**实验二 进程的控制**

# 姓 名 学 号 成绩

实验时间 指导教师(签名)

**（诚信声明：本实验报告内容，均由本人亲自上机完成。 签名：\_\_\_\_\_\_\_\_\_\_\_\_\_）**

一．实验目的

1.熟悉和理解进程和进程树的概念，掌握有关进程的管理机制

2.通过进程的创建、撤销和运行加深对进程并发执行的理解

3.明确进程与程序、并行与串行执行的区别

4.掌握用C 程序实现进程控制的方法

二．实验工具与设备

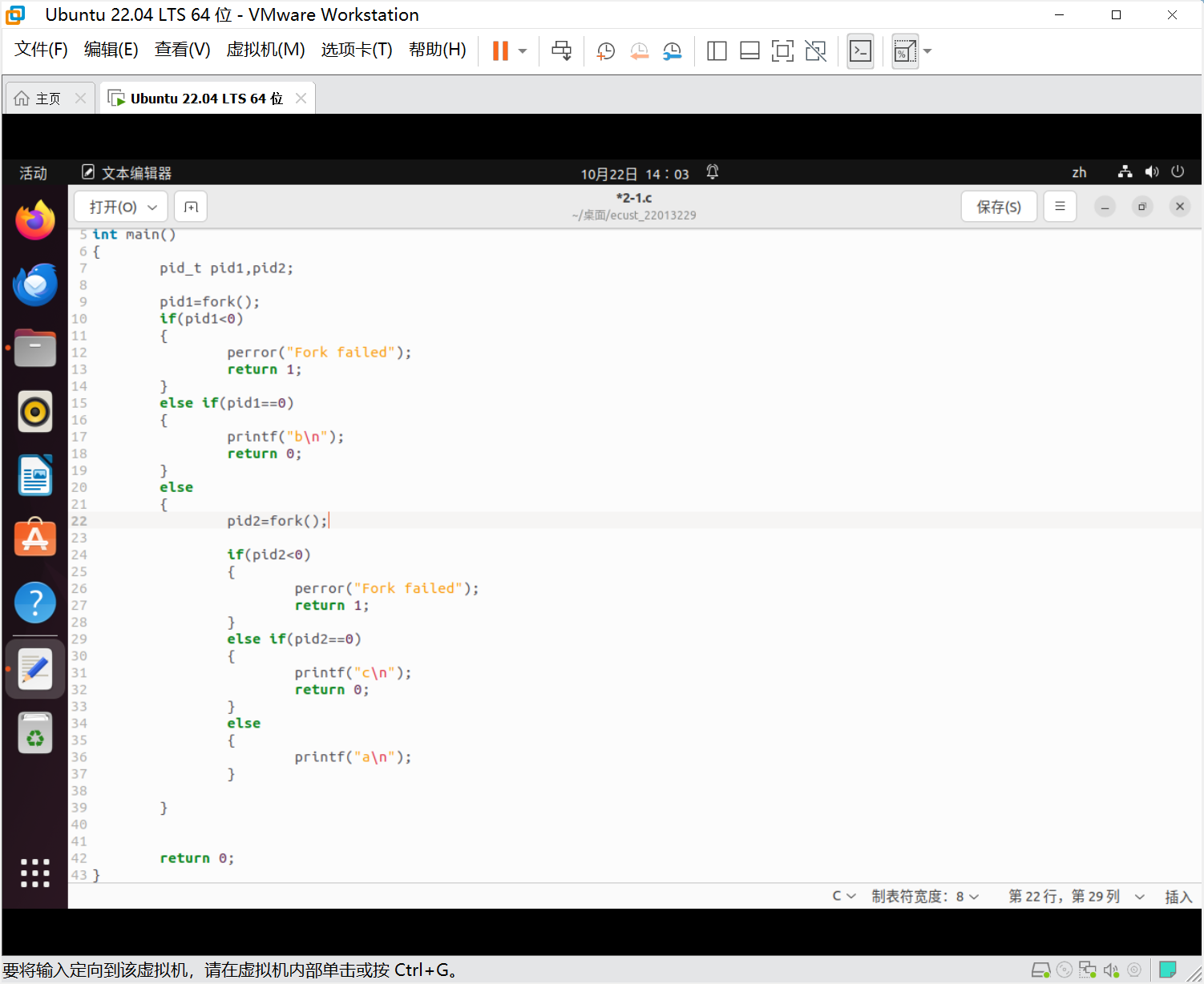
已安装 Linux 操作系统的计算机。

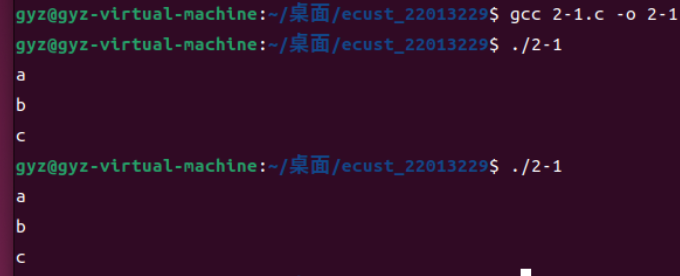
1. 实验内容

## 1. 了解系统调用 fork( ), exec 系列函数, exit( ), waitpid( )的功能和实现过程

## 2．程序设计

* 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’c’。观察屏幕上的显示结果，并分析原因。





输出顺序可能会有所不同，这是由于操作系统在不同时间调度进程的方式导致的。

* 1. 编写一段程序，使用系统调用 fork( )创建一个子进程。子进程通过系统调用 exec 系列函数调用命令 ls，调用 exit( )结束。而父进程则调用 waitpid( )等待子进程结束，并在子进程结束后显示子进程的标识符，然后正常结束。

# 

# 

# 思考题

## 怎样用 C 程序实现进程的控制？当首次调用新创建进程时，其入口在哪里？

（1）进程控制：

使用 fork() 系统调用来创建新进程。fork() 会创建一个与父进程几乎完全相同的子进程。

（2）进程入口：

新创建进程（子进程）的入口点与父进程相同。执行 fork() 后，操作系统会将父进程的代码执行流复制到子进程中。然后根据 fork() 的返回值，程序可以判断当前执行的是父进程还是子进程。

1. **系统调用 fork( )是如何创建进程的？系统调用 exit( )是如何终止一个进程的？**

（1）fork()：

fork() 调用创建一个新的子进程，操作系统在内核中执行以下步骤：

复制当前进程的内存空间（代码、数据、堆栈等）到新的进程。

分配一个新的进程标识符（PID）给子进程。

设置子进程的状态为“就绪”，并将其放入进程调度队列中。

返回两次：在父进程中返回子进程的 PID，在子进程中返回 0。

（2）exit()：

exit() 系统调用用于终止当前进程，具体步骤如下：

释放该进程占用的资源（如内存、打开的文件等）。

更新进程的状态为“终止”。

如果有父进程在等待该进程结束，向父进程发送 SIGCHLD 信号。

通过 exit() 参数传递终止状态，供父进程使用。

1. **系统调用 exec 系列函数是如何更换进程的可执行代码的？**

（1）加载新程序：操作系统会根据给定的程序路径找到新程序的可执行文件，并将其加载到当前进程的内存中。

（2）清空当前进程的地址空间：现有进程的代码、数据和堆栈等内容将被新程序的内容替换。

（3）设置新的入口点：新程序的执行将从其主函数开始（通常是 main 函数），并且所有的全局变量和堆栈将被新程序的初始化值所替换。

（4）返回控制权：如果 exec 调用成功，则该调用不会返回；新程序的代码将开始执行。如果失败（如找不到文件或无执行权限），则会返回 -1，并且原进程继续执行。