Caleb Faruki
Dr. Art Werschulz
CISC3595 Operating Systems
November 4, 2013

Chapter 5 Homework

5.10

a. CPU utilization increases if the context-switching overhead is minimized. Aggregate overhead for context-switching can be minimized by making switches less frequently. This could increase process response times, however.
b. You can minimize average turnaround time by executing the shortest processes first. But this causes starvation for longer processes and thereby increases the running time of these longer processes. This means longer waiting times for longer processes, especially when short processes are being constantly submitted.
c. Occurs when a CPU-bound process holds the CPU and many I/O-bound processes finish I/O. I/O-bound processes wait in the ready queue until the CPU-bound process is finished. During that time, I/O devices will idle. This is called the convoy effect.

5.11

a. No α means that the predicted CPU burst length will always be 100 milliseconds.
b. If α is 1, then the predicted CPU burst length is just just slightly smaller than the length of the previous CPU burst.

5.12

a. see attached photo.
b.

	FCFS	SJF	NPP	RR
P1	10q	19	16	19
P2	11	1	1	2
P3	13	4	18	7
P4	14	2	19	4
P5	19	9	6	14

c.

	FCFS	SJF	NPP	RR
P1	0	9	6	9
P2	10	0	0	1
P3	11	2	16	5
P4	13	1	18	3
P5	14	4	1	9

d. SJF (3.2 milliseconds)

5.13

b. Shorest Job First
d. Priority. (i.e.: NON-PREEMPTIVE PRIORITY)

5.14

a. The process would be executed twice as many times.
b. PROS: You don't have to reimplement the RR scheme. Also, pointers allow the prioritization of more urgent processes.
   CONS: Context-switching might take slightly longer. And removing processes from the ready queue is harder, since the processes are not stored in contiguous memory.
c. You could set the quantum time for processes on an individual basis. More important processes would get larger quanta and minor processes get smaller quanta.

5.15 (note that CPU utilization is a percentage)

a. Processes use 1 quantum per turn and I/O processes perform their I/O-bound operations in time to return to the CPU for their turn. Context-switch overhead occurs outside of CPU utilization time.

Where p = # of processes, q = quantum time (ms), h = overhead (ms),

$$C = CPU\,utilization = p \cdot q = 1 \cdot 11 = 11\,ms$$

$$Overhead_{total} = C \cdot h = 11 \cdot 0.1\text{ms} = 1.1\text{ms}$$

$$Time_{total} = 11\text{ms} + 1.1\text{ms} = 12.1\text{ms}$$

$$\frac{(CPU\,utilization)}{Time_{total}} = \frac{11\text{ms}}{12.1\text{ms}} = 0.909 = \mathbf{91}\,\boldsymbol{percent}$$

b.

$$Time_{total} = 1.1\text{ms} \cdot 10 + 0.1 = 21.1\text{ms}$$

$$C_{CPU-bound} = 10\text{ms},\quad C_{I/O-bound} = 1\text{ms} + context\ switch$$

$$C_{total} = 1 \cdot 10 + 10 = 20\text{ms}\ \frac{C_{total}}{Time_{total}} = 20/21.1 = 0.9478 = \mathbf{95}\,\boldsymbol{percent}$$

5.18

a. FCFS – short processes have disproportionate waiting times, since they may have to wait for long processes to execute before getting CPU time.
b. RR – short processes favored equally since CPU time is cycled among all processes, preventing starvation and short processes terminate first.
c. Multilevel feedback queue – distributes CPU time similar to RR algorithm, meaning that short jobs are executed favorably with minimal waiting time.