OS ASSIGNMENT-2 (PART-1)
WRITE-UP

RAHUL KHATOLIYA
2019265
CSAM ”2023

The Difference :

First of All , it should be very clear that when we pass the parameter (Global variable) , in the sub-processes , i.e child and parent processes , then they both have their own copy for the same variable i.e they are isolated from each other , any change or manipulation inside anyone process doesn’t make any change for the other one . As a result when we fork a child process and pass the global variable to it , then the initial received value of the variable is 10 , and we decrement it with 1 until we get the value -90 , and as soon as we have reached the same value , the child process terminates successfully , and the control is given to the parent process which was waiting for the child process to finish execution. Inside parent process , we again have the same Initialized value (Unchanged) of 10 for the global variable , but here we increment it by 1 until we get 100 , and once it reaches the desired result , it successfully terminates .So the final printed value for this case will always be -90 and 100 respectively .

But Now , for the Thread’s case , we must know that the variables defined within the same process of the thread creation , don’t get isolated , that means suppose one of the thread(say child one) makes an alteration to the global variable by decrementing it by 1 and gets suspended , then once the control is reached to the other thread(say parent) , and it tries to use the same variable , then it will get the manipulated value of the same that other thread have caused , and thus perform its operations on this manipulated value accordingly.

So what happend in our case is that , which ever thread gets triggered first does its operations , makes changes to the global variable perform its operations and gets suspended , and the now when the other thread is triggered , it will continue its operations by first comparing this manipulated value with its loop termination condition and do its operations . Lets make it more clear , suppose parent thread was created first and had an initialised value of 10 , and now suppose before it increment this value by 1 , suppose this thread gets suspended and the other thread(child) gets the control , Now this thread which is still having the Unchanged value i.e 10 which is not yet manipualted by any other thread gets decremented by 1 and gets printed until this thread gets suspended , and once it gets suspended , the parent thread is trigerred which first presumes its initial state at which it was having the value 10 and prints 10, but now since the value was decremented by child thread constantly, So the operations of increment will happen on this changed value of the global variable constantly , and afterwards the threads gets suspended and other thread is triggered.

So the entire functionality of the program keeps on changing from one thread to second , and which ever gets its termination condition like -90 for child thread and 100 for parent thread gets printed , and then this thread exits successfully , and now the other thread gets the charge which performs its operations until the defined termination condition is reached . As a result , we can get any final print value like -90 first and then 100 or 100 first and -90 second, depending on which thread get terminated first .