Please answer the following questions briefly:

1. What is the driven factor of operating system development?
2. Name three abstractions that are provided in process management by modern operating systems?
3. What is multiprogramming?
4. What is the key difference between a process and a thread?
5. How do threads execute conceptually?
6. Why may an execution of a multi-thread program produce incorrect results?
7. What is an atomic operation?
8. What is a semaphore?
9. How does a user program request services provided by OS?
10. What is the three major components in an object file to be loaded into memory while creating a process?
11. Suppose a student writes a UNIX shell program,but instead of calling fork() then exec() to launch a new job,he instead inserts a subtle difference: the code first calls exec() and then calls fork() like the following:

Shell(...){

....

Exec(cmd,args);

Fork();

......

}

Does it work?What is the impact of this change to the shell,if any?(Explain)

1. Suppose a student implements semaphore class using lock and condition variables as follows:

Class Semaphore{

Int count;

Lock\* l;

Condition\_Variable\* c;

Public:

Semaphore(int n){

Count = n;

L = new Lock;;

C = new Condition\_Variable;

}

Void P(){

L->Acquire();

If(count == 0) c->Wait();

Count--;

L->Release();

}

Void V()

{

L->Acquire();

Count++;

C->Signal();

L->Release();

}

}

Does it work?(Explain)if not ,modify it to make it work.

13.given two processes in the READY state,one that is CPU-bound and one that is I/O bound,which process should be given a higher priority for running next(all other things being equal)?Justify your answer.