

ใบงาน 4

วัตถุประสงค์ ศึกษา system call fork() และอื่นๆ ที่เกี่ยวข้อง

4.1 fork()

System call คือ library สำหรับ
นักพัฒนาโปรแกรมเรียกให้ kernel
ให้บริการตามความสามารถของ
system call นั้นๆ

fork() เป็นคำสั่งสร้างโปรเซส โปรเซสในระบบคอมพิวเตอร์มีความสัมพันธ์แม่ลูก สำหรับ โปรเซสที่สร้าง และ โปรเซสที่ถูกสร้าง

การเรียก fork() ต้องใช้ unistd.h และ sys/types.h

fork() ทำสำเนาของ code ของแม่ และคืน process id ของลูกที่สร้างมาให้แม่ ทั้ง 2 โปรเซสจะทำงานต่อใน statement ถัดไป ทำให้ pid ใน copy ของลูกเป็น 0 เราจึงใช้ค่านี้เป็นตัวแยก code ของแม่ และ ลูก หลังจากการ fork()

เนื่องจากการเลือกโปรเซสเข้าไปครอบครองซีพียูเป็นหน้าที่ของโอเอส กล่าวคือลำดับการทำงานระหว่างโปรเซสแม่ กับ โปรเซสลูกไม่จำเป็นต้องแม่ได้ครอบครองซีพียูก่อนลูกเสมอ

Q1 จาก code ที่ให้ เขียน output 3 แบบที่เป็นไปได้

out put 1:

my copy of i is 1

my copy of i is 2

out put 2:

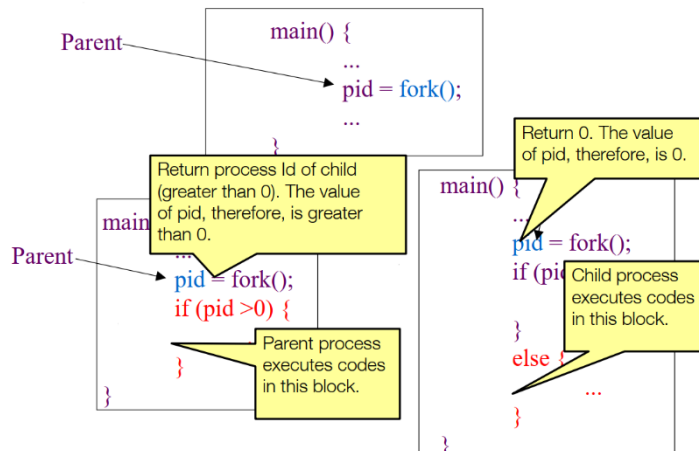
my copy of i is 2

my copy of i is 1

out put 3:

my copy of i is 1

Diagram for fork() example



```

1  #include <stdio.h>
2  #include <unistd.h>
3  #include <sys/types.h>
4
5
6  int main(/* int argv, char** argc */) {
7      pid_t pid;
8      int i = 0;          //prevent garbage
9
10     pid = fork();
11     if (pid > 0) { //parent
12         i = 1;
13         printf("my copy of i is %d\n",i)
14     } else {          //child
15         i = 2;
16         printf("my copy of i is %d\n",i)
17     }
18
19
20     return 0;
21 }
```

4.2 fork() and wait()

จึงมี system call wait() เพื่อให้ นักพัฒนาโปรแกรมสร้างกลไกเพื่อคุมจังหวะการทำงานได้ระดับหนึ่ง โดยแม่จะเป็นผู้เรียก เมื่อเรียก wait() แล้ว แม่จะรอให้ลูกจบการทำงานจึงจะไปทำงานต่อที่ส่วนของโปรแกรมหลัง wait()

ศึกษาว่า child หลุดจาก wait() ได้อย่างไร ที่

<https://man7.org/linux/man-pages/man2/waitpid.2.html>

Q2 จาก code ที่ให้ output มีได้ กี่ แบบ อะไรบ้าง

2 แบบ

out put 1:

my copy of i is 1

my copy of i is 2

out put 2:

my copy of i is 2

my copy of i is 1

```
1  #include <stdio.h>
2  #include <unistd.h>
3  #include <sys/types.h>
4  #include <sys/wait.h>
5
6  int main(void) {
7      pid_t pid;
8      int i;
9
10     pid = fork();
11     if (pid > 0 ) { //parent
12         i = 1;
13         printf("my copy of i is %d\n",i)
14     } else {        //child
15         i = 2;
16         printf("my copy of i is %d\n",i)
17     }
18
19     wait(NULL);
20     return 0;
21 }
```

RETURN VALUE top

wait(): on success, returns the process ID of the terminated child; on error, -1 is returned.

waitpid(): on success, returns the process ID of the child whose state has changed; if WNOHANG was specified and one or more child(ren) specified by pid exist, but have not yet changed state, then 0 is returned. On error, -1 is returned.

waitid(): returns 0 on success or if WNOHANG was specified and no child(ren) specified by id has yet changed state; on error, -1 is returned.

4.3 fork() wait() and exit()

เมื่อใช้ if ในการแยก code ส่วนของแม่กับลูก หากมีโค้ดต่อจาก if ลูกย่อมนำไปด้วย ซึ่งปกติไม่ใช่สิ่งที่นักพัฒนาโปรแกรมต้องการ จึงเรียก exit() เพื่อให้ลูกจบการทำงาน เมื่อทำงานของตนเสร็จ

```
lab4_code > C lab4_31.c
1  #include <stdio.h>
2  #include <unistd.h>
3  #include <sys/types.h>
4  #include <sys/wait.h>
5
6  int main(void) {
7      pid_t pid;
8      int i; int sum = 3;
9
10     pid = fork();
11     if (pid > 0) { //parent
12         i = 1; sum += i;
13         printf("my copy of i is %d\n",i)
14     } else { //child
15         i = 2; sum += i;
16         printf("my copy of i is %d\n",i)
17     }
18     print("my sum = %d\n",sum)
19     wait(NULL);
20     return 0;
21 }
```

```
ab4_code > C lab4_32.c
1  #include <stdio.h>
2  #include <unistd.h>
3  #include <sys/types.h>
4  #include <sys/wait.h>
5
6  int main(void) {
7      pid_t pid;
8      int i; int sum = 3;
9      for (i = 0; i < 3; i++) {
10         pid = fork();
11         if (pid == 0) { //child
12             printf("my copy of i is %d\n",i)
13             exit(0);
14             printf("should not be executed\n")
15         }
16     }
17     while (wait(NULL) != -1); //empty loop
18     print("bye from main \n",sum)
19     return 0;
20 }
```

Q3.1 4_31 บรรทัดที่ 18 ทำงานกี่ครั้ง 2 ครั้ง

Q3.2 4_32 บรรทัดที่ 18 ทำงานกี่ครั้ง 1 ครั้ง