

## CENG 328 OPERATING SYSTEM LAB WORK 2

1. Write a program and create 3 processes where first process is the parent of the second one and the second process is the parent of the third one. Print the pid and parent pid of each process. What is the parent pid of the first process? Print the group id of the processes. Do they belong to the same group? Run the program in two different terminal windows. Do you get same group id in all terminals?

```
#include<stdio.h>
#include<stdlib.h>
int main() {
    int pid,pid1,pid2,pid3,ppid1,ppid2,ppid3,p1,p2,p3,status,i,number;
    int ppgr,p1gr,p2gr,p3gr;
    pid=getpid();
    ppgr=getpgrp();
    printf("group id of the parent process = %d\n",ppgr);
    printf("parents pid = %d\n",pid);
    p1=fork();
    if(p1==0) {
        pid1=getpid();
        printf("child 1s pid = %d\n",pid1);
        ppid1=getppid();
        printf("child 1s parent pid = %d\n",ppid1);
        p2=fork();
        p1gr=getpgrp();
        printf("group id of the 1st child = %d\n",p1gr);
        if(p2==0) {
            pid2=getpid();
            printf("child 2s pid = %d\n",pid2);
            ppid2=getppid();
            printf("child 2s parent pid = %d\n",ppid2);
            p2gr=getpgrp();
            printf("group id of the 1st child = %d\n",p2gr);
            p3=fork();
            if(p3==0) {
                pid3=getpid();
                printf("child 3s pid = %d\n",pid3);
                ppid3=getppid();
                printf("child 3s parent pid = %d\n",ppid3);
                p3gr=getpgrp();
                printf("group id of the 1st child = %d\n",p3gr);
                sleep(10);
                exit(15);
            }
        }
        exit(10);
    }
    exit(5);
}
waitpid(p1,&status,0);
printf("Child1 is returned by %d\n",status>>8);
waitpid(p2,&status,0);
printf("Child2 is returned by %d\n",status>>8);
waitpid(p3,&status,0);
printf("Child3 is returned by %d\n",status>>8);
return 0;
}
```

2. Write a program to create 4 threads. Each thread will increment the value of a global variable by one for 1000 times. After each iteration the thread sleeps for a short while. Test your program several times. Do you always get a correct result?

1st way

```
#include <thread.h>
#include <stdlib.h>
#include <stdio.h>

int global=0;
void * myfunc1() {
    int i;
    for(i=1; i<=10; i++) {
        global=global+1;
        sleep(2);
        printf("Thread loopu %d \n",global);
    }
}

void * myfunc2() {
    int i;
    for(i=1; i<=10; i++) {
        global=global+1;
        sleep(2);
        printf("Thread loopu %d \n",global);
    }
}

void * myfunc3() {
    int i;
    for(i=1; i<=10; i++) {
        global=global+1;
        sleep(2);
        printf("Thread loopu %d \n",global);
    }
}

void * myfunc4() {
    int i;
    for(i=1; i<=10; i++) {
        global=global+1;
        sleep(2);
        printf("Thread loopu %d \n",global);
    }
}

int main() {
    int threadid1,threadid2,threadid3,threadid4;
    thr_create (NULL, 0, myfunc1, NULL, 0, &threadid1);
    thr_create (NULL, 0, myfunc2, NULL, 0, &threadid2);
    thr_create (NULL, 0, myfunc3, NULL, 0, &threadid3);
    thr_create (NULL, 0, myfunc4, NULL,0, &threadid4);
    thr_join(NULL, NULL, NULL); return 0;
}
```

2nd way

```
#include<stdio.h>
#include<thread.h>
#include<stdlib.h>

int x=0;
void *ft() {
    int i;
    for(i=1;i<=1000;i++)
        x+=1;
}

main() {
    int thid1,thid2,thid3,thid4;
    thr_create(NULL,0,ft,NULL,0,&thid1);
    thr_create(NULL,0,ft,NULL,0,&thid2);
    thr_create(NULL,0,ft,NULL,0,&thid3);
    thr_create(NULL,0,ft,NULL,0,&thid4);
    thr_join(NULL,NULL,NULL);
    printf("this is x= %d",x);
}
```