



COMP 346 - Winter 2018
Theory Assignment 3
Due Date & Submission Format: *Please see course outline.*

Answer all questions

Question # 1

Answer the following questions:

- i. (a) What are relocatable programs? (b) What makes a program relocatable? (c) From the OS memory management context, why programs (processes) need to be relocatable?
- ii. What is (are) the advantage(s) and/or disadvantage(s) of small versus big page sizes?
- iii. What is (are) the advantage(s) of paging over segmentation?
- iv. What is (are) the advantage(s) of segmentation over paging?

Explain your answers.

Question # 2

Consider the below implementations of a semaphore's *wait* and *signal* operations:

```
wait () {
    Disable interrupts;
    sem.value--;
    if (sem.value < 0) {
        save_state (current) ; // current process
        State[current] = Blocked; //A gets blocked
        Enqueue(current, sem.queue);
        current = select_from_ready_queue();
        State[current] = Running;
        restore_state (current); //B starts running
    }
    Enable interrupts;
}

signal(){
    Disable interrupts;
    sem.value++;
    if (sem.value <= 0){
        k = Dequeue(sem.queue);
        State[k] = Ready;
        Enqueue (k, ReadyQueue);
    }
    Enable interrupts;
}
```

- a) What are the critical sections inside the *wait* and *signal* operations which are protected by disabling and enabling of interrupts?
- b) Give example of a specific execution scenario for the above code leading to inconsistency if the critical sections inside implementation of *wait()* and *signal()* are not protected (by disabling of interrupts).
- c) Suppose that process A calling semaphore *wait()* gets blocked and another process B is selected to run (refer to the above code). Since interrupts are enabled only at the completion of the wait operation, will B start executing with the interrupts disabled? Explain your answer.

Question # 3

Consider a demand-paged system where the page table for each process resides in main memory. In addition, there is a fast associative memory (also known as TLB which stands for Translation Look-aside Buffer) to speed up the translation process. Each single memory access takes 1 microsecond while each TLB access takes 0.2 microseconds. Assume that 2% of the page requests lead to page faults, while 98% are hits. On the average, page fault time is 20 milliseconds (includes everything: TLB/memory/disc access time and transfer, and any context switch overhead). Out of the 98% page hits, 80 % of the accesses are found in the TLB and the rest, 20%, are TLB misses. Calculate the effective memory access time for the system.

Question # 4

Consider the page reference string $R=\{0, 1, 2, 0, 1, 2, 0, 1, 2, 3, 6, 7, 6, 7, 0, 1, 2, 3, 4\}$ for a given process.

- (a) Show the memory representation of the pages using the LRU algorithm and an allocation of 3 frames. How many page faults are there?
- (b) Show the memory representation of the pages using the Belady Optimal algorithm and an allocation of 3 frames. How many page faults are there?
- (c) Show the memory representation of the pages using the working set model with a window size $\Delta=3$ (Δ indicates the maximum number allowed for a page to be in memory before being replaced; i.e. if a page is not used for 3 consecutive times, then it must either be used/demanded next, or it has to be removed). How many page faults are there?

Question # 5

Consider a system that would implement the page table on the CPU if feasible.

- (a) Give an advantage of this strategy.
- (b) Give a disadvantage of this strategy.

Question # 6

Explain (i) an advantage and (ii) a disadvantage that a global page replacement algorithm has over a local page replacement algorithm.

Question # 7

Consider a system that adjusts the degree of multiprogramming by monitoring the mean time between page faults (i.e. T_{pf}) and the mean time to service a page fault (i.e. T_{fs}). Describe the performance of the paging system in terms of the degree of multiprogramming when (i) T_{pf} is greater than T_{fs} , (ii) T_{pf} is less than T_{fs} and (iii) T_{pf} is equal to T_{fs} .

Question # 8

Some systems automatically open a file when it is referenced for the first time and close the file when the job terminates. Discuss the advantages and disadvantages of this scheme as compared to the more traditional one, where the user has to open and close the file explicitly.

Question # 9

a) What is the difference between preemptive and non-preemptive scheduling? Why is strict non-preemptive scheduling unlikely to be used in a computer system that provides both batch and timesharing service?

b) What is the trade-off used to select the quantum size, say, in pure Round-Robin scheduling?

Question # 10

What advantage is there in having different values of the scheduling quantum on different levels of a multilevel feedback queuing system? Your answer should consider all aspects such as fairness, starvation, efficiency, etc.

Question # 11

Consider the following set of prioritized processes, where a smaller priority value represents a higher priority.

Process	Service Time	Priority
0	20	3
1	15	1
2	21	3
3	7	5
4	12	2

Assume that all processes arrived at the same time, however they are inserted in the ready list in the order indicated in the above table.

a) Draw Gantt charts for the execution scenarios assuming:

- FCFS scheduling
- Non-preemptive SJF scheduling
- Non-preemptive priority scheduling
- Pure Round-Robin scheduling with the quantum = 3

b) What is the waiting time of each process in each case?

c) What is the response time of each process in each case?

d) What is the turn-around time of each process in each case?