Activity 3 Process Concepts

ข้อกำหนด

งานกลุ่มๆ ละ 2-3 คน

ส่งเป็นไฟล์ PDF เพียงไฟล์เดียวที่มีชื่อสมาชิกทุกคน, Source code และผลลัพธ์การแสดงผลของทุกข้อ

ต้องส่งให้ครบถ้วน จึงจะได้คะแนนเต็ม

1.เขียนโปรแกรมสร้างโปรเซส 3 ระดับ โดยที่โปรเซสแต่ละระดับบอกว่า ตัวเองเป็น parent, child หรือ grandchild แล้วแสดงค่า pid ของตนเอง และค่า ppid ด้วยโดยใช้ fork() , getpid() และ getppid()

นี่คือตัวอย่างผลลัพธ์ที่ได้จากโปรแกรมนี้

```
Thong@Debian >sol1
I am the parent process. My PID is 53
I am the child process. My PID is 54 and my parent's PID is 53
I am the grandchild process. My PID is 55 and my parent's PID is 54
Thong@Debian >
```

2. โปรแกรมต่อไปนี้ทำงานสลับระหว่างการคำนวณและการไม่ทำอะไรเลย (sleep)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <signal.h>

      int compute_period = 5;
      int sleep_period = 5;
      int i;

      /* what to do when alarm is on */

      void on_alarm (int signal) {
            printf("Sleep\n");
            sleep (sleep_period);
            printf ("Wake up\n");
            /* activate alarm again */
            alarm (compute_period);
      }

main (int argc, char* argv[])
{
      int i;
      if (argc != 3) {
            printf ("Usage: infinite <compute-period><sleepperiod>\n");
      exit(0);
      }
      else {
            compute_period = atoi (argv[1]);
            sleep_period = atoi (argv[2]);
      }
      /* on_alarm() is signal handler for SIGALARM */
      signal(SIGALRM, on_alarm);
```

```
        /* activate alarm */
        alarm (compute_period);

        /* compute infinitely but can be interrupted by alarm */
        for (i=0; ;i++) {
              if (i==0) printf("computing\n");
        }
}
```

ให้แก้ไขโปรแกรมโดยเปลี่ยนจากการใช้ argument เป็นการถามค่า compute_period และ sleep_period จากแป้นพิมพ์ เหมือนตัวอย่างข้างล่าง

```
thong@Thongchai:~/Activity2$ ./sol1
Enter compute period :
1
Enter sleep period :
1
computing
Sleep
Wake up
Sleep
Wake up
```

3. แก้ไขโปรแกรมข้างล่างนี้ โดยให้กลับลำดับของข้อความที่พิมพ์ออกมา นั่นคือ โปรเซสที่ถูกสร้างขึ้นหลังสุด (pid มากที่สุด) จะพิมพ์ข้อความออกมาแรกสุด และโปรเซสที่ถูกสร้างขึ้นแรกสุดจะพิมพ์ข้อความออกมาหลังสุด

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
main()
{
	int   i;
	int   n;
	pid_t childpid;

	n = 4;
	for (i = 0; i < n; ++i) {
		childpid = fork() ;
		if (childpid > 0)
			break;
	}
	printf("This is process %ld with parent %ld\n",(long) getpid(), (long) getppid() );
	wait(0);
}
```

นี่คือตัวอย่างผลลัพธ์ที่ถูกต้อง

```
Thong@Debian >./sol2
This is process 34 with parent 33
This is process 33 with parent 32
This is process 32 with parent 31
This is process 31 with parent 30
This is process 30 with parent 9
Thong@Debian >
```