实验二 Linux进程控制1

**学号： 20020006107 姓名： 王义钧 年级： 2020**

**一.使用fork系统调用创建多个子进程**

**1.具体要求与步骤**

1. 编写一C/C++语言程序（程序名为fork.c/fork.cpp），使用系统调用fork()创建两个子进程。当程序运行时，系统中有一个父进程和两个子进程在并发执行。父亲进程执行时屏幕显示“I am father”，儿子进程执行时屏幕显示“I am son”，女儿进程执行时屏幕显示“I am daughter”。
2. 多次连续反复运行这个程序，观察屏幕显示结果的顺序，直至出现不一样的情况为止。记下这种情况，试简单分析其原因。
3. 可以使用实验报告模板中所推荐的代码实现，但是要求为代码添加注释，对代码关键逻辑步骤进行解释。在代码头部添加如代码1所示式样的头部版权声明。使用星号、井号、等号、破折号等各类符号对版权声明添加边框，并拼出19os的式样。

**2.实验结果截图**

**结果分析：**

①创建 ~/Father\_Son/fork.c文件，写入文件并添加头部文件注释：

②使用命令 gcc fork.c -o fork 对源文件进行编译。

③执行编译后的文件 ./fork。

连续反复执行多次，会发现 “I am son” 和 “I an daughter” 的出现顺序出现了变化。在大多数执行结果中，“I am son”先执行，而在某些特殊情况下，“I an daughter”先执行。

分析原因：进程的并行执行。

程序首先打印了父进程的创建，优先于pid1和pid2，故先输出“I am father”，紧接着父进程执行第一个fork函数来创建儿子进程，fork函数返回的是该子进程的PID，故肯定不为0。创建完子进程后，父进程与子进程并发执行：父进程执行的是创建女儿进程，而子进程执行的操作是判断儿子进程是否在执行。故而在父进程创建完女儿进程后，父进程，子进程，女儿进程都要执行打印到屏幕的操作，但屏幕只有一个，所以这三个进程会进行竞争，会出现两种情况：

1. 父进程与子进程并发执行，子进程先抢到处理机内容，进行判断，并输出“I am son”，然后交由父进程获得处理机资源，创建女儿进程，女儿进程获得处理机资源，进行判断，并输出“I an daughter”。

此时输出情况为：

“I am father”

“I am son”

“I an daughter”

1. 父进程与子进程并发执行，父进程先获得处理机资源，并创建女儿进程，进行判断，输出“I an daughter”，然后交付子进程获得处理机资源，创建子进程并进行判断，输出“I am son”。

此时输出情况为：

“I am father”

“I an daughter”

“I am son”

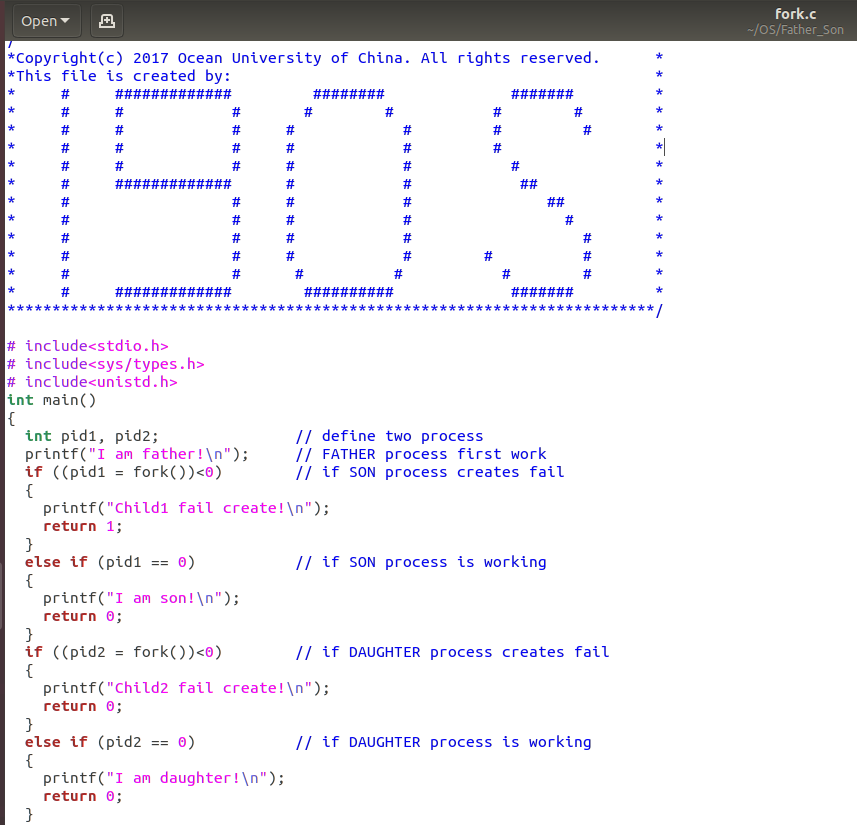


图 1 代码及其头部注释

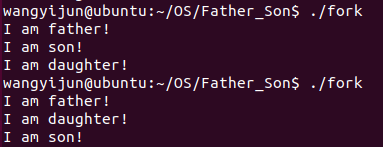


图 2 程序运行的不同结果

**二.使用fork系统调用创建多级子进程**

**1.具体要求与步骤**

1. 编写一C/C++语言程序（程序名为fork.c），使用系统调用fork( )创建一个子进程，然后在子进程中再创建子子进程。当程序运行时，系统中有一个父进程、一个子进程和一个孙子进程在并发执行。父亲进程执行时屏幕显示“I am father”，儿子进程执行时屏幕显示“I am son”，孙子进程执行时屏幕显示“c”。
2. 多次连续反复运行这个程序，观察屏幕显示结果的顺序，直至出现不一样的情况为止。记下这种情况，试简单分析其原因。
3. 可以使用实验报告模板中所推荐的代码实现，但是要求为代码添加注释，对代码关键逻辑步骤进行解释。在代码头部添加如代码1所示式样的头部版权声明。使用星号、井号、等号、破折号等各类符号对版权声明添加边框，并拼出19os的式样。
4. **实验结果截图**

**结果分析：**

按照一中的步骤进行连续编译运行，发现孙子进程与子进程的调用顺序不同，首先，程序优先打印父进程的创建，然后在父进程中创建子进程，其返回值也一定不为0，而子进程中需要做两件事：

①创建孙子进程

②判断子进程是否在执行，如果是，则向屏幕打印“I am son”。

而在孙子进程中，需要判断孙子进程是否在执行，如果是，则向屏幕打印“c”。

因此出现了子进程与孙子进程之间相互竞争临界资源的问题。

如果在子进程创建完孙子进程后，孙子进程便直接抢占了临界资源，那么便会先向屏幕打印“c”，孙子进程释放掉资源后，子进程才会得到临界资源，打印“I am son”。

此时输出结果为

“I am father”

“I am son”

“c”

如果子进程先抢占到了临界资源，便会优先打印“I am son”，然后再把资源分配给孙子进程。

此时输出结果为

“I am father”

“c”

“I am son”

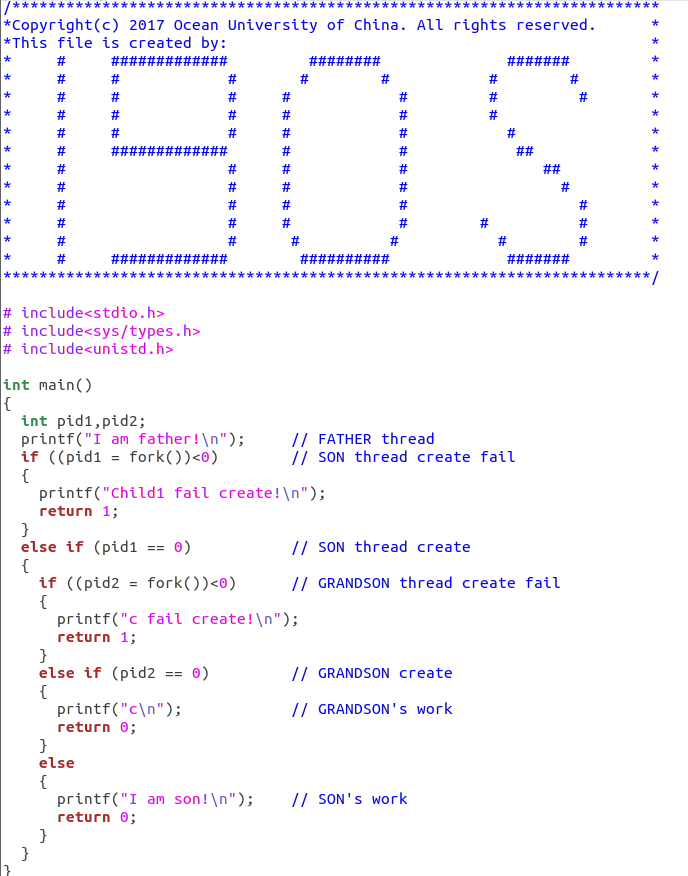


图 3 代码及其头部注释

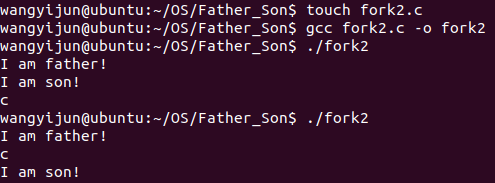


图 4 程序运行的不同结果