实验三 Linux进程控制2

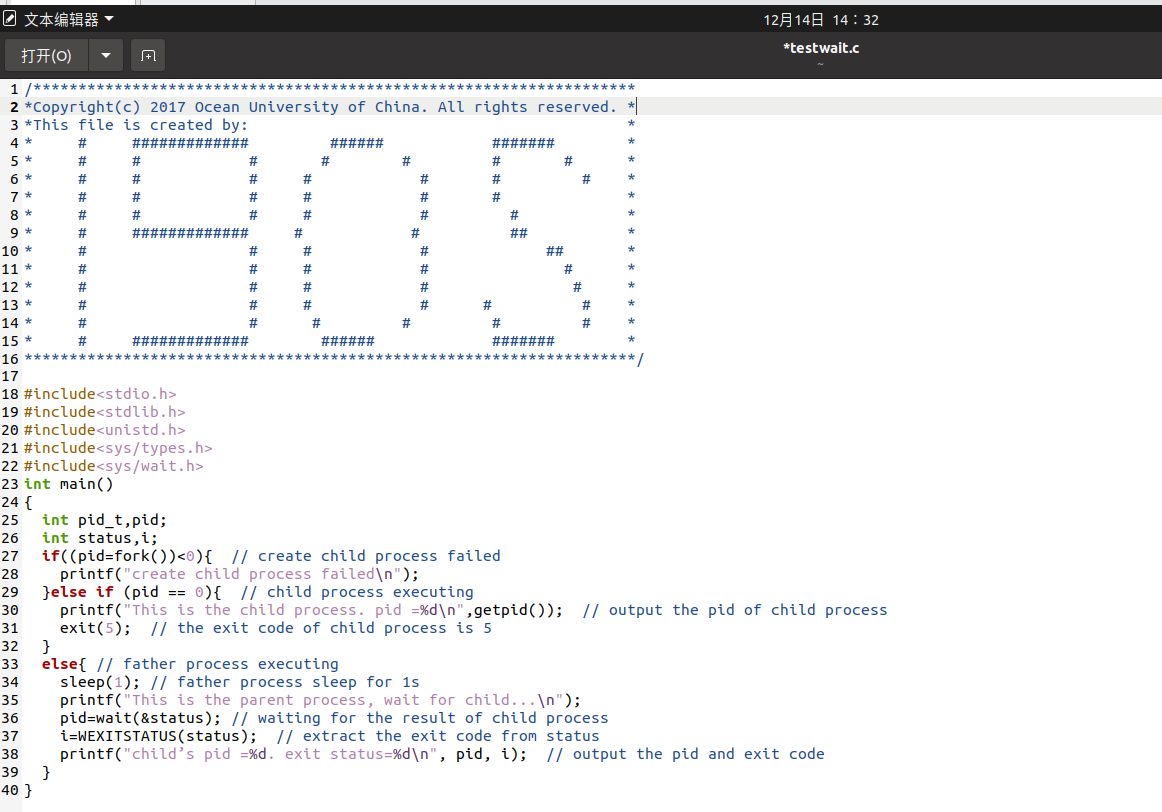
**学号： 19020011038 姓名： 岳宇轩 年级： 2019**

**一.使用wait系统调用**

**1.实验要求**

1. 编写一C/C++语言程序（程序名为testwait.c/testwait.cpp），使用系统调用wait()控制进程的阻塞。
2. 多次连续反复运行这个程序，观察屏幕显示结果的顺序，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，并拼出自己汉字名字的式样。

**2.实验结果截图**

图 1 wait函数代码

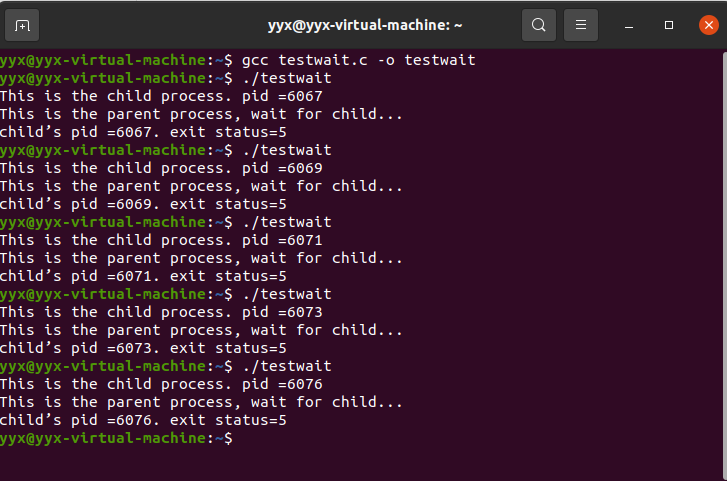


图 2 wait函数实验结果

结果分析：

父进程首先创建了一个子进程，然后sleep 一秒，目的是留给子进程完成工作的时间。

子进程的工作是利用getpid()函数输出自己的进程号，然后结束自己，并设置退出时的参数为5.这一过程在父进程的sleep 1s内完成。

父进程在sleep 1s之后，执行wait函数。由于sleep一秒的时间之内，子进程可以完成其工作并结束，故wait()会立即返回子进程结束状态值，再用WEXISTATUS()函数获取子进程退出时的状态值，接着将其输出。

尝试情况二：

如果强制增加子进程完成任务所需要的时间，比如在子进程中加入一个sleep(3)，如图3所示。那么可以看到，输出顺序发生了改变，如图4所示。

这时，父进程在sleep 1s之后，子进程并没有完成工作，故父进程的wait()函数将父进程阻塞，直到子进程完成，发出信号，父进程才被唤醒。

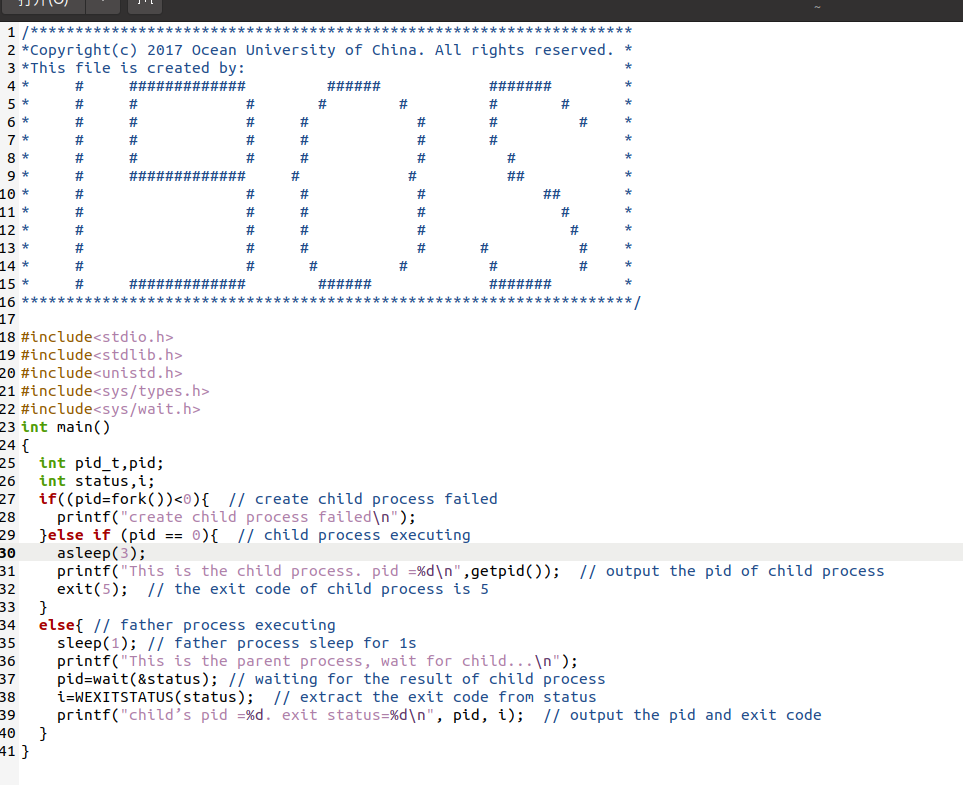


图 3 wait函数代码2

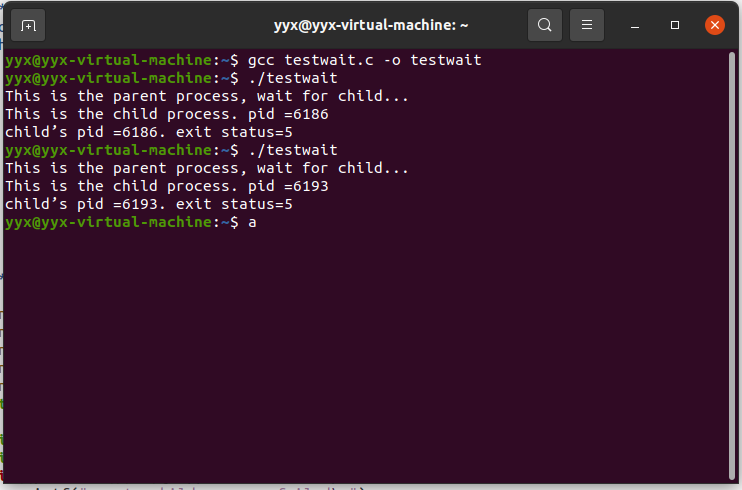


图 4 wait函数实验结果2

**二.使用exit系统调用**

**1.实验要求**

1. 编写一C/C++语言程序（程序名为testexit.c/testexit.cpp），使用系统调用exit( )尝试对进程进行终止操作。
2. 多次连续反复运行这个程序，观察屏幕显示结果的顺序，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，并拼出自己汉字名字的式样。

**2.实验结果截图**

图 5 exit函数代码

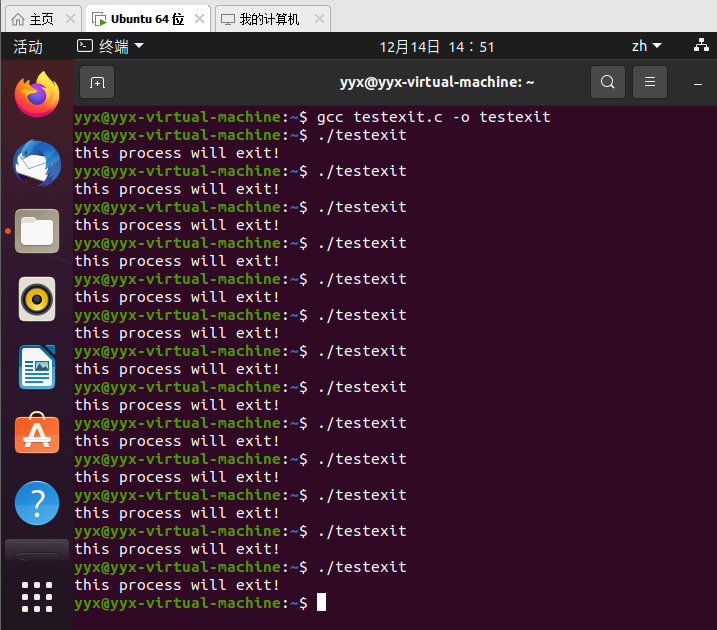


图 6 exit函数代码

结果分析：

父进程首先输出了一句”this process will exit!”。接着结束掉了自己，故之后的的“never be displayed!”将不会被输出。

**三.使用lockf系统调用**

**1.实验要求**

1）编写一C/C++语言程序（程序名为testlockf.c/testlockf.cpp），使用系统调 用lockf( )尝试对程序的指定区域进行加解锁操作。

2）多次连续反复运行这个程序，观察屏幕显示结果的顺序，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，并拼出自己汉字名字的式样。

**2.实验结果截图**

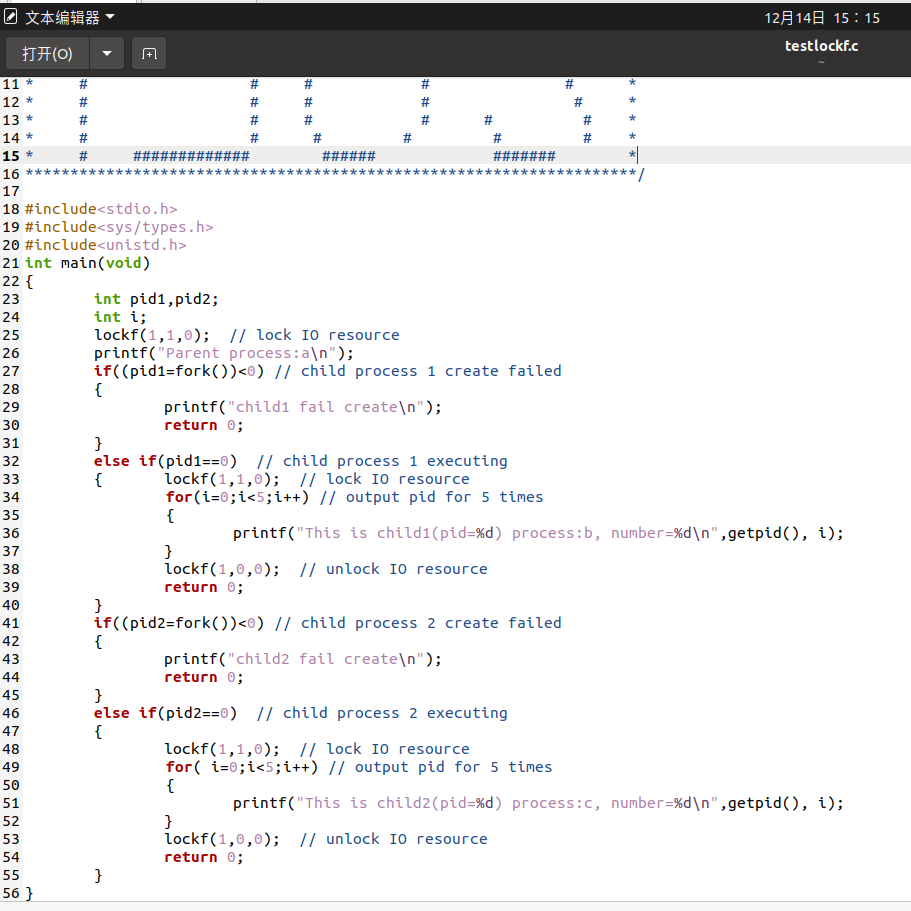
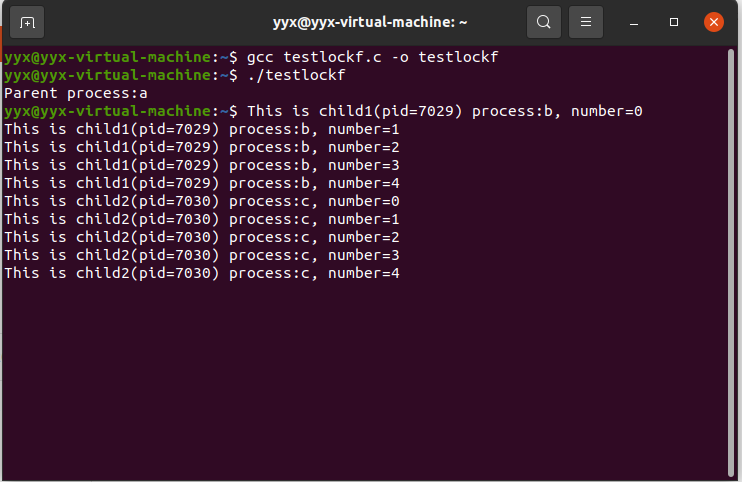


图 7 lockf函数代码

图 8 lockf函数运行结果

实验结果分析：

lockf(1,1,0)的含义是锁定屏幕输出，lockf(1,0,0)的含义是解锁屏幕输出，第一个参数的含义是fd文件；第二个参数，1表示上锁，0表示解锁；第三个参数表示文件范围，0表示全部。

父进程首先锁定了屏幕输出资源，直到父进程结束。故首先输出

Parent process :a

程序中有两个子进程，两个子进程执行相同的操作：1.锁定屏幕输出资源2.循环5次输出自己的pid和循环的number 3.释放屏幕输出资源

考虑进程并发性，两个进程虽然在宏观上是同时进行，但实际上还是有获得处理机的先后顺序的，故输出b,c的顺序有可能是交叉的。但因为b,c进程在一开始就都使用lockf锁定了屏幕输出资源，故即使另一方获得了处理机资源，也无法获得屏幕输出的资源。所以输出b,c的顺序是没有交叉的。

尝试改动如下：

去掉子进程中的lockf，并将循环输出次数均调至500次（增加循环次数的原因是尝试使进程在获得一次处理机资源的时间内不能完成所有输出）。

这时，子进程b,c应当会出现交叉输出的情况。

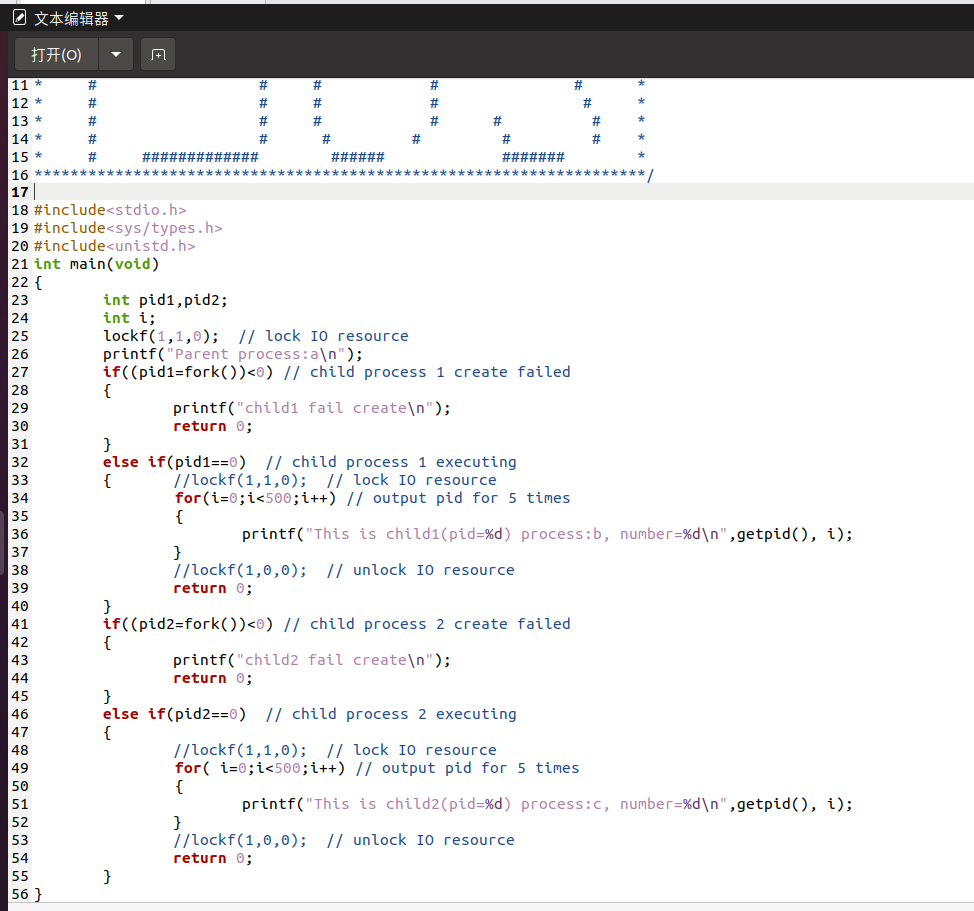


图 9 lockf函数代码2

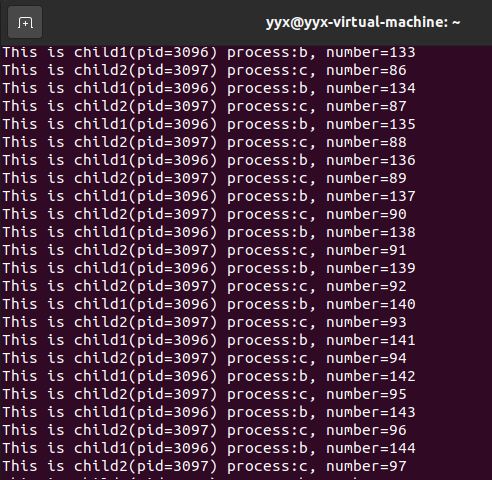


图 10 lockf函数结果2