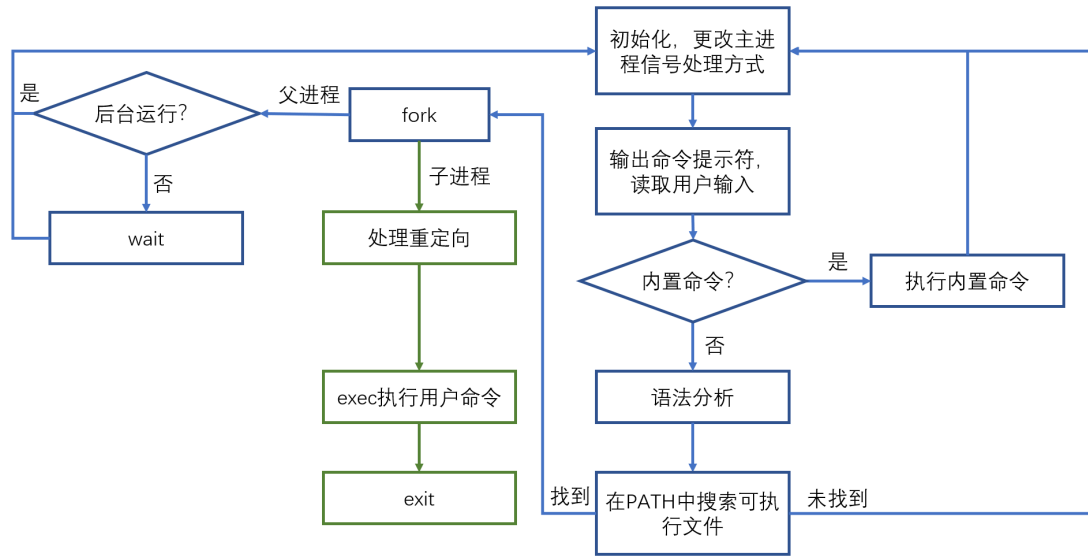


# C/UNIX 程序设计大作业

## Yet Another Unix Shell

无 77 杨昕昊

### 一、 整体流程



### 二、 基本功能要点

1. 用户输入命令与参数，能正常执行命令  
实现内置命令，语法分析，执行外部命令 0
2. 输入、输出重定向到文件  
解析 <, >, >>
3. 管道  
解析 |, 处理依赖关系
4. 后台执行程序  
解析 &
5. 作业控制  
实现内置命令 jobs, bg, fg, 实现对 Ctrl Z 信号 (SIGTSTP) 的处理
6. 历史命令  
调用 readline 库实现
7. 文件名 tab 补全，各种快捷键  
调用 readline 库实现
8. 环境变量、简单脚本  
实现内置命令 addpath, envpath, 实现对 PATH 的更改

### 三、 内置命令说明

该 shell 实现了以下 10 个内置命令：cd、pwd、exit、jobs、kill、bg、fg、addpath、envpath、history。

1. cd  
调用 chdir 函数实现
2. pwd  
调用 getcwd 函数实现
3. exit  
退出 shell，如果此时还有进程在运行，则会提示用户，让用户再次选择是否退出。
4. jobs  
shell 使用一个链表记录所有子进程的信息，jobs 命令遍历这个链表，输出当前存在的各子进程的信息。  
主进程忽略 SIGINT 和 SIGQUIT，将 SIGTSTP（即 Ctrl Z 对应的信号）更换为自己的信号处理函数。该函数会在子进程链表中寻找最后运行的子进程的 pid，如果存在则给这个子进程发送 SIGSTOP 信号，然后打印相应信息，从而实现对 ctrl Z 的处理。
5. kill  
命令格式为：kill [pid]，即杀死对应 pid 的进程。  
通过给对应 pid 的进程发送 SIGQUIT 实现。
6. bg  
命令格式为：bg [pid]，即将对应 pid 的进程移至后台运行。  
遍历子进程信息链表，给对应 pid 的子进程（若存在）发送 SIGCONT 信号。
7. fg  
命令格式为：fg [pid]，即将对应 pid 的进程移至前台运行。  
遍历子进程信息链表，给对应 pid 的子进程（若存在）发送 SIGCONT 信号，主进程等待其完成。
8. addpath  
命令格式为：addpath [target path]，将 target path 加入 \$PATH 环境变量，使得之后可以直接运行路径下的文件，仅对当前终端生效。  
通过调用 putenv 函数实现
9. envpath  
命令格式为：envpath，显示当前 \$PATH 环境变量的值  
通过调用 getenv 函数实现
10. history  
命令格式为：history，显示历史命令  
通过调用 readline 库中的 history\_list 函数实现。

### 四、 外部命令解析方式

Shell 通过以下顺序对用户输入的一个完整的外部命令进行解析：

1. 提取第一条简单命令，简单命令指出现特殊字符之前的命令，特殊字符包含：‘<’，‘>’，‘|’，‘&’，‘\n’。
2. 判断是否存在输入重定向（‘<’），如果存在，则获取输入文件名。
3. 判断是否存在管道（‘|’），如果存在，则提取 ‘|’ 之后的下一条简单命令，直到不

存在管道（'|'）为止。

4. 判断是否存在输出重定向（'>'），如果存在，判断是一个 '>'（写入新文件）还是两个 '>'（追加加入文件），然后获取输出文件名。
5. 判断是否是后台运行（'&'），如果存在，则记录其后台运行的属性。
6. 判断此条命令输入是否结束（'\n'），如果不是，则输出错误信息。

一条符合规则的典型命令输入：`cat < a.txt | grep -n haha > b.txt &`

## 五、 文件说明

文件名	功能
main.c	Shell 主循环
def.h	基本结构体的定义和一些固定参数的 define，用来确定数组大小
externs.h	声明一些在多个文件中用到的变量，比如输入输出文件名，用户输入的指令行等
parse.h	parse.c 中的函数声明
parse.c	初始化函数，输入命令分割，语法分析函数实现
builtin.h	builtin.c 中的函数声明
builtin.c	虽然文件名叫做 builtin，但是实际上不仅包含内置函数的实现与执行，还包含外部命令的执行。总之命令的执行函数都在这个文件里实现。除此之外还有对子进程信息记录链表的操作函数。
test.py	一个用来测试作业控制（jobs、fg、bg）功能的简单 python 脚本。其内容是睡眠几秒然后输出一些文字
b.sh	一个简单的 bash 脚本

## 六、 环境说明

实验环境：Ubuntu 18.04

外部依赖库：路径、文件名自动补全，输入时上下左右键和其他快捷键的支持，历史命令功能均基于 GNU readline 库。从 makefile 也可以看出，编译的时候加入了 -lreadline 选项。

运行：运行 make 编译生成的 shell 可执行文件即可。