

Linux 期末大作业

模仿 bash，完成 具备交互式命令功能的 C 程序 xhell。运行界面示例：

```
gdut:~$ ./xhell
##### Welcome to Xhell! #####
[home\gdut\]# xcd /home
[home\]# xecho this is an example
this is an example
[home\]# xcd
[home\gdut\]# quit
##### Quitting Xhell #####
gdut:~$
```

要求：接收如下 1-4 类合法命令并执行，遇到不合法命令时给出错误信息（见 5）。将代码合理拆分为多份源文件，最终使用 make 编译程序。禁用 system() 接口。

1. 内置命令

| 内置命令 | 调用格式 | 基础功能（鼓励完善更多功能） |
|-------------|----------------------------------------------------------|---------------------------------------------------------------------------------------|
| xpwd | xpwd | 显示当前工作路径 |
| xcd | xcd + 路径 path | 将工作路径转移到 path 下； 若 path 为空字符串，则返回到当前主目录下； 若 path 为“-”，则后退到上一工作目录。 |
| xls | xls + 路径 path | 显示路径 path（path 为空时，则指当前工作目录）下所有文件和文件夹，如： <div>Dir: projects/ File: info.txt</div> |
| xtouch | xtouch + 文件 file | 若文件 file 不存在，则新建文件 file；否则什么都不做 |
| xecho | xecho + 字符串 str | 输出字符串 str |
| xcat | xcat + 文件 file | 显示文件 file 内容 |
| xcp | xcp + file1 + file2 xcp -r + folder1 + folder2 | 新建文件 file2，将 file1 的内容拷贝到 file2 中；拷贝文件夹 folder1 到 folder2 |
| xrm | xrm + file / -r folder | 删除文件 file；删除文件夹 folder |
| xmv | xmv + file1 + file2/folder xmv -r + folder1 + folder2 | 移动文件/文件夹 |
| xhistory | xhistory | 显示使用过的所有历史命令 |
| xtee | xtee + file | 从标准输入读取数据，并将其内容同时写入到标准输出和 file 中 |
| xjournalctl | xjournalctl | 查看 Xhell 运行日志（日志内容执行记录） |
| quit | quit | 退出 Xhell |

以上内置命令模仿的是 Linux shell 中常用命令的基本功能。鼓励大家自行完善更多功能。此外，应充分考虑出错的情况，并将错误信息通过 perror 函数（void perror(const char *msg)）输出到标准错误流。

2. 调用外部程序

提示：常用程序的存放位置见 `$PATH` 变量；非常用命令则需指定完整路径。可通过 `access` 函数¹检查外部程序所在位置，再通过 `fork + exec + wait` 执行。

```
[home\gdut]\# ls -l /home
drwxr-xr-x 1 missing users 4096 Jun 15 2019 missing
```

3. 重定向标准输出流（> 和 >>）和标准错误流（2>）

提示：若文件已存在，> 将清空文件内容（即 `open("file", O_CREAT | O_TRUNC | ..., ...)`），>> 将新内容附加到文件末尾（即 `open("file", O_CREAT | O_APPEND | ..., ...)`）。

```
[home\gdut]\# xls -l /home>info.txt
[home\gdut]\# xcat info.txt
drwxr-xr-x 1 gdut users 4096 Jun 15 2019 missing
[home\gdut]\# xecho test if this can be appended >> info.txt
[home\gdut]\# xcat info.txt
drwxr-xr-x 1 gdut users 4096 Jun 15 2019 missingtest if this can be appended
```

4. 管道命令

提示：习题 7.7。注意 `pipe` 函数，若子进程未关闭 `pipe` 写端，则 `pipe` 读端可能一直堵塞。

```
[home\gdut]\# ps -ef | grep ssh
501 36561 74763 0 10:14PM ttys000 0:00.00 grep ssh
[home\gdut]\# cat /etc/locale.gen | wc -l
506
```

5. 不合法信息处理

接收到不合法命令（包括一切导致错误的命令）时，使用 `perror` 函向标准错误流 `stderr` 输出错误信息，并在日志中记录详细错误信息。

附加题（选作）：其他特色功能（额外加分项，自由发挥创造力）

- 1) 在前文 1-5 描述的基本功能基础上，完善更多实用功能；
- 2) 其他，如 实现更全面的 `bash` 功能（自定义快捷键、接受上下键翻阅/查找历史命令、文本编辑、计算器、小游戏、运行脚本、多任务并行、任务管理、`shell` 编程、后台常驻、远程连接/拷贝等等）；实现命令行下不常见的功能（如浏览网页、聊天功能、图片查看等等）；模拟可视化窗口（彩色、菜单等元素，参考 Windows 1.0/2.0、80 年代的 MacOS 界面）……

¹ <https://www.cnblogs.com/Braveliu/p/10232234.html>