Networking Lab Assignment 3

Familiarization and implementation of programs related to Process
and thread.

Albin Antony

4 February 2019

1   Process And Thread

1.1   Aim

To familiarize and implement programs related to Process and thread.

1.2   Theory

A thread is a path of execution within a process. A process can contain multiple threads.A thread is also known as lightweight process. The idea is to achieve parallelism by dividing a process into multiple threads.A thread is a single sequence stream within in a process. Because threads have some of the properties of processes, they are sometimes called lightweight processes. Threads are not independent of one other like processes as a result threads shares with other threads their code section, data section and OS resources like open files and signals. But, like process, a thread has its own program counter (PC), a register set, and a stack space

1.3   Algorithm

Algorithm 1 Algorithm for creating N threads

```
1  START
2  Let  NUM_THREAD=10
3  Procedure  *MyFunction(void * tid)
4  Begin
5          Print(" Thread ␣:␣" tid)
6  End     MyFunction
7  create  pthread  thread
8  f=fork()  //create fork
9  IF( f==0)
10          PRINT  " child ␣:␣pid"
11          Begin  For  i<NUM_THREAD
12                  call  pthread_create(&thread,NULL,MyFunction,tid)
13          i++
14  ELSE
15          PRINT  " parent ␣:␣pid"
16          Begin  For  i<NUM_THREAD
17                  call  pthread_create(&thread,NULL,MyFunction,tid)
18          i++
19  ENDIF
20  End  For
21  STOP
```

1.4 Program

```cpp
#include<iostream>
#include<cstdlib>
#include<pthread.h>
#include<sys/types.h>
#include<unistd.h>
using namespace std;

#define NUM_THREAD 4

void *Printer(void *threadId){
        printf("Thread_%d_of_process_%d\n",threadId,getpid());
        pthread_exit(NULL);
}

int main(){
    pthread_t thread;
    int rc;
    int f=fork();
    if(f==0){
        printf("cHilD_is_%d_\n",getpid());
        for(int i=0;i<NUM_THREAD;i++){
            rc = pthread_create(&thread,NULL,Printer,(void *)i);
            if(rc){
                    cout<<"ErrOR";
            }
        }
    }
    else{
        printf("pAreNt_is_%d_\n",getpid());
        for(int i=0;i<NUM_THREAD;i++){
            rc = pthread_create(&thread,NULL,Printer,(void *)i);
            if(rc){
                    cout<<"ErrOR";
            }
        }

    }
    pthread_exit(NULL);

}
```

1.5 Output

```
pAreNt is 4105
cHilD is 4106
```

```
Thread 0 of process 4105
Thread 1 of process 4106
Thread 1 of process 4105
Thread 0 of process 4106
Thread 3 of process 4106
Thread 2 of process 4106
Thread 3 of process 4105
Thread 2 of process 4105
```

1.6   Result

Implemented thread in C compiled on gcc 6.3.0 and executed on Debian 4.9 Kernel 4.9 and ouputs were verified.