
《计算机系统软件（1）》实验报告

xv6 系统整合——shell 整合

小组成员

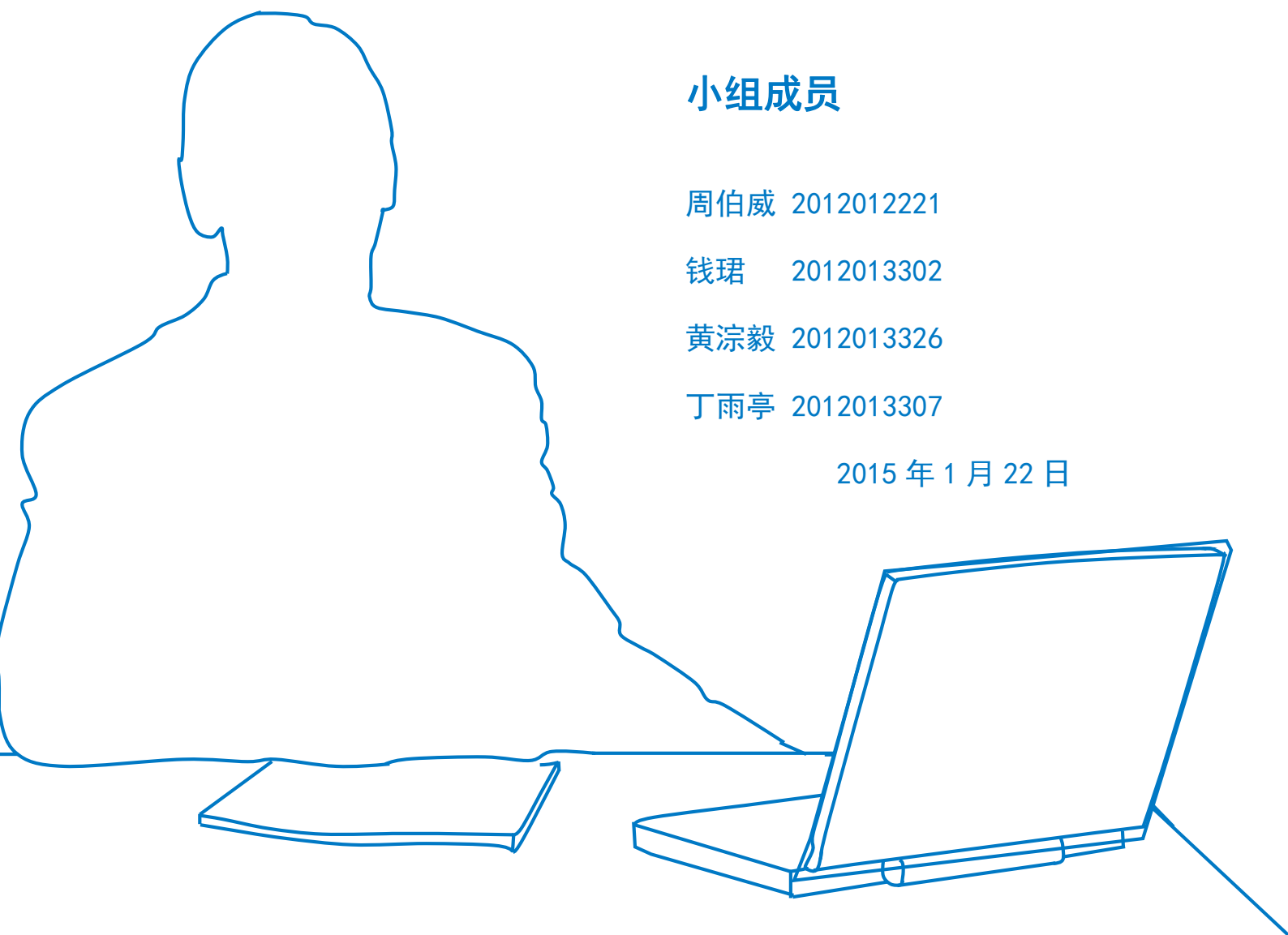
周伯威 2012012221

钱珺 2012013302

黄淙毅 2012013326

丁雨亭 2012013307

2015 年 1 月 22 日



目录

小组成员.....	1
一. 实验背景.....	3
二. 人员分工.....	3
三. 实验环境.....	3
四. 实验成果.....	4
五. 具体实现.....	4
六. 总结.....	8

一. 实验背景

xv6 是由 MIT 开发的用于操作系统课程教学的简单的操作系统。本学期的《计算机系统软件（1）》课程中同样使用该操作系统进行教学。

在之前的教学中，已经对于 xv6 有了一些改进。本次实验要对于这些改进进行整合，以将本系统进行完善，成为一个相对功能更加现代的操作系统。

本次整合我们小组主要负责整合 shell 命令部分。shell 命令部分的基础有洪宇学长和黄怀毅学长的 shell 命令改进版本为基础，并加以整合和改进并且加入新的功能。

二. 人员分工

黄怀毅组整合：周伯威、黄淙毅

洪宇组整合：钱珺、丁雨亭

三. 实验环境

操作系统：Ubuntu 10.04

Ubuntu 12.04

Ubuntu 14.04

改进对象：xv6_rev7

Xv6_rev8

四. 实验成果

本次实验，我们在原有的 xv6_rev7 的基础上进行整合，并且还将功在 xv6_rev8 版本以及 Ubuntu14.04 上完成了内核升级。整合以及升级的功能包括：命令：touch（创建一个或者多个文件）、uptime（获取系统时间）、cp（复制文件）、mv（移动文件）、editor（命令行格式的文本编辑器）、cat（显示和文件操作）、ren（重命名）；功能：代码错误提示，历史记录，命令补全；新功能：splice（拼接文件）、cal（计算器）

五. 具体实现

1. 复用功能

- touch、uptime、mv 命令：本部分复用洪宇组的命令

mv, uptime, touch 命令是通过 open, read, write 等函数实现的。

- cat 命令：本部分复用黄怀毅组命令

修改原有的命令，使得现有的命令具有与 Linux 标准的 cat 命令相同的功能，不但可以读文件，还可以创建、拷贝以及为文件加行号等。

- 错误命令提示：本部分复用黄怀毅组功能

本功能主要通过修改 sh.c 文件中的 runcmd 函数（加入静态可接

受命令集）使得用户可以在输入错误命令时有系统自动到寻找相似命令进行并且提醒用户。（注：相似命令：如果用户输入的命令与系统已有命令有至少两个字母相同则认定为相似。）

2. 改进功能

- 在原来的版本中，shell 命令是只可以在根目录下面运行的，然后我们通过修改（主要是修改 sh.c 文件）使得在我们的版本中 shell 命令可以在任意子文件夹下面运行。

- cp 命令：修正原版本中文件夹复制成文件的错误使得命令更加实用

- editor 命令：editor 的功能是按行处理一个文件，实现了添加行、删除行、修改行、保存与退出等基本功能以及提供给用户 show/hide 的设置，并提供了 Esc 快捷键和 Ctrl+S 快捷键的支持。首先，editor 使用 write 函数将文件内容读到内存，并按照 \n 分隔符对文件进行分行，分行后的文本内容用 text 变量存储，这是一个存储指针的数组，可以动态适应行数的变化。而插入、删除、修改功能则都是实现了相关函数 com_ins、com_mod、com_del。实现本身并不复杂，就是对 text 数组进行处理，在此不再赘述。

快捷键的实现思路是利用 gets 函数循环等待输入，当按下快捷键时从内核返回“特殊的缓冲区值”。通过判定该值决定是否执行事件处理函数。

最后，考虑到用户习惯的因素，在插入时，如果最后一行是空行，则插入到最后一行并在下面新增一行。另外，程序记录了当前文

本文件是否是最新的，如果不是，在退出时会提示用户保存。最后，程序还提供给用户一个方便的帮助。

我们还在之前版本基础上做了一些改动，修复了其中的一些 bug，比如指定行数和实际操作行数不吻合等。

现在的 editor 命令是一个更加全面与完善的命令，用户的体验度更高

- ren 命令：重新命名了 rename 命令为 ren 命令。rename.c 重命名为 ren.c，在新的 ren.c 文件中注释掉

```
//char * dir = argv[3];
```

```
//chdir(dir);
```

并在第 9 行中添加 `argc!=3`，并且修改相关 makefile

xv6 中的文件与文件名并不是一一对应的。多个文件名可以对应同一个文件。（当一个文件的所有文件名都与此文件失去关联时，此文件会被删除。）可以通过建立新文件名与文件的关联，取消旧文件名与文件的关联来实现对文件的重命。所以通过使用系统调用的 `link` 函数，将文件与新的文件名关联，再使用系统调用的 `unlink` 函数，取消文件与旧的文件名的关联。使用命令 `ren old_filename new_filename` 就可以将一个名为 `old_filename` 的文件转换成为 `new_filename`

- 查看历史命令

如果我们想再次运行以前运行过的命令，我们希望通过快捷键执

行，而不是重新打一遍该命令，就像 linux 的 terminal 或者 windows 的 cmd 中的上下键一样，可以直接调出之前执行过的命令。而 xv6 操作系统的 shell 并没有这样。所以我们对于该命令进行了改动。

首先，当用户每次键入回车后，系统将会将该命令储存到一个数组中。当用户在非命令提示的过程中（即界面中没有白底灰字的命令提示）键入上下键，用户即可相应地跳转至其他命令。

● 自动补全

像一些很友好的 console、以及 sublime text 一样，我们也希望在输入命令的时候能够更加快捷，比如上下键可以查看命令、右键可以补全该命令。本功能就实现了这样的需求。用户在输入命令的时候，甚至只需命令的第一个字母的键入加上一个右键按钮的点击就可以实现了命令的输入。

在每个空格或者回车之后，系统会重新记录用户的输入字符串，并根据用户输入的字符寻找匹配的所有命令。举个例子，当用户在空格或回车之后，输入了“s”，则系统找出所有以“s”为开头的命令：“sh”、“stressfs”；再输入“t”（此时输入字符串为“st”），则系统找出所有以“st”开头的命令：“stressfs”。当用户按上下键的时候可以翻看历史命令，然后按右方向键可以补全。

3. 新增功能

● splice 命令：

在文件操作中，cp 命令只能整个复制文件，然而这对于一个可扩展的良好的系统来说，我们应该有对文件更多的操作。尤其是对于我们整个要实现的达到文件系统来说，我们应该在文件操作中多加入一些功能从而使其更像现代操作系统。于是我们在 cat 命令的启发下，做出了文件的拼接功能，这样我们就可以在问价中实现类似于 word 的将其他部分的内容合并到同一文件下。这为以后的有 GUI 的文件中的操作提供了可能。

● cal 命令

由于 shell 命令到后来需要合成到 GUI 界面上，所以我们在这里做了一个简单运算命令，再结合 GUI 之后就可以成为一个小的计算器程序。我们的 cal 命令通过对于参数 argv[1]的解析，在进行程序求值就可以运行出结果。另外，我们的 cal 命令可以处理“()”使得输入更加方便。

六. 总结

本次实验，是由三个组一起联合做的，对于各组之间的协调其实很难，所以则其实是本次最大挑战之一。

其次由于我们是整合学长的代码，所以队友其存在 bug 的可能性估计不足。导致整个组的进程在某些程度上被拖慢。所以在之后的开发中，很多东西要重头开始，这也和预期有较大的差别。

总的来说，本次的实验还是取得了预期的效果，但是还是存在很

多的问题，等待之后的人去探索 and 解决。