Eddie Gurnee
COSC 592
Operating Systems

Chapter 6 Reading Questions:

6.11) 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
c) I/O device utilization and CPU utilization

Response:

a) CPU utilization can be increased if context-switching occurs less frequently; the overhead required for context-switching contributes nothing to CPU utility. However if context-switching were to occur less frequently there would be an increase in the response times in processes.
b) The average turnaround time can be minimized by implementing a shortest-job-first scheduling algorithm and completing the shortest tasks available, doing that however could starve some of the longer running tasks which would increase the maximum waiting time.
c) Maximum utilization of I/O devices can be achieved by prioritizing I/O jobs so that they execute as soon as they are ready to run. This execution prioritization results in frequent context-switching which of course decreases the CPU utilization.

6.14) Consider the exponential average formula used to predict the length of the next CPU burst. What are the implications of assigning the following values to the parameters used by the algorithm?

a) $\alpha = 0$ and $\tau_0 = 100$ milliseconds
b) $\alpha = 0.99$ and $\tau_0 = 10$ milliseconds

Response: The exponential average formula is given to be:

$$\tau_{n+1} = \alpha \cdot t_n + (1 - \alpha) \cdot \tau_n$$

With a α value of 0 as presented in part (a), the formula will always predict the length of the next CPU burst the be the same as the current length; the time would always be predicted to be 100 milliseconds. The α-value differentiates the value of recent history and past history; a value of 0.99 (very close to 1) places very high importance on the most recent history and the formula would predict values to be very close to the current CPU burst time (i.e. τ_1 would be 10 milliseconds (okay, bad example but you know what I mean)).

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

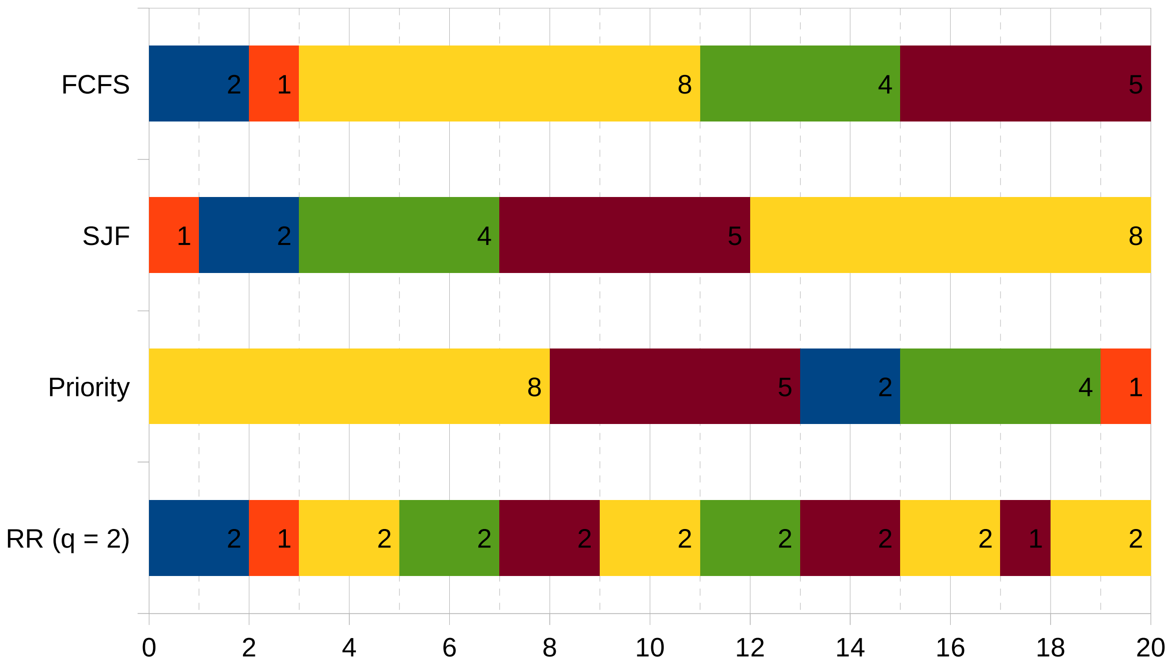
Process	**Burst Time**	**Priority**
P_1	2	2
P_2	1	1
P_3	8	4
P_4	4	2
P_5	5	3

The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5, all at time 0.

a) Draw four Gantt charts that illustrate 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)?

Response:
a)



b(x) is the turnaround time for process x
c(x) is the waiting time for process x

	FCFS		SJF		Priority		RR (q = 2)	
x	b(x)	c(x)	b(x)	c(x)	b(x)	c(x)	b(x)	c(x)
P_1	2	0	3	1	15	13	2	0
P_2	3	2	1	0	20	19	20	2
P_3	11	3	7	3	8	0	3	12
P_4	15	11	12	7	19	15	13	9
P_5	20	15	20	12	13	8	18	13
Average	10.2	6.2	8.6	4.6	15	11	11.2	7.2

d) The algorithm with the minimum average wait-time is the shortest job first algorithm.

6.21) Consider a system running ten I/O-bound tasks and one CPU-bound task. Assume that the I/O-bound tasks issue an I/O operation once for every millisecond of CPU computing and that each I/O operation takes 10 milliseconds to complete. Also assume that the context-switching overhead is 0.1 millisecond and that all processes are long-running tasks. Describe the CPU utilization for a round-robin scheduler when:

a) The time quantum is 1 millisecond

b) The time quantum is 10 milliseconds

Response:
With ten I/O-bound tasks each taking one millisecond, and one CPU-bound I/O operation taking ten milliseconds, you'd experience 20 milliseconds of actual time spent on tasks.

- a) At quantum of one millisecond, every millisecond spent on a task will be swapped if another task remains undone, so: 20 / (20 * 1.1) ⇒ $90.\overline{90}$% CPU utilization
- b) At quantum of ten milliseconds, each of the ten I/O-bound tasks would incur the context-switching overhead, but the I/O operation would be able to complete before switching, therefore: 20 / (10 * 1.1 + 10.1) ⇒ ~94.7867% CPU utilization

6.23) Consider a preemptive priority scheduling algorithm based on dynamically changing priorities. Larger priority numbers imply higher priority. When a process is waiting for the CPU (in the ready queue, but not running), its priority changes at a rate α. When it is running, its priority changes at a rate β. All processes are given a priority of 0 when they enter the ready queue. The parameters α and β can be set to give many different scheduling algorithms.

a) What is the algorithm that results from $\beta > \alpha > 0$?
b) What is the algorithm that results from $\alpha < \beta < 0$?

Response:

a) First come, first serve – look at part b and swap some of the words around
    { swap(α, β); swap(lowers, raises); swap(higher, lower); swap(running, waiting); }
b) Last in, first out – the α rate lowers the priority for the waiting tasks faster than the running priority rate β, leading to the most recently added (starting at a value of 0) to have higher priority than those that have been there the longest