

```
#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/wait.h>
```

```
void child(); //Voidเป็นการประกาศฟังก์ชันโดยที่childคือตัวเลข
```

```
void Grandparent(); //Voidเป็นการประกาศฟังก์ชันโดยที่Grandparentเป็นการเรียกใช้งานฟังก์ชันตัวแรก
```

```
void parent(pid_t,int); //Voidเป็นการประกาศฟังก์ชันโดยที่parentเป็นการเรียกใช้งานฟังก์ชันตัวถัดไปจากGrandparent
```

```
int main(int argc,char*argv[]) //รับค่า input จาก Terminal
```

```
{
```

```
printf("OS 2563 Assignment 1\n"); //ปริญการมอบหมายงาน os 2563
```

```
printf("1. 6140203917 Pornthip Komresu\n"); //ปริญรหัสนิสิตและชื่อ
```

```
printf("2. 6140204428 Phanumas Tangon\n"); //ปริญรหัสนิสิตและชื่อ
```

```
printf("3. 6140204832 Rattanasuk Sanrang\n"); //ปริญรหัสนิสิตและชื่อ
```

```
printf(" \n");
```

```
pid_t pid;
```

```
int number = atoi(argv[argc=1]); //Set an int value as Number.
```

```
switch (pid = fork()) //คำสั่ง switch จะเปรียบเทียบกับค่าที่อยู่หลัง case แต่ละคำสั่ง ถ้าตรงกัน ก็จะทำงานที่อยู่ใน case นั้น ๆ แต่ถ้า  
ไม่ตรงกับ case ไหนเลย จะไปทำหลังคำสั่ง default
```

```
{
```

```
case 1:
```

```
exit(EXIT_FAILURE);
```

```
case 0:
```

```
parent(pid,number);
```

default:

Grandparent(number);

}

}

void child(int num) //Voidเป็นการประกาศฟังก์ชันโดยที่childคือตัวเลข(intประกาศตัวแปรชนิดจำนวนเต็ม ชื่อ num)

{

printf("pid is %ld %d\n", (long int) getpid(), num*3); //ป้อนค่า pid จากนั้นนำ input มาคูณ3

exit(EXIT_SUCCESS);

}

void Grandparent(int num) //Grandparentเป็นการเรียกใช้งานฟังก์ชันตัวแรก(intประกาศตัวแปรชนิดจำนวนเต็ม ชื่อ num)

{

sleep(5);

printf("Grandparent is %ld %d\n", (long int) getpid(), num); //ป้อนค่า Grandparent จากนั้นนำค่า input มาแสดง

exit(EXIT_SUCCESS);

}

void parent(pid_t pid, int num) //parentเป็นการเรียกใช้งานฟังก์ชันตัวถัดไปจากGrandparent(intประกาศตัวแปรชนิดจำนวนเต็ม ชื่อ num)

{

switch (pid = fork()) //คำสั่ง switch จะเปรียบเทียบกับค่าที่อยู่หลัง case แต่ละคำสั่ง ถ้าตรงกัน ก็จะทำงานที่อยู่ใน case นั้น ๆ แต่ถ้าไม่ตรงกับ case ไหนเลย จะไปทำหลังคำสั่ง default

{

case 1:

exit(EXIT_FAILURE);

case 0:

```
child(num);

default:

sleep(4);

}

printf("Parent pid is %ld %d\n", (long int) getpid(), num*2); //ปรี้นค่า Parent pid จากนั้นนำ input มาคูณ2


exit(EXIT_SUCCESS); //หยุดการทำงานการตามกระบวนการและสถานะการส่งคืน


}
```