layout: post

title: Lab 06

本实验是最后一个操作系统实验，并且是选做，即不做不扣分，做了可以加分。

实验提交

截止时间: 2016/07/09 23:59:59 (**注意：因为课程成绩提交的时间限制，本次实验迟交没有成绩！**)

请大家在提交的实验报告中注明你的邮箱，方便我们及时给你一些反馈信息。

学术诚信: 如果你确实无法完成实验，你可以选择不提交，作为学术诚信的奖励，你将会获得10%的分数；但若发现抄袭现象，抄袭双方(或团体)在本次实验中得 0 分。

提交地址：http://cslabcms.nju.edu.cn/

提交格式: 你需要将整个工程打包上传，特别地，我们会清除中间结果重新编译，若编译不通过，你将损失相应的分数（请在报告中注明你实验所使用的 gcc 的版本，以便助教处理一些 gcc 版本带来的问题）. 我们会使用脚本进行批量解压缩. 压缩包的命名只能包含你的学号。另外为了防止编码问题，压缩包中的所有文件都不要包含中文.如果你需要多次提交，请先手动删除旧的提交记录(提交网站允许下载，删除自己的提交记录)，否则若脚本解压时出现多次提交相互覆盖的现象，后果自负.我们只接受以下格式的压缩包：

- tar.gz
- tar.bz2
- zip

若提交的压缩包因格式原因无法被脚本识别，后果自负。

请你在实验截止前务必确认你提交的内容符合要求(格式、相关内容等)，你可以下载你提交的内容进行确认。如果由于你的原因给我们造成了不必要的麻烦，视情况而定，在本次实验中你将会被扣除一定的分数，最高可达 50% 。

git 版本控制：我们建议你使用 git 管理你的项目，如果你提交的实验中包含均匀合理的 git 记录，你将会获得 10% 的分数奖励（请注意，本实验的 Makefile 是由你自己准备的，你可以选择像 PA 中一样在每一次 make 后增加新的 git 记录作为备份，但是请注意，这样生成的 git log 一般是无意义的，所以不能作为加分项）。为此，请你确认提交的压缩包中包含一个名为 .git 的文件夹。

实验报告要求：本次实验你必须在实验报告中写明自己为shell实现了哪些功能，以方便助教测试。**如果你实现了某个功能但是没有在报告中注明，将不被记分。**然后你还应该在实验报告中写明实现过程。

分数分布： 助教根据实现情况给分

解释： - 本次实验的加分加到整个实验上

实验要求

本次实验只有实验要求，不提供具体的实现方案。

本次实验的目标是在文件系统的基础上实现一个简单的shell，在shell中你可以选择完成这些功能：

- ls
- cat
- touch
- echo
- 重定向
- cd：
- 执行程序
- readline
- [Ctrl-C]

以下是对上述指令的解释。

ls [注意！后面所有的命令全部依赖于ls，即必须在实现了ls之后才能获得加分]

ls是Linux中的一个常用指令，其基本功能就不必赘述了。ls有很多选项，最常用的是-a, -t和-h。

- 必须实现： [20分]
  - ls -a：显示目录下所有文件；ls显示目录下除了以"."开头的文件外的其它文件；
  - ls -l：输出文件的详细信息；
  - 选项的正交性：比如，ls -a -l输出所有文件的详细信息；
- 可选实现： [10分]
  - ls -t：按最后修改时间顺序输出；
  - ls -h：输出的文件大小用B, KB, MB等单位表示。

cat [5分]

实现基本功能即可

touch [10分]

Linux的touch的功能详见链接，必须实现-c选项，其它选项可选。

echo [5分]

实现基本功能即可

>, >> [20分,依赖于cat和echo]

实现重定向功能。考虑到如果程序结构设计不合理，实现重定向功能十分困难，因此我们只要求对echo和cat指令做重定向。 在这种放宽的条件下，你可以将重定向的符号看作是cat或者echo的一个选项。

cd [15分，依赖于lab4的选做]

如果你在lab5中完成了多级目录，那么你可以在ls的基础上可以选择完成cd命令。 需要注意的是，如果要完成cd，那么必须支持文件的“相对路径”访问，这是一个挑战。

执行程序 [10分]

执行用户给定的可执行文件，若给定的字符串不是可执行文件，那么进行错误提示

readline [20分，依赖于上面所有的命令，因为指令太少的话readline没有意义]

即在NEMU中使用到的libreadline的部分功能。

- 必须完成：通过“↑”键得到历史输入的命令
- 其它自选

[Ctrl-C] [30分，依赖于“执行程序”]

终止正在执行的程序，或清除正在输入的命令，有相当大的挑战性。

实验参考资料

1. JOS中从lab1开始便有一个基本的shell，有基本的执行逻辑，可以选择直接继承JOS的响应代码，也可以借鉴其中的一些设计。
2. NEMU中的UI设计也是一个不错的参考，包括用户命令的解析。注意，我们没有正则表达式库可以使用，因此可以对文件名做出限制以简化实现。
3. 关于Linux命令的功能，请RTFM