

Day 10 – Assignments for 21-Dec-17 (Multi-threading, Multi-processing and Network Programming)

1. Implement a sample program as follows to demonstrate multithreading and synchronization:
 - Create two or more threads with argument (minimum number of threads are 2)
 - Pass a value through command line argument, if user not passed any argument then take default value as 1 lakh
 - Take a global variable
 - In the thread function receive the value (which was passed through command line or default value) and increment that many times with the global variable. Print the final value.
 - Write the program without synchronization mechanism
2. Implement the above sample program along with synchronization
3. Write a program in Python to perform the following:
 - Create first child process, say child1
 - Create second child process, say child2
 - Child1 need to call the function to implement the below functionality, say functionality1
 - Child2 need to call the function to implement the below functionality, say functionality2
 - Don't read input from user from function. Instead, accept input and then pass to the function as arguments accordingly. Need to print the result after receiving from function.
 - Parent process need to wait until the child processes completes their tasks (Don't use sleep() or delay() functions)
 - Functionality 1

To perform maximum product of two numbers as follows (example is only for understanding, need to write generic code. You can take values as below but code should be generic)

- Implement function called `maximum_product_two_numbers()`, which accepts two arguments, first one is the length of list and second one is `elems`, which is variable-length argument. In other words, the function need to accept variable length arguments or variable number of arguments i.e., first argument is length of variable arguments, second argument is elements themselves. The function need to return product of first two maximum numbers.
- Call from main program such as
Ex:
`maximum_product_two_numbers(5, 1, 100, 42, 4, 23)`
or
`maximum_product_two_numbers(9, 1, 100, 42, 4, 23, 123, 45, 92, 99)`
Print the received value from function, which is maximum product.
- Functionality 2

Given an unsorted list having both negative and positive integers. The task is to place all negative element at the end of list without changing the order of positive element and negative element. No need to modify the existing list, you can create new list for output.

- Accept user input of +ve and -ve values in a list as user input
- Move all negative elements to end.
- Call from main program such as:

```
move_negative_elements_to_end(mylist)
```

Where mylist is list of +ve and -ve numbers as entered by user

NOTE: User would enter input of list consisting of +ve and -ve numbers

Examples (Code should be generic, this is just an example)

Input : mylist = [1, -1, 3, 2, -7, -5, 11, 6]

Output : [1 3 2 11 6 -1 -7 -5]

Input : mylist = [-5, 7, -3, -4, 9, 10, -1, 11]

Output : [7 9 10 11 -5 -3 -4 -1]

4. Write a simple server and client to demonstrate connection oriented (TCP/IP) protocol. The client need to send a string to server, the server has to reverse the string and send back to client. After receiving the response from server, client need to exit.
5. Write a simple server and client to demonstrate connection less (UDP/IP) protocol. The client need to send a number to server, the server has to perform the factorial of that number and send back the result to client. After receiving the result from server, client need to print the output and exit.