

实验一 多进程实验

姓名：吴侃 学号：14348134

班级：14 级计算机系一班 邮箱：wkc@live.cn

进程创建的例 1

屏幕输出两行，依次是 1 和 0

解释：调用 fork() 函数时，系统创建新进程并为其分配资源，然后将原进程的所有值都复制到新进程中，只有少数值和原来进程的值不同。

父进程的 fork() 返回的是子进程的 id 值

子进程中的 fork() 返回的是 0;

参考资料：<http://blog.csdn.net/jason314/article/details/5640969>

进程创建的例 2

屏幕输出了：

```
**1**  
**3**  
**1**  
**2**  
**2**
```

分析：

```
1 void main(){  
2  int pid1=fork();  
3  printf("***1**\n");  
4  if(pid1==0){int pid2=fork();printf("***2**\n");}  
5  else printf("***3**\n");  
6 }
```

记父进程为 A

第 2 行的 fork() 创建了一个子进程 b

对于进程 A, fork() 返回的值非 0,

在第 3 行输出 **1**

经过第 4 行条件判断，转到 else 分支，输出 **3**

对于进程 b, fork() 返回的值为 0

到第 3 行，输出 **1**

第 4 行，条件判断后转到 yes 分支，

执行 `int pid2 = fork();`
创建一个子进程 c

进程 b 输出 `**2**`

进程 c 输出 `**2**`

进程创建的例 3

输出了 XXXX0

4 个 X, 1 个 0

分析:

```
1 int i, a=0;
2 pid_t pid;
3 if((pid=fork()))a=1;
4 for(i=0; i<2; i++){
5     printf("X");
6 }
7 if(pid==0)printf("%d\n",a);
```

记父进程为 A,

当执行到第 3 行时, 创建一个子进程 b

父进程 A 中, `pid=id of process b`, 不为 0

因此 `a = 1`

进入第 4 行循环, 输出 2 个 X

第 7 行中不满足条件

子进程 b 中, `pid=0`, 不满足第 3 行的条件, a 仍为 0

在第 4 行的循环中输出 2 个 X

在第 7 行满足条件, 输出 `0\n`