

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



Fork VS. Threads

Lab 05



اللهم علمنا ما ينفعنا،، وانفعنا بما علمتنا،، وزدنا علماً



Lab Objective

- To understand the deference between thread and fork.



Practice

- In the following C program, the main process creates one thread of the function `doit` and forks one child.
- Both the `doit` function and the child code increment and display the global variable `counter`.
- Write, compile, and run the program.



```

#include <iostream>
#include <stdlib.h>  /* exit() */
#include <unistd.h>  /* fork() */
#include <sys/types.h>      /* pid_t */
#include <sys/wait.h> /* wait() */
#include "pthread.h"
using namespace std;
int counter = 0; //Incremented by the threads and child
void * doit(void *);
int main()
{
    pthread_t tid;
    pid_t pid, cpid;
    int status;
    // Start the thread
    pthread_create(&tid, NULL, doit, NULL);
    //Delay between starting the thread and forking the child
    sleep(2);
    pid = fork(); //Fork the Child

    if(pid<0) //error occurred
    {
        cout<<"Fork failed\n";
        exit(-1);
    } //end if

```

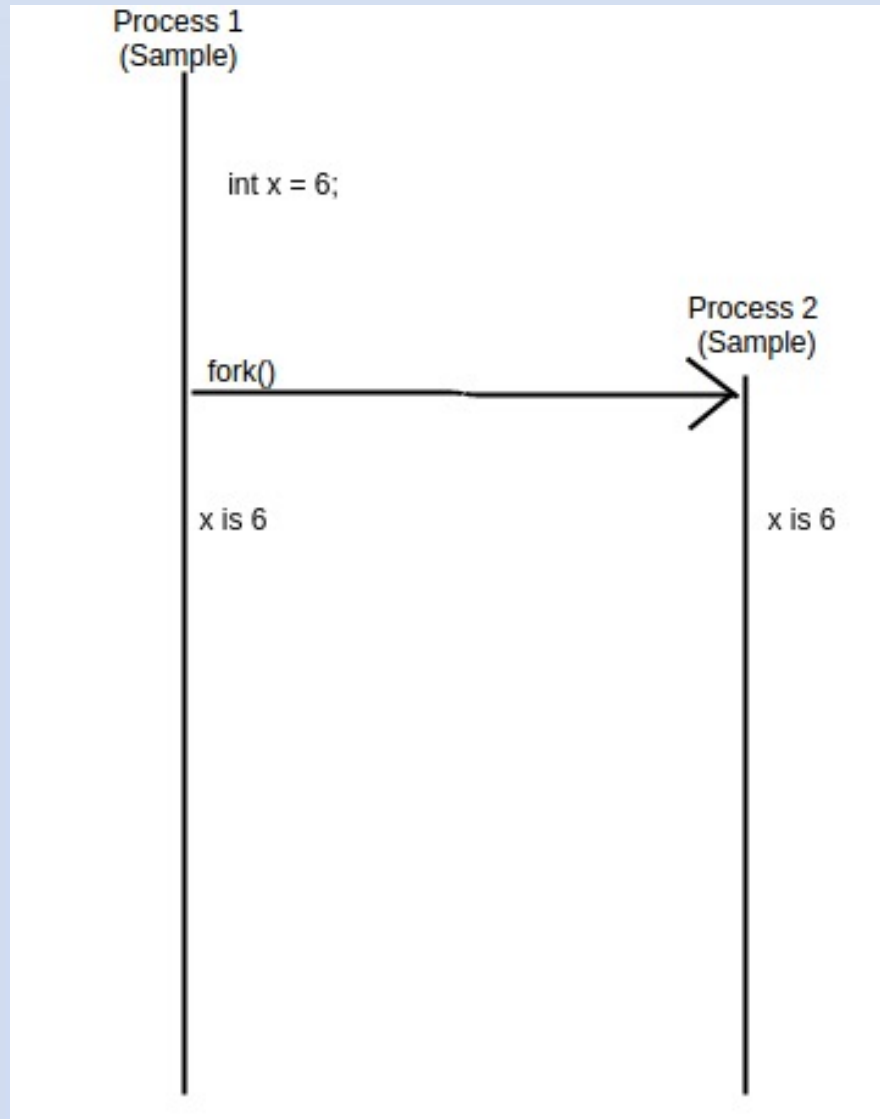


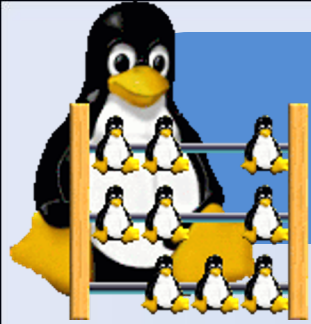
```
else if (pid == 0) { // child process
    sleep(2);
    counter++;
    cout << "Child Counter = " << counter << endl;
}
else // parent process
{
    // parent will wait for the child to complete
    cpid = wait(&status);
    // wait for the thread to terminate
    pthread_join(tid, NULL);
    exit(0);
} //end else
} // End main

void * doit(void *vptr)
{
    sleep(1);
    cout << "Thread First Counter = " <<
        ++counter << endl;
    sleep(5);
    cout << "Thread Second Counter = " <<
        ++counter << endl;
    return(NULL);
}
```



- **fork() creates a clone!**
- A child process created by fork() is a copy of the parent process but they **do not** share any memory





Check Off

- 1) What are the printed values of **counter**? Explain why **counter** gets these values from the child and the thread.
- 2) Remove the **sleep(2)** line that delays between starting the thread and forking the child. Recompile the program and run it. What are the new printed values of **counter**? Explain why **counter** gets these values from the child and the thread.





??? ANY QUESTIONS ???

