1. Suppose our OS does not have kernel threads, and the LOOKITUP thread of the ANSWER process needs to read a value form a file, so LOOKITUP will be waiting on this input operation.
    a. Then the other threads belonging to ANSWER will not also be waiting.

2. A critical section of code must be managed so while it is executing, no other process can execute it's critical section.

3. Interrupts are not always handled in the order of their occurrence.

For questions 4 -7 suppose (human)user writes this at the LINUX prompt:

Program1 | Program2

4. Program 1 can accept input from the user using the standard input

5. Program2 can ask the user for input using standard output but the user cannot provide it directly to Program2.

6. The system will create two processes and a buffer

7. Output from Program1 must be in a format that makes sense as input for program2.

8. The argument for the pipe() function is an array of 2 integers.

________short__________________Answers_____________________________________

8.. Suppose we are executing processes one after another not using time slices. We have these 4 process (arriving in order A B C D order) to schedule:

PROCESS	PRIORITY	NEXT CPU BURST
A	29	6 time units
B	7	21 time unit
C	22	13 time unit
D	18	27 time unit

For each of these scheduling methods, draw a gantt chard (a timeline) to illustrate the sequence of events and compute the average time each process spends waiting before it begins execution.

a. Shortest job first

b. Priority Scheduling (high priority numbers first)

c. First Come First Serve

9. Suppose we call fork() function
    a. Return value if the call to fork () fails?
    b. If the call succeeds, what is the return value in the child?
    c. If the call succeeds, what is the return value in the parent?

10. Suppose Curly, Moe, Larry and Sheep are each trying to register for CSCI480, which promises to be a fun course. The class size for already 26 students registered. In the registration system, such requests are handled by separate threads, the code involved looks like this:

```
        If (NumberEnrolled < CLASS_SIZE)
        {
        CanAddSudent = true;
        AddStudentToClass(CourseNumber, StudentID);
        NumberEnrolled++;
        }
        Else
                CanAddStudent = false;

If (CanAddStudent)

        Cout << “adding a student to CSCI “ << courseNumber << endl;

Else

        Cout <<”unable to add the student” << endl;
```

NO one has made any attempt (yet) to synchronize these threads, which are executing concurrently.

a. If this is not fixed, is it possible for us to end up with too many students in CSCI 480?

b. A semaphore is normally used to control access to some resource or variable. If we want to use a semaphore here, what resource of variable should it control?

c. A programmer decides to fix this using a semaphore called CheckFull, as in (b). Assume the semaphore has already been defined somewhere. What line of the code does the programmer need to add before the first "if" statement above?

d. What line of code does the programmer need to add after the end of the first "if" statement above?

11. Suppose our program hash used fork() twice and we now have four processes as follows:

Parents
PID = 240

Child1 Child2

PID = 241 PID = 242

Grandchild

PID = 243

a) In which two processes was fork () executed

b) Suppose in the process with PID = 241 we execute this line of code

Exit (999);
What is now the status of the process with PID = 243

c) After (b), what process is now the parent of the process with PID = 243?

d) Suppose the process with PID = 240 does not have a line that uses wait() or waitpid(). What is now the status of the process with PID = 241

12. Suppose we have two threads, ARTIST and ARTSHOW. ARTIST creates small graphics images as its output. ARTSHOW uses the images as its images and displays them on a screen. They may be dealing with the images at different speeds. It is not guaranteed that ARTSHOW can display an images as fast as ARTIST creates it. It has been arranged that there is a buffer between the two:

ARTIST ------→ bugger ----→ ARTSHOW

We can think of BUFFER as a fixed-size queue with the usual queue operations. In particular we have Full() and Empty().

(*Hints:) You may need busy waiting at some point. Only part C requires anything complicated.*

a) In artist we have a loop {

```
While (true)
{
NewImages = MakeImage ();

_______________________

WriteToBUffer(newImage);

}
```

What code do we need to put in the blank space to avoid averfilling BUFFER?

b) In ARTSHOW we have a loop

```
While (true)
{
___________________________________________

NewImage = ReadFromBuffer(); }
```

What code do we need to put in the blank space to avoid trying to obtain an item from BUFFER when BUFFER is empty?

c) Suppose instead that we have several ARTIST threads (and only one BUFFER and one ARTSHOW). To make sure only one ARTIST can check for a full BUFFER at one time. What tool do we need to use?

13. We use various benchmarks in scheduling. Name and describe two of these and state whether we want to minimize or maximize them.

14. Suppose my program is interactive (live user). (Keyboard, monitor) and it needs to use a LINUX command-line utility called "date" to display the date and time. It can use "date" either by means of the system() function of by means of the execvp() functions. The program will have to work to do after displaying the date.

    a. Which should I use and why?

15. One of our scheduling methods is knows as "first-come-first-served".

    a. Name an advantage of this method

    b. Name a disadvantage of this method.

16. As a process works it's way through a multiprocessing system, it may be in various different states. Identify two of these states.

17. Suppose I happen to know that most of the processes running on my OS have CPU bursts taking between 4 & 9 time units. I am using round-robin scheduling. Switching between processes takes about 1 time unit. Would it make more sense to have a time slice of 8 time units or a time slice of 12 time units? Why?

18. Suppose we have processes G1, G2, and G3, each of which wants to use files PQR1, PQR2 and PQR3 there is a danger of deadlock.
    a. There are several conditions which are necessary to have a deadlock. Describe two of these

    b. Suppose we want to prevent deadlock. What would be as strategy we could use?

19. A thread shares various resources with all other threads of the same process/
    a. Name two items that are shared.

    b. Name two items that are not shared.

20. Name two items that are included in the process control block.

21. In the code for a semaphore's Wait function. We may see a loop such as
    (While N<=1);

Which are called "busy waiting" and may not be the best use of the CPU's time.

An alternative is to put the process into a queue to wait, allowing some other process to execute in the meantime.

Suppose experience tells us that the time spent waiting is typically about 7 time units and the time needed to switch contexts (moving a process to or from a queue) is about 1.2 time units.

Should the busy waiting be replaced by waiting on the queue? Answer yes or no and explain very briefly why?

22. A programmer wants to have a process p1 with two child processes p2 and p3, and p1 should be able to send the same message to both p2 and p3. (All the messages are short) He tries using pipe() and two fork() calls (borth in parent process.
    We will assume the appropriate ends of the pipe have been closed.

    However, when P1 sends a message through the pipe, only one of the two child processes is able to read it. It does not work.

    a. Why doesn't it work?

b. The other ways to do this are:

- Use two pipes (one to connectP1 to p2 and one to connect p1 to p3) and write the message twice (once into each pipe)
- Have p1 write the message into a disk file and have p2 and p3 read the disk file (there may be other methods as well)

In terms of execution speed which of these two is a better choice?