**OS project two 实验报告**

**——UNIX Shell and History Feature**

**5140309201**

**黄晟**

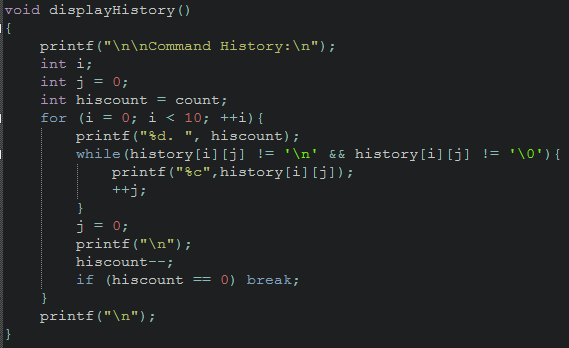
1. **实验要求**

在linux中实现一个简单的shell，能够实现程序的后台执行，更改信号量，并且能够显示最近执行的命令的历史特性。

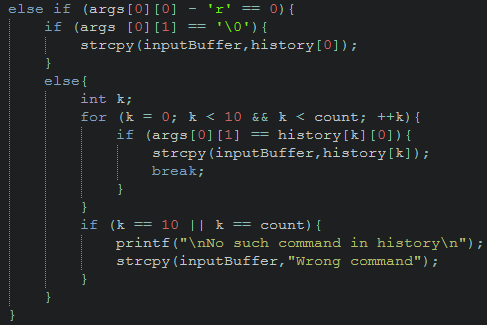
1. **实现过程**
2. 完成setup函数以及程序的执行函数，完成对于新读入的命令的预处理；
3. 定义一个二维数组，用以存放历史进程；



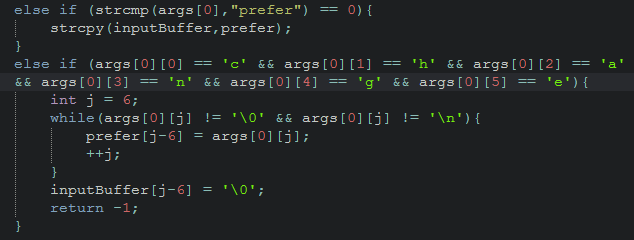
1. 编写完成打印历史特性函数，定义hiscount来记录当前要打印的命令；



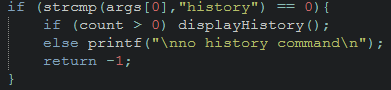
1. 为了完成用r来调用最近执行的指令以及rx调用最近的首字母为x的一条命令，在setup函数读取输入之后加条件判断；



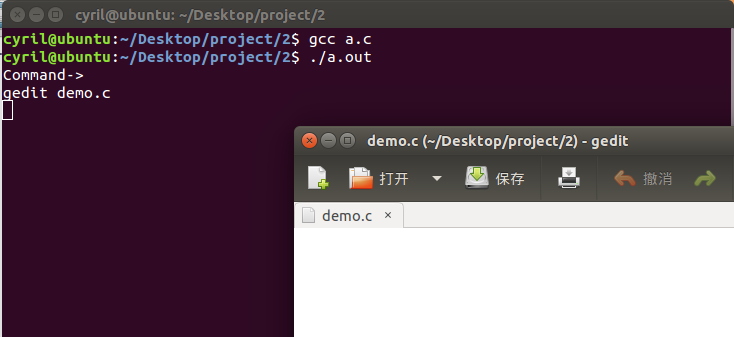
1. 同时，我添加了一项类似于收藏夹的功能，即利用‘prefer’快速调用用户想长期的某项指令，若用户想替换当前指令时，可以用“change+新的指令”来替换。Prefer初始自动设置为cal；



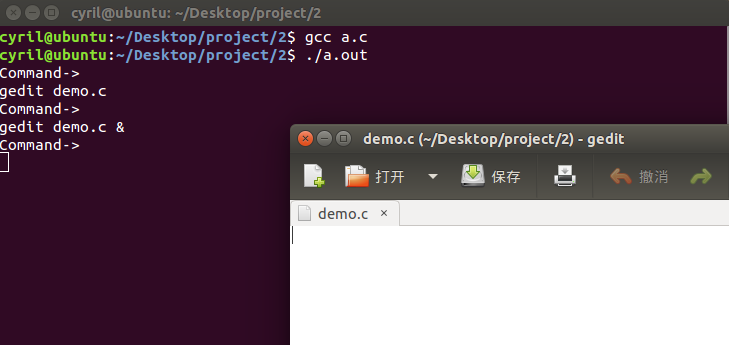
1. 本实验要求采用“ctrl+c”来调用历史特性，但在实际实验过程中，我发现在用户模式下“ctrl+c”很不稳定，所以为了较好的显示历史特性，我又添加了同步调用历史特性的方法，即手动输入history命令输出历史特性。



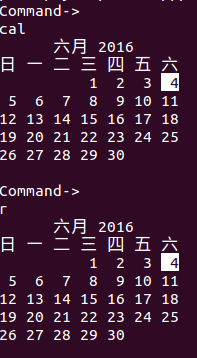
1. **实现效果**
2. 在shell中键入不带‘&’命令时，会等待改程序执行完后再键入新的指令；



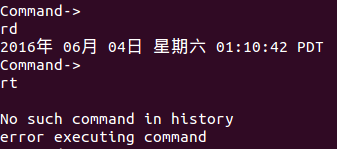
1. 在shell中键入带有‘&’命令时，会将该指令后台执行，shell可以接受新的指令；



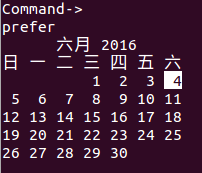
1. 单独输入r时，会执行最近执行的指令；



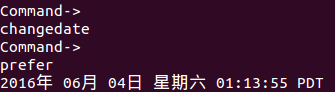
1. 输入r+x时，会搜索并执行最近的以x开头的指令，若没有这样的指令，输出错误提示；



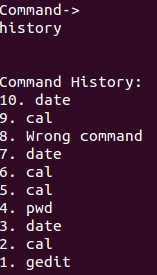
1. 键入prefer后会输出预存指令；



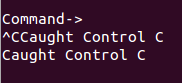
1. 使用change+指令后可以改变prefer，此时再调用prefer会执行新的指令；



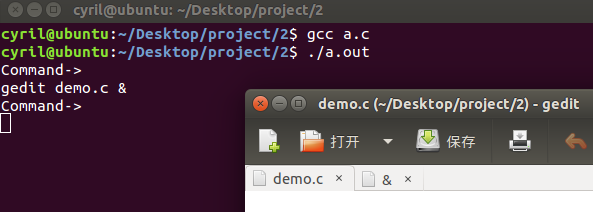
1. 键入history后可以显示历史特性；



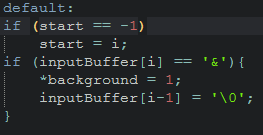
1. 而当使用“ctrl+c”时，shell会变得不稳定，如下图，当我输入异步ctrl+c时，系统会提示接收到两次ctrl+c，因而会同时输出两个历史特性，但只有一个是正确的。



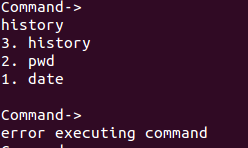
1. **心得与体会**
2. 最初在创建子进程时，等待是用的wait(NULL)，这样就导致了执行过一次后执行程序后，之后的程序就都是后台执行的了。为了解决这个问题，首先我是采用了键入exit命令，手动退出后台执行程序；但是后来觉着这样做不方便，于是将wait(NULL)替换成waitpid(pid,NULL,0)，即等待特定进程。因为wait(NULL)函数虽然会等待，但是它不关心返回状态，所以系统也不知道后台程序什么时候被终止，即之后的程序都是后台执行了。
3. 在执行后台文件时，会多出来一个文件&，如图所示：



这是因为在预处理指令时，检查到&后立即结束，添加‘\0’，然而由于i++后，使得指令检测不到‘\0’了，所以要将inputBuffer[i-1] = '\0';改为inputBuffer[i] = '\0';即可解决该问题。



1. 在执行同步显示历史特性指令history时，会出现如图所示



这是因为history不是shell固有指令，为了执行history并且不再出现错误提示，将setup函数的void改成int，检测到输入history是返回-1，而在创建子进程时若setup函数返回-1，即不执行接下来的fork等语句。