

计算机系统软件(1)大作业方案二用户手册

源代码分为Shell部分和GUI部分

实验环境

- Ubuntu 16.04 LTS
- QEMU虚拟机
- 依赖texinfo (用 `sudo apt-get install texinfo` 安装)

运行

用 `sudo make qemu` 运行 (注意需要管理员权限), 如要清空二进制文件, 用 `sudo make clean`, 需要依赖 `rsc` (用 `sudo apt-get install rsc` 安装)。

在第一次开机时会出现设置用户名和密码的命令, 以后每次登陆都需要输入正确的用户名和密码才可登录。

Shell操作

1. mv命令 (文件重命名): `mv [source_file] [destination_file]`
2. touch命令 (若文件不存在, 则创建一系列新文件): `touch [file_name1] [file_name2] ...`
3. cp命令 (文件拷贝):

`cp -h`: 显示帮助

`cp -n [src_path] [dest_path]`: 复制过程中不覆盖任何文件

`cp -p [src_path] [dest_path]`: 复制过程中不覆盖任何比源文件新的文件

4. head命令 (显示文件前十行): `head [file]`
5. tail命令 (显示文件后十行): `tail [file]`
6. splice命令 (拼接文件): `splice [file1] [file2] [file3] ... [destination_file]`
7. shutdown命令 (关机): `shutdown`
8. more命令 (按页/段阅读文件): `more [file]`
9. `!!` 命令: 运行上一条命令
10. `history` 命令 (显示历史命令):
 1. `history (N)`: 列出最近的N条命令 (无参数输入则展示10条, 若全部命令少于10条则全部输出)
 2. `history -c`: 清空命令历史
11. delete命令 (类似于回收站的删除机制):

第一次 `delete [file]`: 删除一个文件, 将其属性设置为隐藏

第二次 `delete [file]`: 将其彻底删除 (此时功能上相当于rm)。

12. showdeled命令（显示隐藏的文件）：`showdeled [file]`

13. refresh命令（还原一个文件）：`refresh [file]`

14. 方向键的`↑`：替换当前输入为上一条指令。

15. 方向键的`↓`：替换当前输入为下一条指令。

16. vim使用：

xv6的vim的使用方法与真vim的使用方法基本一致（以下说明来自三字班的vim说明,有部分修改）。Vim的输入有两种模式，一种是控制模式，一种是输入模式。输入状态又分为插入模式和替换模式。

控制模式，用户可以输入各种控制指令，支持的指令列表如下：`i` 在当前光标处开始插入字符。`o` 新建一行并开始插入字符。`a` 在当前光标下一个位置开始插入字符。`A` 在行尾开始插入字符。`r` 替换光标处的字符，需要连续输入，例如输入`ra`则将光标处字符替换为`a`。`R` 进入替换模式。`x` 删除光标处的字符。`h` 光标向左移动一个字符。`j` 光标向下移动一个字符。`k` 光标向上移动一个字符。`l` 光标向右移动一个字符。`d` 删除光标所在行。`:q` 退出。`:w` 保存，可以加参数指定保存到哪个文件，如`:w test`，默认为打开的文件。`:wq` 保存并退出。`:q` 不保存退出。在**输入模式**下的操作和一般的文本编辑器类似，可以用键盘的上下左右移动光标。在**输入模式**下按`Esc`可以退回到**控制模式**。

17. find命令

按照文件名搜索文件（不忽略大小写，支持通配符），命令格式：`find 目录 -name 文件名`

按照文件名搜索文件（忽略大小写，支持通配符），命令格式：`find 目录 -iname 文件名`

查找特定大小的文件（大于或小于一定值），命令格式：`find 目录 -size 文件大小范围(+n表示文件大小大于n, -n表示文件大小小于n)`

查找特定创建时间的文件（n分钟前或n天前），命令格式：

```
find 目录 -cmin 时间 (n分钟前)
```

```
find 目录 -ctime 时间 (n天前)
```

删除查找到的文件，命令格式：`find 目录 XX XX delete`

查找所有空文件，命令格式：`find 目录 -empty`

18. bi命令(BASIC解释器)

输入文件执行：`bi 文件名(可在vim中编辑代码文件)`

在shell中输入并执行：输入`bi`，在shell中输入代码，最后一行输入`RUN`并按回车执行。

支持的BASIC语法有：`LET`，`INPUT`，`PRINT`，`IF-ELSE`，`FOR-ENDFOR`，`WHILE-WEND`，`LIST`，`GOTO`

变量名必须为A-Z，且均为全局变量，保存字符串或整数。输出必须为字符串和变量，不能输出一个表达式的结果。

支持算术运算和与或非逻辑运算。

样例代码：

```
INPUT A
WHILE A < 5
  LET B = 1
  WHILE B < 5
    LET B = B + 1
  PRINT B
WEND
WEND
```

19. pwd命令：输入 `pwd`，输出系统当前工作目录。

20. date命令：输入 `date`，输出当前系统时间。

21. ls命令改进：输入 `ls -a`，输出包含以'.'为开头的隐藏文件。

22. cat命令改进：当cat参数中的文件不存在时，自动创建该文件

23. 子目录下执行shell指令

原有xv6版本切换至子目录后无法直接运行shell命令，需要输入绝对路径，我们使这一过程在shell中自动完成，从而在子目录下可直接运行shell命令。

24. rename命令，用于文件重命名。输入格式：`rename 旧文件名 新文件名`

25. TAB自动补全

按下TAB键时，shell会按照用户输入的字符串匹配前缀为该字符串的shell命令，并输出所有匹配到的shell命令名，如果没有找到则提示用户未找到。此时，缓冲区被清空，用户需要完整输入所要执行的shell命令并按回车执行。

26. 后台进程使用方法：

有两种方式实现后台进程：

1. 命令+&

2. 前台程序使用 `ctrl+c` 挂起，使用 `bg` 转入后台，之后使用 `jobs`，`kill` 等配合进行查看等操作。

在命令行输入 `jobs`，即可打印出后台进程信息。注意使用时应确保有进程在后台运行，否则无输出。

扩展 `kill` 指令后原 `kill` 功能不受影响。如需杀死后台进程，在命令行输入 `kill %+后台进程序号`。成功杀死进程无提示，否则报异常错误。

27. lua运行说明

输入 `lua` 进入交互式界面，`Ctrl+D` 退出交互式界面。运行 `lua testmath.lua` 等脚本进行脚本测试。

28. 任务管理器使用说明

1. 输入 `taskmgr` 指令，进入任务管理器界面




2. 任务管理器会列出当前所有进程（xv6进程数目上限为64）的基本信息，包括进程的id、名称、状态和占用内存的大小

3. 选中的进程会被高亮显示，按上下键可移动选中的进程，按左右键可翻页，按F键可进行刷新，按K键可以删除选中的进程

GUI操作

图片浏览器使用说明

1. 单击 `↑` 键，播放上一张图片

2. 单击  键, 播放下一张图片
3. 单击  键(位于右侧数字键盘), 将当前图片放大到3/2倍
4. 单击  键(位于右侧数字键盘), 将当前图片缩小到2/3倍

文档浏览器&文本编辑器

1. 文档浏览器打开文件, 并用文本编辑器浏览:

打开文档浏览器, 双击相应文件即可。

2. 文档浏览器返回上一级:

1. 打开文件浏览器。
2. 打开一个文件夹。
3. 从工具栏里选择operations->back即可。

3. 文本编辑器打开文件:

1. 打开文档浏览器。
2. 从工具栏里选择open->allText->scanTextInXV6即可。
3. 双击任意文档。
4. 使用open->allText->return从文档选择中离开

3. 文本编辑器退出前保存:

1. 在未保存的情况下点击退出即可。