

**软件学院学生实验报告**

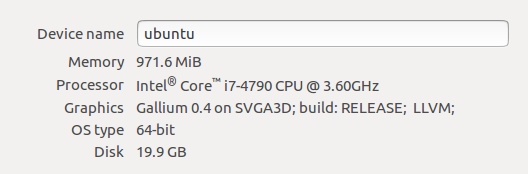
**实验课程名称：**操作系统**教师：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验项目名称** | **命令行解释程序的模拟** | | | **实验成绩** |  |
| **学生姓名** |  | **学号** | **053** | **年级专业班级** |  |
| **小组成员** |  | | | **实验日期** | **2017年10月10日** |

**一、实验目的**

**命令行解释程序的模拟**

**二、实验软硬件环境**



**三、实验内容及要求**

* 使用高级语言编写一个**命令行解释程序**
* **实现**至少三种shell命令

**四、实验记录**

[操作系统](https://baike.baidu.com/item/%25E6%2593%258D%25E4%25BD%259C%25E7%25B3%25BB%25E7%25BB%259F)的主要功能是为管理硬件资源和为[应用程序](https://baike.baidu.com/item/%25E5%25BA%2594%25E7%2594%25A8%25E7%25A8%258B%25E5%25BA%258F)开发人员提供良好的环境来使应用程序具有更好的兼容性，为了达到这个目的，[内核](https://baike.baidu.com/item/%25E5%2586%2585%25E6%25A0%25B8)提供一系列具备预定功能的[多内核](https://baike.baidu.com/item/%25E5%25A4%259A%25E5%2586%2585%25E6%25A0%25B8)[函数](https://baike.baidu.com/item/%25E5%2587%25BD%25E6%2595%25B0)，通过一组称为系统调用（system call)的接口呈现给用户。系统调用把[应用程序](https://baike.baidu.com/item/%25E5%25BA%2594%25E7%2594%25A8%25E7%25A8%258B%25E5%25BA%258F)的请求传给[内核](https://baike.baidu.com/item/%25E5%2586%2585%25E6%25A0%25B8)，调用相应的的内核[函数](https://baike.baidu.com/item/%25E5%2587%25BD%25E6%2595%25B0)完成所需的处理，将处理结果返回给应用程序。

命令介绍

在我们现在使用的linux的终端中，也就是在POSIX标准下，命令分为内部命令和外部命令，当命令解释器遇到内部命令的之后，便直接运行内部命令，也就是说，内部命令是我们自己的实现的。而外部命令，当命令解释器遇到外部命令的时候，直接从系统中找到相应的应用程序，然后开启一个子线程来执行该程序。

下面是我们知道的相应的内部命令：

• exit : 退出该shell程序

• cd : 改变当前的工作目录

• echo : 回显命令后面的字符串

• export : 设置环境变量

• history : 显示历史命令

外部命令

linux下C语言可以用fork()建立子进程。

fork函数返回两个值，对于子进程，返回0; 父进程，返回子进程[ID](https://www.baidu.com/s?wd=ID&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLrH7hmv79mHuBrjcdujbs0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPHTknj6LP1c1). 所以用

if(fork()==0)

{子进程执行的代码段；}

else

{父进程执行的代码段；}

    创建新进程成功后，系统中出现两个基本完全相同的进程，这两个进程执行没有固定的先后顺序，哪个进程先执行要看系统的进程调度策略。

    每个进程都有一个独特（互不相同）的进程标识符（process ID），可以通过getpid（）函数获得，还有一个记录父进程pid的变量，可以通过getppid（）函数获得变量的值。



说是exec系统调用，实际上在Linux中，并不存在一个exec()的函数形式，exec指的是一组函数，一共有6个，分别是：

#include <unistd.h>

extern char \*\*environ;

int execl(const char \*path, const char \*arg, ...);

int execlp(const char \*file, const char \*arg, ...);

int execle(const char \*path, const char \*arg, ..., char \* const envp[]);

int execv(const char \*path, char \*const argv[]);

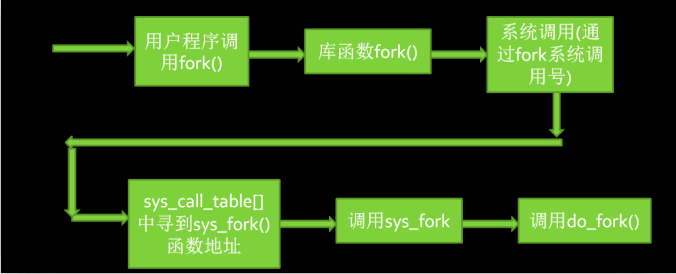
int execvp(const char \*file, char \*const argv[]);

int execve(const char \*path, char \*const argv[], char \*const envp[]);

其中只有execve是真正意义上的[系统调用](https://baike.baidu.com/item/%25E7%25B3%25BB%25E7%25BB%259F%25E8%25B0%2583%25E7%2594%25A8)，其它都是在此基础上经过包装的[库函数](https://baike.baidu.com/item/%25E5%25BA%2593%25E5%2587%25BD%25E6%2595%25B0)。

这里使用execvp，带v可以使用不定长参数，带p是因为没有重新定义路径的必要，而且也更符合shell的语法。

Linux下用于创建进程的API有三个fork,vfork和clone，这三个函数分别是通过系统调用sys\_fork,sys\_vfork以及sys\_clone实现的

（这里目前讨论的都是基于x86架构的）。而且这三个系统调用，都是通过do\_fork来实现的，只是传入了不同的参数。

execve（执行文件）在[父进程](https://baike.baidu.com/item/%25E7%2588%25B6%25E8%25BF%259B%25E7%25A8%258B)中fork一个子进程，在子进程中调用exec函数启动新的程序。exec函数一共有六个，其中execve为内核级[系统调用](https://baike.baidu.com/item/%25E7%25B3%25BB%25E7%25BB%259F%25E8%25B0%2583%25E7%2594%25A8)，其他（execl，execle，execlp，execv，execvp）都是调用execve的[库函数](https://baike.baidu.com/item/%25E5%25BA%2593%25E5%2587%25BD%25E6%2595%25B0)。



外部命令的实现，重点在于

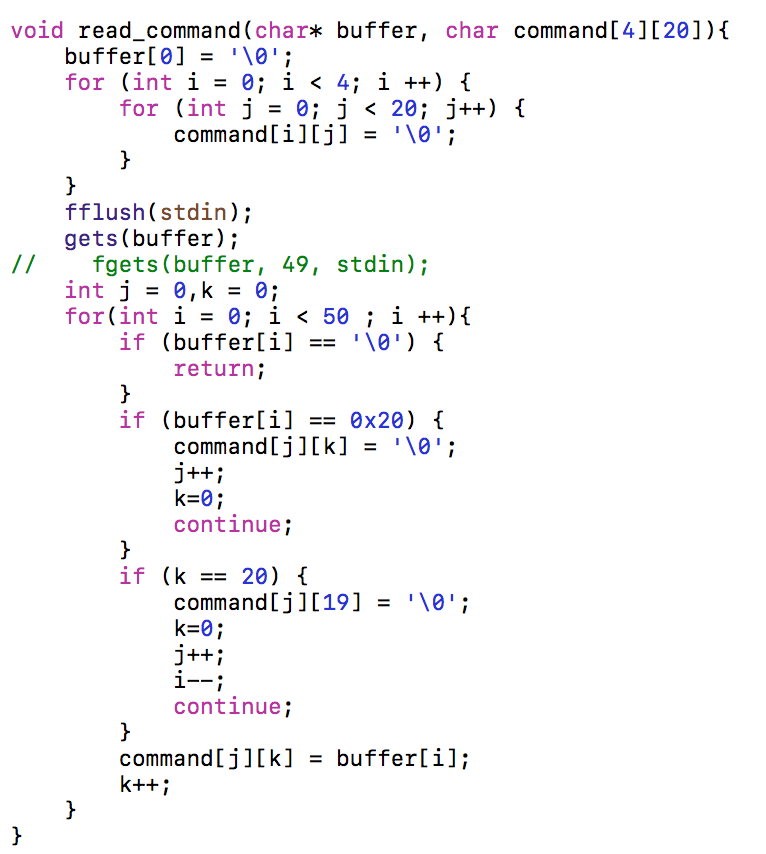
int result = execvp(argument[0],argument);



内部命令

实现shell命令：ls, cd,echo,exit

实现了对一系列命令的分词，以及外部命令的调用，学习exec系列函数的使用，复习了C语言的指针，二维指针，NULL，防止数组溢出和字符串操作。



部分代码参考：[vvy](https://github.com/vvy) / [wshell](https://github.com/vvy/wshell)

 about命令很简单，使用了c函数库的printf()函数，它的定义在文件printf.c中。查看printf()的定义，函数中调用了putc()函数来进行输出，继续跟踪putc()函数的定义，write函数被调用了，继续跟踪write函数，会发现它的声明在user.h中: int write(int,void\*,int)，这是一个系统调用。在Linux系统下，真正的系统调用是sys\_write()，经过简单的封装后sys\_write()又调用了vfs\_write()，vfs\_write()会继续根据写入的文件类型来调用驱动提供的底层函数，这些函数将真正的完成写入操作。

chdir 是[C语言](https://baike.baidu.com/item/C%25E8%25AF%25AD%25E8%25A8%2580)中的一个系统调用函数，用于改变当前工作目录，其参数为Path 目标目录，可以是绝对目录或相对目录。

exit的话，它会先刷新所有的流，调用一些函数，执行当前系统定义的其他和exit相关的操作。最后，调用\_exit这个内核操作，来进行释放内存，关闭相关文件这些善后工作。

可以改进的地方：可以通过内存分配实现动态参数数组指针，可以加入lex和flex来进行分词，可以加入重定向和管道的功能



实现cd的功能，主要调用了chdir函数，显示的时候附带了一些调试用的输出。



****

**五、实验小结（心得体会、遇到的问题及解决方式、未解决/需进一步研讨的问题或建议新实验方法等）**