**操 作 系 统**

**实 验 报 告**

**实验名称：实验一 进程的创建实验**

**姓名： 王迎旭**

**学号： 16340226**

实验名称：**进程的创建实验**

1. **实验目的：**

加深对进程概念的理解，明确进程和程序的区别。进一步认识并发执行的实质。

1. **实验要求：**

认识进程生成的过程，学会使用fork生成子进程，并知道如何使子进程完成与父进程不同的工作。

**三、实验过程：**

1. **编译并运行PPT中所给C代码并解释现象**

这周的作业主要是尝试使用fork创建简单的进程并且通过在linux环境下编译运行，进而去了解进程之间的相互关系。

所以需要先了解fork这个函数的基本特性，如下：

1. 调用fork()可以创建进程。不过，fork是把进程当前的情况拷贝一份，执行fork时，进程只拷贝下一个要执行的代码到新的进程。

2. fork调用的一个奇妙之处就是它仅仅被调用一次，却能够返回两次，它可能有三种不同的返回值：

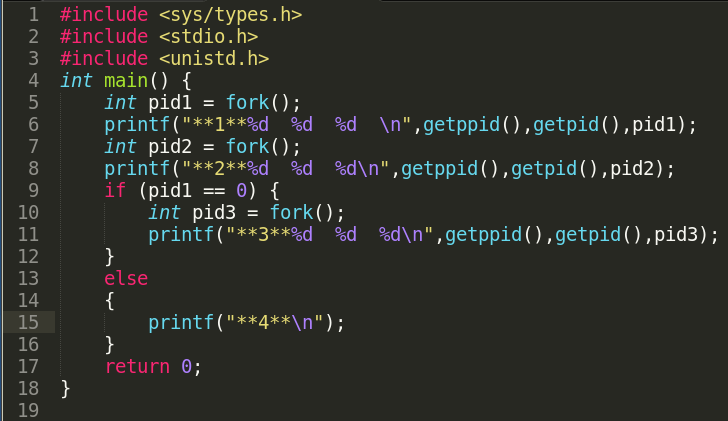
　  ① 在父进程中，fork返回新创建子进程的进程ID；  
    ② 在子进程中，fork返回0；  
    ③ 如果出现错误，fork返回一个负值；

3.fork出错可能有两种原因：

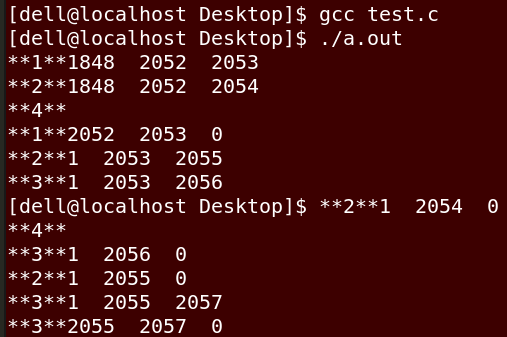
    ① 当前的进程数已经达到了系统规定的上限，这时errno的值被设置为EAGAIN。  
    ② 系统内存不足，这时errno的值被设置为ENOMEM。

在完成实验时候为了更好的追踪进程之间的关系，我使用了**getppid()查询父进程函数**与**getpid()查询当前进程函数**，同时也**输出相应的子进程的编号**，所以对程序也进行了相应的修改。

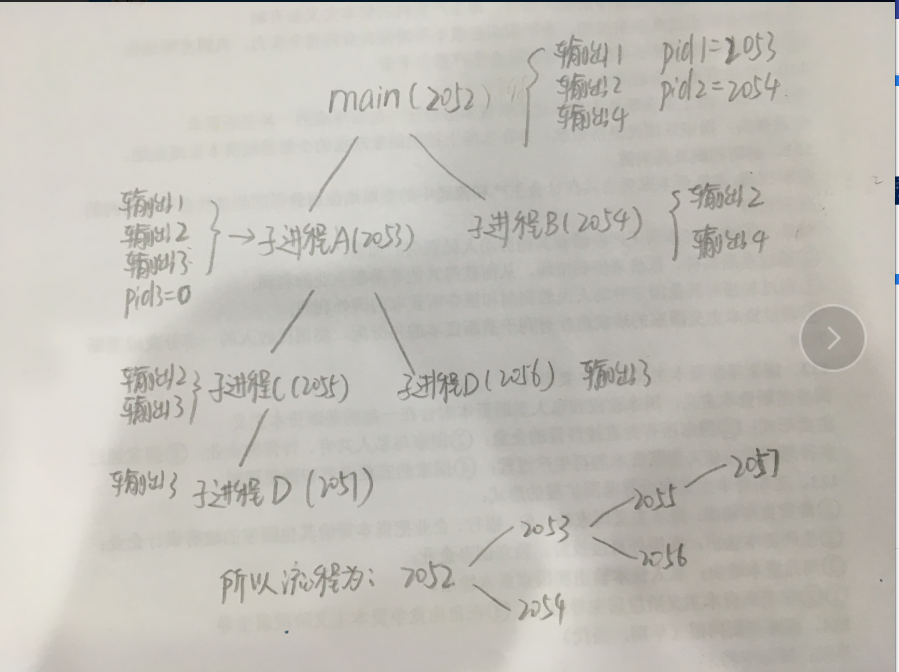
修改后如下：



编译运行之后命令框显示如下：



所以这样就很容易可以画出进程之间的相互联系关系：



相关解释：

①

Ⅰ、程序开始main函数执行：

创建进程A,输出1，pid1 = 2053，创建进程B ，pid2 = 2054 ，输出2 ，此时pid1 ！= 0 ，所以输出4 ，main函数结束。

Ⅱ、进程A开始执行，此时pid1 = 0

输出1，创建进程AB，pid2 = 2055，输出2，由于pid1 = 0 ，所以创建进程ABC1，输出3，进程A结束

Ⅲ、进程B开始执行，此时pid1 = 2053，pid2 = 0

输出2，输出 4 ，进程B结束

Ⅳ、进程ABC1开始执行

输出3，进程ABC1结束

Ⅴ、进程AB开始执行，此时pid1 = 0 ，pid2 = 0

输出2，创建进程ABC2，输出3，进程AB结束

Ⅵ、进程ABC2开始执行

输出3，进程ABC2结束

Ⅶ、所以输出顺序为 124123243233

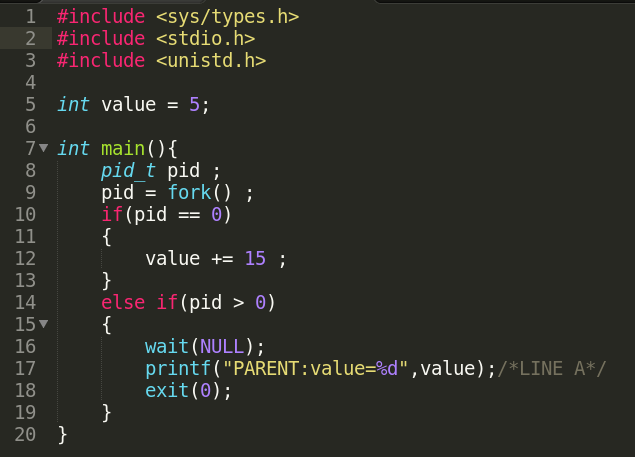
② 观察截图会发现，进程2053、2055、2056的父进程是1，并不是2053对应的2052以及2055与2056对应的2053，这是因为2053执行完之后，这个main已经结束了，所以整个进程相当于已经结束也就是父进程已经死亡，但是子进程一定要有对应的父进程，所以系统默认给这三个子进程的父进程设为1.

③ 观察截图还会发现，有些进程的子进程为0，这是因为这个进程并不会生成子进程，所以就置为0.

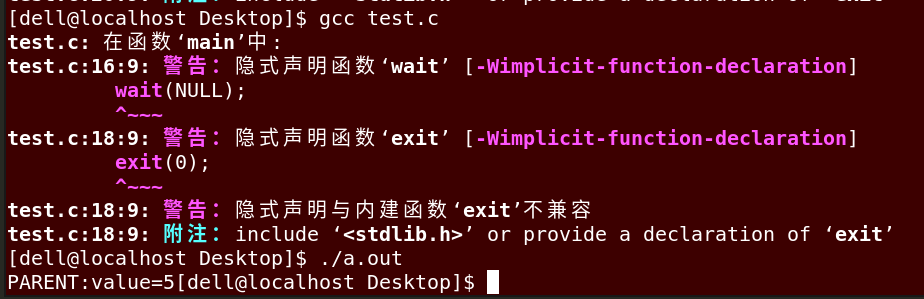
④ 同时我发现了一个问题，就是多次执行可执行文件，输出123的总个数不变，但是输出123的顺序却不同，这说明所有的进程都是可以被执行的，但是进程被执行的顺序是有差异的；刚开始做到这个地方时候不太懂为什么同样的可执行程序输出结果不同，后来查询资料之后发现，进程之间存在竞争，并且这个存在于进程被创建之后的运行过程中，竞争体系的存在也就导致了执行可执行文件时候输出的结果不相同。

1. **通过实验完成教材上的习题3.4**

代码：



运行结果：



相关解释：

① 由于题中给的头文件与所要使用的函数搭配出了点冲突，所以会出现警告信息，但是并不影响程序的正常运行。

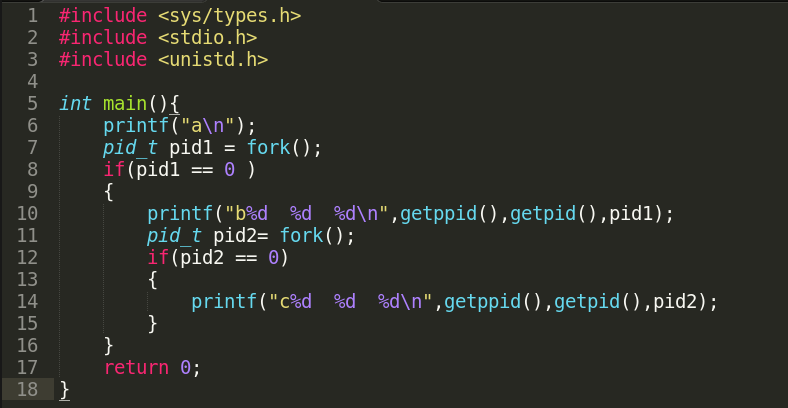
② 至于输出value = 5，是因为当程序的控制权返回到父进程的时候，这个value的值仍会保持为进入子进程之前的那个初始值。

③ 这里我把value放在了main函数中再次进行了编译并运行（如图），发现输出的结果仍然是5，这就证明不管是全局变量还是局部变量，子进程对参数更改之后，父进程使用参数时候仍然是使用的参数的初始值。



1. **编写一段程序，使用系统调用fork()创建两个子进程。当此程序运行时，在系统中有一个父进程和两个子进程活动。让每一个进程在屏幕上显示一个字符；父进程显示字符“a”；子进程分别显示字符“b”和字符“c”。试观察记录屏幕上的显示结果，并分析原因。**

代码:



运行结果:



相关解释：

1. main函数执行，输出a ，创建进程A，pid1 = 2214 ，main结束
2. 进程A执行 ，输出b，创建进程AB，此时pid1 = 0 ，pid2 = 2015，进程结束
3. 进程AB执行，pid1 = 0 ， pid2 = 0 ， 输出c ，进程AB结束

**四、实验感想：**

① 第一次研究进程之间的相互联系，如果没有借助两个辅助函数帮助debug，真的是十分难以寻找对应的父子进程关系。

② 即使是完成了此次实验作业，但是仍然还是要对进程之间的联系做进一步研究，力求达到不使用辅助函数就可以搞清彼此之间的关系。

③ 作业布置的那天就抽了时间做完了三个题写下了前两个感想，但是感觉收获并不是很大，隔了大概3天又重新做了一遍，这次没有借助辅助函数，但是这次靠着自己的理解弄清楚程序执行的顺序以及机理，同时也对进程之间的关系有了更深层次的理解。**不过进程之间的竞争问题，还是没有搞清楚**