

CSE 321 Operating Systems

Lab Assignment 2

Total Marks: 20

The following questions are related to C programming (struct and function)
[2X2= 4 marks]

Task-1:

Suppose you and your friends go to have breakfast in a restaurant. Each of you orders Paratha, Vegetable, and Mineral Water. Treat each of the ordered items as structures and each of the structures will have two properties which are: quantity and unit price. Each property of the structure will be taken as input from the user. After taking all the inputs calculate what is the total bill and also, take input from the user on how many people are there in total. Lastly calculate how much each person will have to pay and print it (Note: This value will be a float).

Sample Input:

Quantity Of Paratha: 25

Unit Price: 10

Quantity Of Vegetables: 5

Unit Price: 20

Quantity Of Mineral Water: 20

Unit Price: 20

Number of People: 6

Sample Output:

Individual people will pay: 125.00 tk

Task-2:

Write a C program to print perfect numbers between given intervals using a function. A perfect number is a positive integer equal to the sum of its positive divisors, excluding the number itself.

Sample Input:

1

10000

Sample Output:

6

28

496

8182

The following questions are related to system call [5X2= 10 marks]

Task-1:

Write a c program that will open a file given from the command line argument and then it will ask the user to input strings that will be written to that file. It will continue to ask the user to enter a string as long as the user enters “-1”. If the given file does not exist in the directory, then your program will automatically create the file.

Task-2:

Write a program that will create a child and another grandchild process. Every process will print a line.

The parent process will print, “I am parent”

Child process will print, “I am child”

The grandchild process will print, “I am grandchild”

Now, write the program in such a way that the following output will be shown -

I am grandchild

I am child

I am parent

Task-3:

Consider the following code snippet in your main function -

```
a = fork();
```

```
b = fork();
```

```
c = fork();
```

Now, write the full program, that will check the children's PID and if it is odd then the process will create another child process. Lastly, print how many processes have been created considering the first parent process.

Task-4:

Write a program named "sort.c" where you will give some number from the command line argument and the program will print the sorted array in descending order. Then, write another program named "oddeven.c" which will take some numbers from the command line, then check and print whether the numbers in the array are odd or even.

Now, you have to write a program that will create a child process and the child process will first sort the array that you have declared in this program. And then, the parent process will print the odd/even status for each number in the array.

Task -5:

Write a program in c that the parent process will create one child process and 3 grandchild processes and print their IDs

Output: 1. Parent process ID : 0
2. Child process ID:

3. Grand Child process ID: ...
4. Grand Child process ID:
5. Grand Child process ID:

The following questions are related to Threading [3X2= 6 marks]

Task-1:

Write a c program that creates 5 threads and prints which thread is running and after the thread is closed, a new thread starts its execution. Each thread should run sequentially one by one.

OUTPUT:

thread-1 running

thread-1 closed

thread-2 running

thread-2 closed

...

Task-2:

Write a program in c using 5 threads where each thread will print 5 integers

The outputs will look like this:

Output:

Thread 0 prints 1

Thread 0 prints 2

Thread 0 prints 3

Thread 0 prints 4

Thread 0 prints 5

Thread 1 prints 6

Thread 1 prints 7

.

.

.

.

Thread 4 prints 21

.

.

Thread 4 prints 25

Task-3:

Write a program in c that has a function that takes the name of the user and adds all the ASCII value of the characters and returns it. Now create 3 threads that run the function using 3 different user names. Now print “Youreka” if all the returned values are equal, print “Miracle” if the 2 returned values are equal, and print “Hasta la vista” if the values don’t match using another thread.