

## Activity 3 Process Concepts

1. 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 SIGALRM */
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 แล้วแสดง source code และอธิบายสิ่งที่เปลี่ยนแปลง

```

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

```

2. แก้ไขโปรแกรมข้างล่างนี้ โดยให้กลับลำดับของข้อความที่พิมพ์ออกมา นั่นคือ โปรเซสที่ถูกสร้างขึ้นล่าสุด (pid มากที่สุด) จะพิมพ์ข้อความออกมาแรกสุด และโปรเซสที่ถูกสร้างขึ้นแรกสุดจะพิมพ์ข้อความออกมาหลังสุด ให้อธิบายสั้น ๆ ว่ามีแนวคิดในการแก้ปัญหานี้อย่างไร แก้ไขโปรแกรมที่จุดไหน อย่างไร และแสดงผลพร้อมหน้าจอ

```
#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);
}
```

3. แก้ไขโปรแกรมข้างล่างนี้ โดยทำให้โปรเซสที่ถูกสร้างขึ้นกลายเป็น Zombie ด้วยการให้ child process จบการทำงานก่อน parent process เมื่อใช้คำสั่ง ps -al ในขณะที่โปรแกรมทำงานอยู่ (เปิดอีก terminal หนึ่ง) ในคอลัมน์ที่ 2 ซึ่งแสดง Status ของ child process จะเป็น Z ให้อธิบายสั้น ๆ ว่ามีแนวคิดในการแก้ปัญหานี้อย่างไร แก้ไขโปรแกรมที่จุดไหน อย่างไร และแสดงผลพร้อมหน้าจอ

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

```
n = 4;

for (i = 0; i < n; ++i) {
    childpid = fork() ;
    if (childpid == 0) break;
    wait(0);
}

printf("This is process %ld with parent %ld\n", (long) getpid (), (long)
getppid());
}
```