**操作系统课程设计实验报告**

——实验一：shell实验

负责人姓名：段瑞东

学号：

日期：2016.4.1

**小组成员**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 姓名 | 学号 | 实验分工 |
| 1 | 段瑞东 | 14061140 | 实验一 |
| 2 | 符耀仁 | 14061143 | 实验二 |
| 3 | 朱瑞江 | 14061132 | 实验三 |
| 4 | 凌杰 | 14061128 | 实验四 |

目录

[1.实验目的 4](#_Toc446710167)

[2.需求说明 4](#_Toc446710168)

[2.1基本要求 4](#_Toc446710169)

[2.2 提高要求 4](#_Toc446710170)

[2.3 完成情况 4](#_Toc446710171)

[3.设计说明 5](#_Toc446710172)

[3.1 程序流程图 5](#_Toc446710173)

[3.2基本要求实现说明 5](#_Toc446710174)

[3.3 提高要求实现说明 5](#_Toc446710175)

[4.收获和感想 5](#_Toc446710176)

[5.收获和感想 5](#_Toc446710177)

# 1.实验目的

1. 学习Linux相关软件工具的使用（如gcc、gdb和make）
2. 熟悉使用Linux中YACC工具进行语法分析的基本方法
3. 运用man帮助手册查询相关命令
4. 理解并发程序的同步问题
5. 学习POSIX/UNIX系统调用的使用
6. 掌握进程控制和进程间通信的方法

# 2.需求说明

## 2.1基本要求

1.程序能够正常运行

2.能够执行fg、bg、cd、history、exit等内部命令

3.能够执行外部程序命令，命令可以带参数

4.使用I/O重定向

5.支持前后台作业，提供作业控制功能，包括打印作业的清单，改变当前运行作业的前台/后台状态，以及控制作业的挂起、中止和继续运

## 2.2 提高要求

1.尝试对YACC语法分析的文法进行进一步的修改与完善

2.尝试在Linux下将Lex和YACC结合起来使用进行词法和语法分析

3.对其他常用的内部命令进行实现，并可以尝试考虑对通配符的支持与实现

4.实现对管道的支持

5.考虑并实现组合键ctrl+c命令

## 2.3 完成情况

【简述实验完成过程】。完成了以下功能：

【简述实验完成过程】。完成了以下功能：

1. 能够执行fg、bg、cd、history、exit、echo、which等内部命令

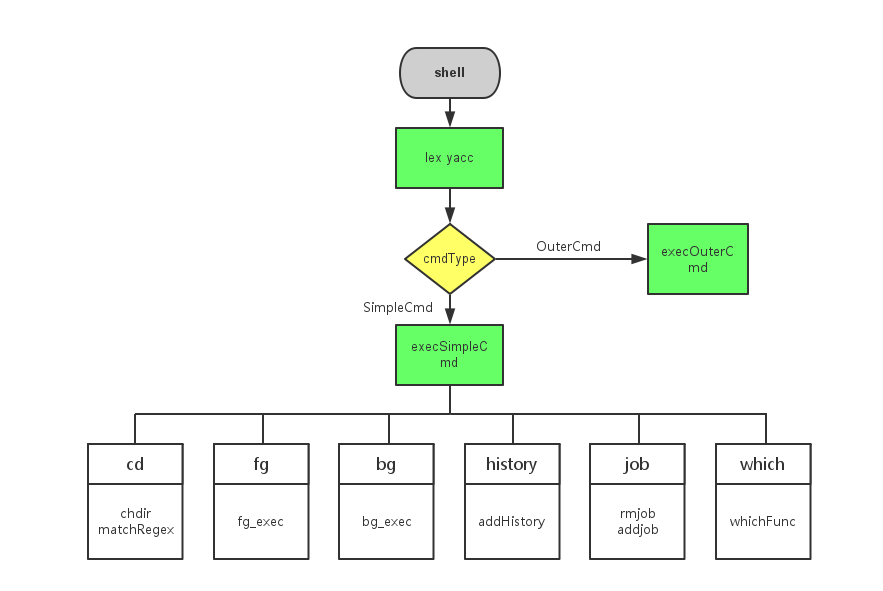
2. 能够执行外部程序命令，命令可以带参数

3. 使用I/O重定向

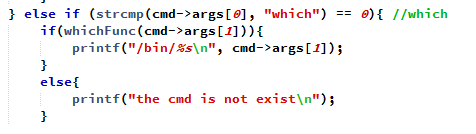
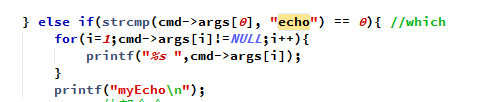
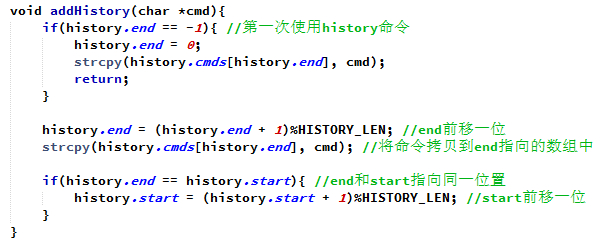
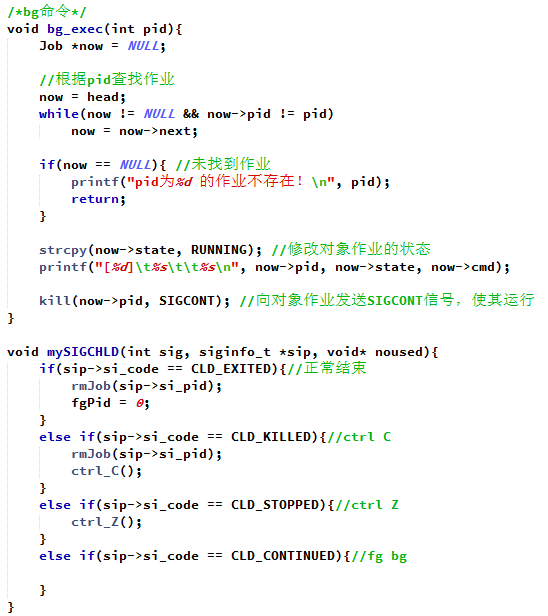
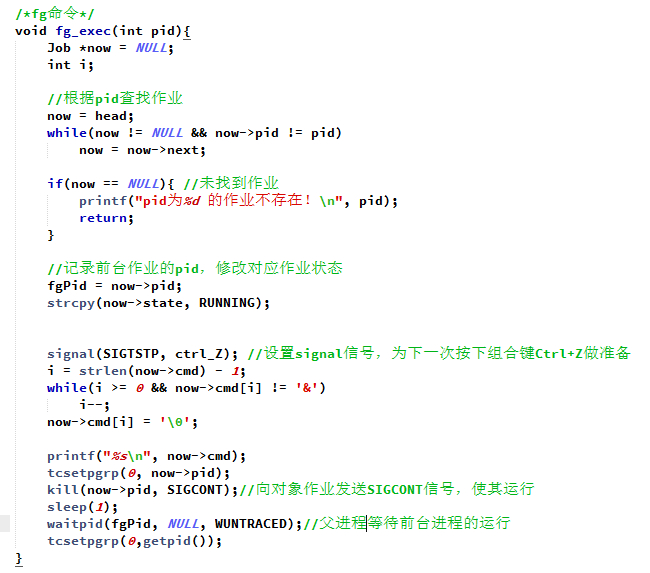
4. 支持前后台作业，提供作业控制功能，包括打印作业的清单，改变当前运行作业的前台/后台状态，以及控制作业的挂起、中止和继续运行

# 3.设计说明

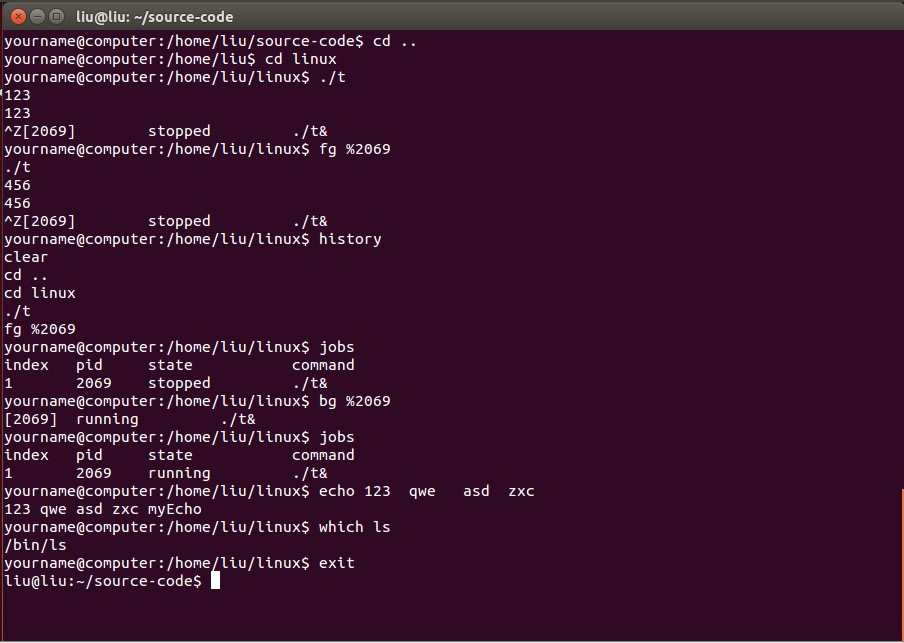
## 3.1 程序流程图



## 3.2基本要求实现说明



命令测试：



# 3. 实验遇到的问题及解决方法

坑1：

问题：在后台挂起程序之后会卡死

解决办法：shell卡死是因为当父进程发出SIGUSR1信号之后子进程没有及时注册这个信号，导致setgoon()函数没有被执行，导致父进程卡死在while（goon==0）的循环里，所以将子进程的signal(SIGUSR1,setgoon)函数放在刚进入execOuterCmd函数里。

坑2：

问题：运行前台作业挂起之后使用fg放在前台，结果命令提示符先打出来了，没有等在子进程执行完。

解决方法：在fg\_exec()函数里的kill之后加上sleep(1)，使得父进程能够等待到子进程结束才输出



坑3：

问题1：按Ctrl Z的时候会将后台的进程也一起杀死。

解决办法：用setpgid(0,getpid())将每次fork出来的新的子进程放进新的进程组里，使得对于前台的操作不会影响后台的进程。

# 5.收获和感想

收获：

通过这次shell实验，我们明白了系统的shell是怎么实现的，明白了一些系统调用的使用方法例如signal等等，以及进程之间是如何通信并且共享资源的，也知道了在终端里前后台程序的区别。除了这些，这次也是我们几个同学第一次合作完成一个作业，通过相互之间的合作学习，互相帮助，最终完成了任务，为我们的接下去的几次合作实验打下来良好的基础。同时，这次的实验用到了好多老师上课的时候没有教过的知识，我们通过自己上网搜索答案，自己学习来解决问题，很好地锻炼了独立自主的学习能力。

感想：

以前在Linux里用shell的时候感觉特别神奇，不知道为什么通过简单的终端就可以实验那么多的功能，通过这次实验，我们明白了shell的基本工作原理，明白了无论看起来多么酷炫复杂的功能实现，都是用代码一砖一瓦搭建而成的，而我们也是可以通过学习来实现属于自己的程序。

**会议记录**

我们小组每周开一次会议

第一次会议

阅读源代码，确定每个人的工作，确定这次实验的难点重点进行重点学习。

第二次会议

交流完成的基本功能，发现在fg，bg的时候有一些奇怪的bug，组员们进行集中讨论，面对面的交流debug，最终解决了这些问题，最后将完成的代码进行合并。并且确定将要做哪些提高要求。

第三次会议

交流完成的提高要求，讨论一下具体实现了哪些功能，遇到了哪些调不出来的bug，进行集中处理，合并代码进行测试，讨论并完成实验报告。