National University of Computer & Emerging Sciences, Karachi

2018 CS-Department

Lab Mid

Course Code: CL205	**Course Name: Operating Systems Lab**
Instructor Name: Sumaiyah Zahid	
Student Roll No:	**Section:**

"If there is something, you don't know today. You will surely learn afterwards. Life is not an exam hall."

BEST OF LUCK!

Instructions

- Rules are made to break them. So, invent yours and I'll break.

Time: 90 minutes **Max Marks:** 60 points

Write single bash command in first line and also execlp() system call for that command in second line for each of the following: **(10 marks)**

1. Display first 10 lines of the file name 'statistic.txt'

____

____

2. List files which starts from either 1,2 or 3 and afterward have term 'file' and have an extension .mp3, from the directory /opt/usr/myData

____

____

3. Make a hard link of the file 'plants' which is present in /logs/data to /home/student/Desktop assume that you are the owner of the file.

____

____

4. List files and folders in long list format and in recursive order of the directory 'planetData'

____

____

5. Change owner of the directory 'Idea99' to the username 'peter'

____

____

Below is the code of shell script which is incorrect. Mark the error and correct them. **(5 marks)**

```
void get2Num() {
        read "Enter First Number: " firstNum
        read "Enter Second Number: " secondNum
}
echo Select an Option
echo '+ or 1 for Addition'
echo '- or 2 for Subtraction' echo
read "Your Selection: " sel
if (( sel = '1' || sel = '+' )); then
        get2Num()
         result = firstNum + secondNum
else if (( $sel = '2' || $sel = '-' )); then
        get2Num()
         result = firstNum - secondNum
else
        echo 'Error, Invalid Selection'
fi
echo "The result is: $result"
result=""
read "Do you want to Continue? [Y/N]" e
```

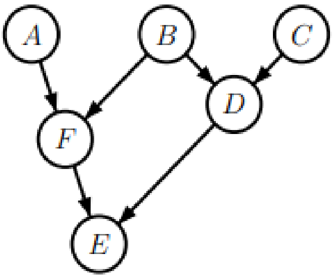
What output do the following 2 programs produce and why? **(3 marks)**

```
int counter;
static void * thread_func(void * _tn)
{
 int i;
 for (i = 0; i < 100000; i++)
 counter++;
 return NULL;
}
Int main()
{
 int i, N = 5;
 pthread_t t[N];
 for (i = 0; i < N; i++)
 pthread_create(&t[i], NULL,
 thread_func, NULL);
 for (i = 0; i < N; i++)
 pthread_join(t[i], NULL);
 printf("%d\n", counter);
 return 0;
}
```

```
int counter;
static void * thread_func(void * _tn)
{
 int i;
 for (i = 0; i < 100000; i++)
 counter++;
 return NULL;
}
Int main()
{
 int i, N = 5;
 pthread_t t[N];
 for (i = 0; i < N; i++) {
 pthread_create(&t[i], NULL,
 thread_func, NULL);
 pthread_join(t[i], NULL);
 }
 printf("%d\n", counter);
 return 0;
}
```

Suppose that we have six C functions that together solve some problem. Suppose these function depend on each other according to the following dependency graph. For example, the edge from node A to node F means that functionA must be called, and must return, before functionF can be called.

Write a sketch of a C program that uses Pthreads to execute the six functions in a way that is maximally parallel, but adheres to the above dependency graph. **(7 marks)**



True or false: Code in an OpenMP program that is not covered by a pragma is executed by all threads. **(1 marks)**

You have a computer with 4 cores. Use OpenMP to parallelize a for-loop that initializes a 100× 100 matrix with 0 and 1 such that it makes an identity Matrix **(5 marks)**

The following code outlines a synchronization pattern. Assume that the two threads begin at the same time. In what way are the two threads synchronized? Give your answer in terms of how the three calculations, A, B, and C, are ordered in time. Explain carefully what role each of the three semaphores plays in the synchronization. **(3 marks)**

```
void *thread1(void *vargp)
{ while(1)
{ << do Calculation A >>
sem_post(&semaphore1);
<< do Calculation B >>
sem_post(&semaphore2);
sem_wait(&semaphore3);
} }
void *thread2(void *vargp)
{ while(1)
{ sem_wait(&semaphore1);
<< do Calculation C >>
sem_post(&semaphore3);
sem_wait(&semaphore2);
} }
```

```
sem_t semaphore1, semaphore2, semaphore3;
int main()
{ pthread_t tid;
sem_init(&semaphore1, 0, 0); // not signaled
sem_init(&semaphore2, 0, 0); // not signaled
sem_init(&semaphore3, 0, 0); // not signaled
pthread_create(&tid, NULL, thread1, NULL);
pthread_create(&tid, NULL, thread2, NULL);
while(1){ Sleep(1000); }
}
```

A car is manufactured at each stop on a conveyor belt in a car factory. A car is constructed from the following parts - chassis, tires, seats, engine, the top cover, and painting. Thus there are 6 tasks in manufacturing a car. However, tires, seats or the engine cannot be added until the chassis is placed on the belt. The car top cannot be added until tires, seats and the engine are put in. Finally, the car cannot be painted until the top is put on.

A stop on the conveyor belt in your car company has four technicians assigned to it - Abe, Bob, Charlie, and Dave. Abe is skilled at adding tires and painting, Bob can only put the chassis on the belt, Charlie only knows how to attach the seats, and Dave knows how to add the engine as well as how to add the top.

Write code for Abe, Bob, Charlie and Dave to be able to work on the car, without violating the task order outlined above. **(7 marks)**

How /proc is different from others? **(15 marks)**

1.
2.

What is the sequence of start, stop, next, show in any sequence file execution?

What is the contents of /sys/module directory?

What is the difference between pos and v ?

Inode stores?

What is the purpose of

module_init(ct_init) ______________________________
MODULE_LICENSE() ______________________________
MODULE_DESCRIPTION() ______________________________
KERN_WARNING ______________________________
KERN_EMERG ______________________________

What is the difference between SIGINT and SIGSTOP?

What is the difference between SIGKILL and SIGTERM?

Write a code snippet which sets default behavior of ctrl+\, ignores ctrl+Z and assign func to ctrl+C.

What is the command of communication between two processes using signals?

This program will create ____ child processes and ____ threads? **(4 marks)**

```
int main()
{
    fork();
    pthread_create(&tid, NULL, thread, NULL);
    fork();
    pthread_create(&tid, NULL, thread, NULL);
    fork();
    fork();
    pthread_create(&tid, NULL, thread, NULL);
    return 0;
}
```