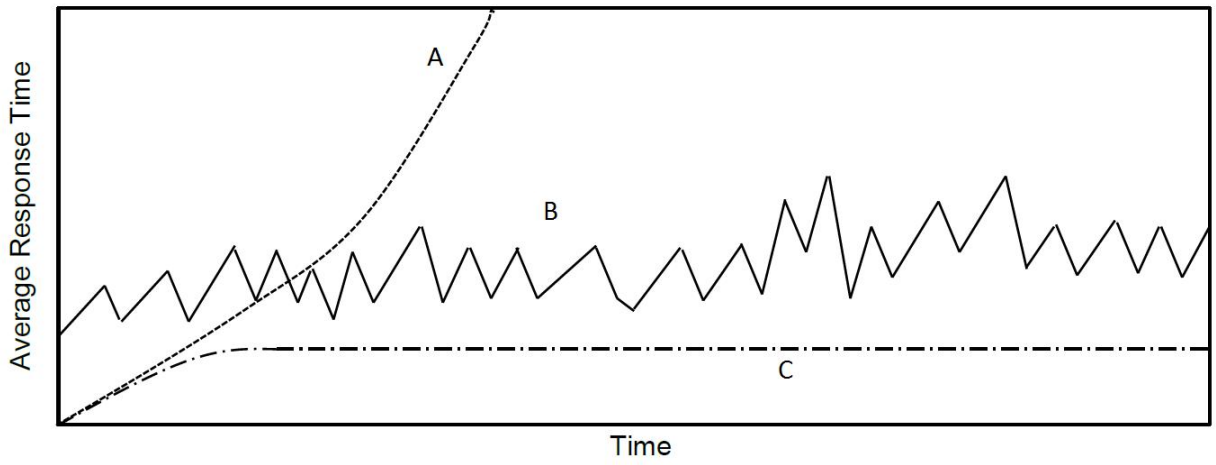
D. RR is a good scheduling algorithm for user interface applications

True / ~~False~~

As RR is described, we "jump" between processes quickly, hence minimizing response time. Whenever we deal with UI or Real time systems, we will prefer RR as our scheduler since it updates the processes most frequently therefore keeping the user (or system) up to date.

Part 3 (24 points)

The graph below represents a test with three process scheduling algorithms: PSJF, FIFO and RR. The X axis represents time and the Y axis represents the average response time of the processes that were queued.



In the questions below, fill in the scheduling algorithm used and explain why you think that was the one.

1. A was using:
   PSJF. This curve could represent starved processes as we can see that the avg. response time keep increasing.

2. B was using:
   FIFO. The curve shows us that every process waits for his previous to end, without having constant time as an average, thus creating "edges" in the graph.
3. C was using:
   RR. As it is shown in our previous calculations, RR have the lowest response time thus resulting in almost constant (and linear) response time from some point X on the Time axis.

Part 4 (20 points)

Prove formally or disprove using a detailed counterexample:
(In this part you must provide a complete formal proof or a complete and clear counterexample. Other explanations will not be accepted)

Round Robin satisfies starvation freedom

Let $p_1, p_2 \dots p_n$ be processes.
It is known that RR switches between existing p_i's according to some final timespan q (quantum) and in a circular way.
Assuming some process was paused by RR and moved to the queue's end, we can say that after $n * q$ time units we will return to the same location in the processes circle.
By this property it is implied that there is no process p_i that is not being processed due to algorithm prioritization.
Concluding that we can say that this algorithm satisfies the starvation freedom property $Q.E.D$