Bronson Edralin
EE 468
Homework 4
9/24/14

Problem 6.11 a and b (1 pt)
Discuss how the following pairs of scheduling criteria conflict in certain settings.

a) *CPU utilization and response time*
b) *Average turnaround time and maximum waiting time*

a) CPU utilization is maximizing while response time is minimizing. While utilization is high, response time won't be as fast. CPU utilization means keeping the CPU busy as possible by giving it more processes to execute. A low response time is something desired inside an interactive system that is waiting for the user.
b) Turnaround time is the sum of the periods spent waiting to get into memory, waiting in the ready queue, executing on the CPU and doing I/O. Waiting time is the sum of the periods spent waiting in the ready queue. It is highly desirable to have both times minimized. However, CPU scheduling can only minimize the waiting time.

Problem 6.15 (1 pt)
A variation of the round-robin scheduler is the regressive round-robin scheduler. This scheduler assigns each process a time quantum and a priority. The initial value of a time quantum is 50 milliseconds. However, every time a process has been allocated the CPU and uses its entire time quantum (does not block for I/O), 10 milliseconds is added to its time quantum, and its priority level is boosted. (The time quantum for a process can be increased to a maximum of 100 milliseconds.) When a process blocks before using its entire time quantum, its time quantum is reduced by 5 milliseconds, but its priority remains the same. What type of process (CPU-bound or I/O-bound) does the regressive round-robin scheduler favor? Explain.
It's definitely CPU-bound because the algorithm says to keep adding 10ms if the process hasn't finished in the given quantum time. If all process are I/O-bound, the ready queue will almost always be empty, and the short-term scheduler will have little to do. If all processes are CPU bound, the n I/O waiting queue will almost always be empty, devices will go unused and again the system will be unbalanced.

Problem 6.16 (2 pts)
Consider the following set of processes, with the length of the CPU burst given in milliseconds:

Process	*Burst Time*	*Priority*
P1	*2*	*2*
P2	*1*	*1*
P3	*8*	*4*
P4	*4*	*2*
P5	*5*	*3*

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.

a) *Draw four Gantt charts that illustrated the execution of these processes using the following scheduling algorithms: FCFS, SJF, nonpreemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2).*
b) *What is the turnaround time of each process for each of the scheduling algorithms in part a?*
c) *What is the waiting time of each process for each of these scheduling algorithms?*
d) *Which of the algorithms results in the minimum average waiting time (over all processes)?*

a)

First Come First Serve (FCFS)				
P1	P2	P3	P4	P5
Shortest Job First (SJF)				
P2	P1	P4	P5	P3
Nonpreemptive priority				
P3	P5	P1	P4	P2

For Round Robin (RR)….The highlighted is when process is being executed. The burst time is subtracted by 2 quantum.

RR	Initial	Rnd 1	Rnd 2	Rnd 3	Rnd 4	Rnd 5	Rnd 6	Rnd 7	Rnd 8	Rnd 9	Rnd 10	Rnd 11
Quant:		2	1	2	2	2	2	2	2	2	1	2
P1:	2	0	0	0	0	0	0	0	0	0	0	0
P2:	1	1	0	0	0	0	0	0	0	0	0	0
P3:	8	8	8	6	6	6	4	4	4	2	2	0
P4:	4	4	4	4	2	2	2	0	0	0	0	0
P5:	5	5	5	5	5	3	3	3	1	1	0	0

b) Turnaround time:

First Come First Serve (FCFS)				
P1=2	P2=3	P3=11	P4=15	P5=20
Shortest Job First (SJF)				
P1=3	P2=1	P3=20	P4=7	P5=12
Nonpreemptive priority				
P1=15	P2=20	P3=8	P4=19	P5=13
Round Robin (RR)				
P1=2	P2=3	P3=20	P4=13	P5=18

c) Waiting time:

First Come First Serve (FCFS)				
P1=0	P2=2	P3=3	P4=11	P5=15
Shortest Job First (SJF)				
P1=1	P2=0	P3=12	P4=3	P5=7
Nonpreemptive priority				
P1=13	P2=19	P3=0	P4=15	P5=8
Round Robin (RR)				
P1=0	P2=2	P3=12	P4=9	P5=13

d) SJF had the winning minimum Avg Waiting Time!!!

	Avg Waiting Time
First Come First Serve (FCFS)	6.2
Shortest Job First (SJF)	4.6
Nonpreemptive priority	11
Round Robin (RR)	7.2

Problem 6.17 (2 pts)

The following processes are being scheduled using a preemptive, round- robin scheduling algorithm. Each process is assigned a numerical priority, with a higher number indicating a higher relative priority. In addition to the processes listed below, the system also has an ***idle***

Thread	*Priority*	*Burst*	*Arrival*
P1	*40*	*20*	*0*
P2	*30*	*25*	*25*
P3	*30*	*25*	*30*
P4	*35*	*15*	*60*
P5	*5*	*10*	*100*
P6	*10*	*10*	*105*

task *(which consumes no CPU resources and is identified as P_{idle}). This task has priority 0 and is scheduled whenever the system has no other available processes to run. The length of a time quantum is 10 units. If a process is preempted by a higher-priority process, the preempted process is placed at the end of the queue.*

a) Show the scheduling order of the processes using a Gantt chart.
b) What is the turnaround time for each process?
c) What is the waiting time for each process?
d) What is the CPU utilization rate?

a) Scheduling order of processes…

RR	**Init**	**Rd 1**	**Rd 2**	**Rd 3**	**Rd 4**	**Rd 5**	**Rd 6**	**Rd 7**	**Rd 8**	**Rd 9**	**Rd 10**	**Rd 11**	**Rd 12**	**Rd 13**	**Rd 14**	**Rd 15**
Time:		0-10	10-20	20-25	25-35	35-45	45-55	55-60	60-70	70-75	75-80	80-90	90100	100-105	105-110	110-115
Quant:		10	10	5	10	10	10	5	10	5	5	10	10	5	5	5
P1:	20	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P2:	25	25	25	25	15	15	5	5	5	5	0	0	0	0	0	0
P3:	25	25	25	25	25	15	15	10	10	10	10	0	0	0	0	0
P4:	15	15	15	15	15	15	15	15	5	0	0	0	0	0	0	0
P5:	10	10	10	10	10	10	10	10	10	10	10	10	10	5	5	0
P6:	10	10	10	10	10	10	10	10	10	10	10	10	10	0	0	0

b) Turnaround time….
P1= 20ms, P2=55ms, P3=60ms, P4=15ms, P5=20ms, P6=10ms

c) Waiting time…
P1=0s, P2=30ms, P3=30ms, P4=0s, P5=10ms, P6=0s

d) CPU utilization rate = $\frac{105ms}{115ms} = 91.3\%$

Problem 6.19 (1 pt)
Which of the following scheduling algorithms could result in starvation?

a) *First-come, first-served*
b) *Shortest job first*
c) *Round robin*
d) *Priority*

a) The CPU's resources are allocated to the processes that requests the resources first. Every process gets a chance.
b) The processes with the smallest burst times get allocated to the CPU first. If there are too many processes with small burst times, then other processes with longer burst times may starve.
c) CPU's resources are allocated to each processes in a circular-like time queue. The time queue may consist of 1 quantum which may be from 10ms to 100ms. Every process gets a chance.
d) The process with the highest priority is processed first. If too many processes with higher priority keep popping up, the lower priority queues may starve.

Problem 6.20 (1 pt)
Consider a variant of the RR scheduling algorithm in which the entries in the ready queue are pointers to the PCBs.

a) *What would be the effect of putting two pointers to the same process in the ready queue?*
b) *What would be two major advantages and two disadvantages of this scheme?*
c) *How would you modify the basic RR algorithm to achieve the same effect without the duplicate pointers?*

a) By doing this, the process have increased its priority. With two pointers in the same process in the ready queue and with time quantum t, each process gets 50% of CPU time.
b) Advantage is that you can give higher priority to more important jobs. The consequence is that shorter jobs will suffer. Could result in starvation.
c) Give processes, that have more priority, more time to finish. This means that priority processes may have access to more than 1 quantum.

Problem 6.22 (1 pt)
Consider a system implementing multilevel queue scheduling. What strategy can a computer user employ to maximize the amount of CPU time allocated to the user's process?

If program uses only a large fraction of its time quantum and not entire time quantum, it may increase priority for that process. Program could maximize its CPU time allocated to it by not fully utilizing its time quantums. If users can label their processes so the processes may enter the correct queues, which have higher priority and use Round Robin algorithms, the users processes will go through faster.

Problem 6.30 (1 pt)
Under what circumstances is rate-monotonic scheduling inferior to earliest-deadline-first scheduling in meeting the deadlines associated with processes?

Earliest-deadline-first scheduling will give priority to process according to their deadlines. Rate-monotonic scheduling will cuts off when a higher priority process arrive.

Problem 6.31 (1 pt)
Consider two processes, P_1 and P_2, where $p_1 = 50$, $t_1 = 25$, $p_2 = 75$, and $t_2 = 30$.

a) *Can these two processes be scheduled using rate-monotonic scheduling? Illustrate your answer using* *a Gantt chart such as the ones in Figure 6.16–Figure 6.19.*

b) Illustrate the scheduling of these two processes using earliest- deadline-first (EDF) scheduling.

a) No, because either process will meet the deadline with either one having the higher priority.
b) This one can work but you have to use context switching a lot. P1 must go first so it may finished first and then P2 will finish.