实验一 进程创建

一、实验目的： 1.学习Linux中创建进程的方法

2.学习系统调用fork()的使用方法

3.学习系统调用getpid()的使用方法

3.学习系统调用exec()族的使用方法

4.学习系统调用wait()的使用方法

5.学习系统调用exit()的使用方法

6.理解父进程和子进程并发执行

二、实验内容

1.将以下代码输入，在linux中运行，观察输出结果，理解程序功能和父子进程并发执行过程。

#include <stdio.h>

#include <unistd.h>

#include <sys/types.h>

int main()

{

pid\_t pid;

/\* fork a child process \*/

pid = fork();

printf(“process id:”,pid);

if (pid < 0) { /\* error occurred \*/

fprintf(stderr, "Fork Failed\n");

return 1;

}

else if (pid == 0) { /\* child process \*/

printf("I am the child %d\n",pid);

execlp("/bin/ls","ls",NULL);

}

else { /\* parent process \*/

/\* parent will wait for the child to complete \*/

printf("I am the parent %d\n",pid);

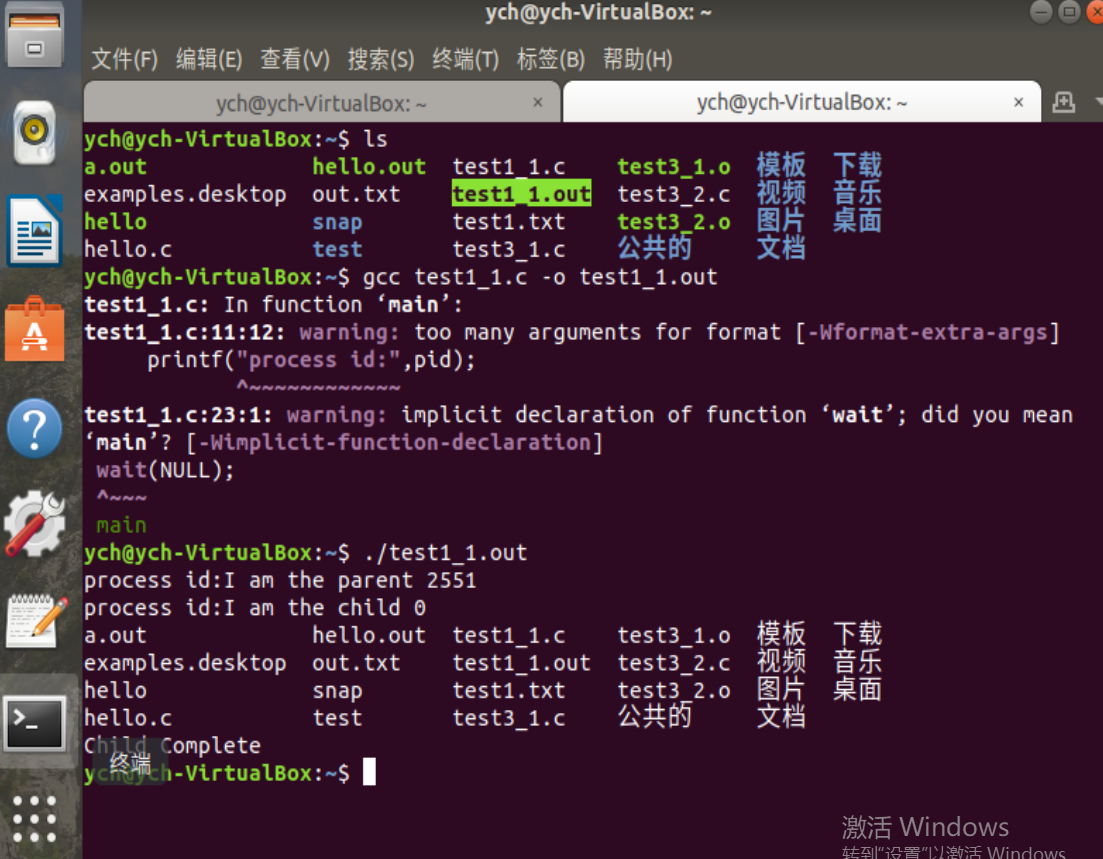
wait(NULL);

printf("Child Complete\n");

}

return 0;

}



2.将以下代码输入，在linux中运行，观察LINE A(即value的)输出结果，理解程序功能和父子进程并发执行过程。

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

int value = 5;

int main()

{

pid\_t pid;

pid = fork();

if (pid == 0) { /\* child process \*/

value += 15;

return 0;

}

else if (pid > 0) { /\* parent process \*/

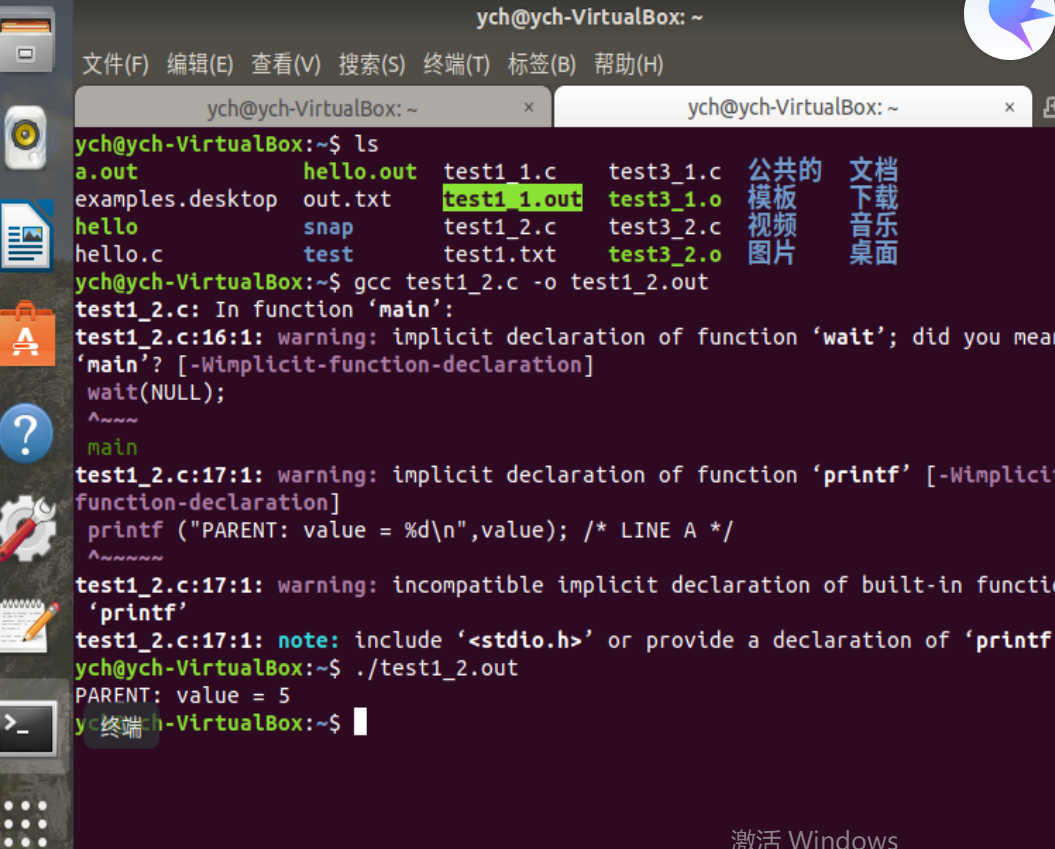
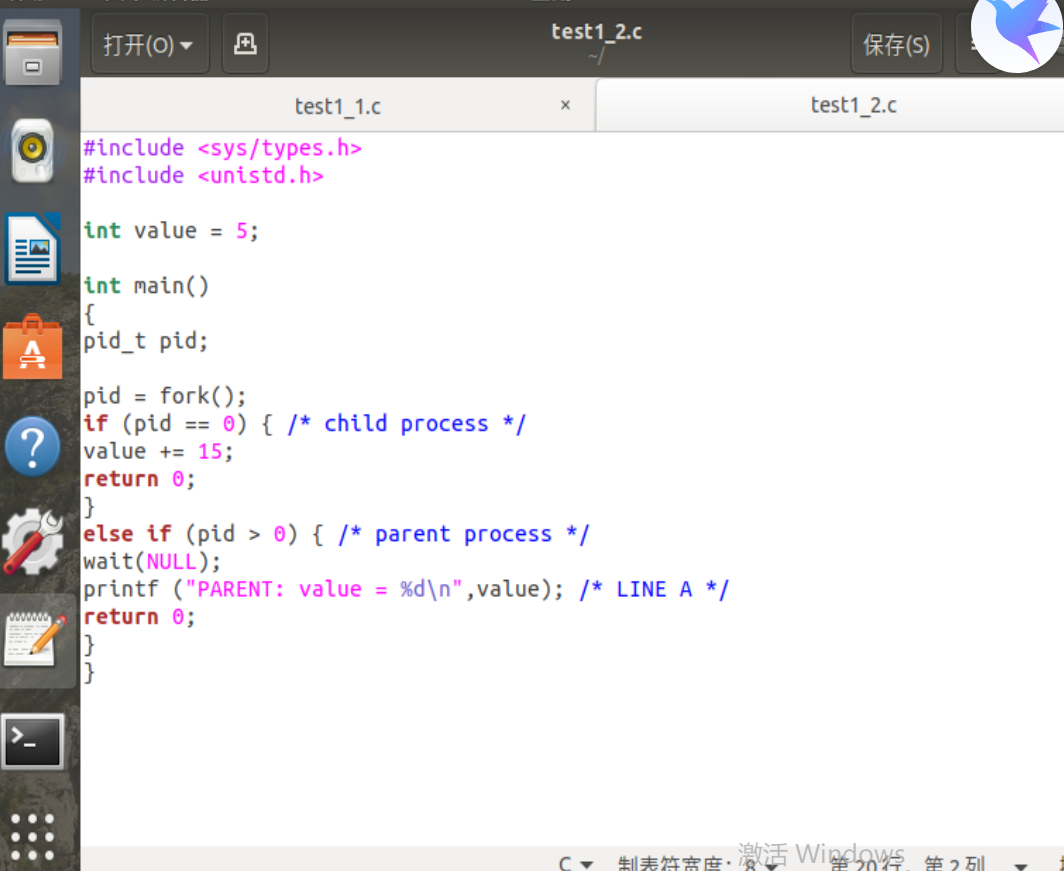
wait(NULL);

printf ("PARENT: value = %d\n",value); /\* LINE A \*/

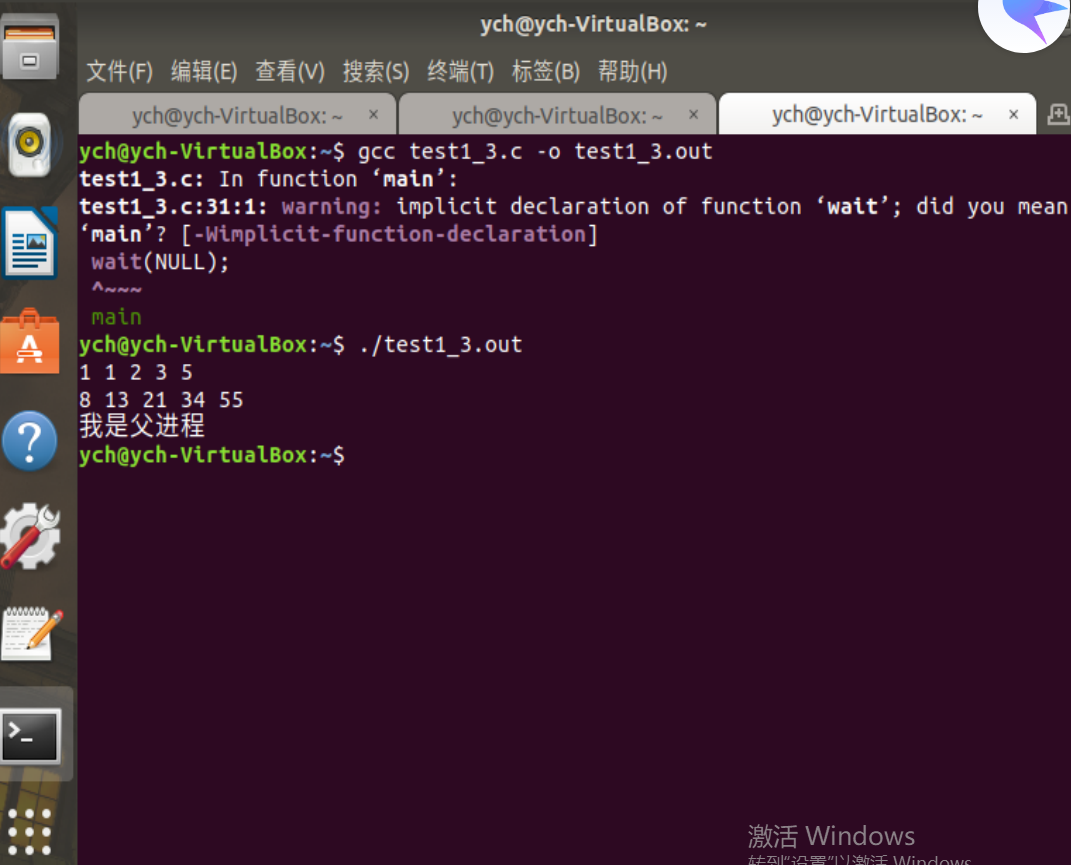
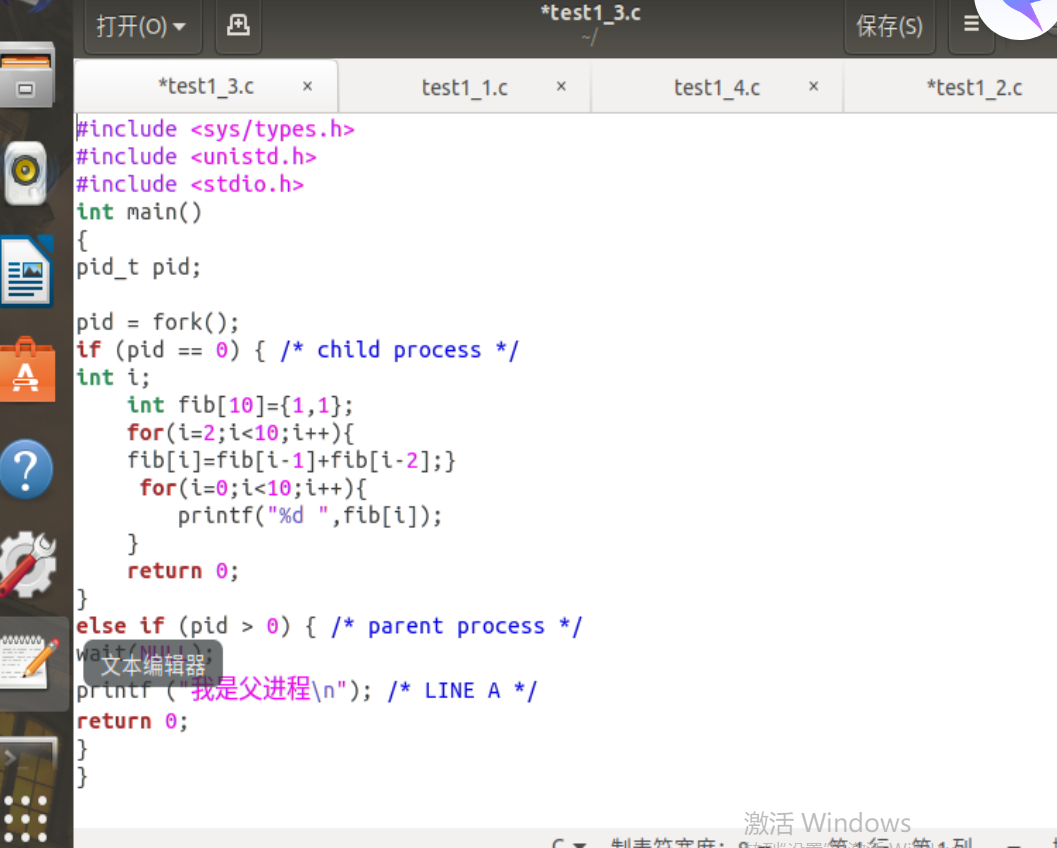
return 0;

}

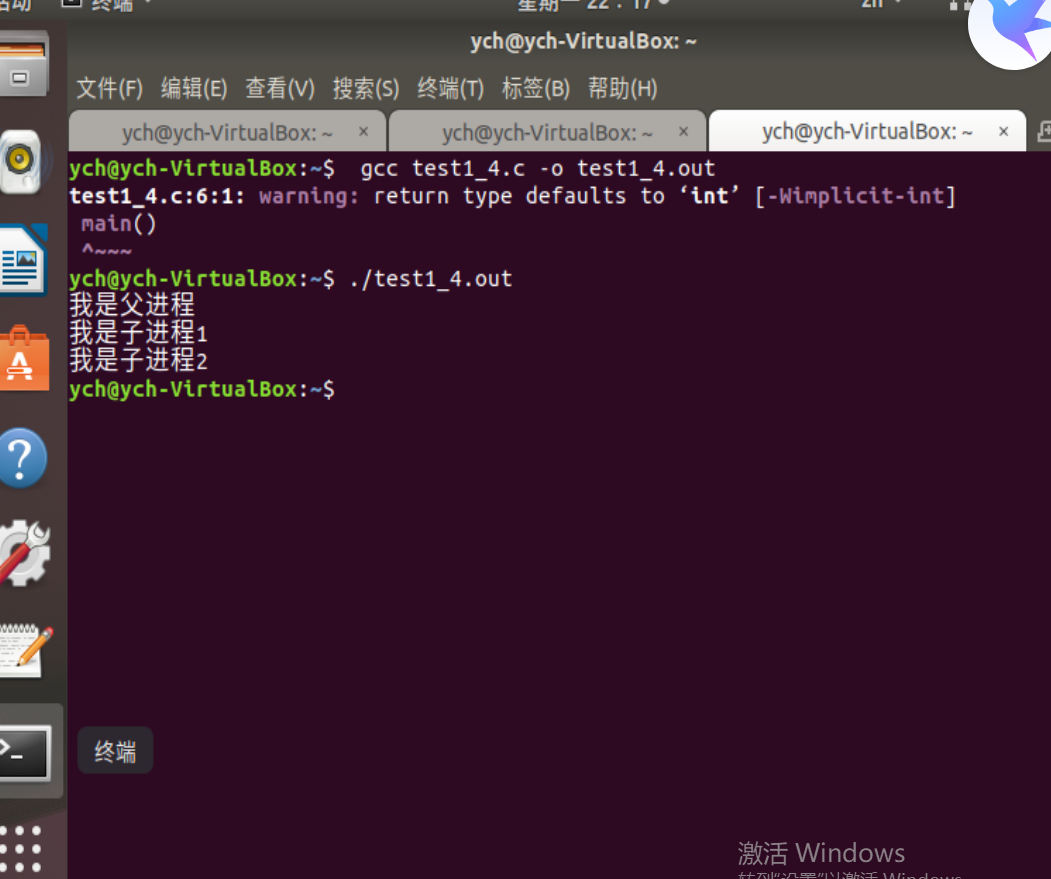
}



3.编写在父进程中创建一个子进程，在子进程中完成10个斐波那契数列的输出。



4.编写一个在父进程中输出我是父进程，在父进程中创建两个子进程，分别输出我是子进程1、我是子进程2。



三、实验要求

1.独立完成

2.按模板撰写实验报告

3.于规定时间提交校网络教学平台