COP4600 Homework 3

Due Tuesday, October 1st by midnight

1. As we discussed in class the operating system uses the system clock interrupts to wake up periodically and check whether it is time to schedule another process when round-robin algorithm is used. Let the system clock frequency of a system denoted by f, the quantum value that is used by the Round-robin algorithm denoted by q, and the interrupt handling overhead of the kernel denoted by i.

   Do you think the operating system can make sure that a process runs at most q amount of time when it is scheduled on the processor based on the Round-robin algorithm? Explain.

   If your answer is no, give an upper bound for the actual amount of time a process may use the CPU in terms of q, f, and i.

2. For each of the following performance goals explain what type of processes should be given the highest scheduling priority:

   - Shorter response time (improving interactivity of applications)
   - High throughput
   - Increasing resource utilization

   In your answers feel free to use your own heuristics and provide specifics of the characterizations of the processes.

3. Answer the following:

   (a) What does a relocating loader do?

   (b) Is relocation an alternative technique to using boundary (base and limit) registers? Explain.

4. Assume that a computer has a 32K physical memory (RAM, main memory) and processes running on that system can use a virtual address space of size 64K. The page size is 4K.

   Let (p, pf) denote the mapping of virtual page p to page frame pf. A process's virtual pages that are currently in the main memory have the following mappings: $\{(0,2),(1,1),(2,6),(3,0),(4,4),(5,3),(9,5),(11,7)\}$.

For each of the following virtual addresses i) explain whether referring to that virtual address in an instruction would cause a page fault or not, ii) if your answer to part i) is no, give the corresponding physical address and specify the page no, page frame no, and the offset (displacement):

(a) Virtual address = 31460

(b) Virtual address = 17400

(c) Virtual address = 45100

(d) Virtual address = 25000

5. Assume that the virtual to physical memory mapping for Process 1 and Process 2 are shown in Tables 1 and 2, respectively. Draw the equivalent inverted page table that uses hashing with chaining. Assume that $x\%4$ is used as the hash function, where x is the virtual page no. In each node of a linked list per inverted table entry you should store the following information: processID, page no, page frame no.

Virtual page no	Page frame no	Present
0	3	1
1	2	0
2	0	1
3	1	0
4	2	0
5	3	0
6	3	0
7	0	0

Table 1: Process 1's page table

Virtual page no	Page frame no	Present
0	3	0
1	2	0
2	1	1
3	2	1
4	0	0
5	1	0
6	2	0
7	3	0

Table 2: Process 2's page table