
目录

1	引言.....	2
2	运行环境.....	2
3	主要功能总体描述.....	2
4	命令详解.....	3
4.1	ls 命令.....	3
4.2	pwd 命令.....	3
4.3	cp 命令.....	3
4.4	rm 命令.....	3
4.5	mv 命令.....	3
4.6	cat 命令.....	3
4.7	tail 命令.....	4
4.8	touch 命令.....	4
4.9	vim 命令.....	4
4.10	mkdir 命令.....	4
4.11	who 命令.....	4
4.12	echo 命令.....	5
4.13	uname 命令.....	5
4.14	find 命令.....	5
4.15	su 命令.....	5
5	实现流程图.....	5
5.1	主程序运行流程图.....	5
5.2	用户输入.....	6
5.3	命令行分析.....	7
5.4	各个命令的程序框图.....	9
5.4.1	ls 命令.....	9
5.4.2	pwd 命令.....	9
5.4.3	cp 命令.....	11
5.4.4	rm 命令.....	12
5.4.5	mv 命令.....	12
5.4.6	cat 命令.....	14
5.4.7	tail 命令.....	16
5.4.8	touch 命令.....	17
5.4.9	vim 命令.....	18
5.4.10	mkdir 命令.....	19
5.4.11	who 命令.....	19
5.4.12	echo 命令.....	21
5.4.13	uname 命令.....	22
5.4.14	find 命令.....	23
5.4.15	su 命令.....	24

1 引言

《操作系统》是本专业的核心课程，对于它的认知仅靠学习课本内容的理论知识是远远不够的。在学习了进程部分内容后，为了更加深入地了解计算机操作系统的一些实现原理和操作过程，我们需要通过对最经典的操作系统——Linux 操作系统下常用命令的实现过程进行探究，编写 c 语言程序实现和使用 shell 命令相同的作用，当用户输入一个命令，反馈给用户想要获得的信息，从而帮助我们进一步体会操作系统底层是怎样运行的。

2 运行环境

硬件平台：能满足安装 Linux 系统所需的最低硬件

软件平台：Linux 操作系统

3 主要功能总体描述

运行编写的 C 语言文件，通过输入指定命令达到与 shell 命令相同的目的。例如实现 Linux 系统命令中：显示当前目录——pwd，列出当前目录下的文件（或含隐藏文件或包括列出文件的具体信息等）——ls 系列，以及进入编辑模式——vim，查看文件内容——cat，复制文件命令——cp，创建文件——touch 命令等。

1. 实现输入输出重定向：

1.1 关于输出重定向：

通俗来讲，重定向输出就是把要输出得文件信息写到一个文件中，而不是将要输出得文件信息输出到控制台（也就是终端）。

1.2 关于输入重定向：

是指不使用系统提供的标准输入端口，而进行重新制定。也就是说，输入重定向就是不使用标准输入输出端口输入文件，而是使用指定的文件作为标准输入设备。

2. 打印终端提示符

每行命令由 myshell_17> 开始，然后由用户输入命令行，例如 cat os.c，该命令将 os.c 文件展示于终端上。

eg.

```
myshell_17> cat os.c
```

3. 获取用户输入命令，注意区分内部命令和外部命令

3.1 外部命令，也就是对应着磁盘上的某个程序，例如 wc、ls 等等。对这种外部命令，我们首先要到指定的路径下找到它，然后再执行它。

3.2 内部命令，内部命令并不对应磁盘上的程序，例如 cd、echo 等等，它需要 shell 自己来决定该如何执行。例如对 cd 命令，shell 就应该根据它后面的参数改变当前路径。

4. 支持&修饰命令行，使命令进入后台运行（本质上是使命令行在子进程运行于后台，或者子进程和父进程同时运行。）

5. 界面美观

6. 当用户输入 exit 或 logout 时，退出该 shell 程序

4 命令详解

4.1 ls 命令

功能：显示文件和目录的信息，和 Windows 系统下 DOS 命令 dir 类似。当执行 ls 命令时，默认显示的只有非隐藏文件的文件名、以文件名进行排序及文件名代表的颜色显示。当不加参数时，默认列出当前目录的列表信息。

ls - list directory contents

4.2 pwd 命令

功能：打印当前的工作目录

pwd - print name of current/working directory

4.3 cp 命令

功能：复制一个源文件到目标文件（夹）；同时复制多个文件到目标文件（夹）下；复制源文件夹到目标文件夹下。

cp - copy files and directories

4.4 rm 命令

功能：删除一个目录中的一个或多个文件或目录，它也可以将某个目录及其下的所有文件及子目录均删除。对于链接文件，只是断开了链接，原文件保持不变。

rm - remove files or directories

4.5 mv 命令

功能：将文件移动走，或者改名，在 unix 下面没有改名的命令，如果想改名，可以使用该命令

mv - move (rename) files

4.6 cat 命令

功能：文本输出命令，显示文件的内容，和 DOS 的 type 相同。cat 主要有三大功能：
1. 一次显示整个文件。
2. 从键盘创建一个文件。只能创建新文件，不能编辑已有文件。
3. 将几个文件合并为一个文件。

cat - concatenate files and print on the standard output

4.7 tail 命令

功能：依照要求将指定的文件的最后部分输出到标准设备，通常是终端，通俗讲来，就是把某个档案文件的最后几行显示到终端上，假设该档案有更新，tail 会自己主动刷新，确保你看到最新的档案内容。

```
tail - output the last part of files
```

4.8 touch 命令

功能：一是用于把已存在文件的时间标签更新为系统当前的时间（默认方式），它们的数据将原封不动地保留下来；二是用来创建新的空文件。

```
touch - change file timestamps
```

4.9 vim 命令

功能：在系统提示符号输入 vim 及文件名称后，进入 vim 之后，是处于命令行模式。按一下字母 i 就可以进入插入模式，这时候你就可以开始输入文字。处于「插入模式」，就只能一直输入文字，如果发现输错了字，想用光标键来回移动，将该字删除，就要先按一下「ESC」键转到命令行模式再删除文字。在命令行模式下，按一下冒号键进入「Last line mode」，例如：

```
: w filename （输入「w filename」将文章以指定的文件名 filename 保存）
```

```
: wq （输入「wq」，存盘并退出 vi）
```

```
: q! （输入 q!，不存盘强制退出 vi）
```

```
vim - Vi IMproved, a programmer's text editor
```

4.10 mkdir 命令

功能：用来创建目录。该命令创建由 dirname 命名的目录。如果在目录名的前面没有任何路径名，则在当前目录下创建由 dirname 指定的目录；如果给出了一个已经存在的路径，将会在该目录下创建一个指定的目录。在创建目录时，应保证新建的目录与它所在目录下的文件没有重名。

```
mkdir - make directories
```

4.11 who 命令

功能：显示关于当前在本地系统上的所有用户的信息。显示以下内容：登录名、tty、登录日期和时间。输入 whoami 显示您的登录名、tty、您登录的日期和时间。如果用户是从一个远程机器登录的，那么该机器的主机名也会被显示出来。who 命令也能显示自从线路活动发生以来经过的时间、命令解释器（shell）的进程标识、登录、注

销、重新启动和系统时钟的变化，还能显示由初始化进程生成的其它进程。

`who - show who is logged on`

4.12 echo 命令

功能：将要输出的文本打印出来，后面跟不同的参数表示打印出的文本格式不同。

`echo -n`：不换行输出

`echo -e`：输出转义字符

`echo - print the same content`

4.13 uname 命令

功能：用于打印当前系统相关信息（内核版本号、硬件架构、主机名称和操作系统类型等）。

`uname - print system information`

4.14 find 命令

功能：用来在指定目录下查找文件。任何位于参数之前的字符串都将被视为欲查找的目录名。如果使用该命令时，不设置任何参数，则 `find` 命令将在当前目录下查找子目录与文件。并且将查找到的子目录和文件全部进行显示。

`find - find all files that match the criteria`

4.15 su 命令

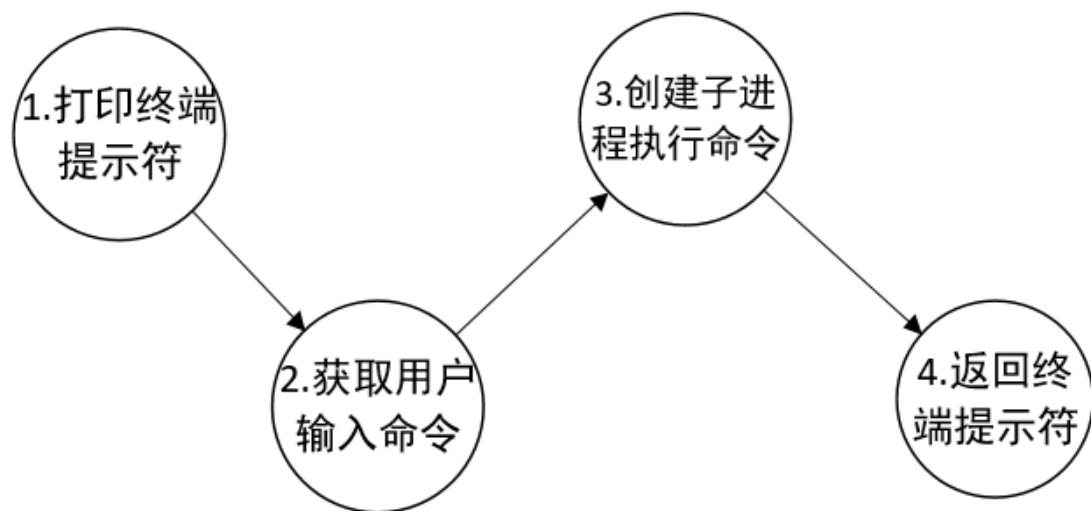
功能：用于变更为其他使用者的身份，除 `root` 外，需要键入该使用者的密码。使用权限：所有使用者。

`su - 更改用户 ID 或成为超级用户`

5 实现流程图

5.1 主程序运行流程图

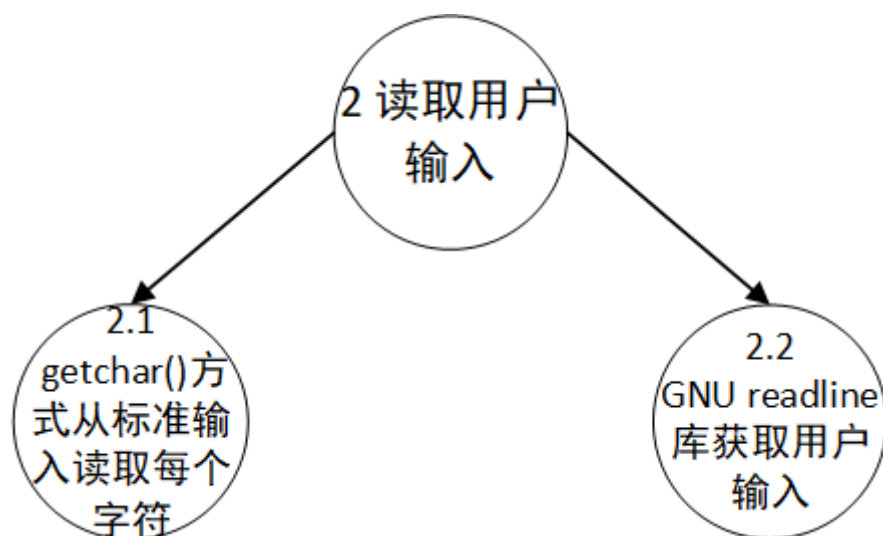
做完一些必要的初始化工作之后，进入这样的一个循环：



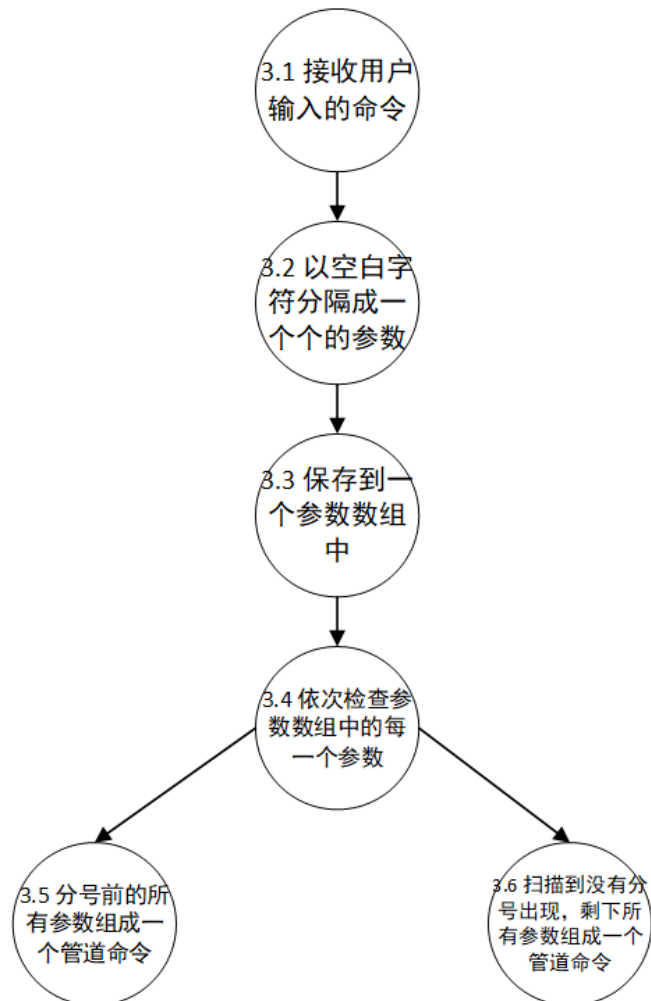
如果用户输入 `logout` 或者 `exit` 之后，才退出这个循环。类似伪代码的形式表示如下：

```
while(1) {  
    print_prompt();  
    get_input();  
    parse_input();  
    if(“logout” || “exit”)  
        break;  
    do_cmd();  
}
```

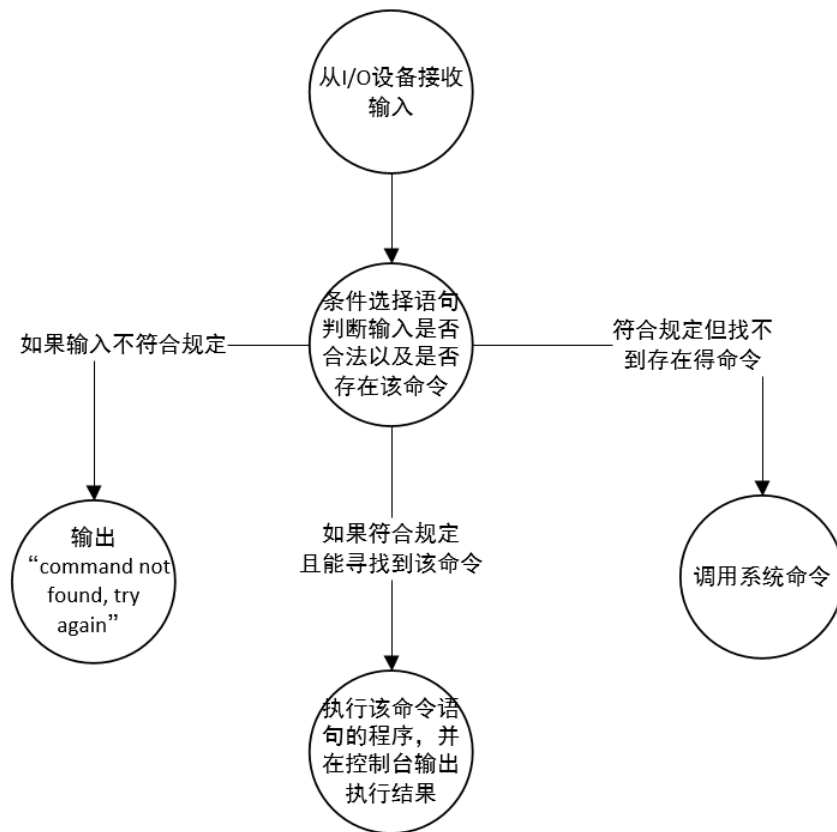
5.2 用户输入



5.3 命令行分析



输入之后，创建子进程之前
做的条件判断

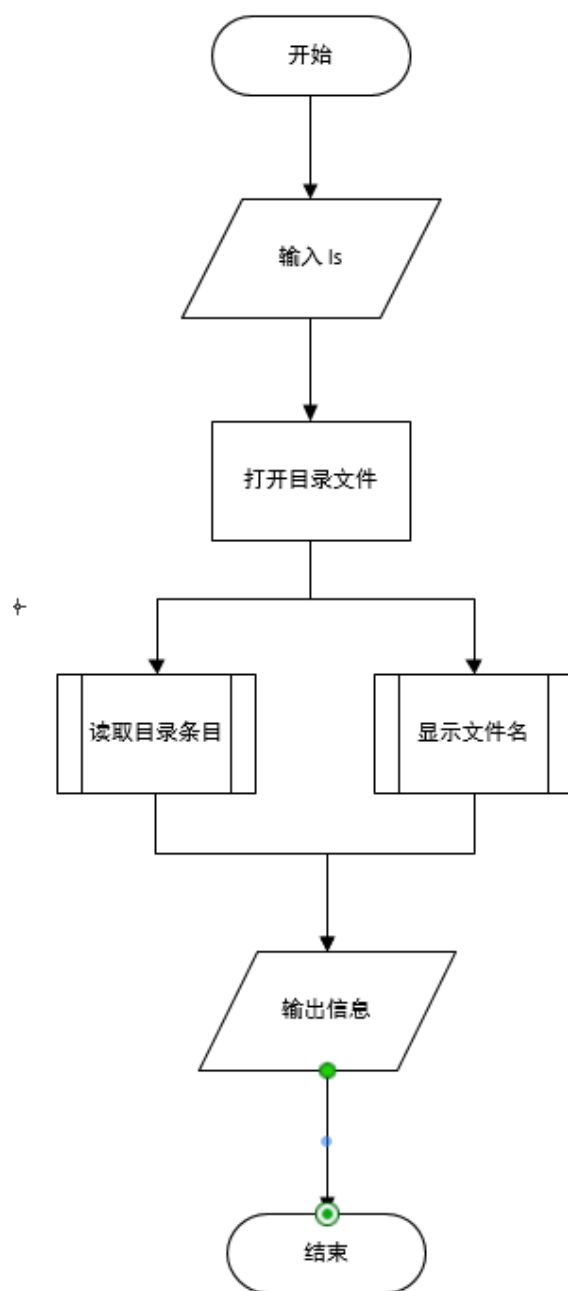


这里输入不符合规定大概指的就是输入字母错了，单词错了，参数顺序错了等等等等乱七八糟的错误。

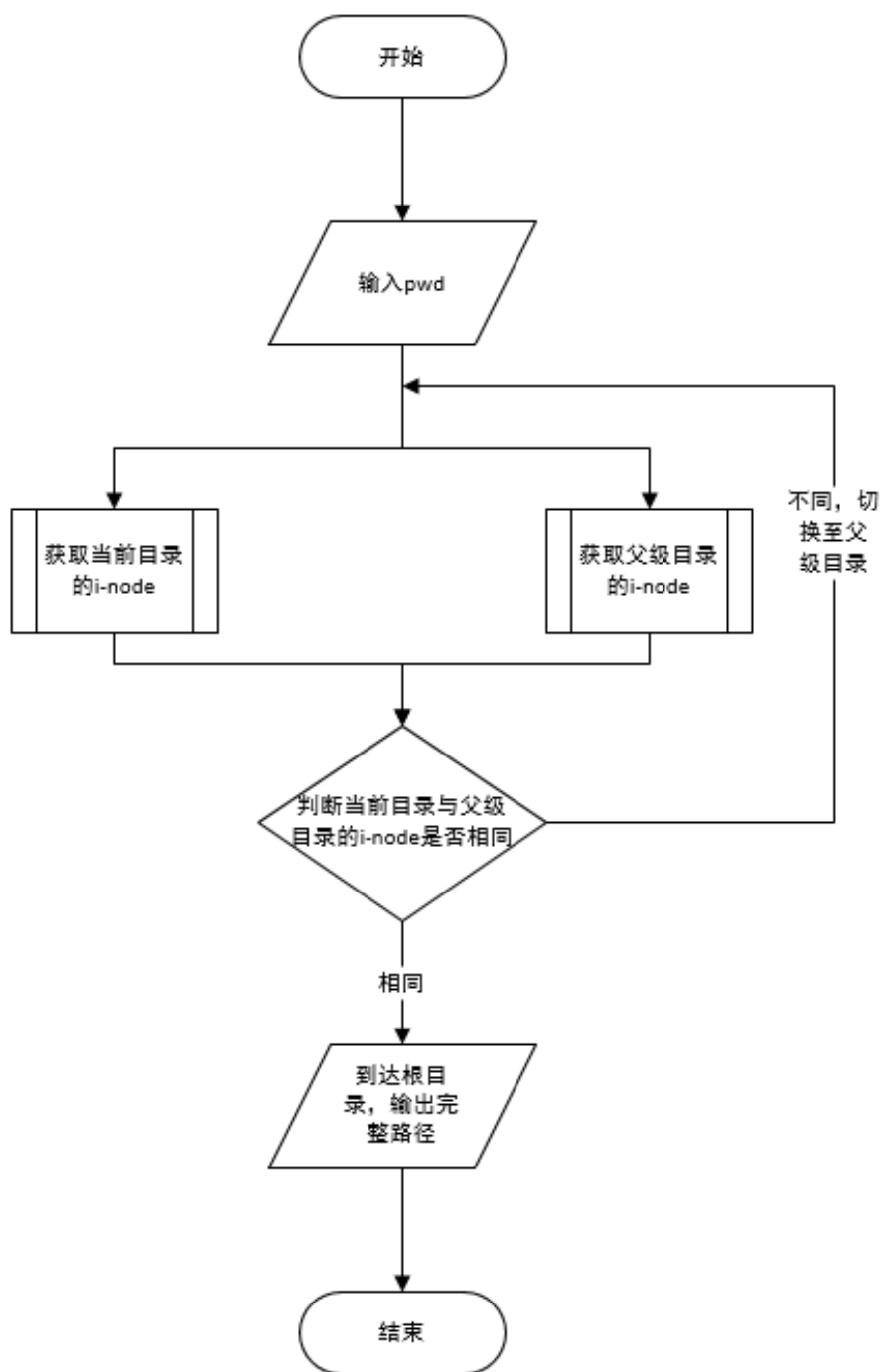
符合规定但找不到存在的命令意为输入的命令在我们自己写的 C (C++) 程序找不到，这个时候可以调用系统命令。

5.4 各个命令的程序框图

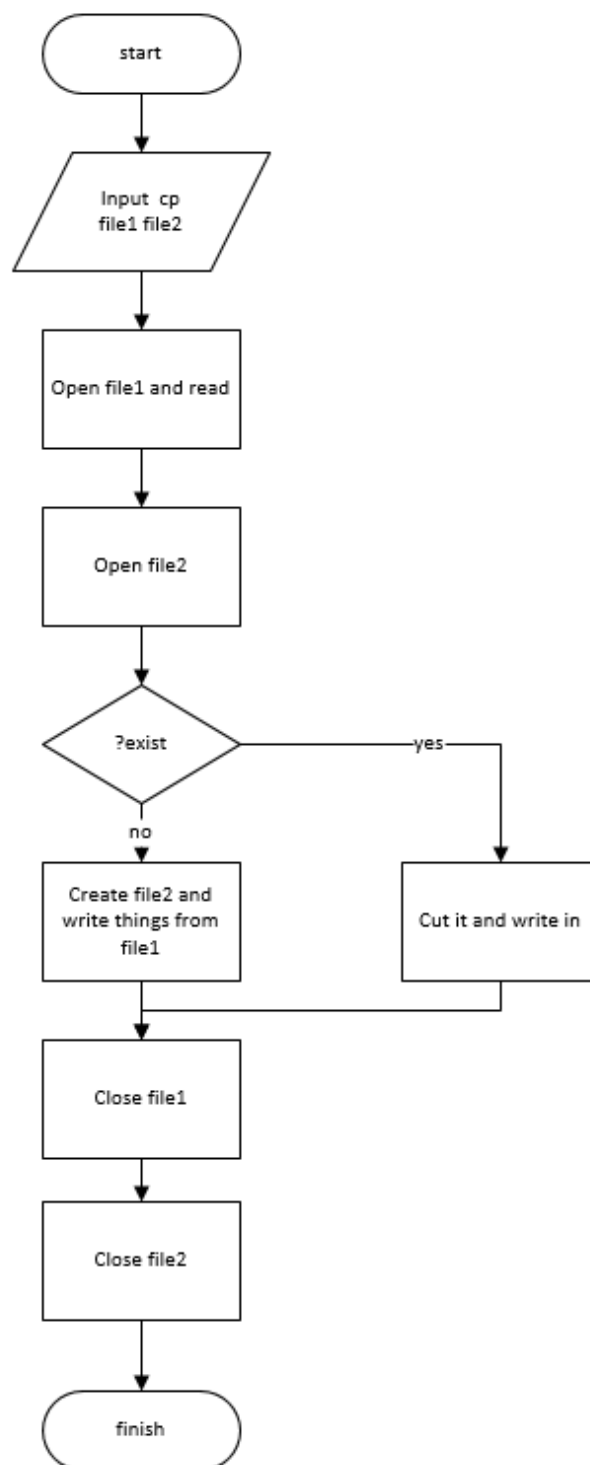
5.4.1 ls 命令



5.4.2 pwd 命令

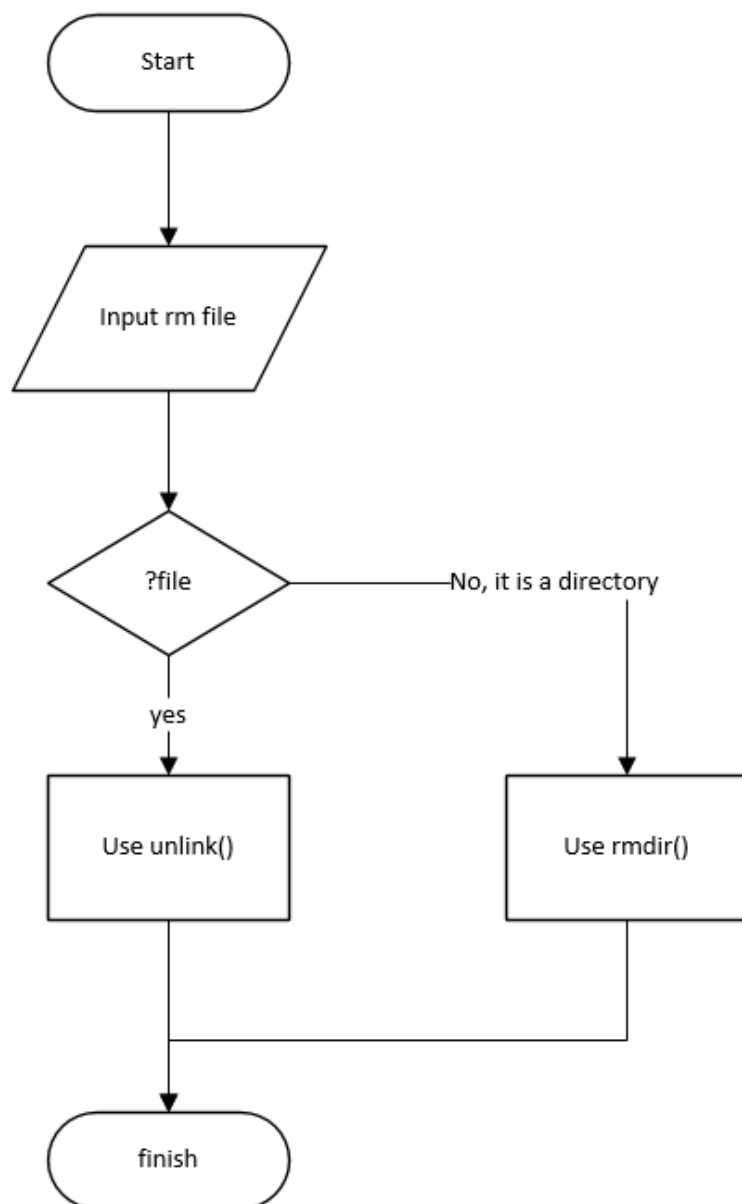


5.4.3 cp 命令

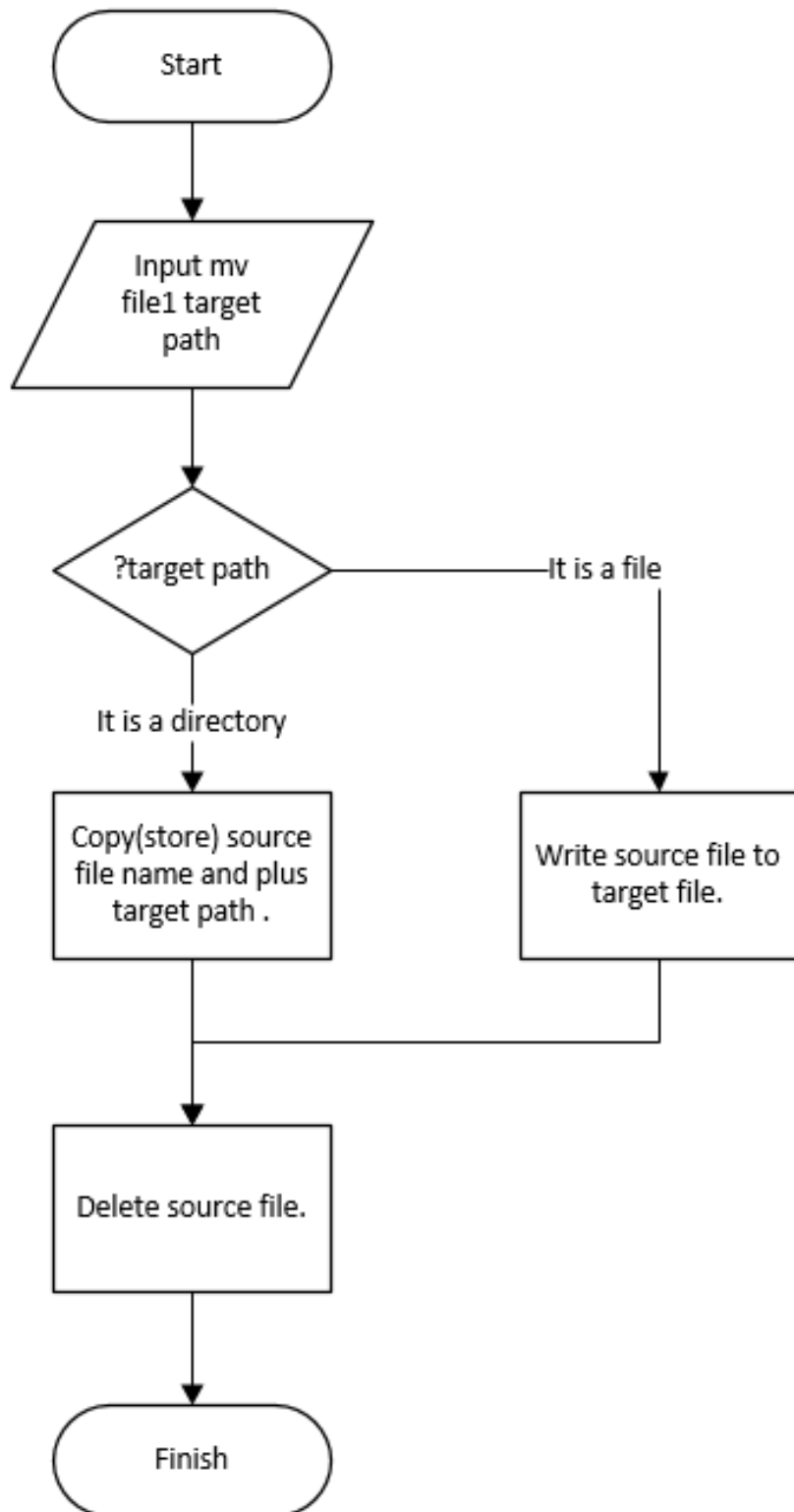


✦

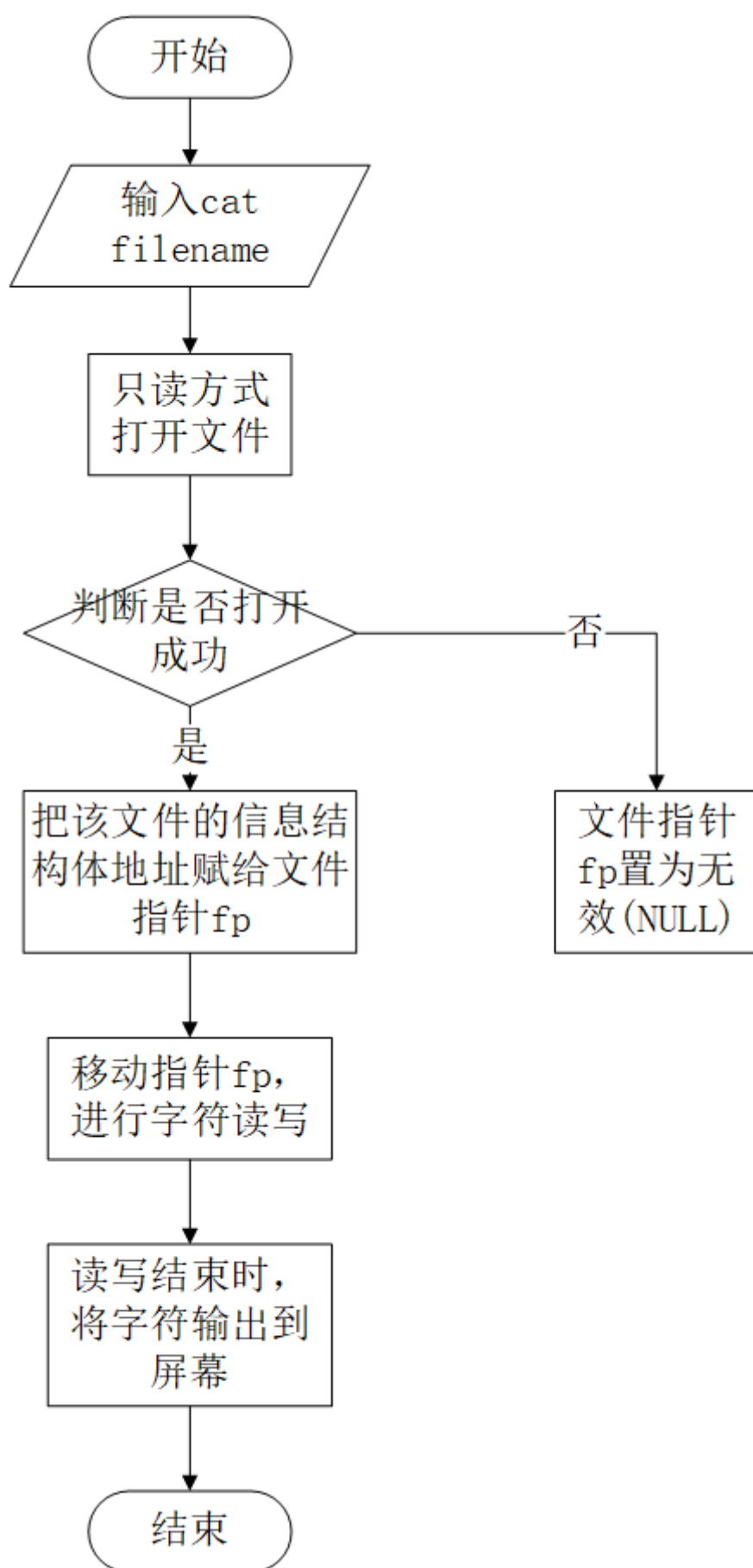
5.4.4 rm 命令



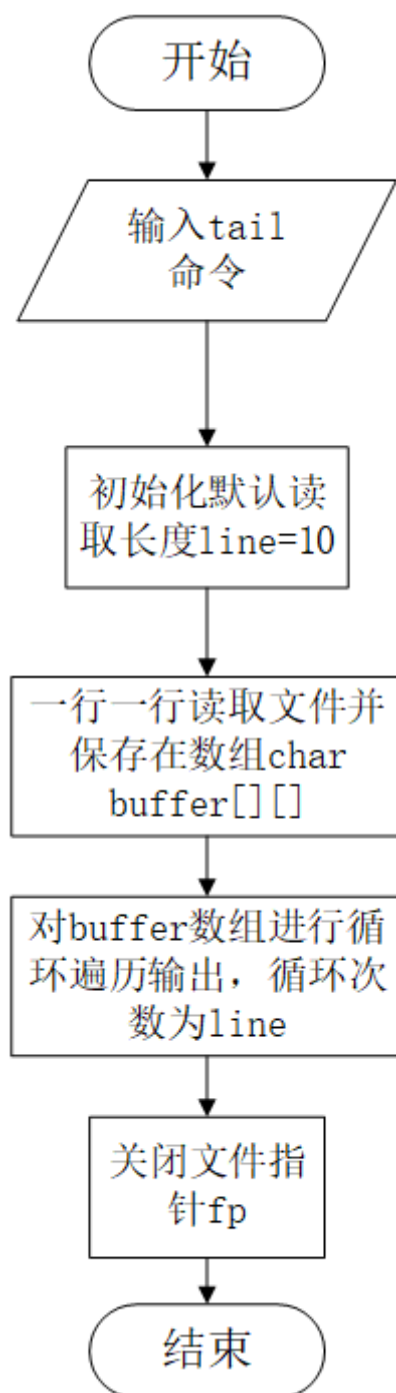
5.4.5 mv 命令



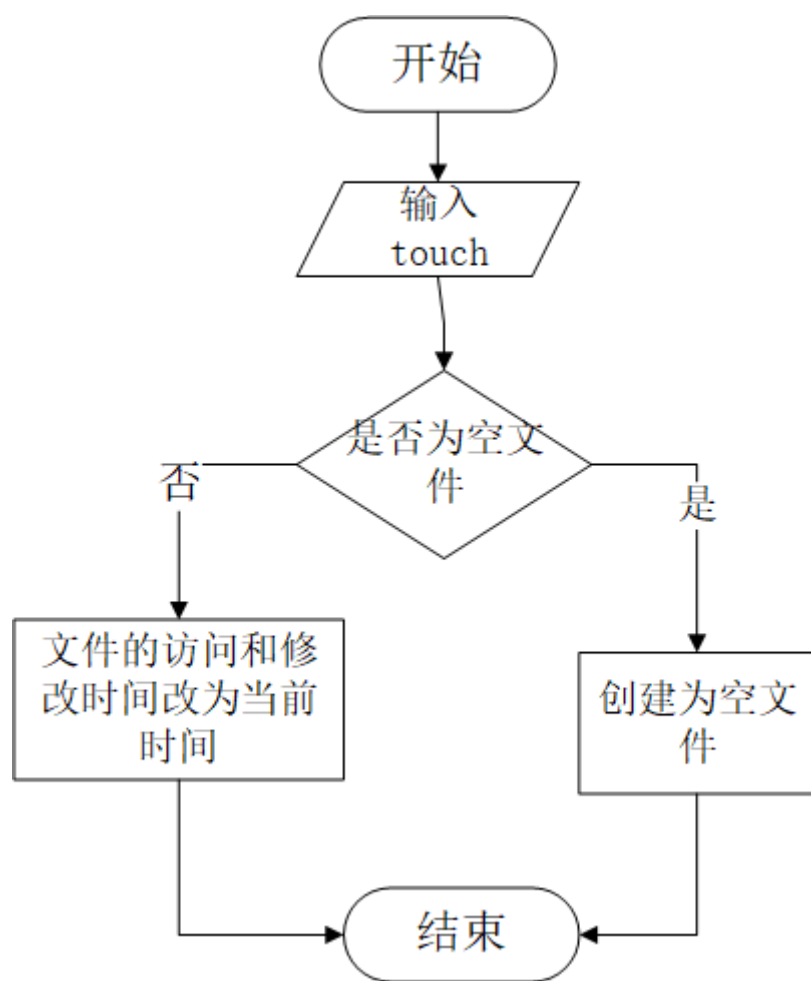
5.4.6 cat 命令



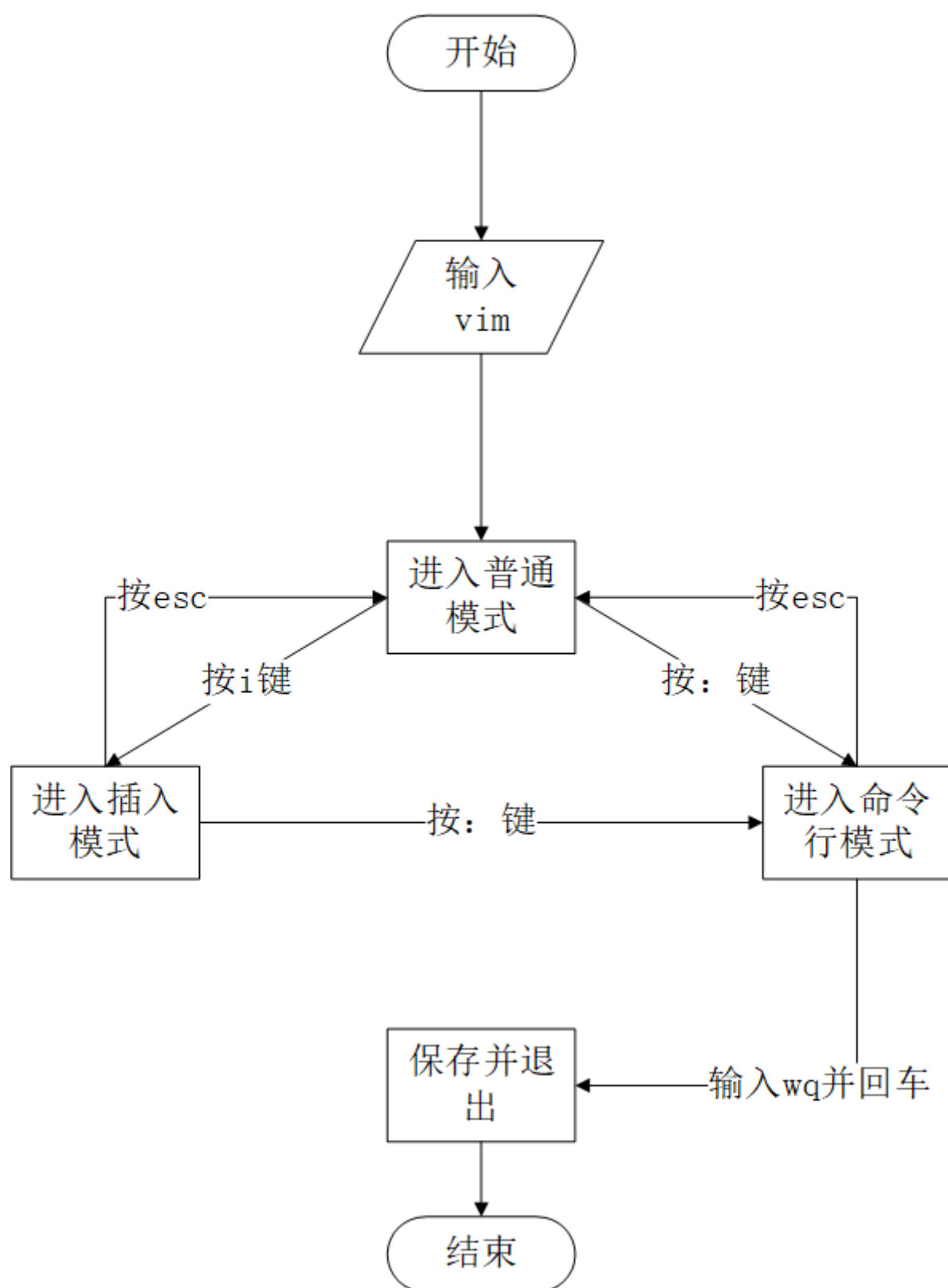
5.4.7 tail 命令



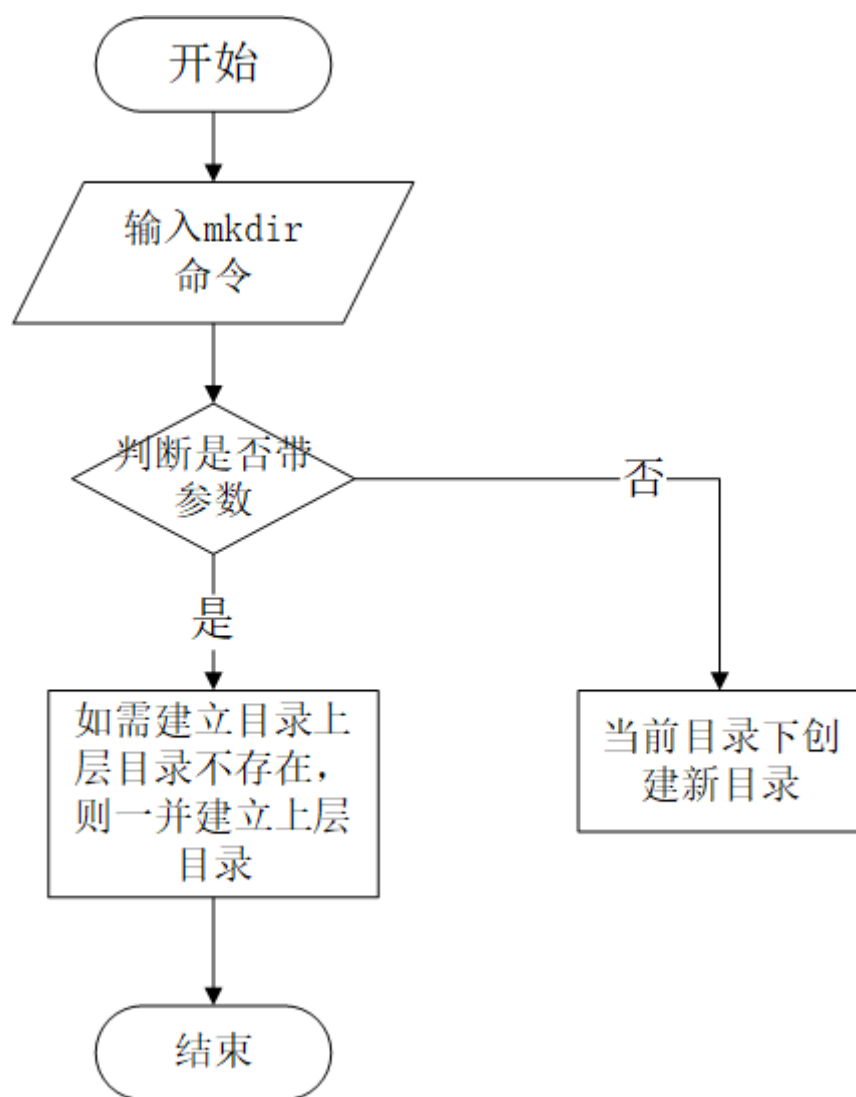
5.4.8 touch 命令



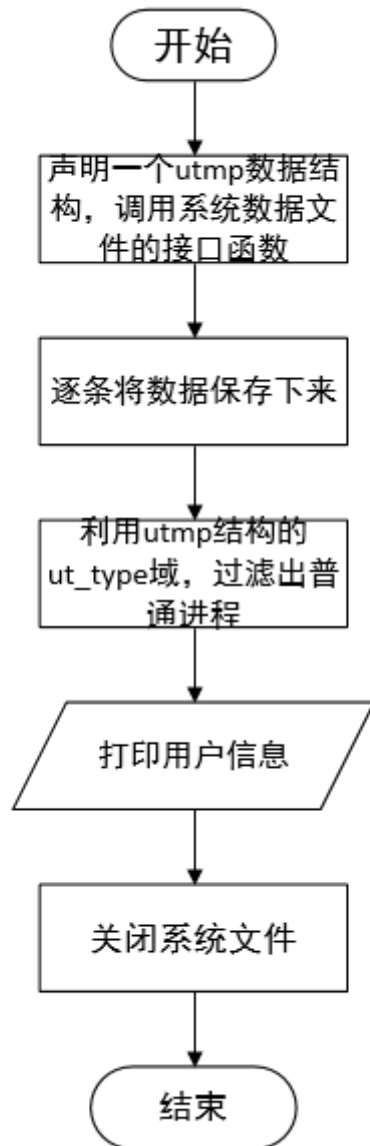
5.4.9 vim 命令



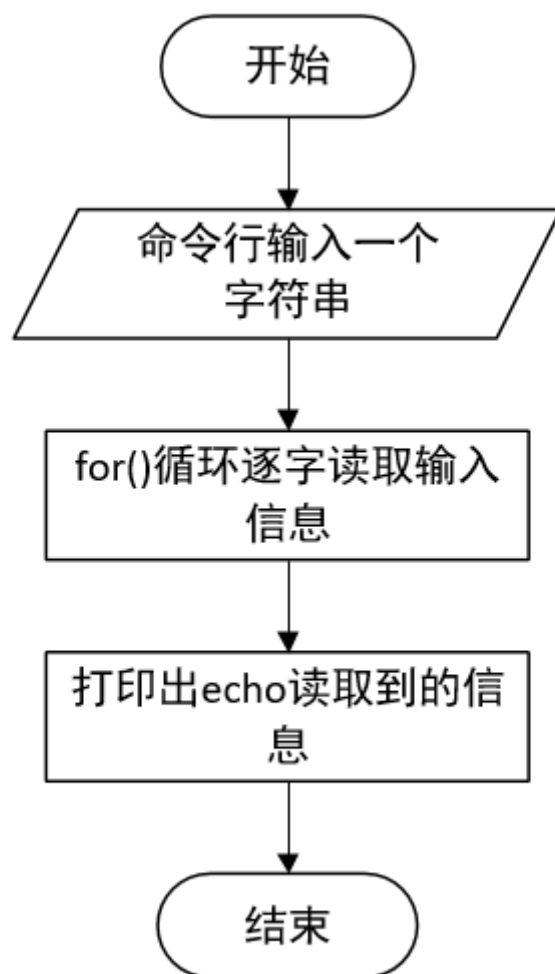
5.4.10 mkdir 命令



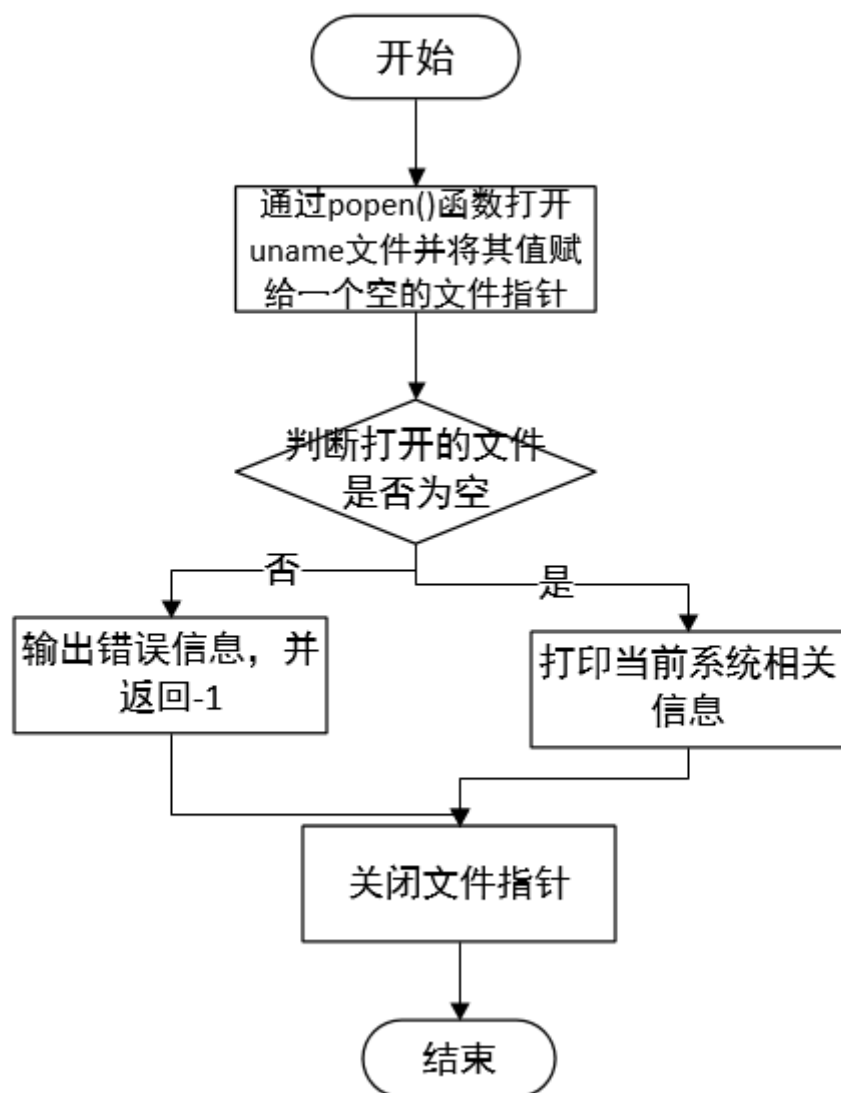
5.4.11 who 命令



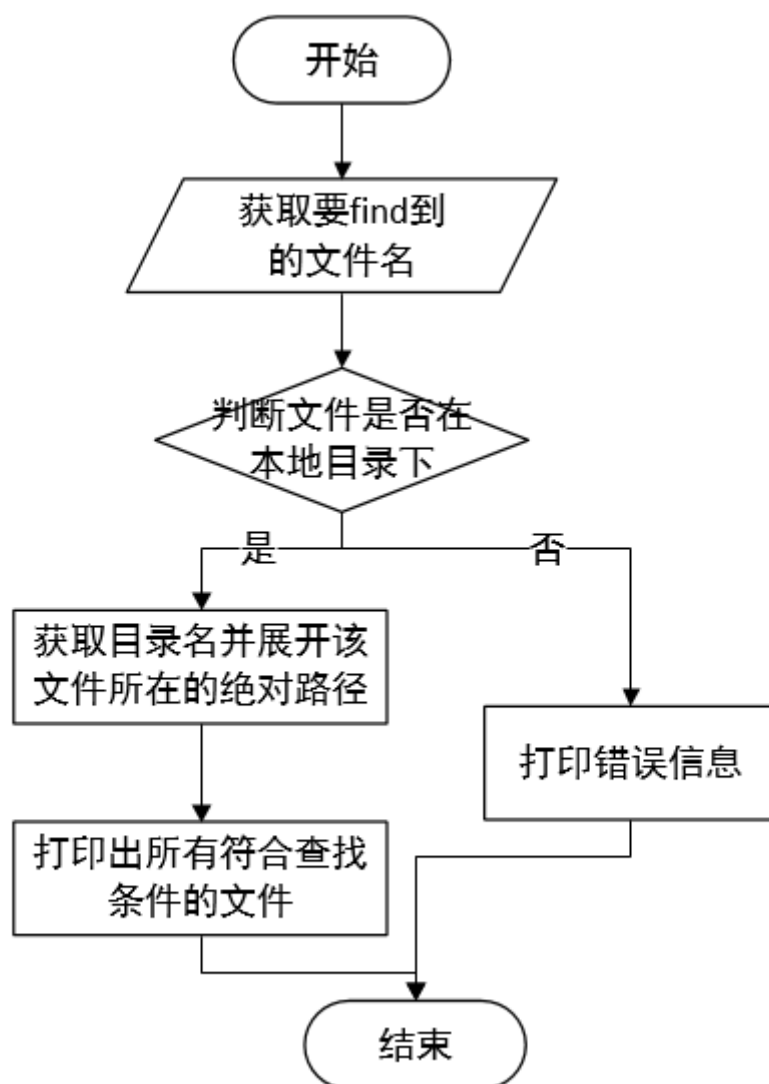
5.4.12 echo 命令



5.4.13 uname 命令



5.4.14 find 命令



5.4.15 su 命令

