

Q5. Consider a program that uses the fork() system call to create a child process. Initially, the parent process has a variable x with a value of 5. After forking, both the parent and child processes increment the value of x by 1. What will be the final values of x in the parent and child processes after the fork() call?

-> Initially x=5 in the process. After fork() is called, a duplicate (child) process will be created which will get its **own Memory Space** on RAM which is **independent** from the Memory space of the process through which fork() was called (which now becomes the Parent process).

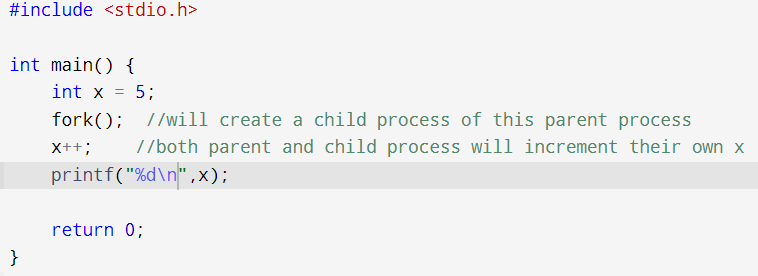
i.e. Each process (child and parent) will be stored at different Memory Addresses on the RAM.

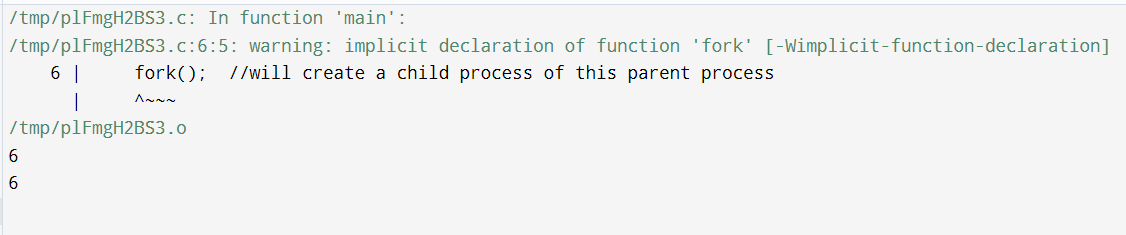
So, each process will have its **own stack, data, heap memory**

So, if the variable x is incremented in child process, it will increment the variable x which is stored on its own stack memory segment. Therefore **x=6 in the child process**.

Similarly, the x which got incremented in the parent process will get incremented in its own stack memory segment and **x=6 in the parent process**.

Therefore, x=6 in both Parent and the Child Process.





6 is printed twice. Once from the parent process and once from the child process.