

Lab01: 简单的shell

NJU Fall 2018

截止日期: 2018年9月28日 23:59PM

Shell是Linux/UNIX系统下传统的用户和计算机的交互界面。Bash就是一个常见的shell, 其可执行文件位置在/bin/bash, 是一个普通的Linux应用程序。运行bash即可简单了解其功能, Lab01要求实现一个简化版的shell。

1 实验要求

对于所有实验作业, 需要提交源代码, Makefile和README文件。README文件应解释你的方案与指定的解决方案不同的地方, 并说明所做的假设和做过的程序测试。此外还需要包括一个references.txt文件, 包含用于完成作业的材料参考列表, 包括网站的URL和帮助你其他学生的姓名。

1.1 基本的shell功能

1. shell可执行文件命名为njush。源代码应主要在shell.c中。可自由添加其他源代码文件, 但需要让Makefile正常工作, 运行make后即可运行njush。

2. 等待输入时显示提示符\$, 例如: `$/bin/ls -la /home/nju`

3. 一次从stdin中读取一行, 将该行解析为命令及其参数。可以假设唯一支持的分隔符是空格符(ASCII编码32)。不需要处理特殊字符, 如引号, 反斜杠和制表符。可以在命令行参数的数量上设置合理的最大值。

4. 在解析获得命令之后, njush应执行该命令。命令可以是可执行文件, 也可以是内置的shell命令(请参阅第1.2节)。运行可执行文件需要通过调用fork()然后调用exec()来完成, 可执行文件不需要输入其完整路径, 只输入文件名即可。

5. 在输入最后添加&应使得该命令在后台运行。

1.2 内置命令

实现下列内置命令。

1. exit: 清理资源退出程序

2. cd [dir]: change directory, 改变shell当前工作目录, 可假设只有一个参数

3. `history [-c] [number]`: 简化的bash history命令

- `history` 无参数: 显示最近执行的100条命令
- `history -c`: 清除所有历史命令
- `history [number]`: 显示最近执行的number条命令
- `!!`: 执行最近一条命令
- `!string`: 找出历史命令中最近的以string开头的命令

1.3 管道重定向

1. 扩展njush使程序可以通过管道进行通信。例如, 输入`ls -l | wc -l`, njush应使用fork创建两个进程, 并用dup2将stdin和stdout替换为管道的文件描述符。

2. 支持多个进程之间的管道。例如: `sleep 10 | sleep 10 | sleep 10 &`。

3. 支持内置命令使用管道。因为内置命令不需要使用fork, 此时需要额外进行处理。

4. 扩展njush使得程序的输入输出可以重定向到文件。例如, 输入`ls > junk`, njush应该将ls的输出写入junk中。

5. 命令没有输入输出时保持管道和重定向正常工作。例如, `sleep 10 | sleep 10 | sleep 10 | sleep 10 > empty-output.txt`。

2 评分标准

1. 正确编译make: 5分
2. 内置命令正常工作: 15分
3. 前台和后台进程正常工作: 10分
4. 具有两个进程的管道正常工作: 15分
5. 具有三个或更多进程的管道正常工作: 25分
6. 一个进程的IO重定向: 15分
7. 两个或多个进程的IO重定向: 15分

3 注意事项

1. 遵循Linux内核编码风格, 利用checkpatch脚本检查代码规范<https://www.kernel.org/doc/Documentation/process/coding-style.rst>
2. 使用gcc编译源码时使用-Wall显示所有警告, 编译产生警告会酌情扣分。
3. 不能使用system()函数, 因为它只调用/bin/sh shell来完成所有工作。
4. 学会利用man命令和google。