Saturday, October 23, 2021 11:14 AM

At the time of creation each child has a parent process id.
If parent completes before the child then the process is called as Orphan process!!!

Then the Kernel init process is assigned as the parent of that orphan process!!!!

Parent process clears the **process table entries** of the child process once it has terminated!!!

Process runs the LAST instruction in MAIN } Logically the process END
|
| ZOMBIE process!!!
|
The process table entries must be cleared } Logistically the process ends here

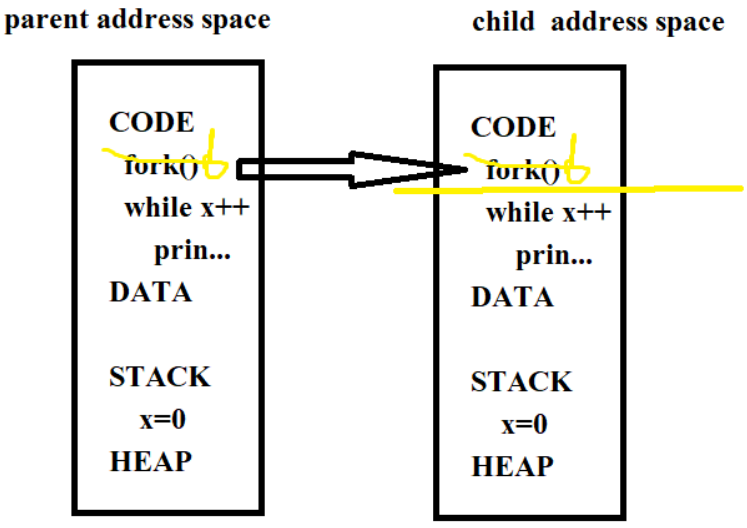
If the process remains ZOMBIE for a long then there is a **RESOURCE LEAK** !!!
Process table

Process	PCB
P1	
zombie	
P2	
P3	
zombie	

LINUX WILD CARD CHARACTER = * -matches for 0 or more occurrences of any charachter

FORK ----------

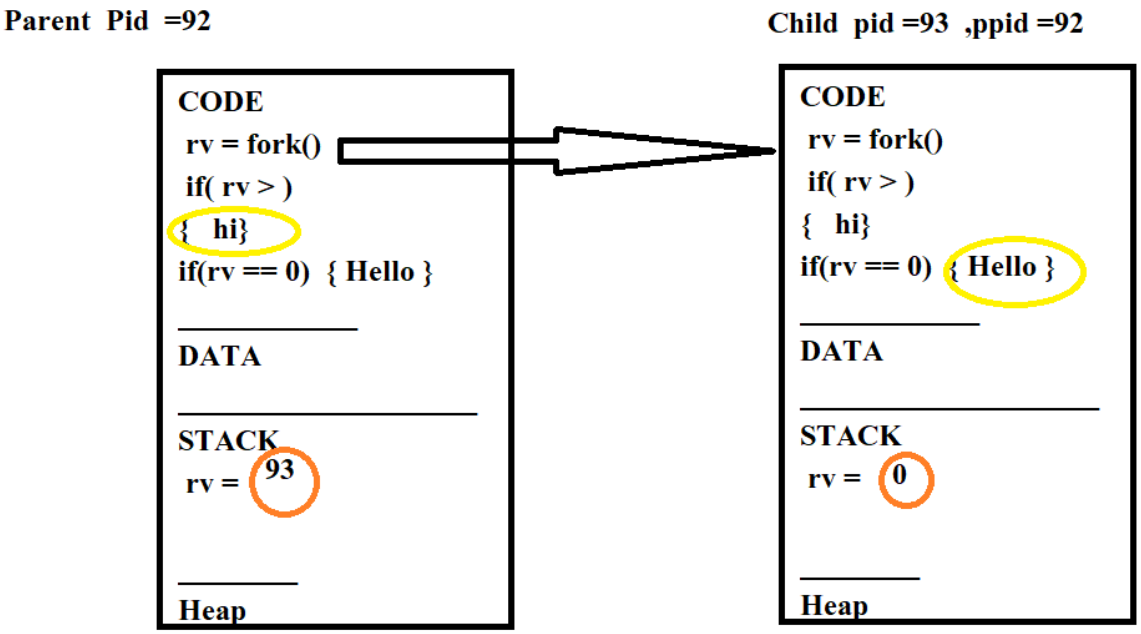
FORK makes a COPY of the CURRENT ADDRESS SPACE !!!!!



FORK returns an integer values which is PID

1. Childs pid to parent
2. 0 to child

The parent process says Hi infinitely
The child process says Hello infinitely !!!!



HW --- write a program that will make the parent print the fibonacci series

1
1 1
1 1 2
1 1 2 3
1 1 2 3 5
....

And child will print the table of each number

2 ,4,6,8,10,12,14,16,18,20
3,6,9,12,15,18,21,24,27,30
...
...

If you see **consecutive** forks() then the formula is
2 raised to number of forks () = number of processes created.

n = no of fork

2^n = no of process

TRY this in the LAB ----------

```
Main ----------
{
        Fork()
        Fork()
        Fork()
        Fork()
        Print( pid )
}
```

TRY this in the LAB ----------

```
Main ----------P1
{
        Rv = Fork() ----------P11
        If(rv >0 )
        {
                Fork() ------P12
                Print("TEST1");   ---------- P1 ,P12
        }
        Else
        {
```

```
            Print("Test2")  ----------------P11
      }
}
```
