Activity 3

กลุ่ม :(){:|:&};:

สมาชิก:

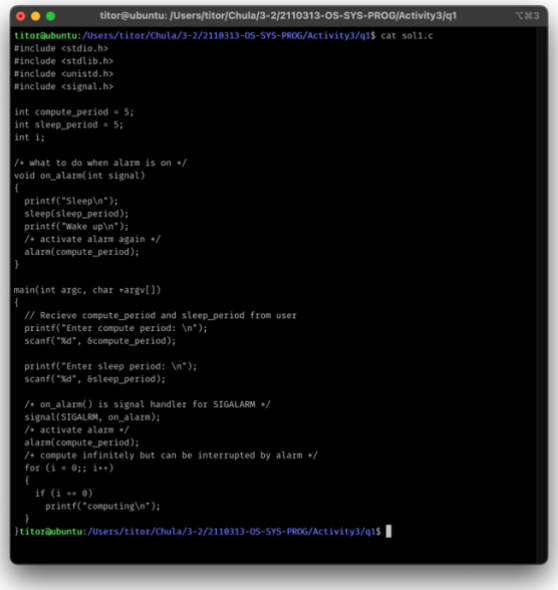
1. ธนัส วงศ์สมุทร 6432067021
2. ก้องภพ จริยาสถาพร 6430014321

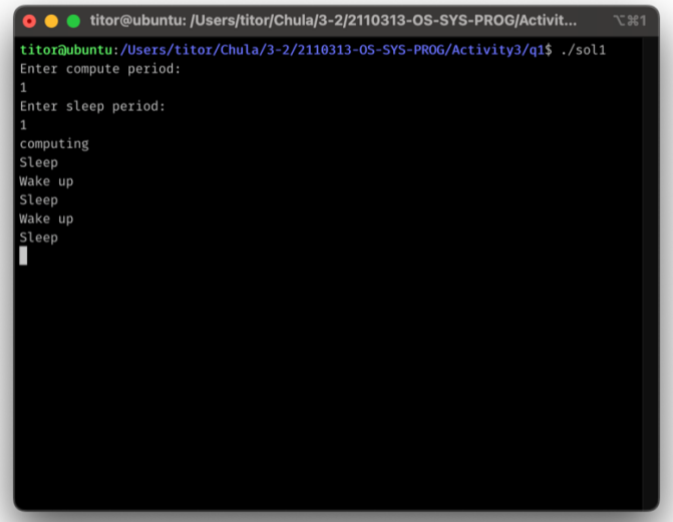
1.

เปลี่ยนการรับ compute_period และ sleep_period จากการรับจาก argument เป็นรับจาก user input แทน

```c
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");
  }
}
```

```c
main(int argc, char *argv[])
{
  // Recieve compute_period and sleep_period from user
  printf("Enter compute period: \n");
  scanf("%d", &compute_period);

  printf("Enter sleep period: \n");
  scanf("%d", &sleep_period);

  /* 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");
  }
}
```



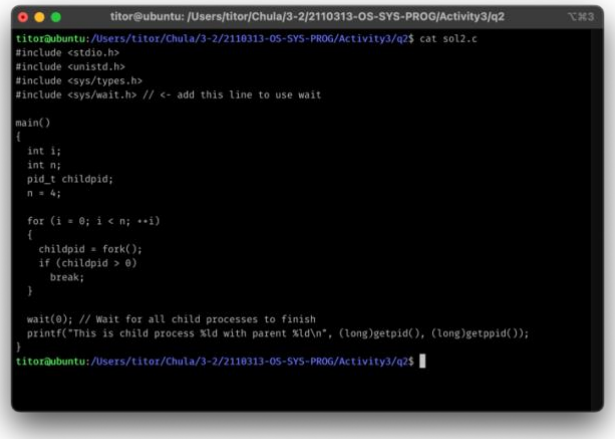


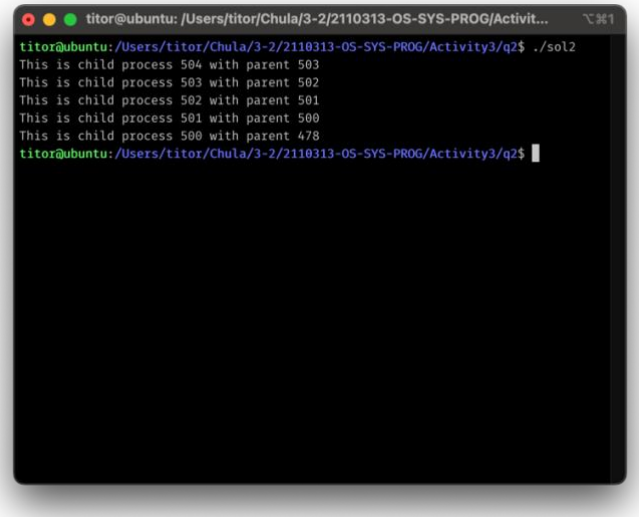
2.

ทำการสลับระหว่าง wait(0) และ printf เพื่อรอให้ child process ทั้งหมดทำงานเสร็จก่อน เวลาแสดงผลจะได้แสดงเรียงจาก pid ที่มีค่ามากๆ แสดงผลออกมาก่อน

```c
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);
}
```

```c
main()
{
  int i;
  int n;
  pid_t childpid;
  n = 4;

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

  wait(0); // Wait for all child processes to finish
  printf("This is child process %ld with parent %ld\n", (long)getpid(), (long)getppid());
}
```





3.

Zombie process is fork process that child process died or exit before their parent does, to accomplish that by modify given source code. I simply add exit(0) and print the PID instead of break when detect that it's child process, meanwhile I tell parent to sleep, to make sure that it won't exit before child does. (remove wait(0) to tell the parent not to wait for child process)

```c
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());
}
```

```c
main()
{
  int i;
  int n;
  pid_t childpid;
  n = 4;
  for (i = 0; i < n; ++i)
  {
    childpid = fork();
    if (childpid == 0)
    {
      printf("This is process %ld with parent %ld\n", (long)getpid(), (long)getppid());
      exit(0);
    }
    sleep(30);
  }
}
```

```
titor@ubuntu: /Users/titor/Chula/3-2/2110313-OS-SYS-PROG/Activity3/q3
titor@ubuntu:/Users/titor/Chula/3-2/2110313-OS-SYS-PROG/Activity3/q3$ cat ./sol3.c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h> // <- add this line to use wait
#include <stdlib.h>

main()
{
  int i;
  int n;
  pid_t childpid;
  n = 4;
  for (i = 0; i < n; ++i)
  {
    childpid = fork();
    if (childpid == 0)
    {
      printf("This is process %ld with parent %ld\n", (long)getpid(), (long)getppid());
      exit(0);
    }
    sleep(30);
  }
}
titor@ubuntu:/Users/titor/Chula/3-2/2110313-OS-SYS-PROG/Activity3/q3$
```

```
titor@ubuntu: /Users/titor/Chula/3-2/2110313-OS-SYS-PROG/Activit...
titor@ubuntu:/Users/titor/Chula/3-2/2110313-OS-SYS-PROG/Activity3/q3$ ./sol3
This is process 603 with parent 602
```

```
titor@ubuntu: /Users/titor/Chula/3-2/2110313-OS-SYS-PROG/Activity3/q3
titor@ubuntu:/Users/titor/Chula/3-2/2110313-OS-SYS-PROG/Activity3/q3$ ps -al
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S   501     602     478  0  80   0 - 298305 hrtime pts/1    00:00:00 sol3
1 Z   501     603     602  0  80   0 -     0 -      pts/1    00:00:00 sol3 <defunct>
0 R   501     604     525  0  80   0 - 300944 -      pts/2    00:00:00 ps
titor@ubuntu:/Users/titor/Chula/3-2/2110313-OS-SYS-PROG/Activity3/q3$
```