

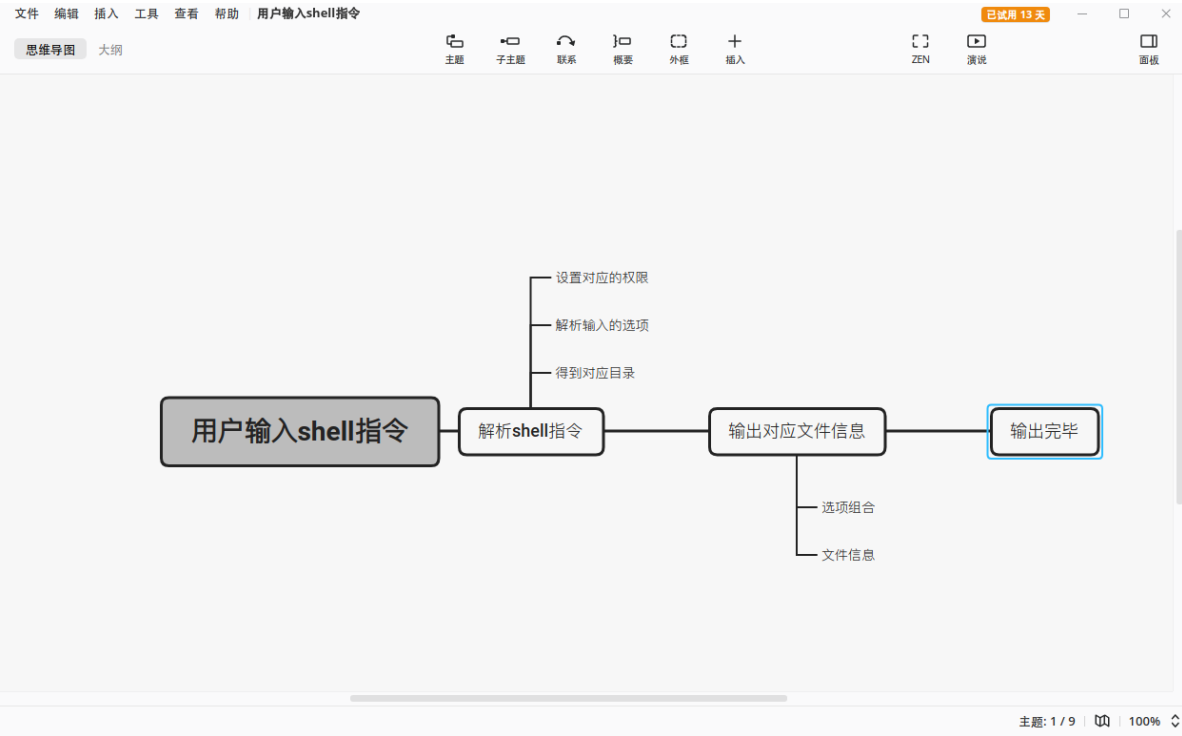
my_ls实验报告

1.项目概述

该实验基于本学期学习的linux程序设计课程文件相关知识，通过所学知识编写实现shell命令行中的ls命令，熟悉ls的相关指令及其操作。体会学习命令行编程，通过ls输出目录文件中的文件相关信息，并将特定的输出形式转换为标准的输出模式。综合所学知识完成ls的模仿开发。

2.功能流程图

功能视图

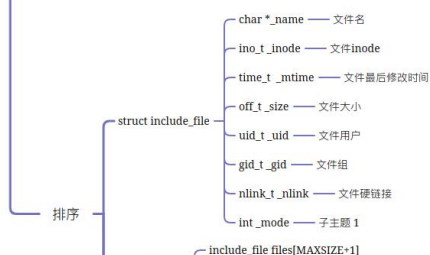
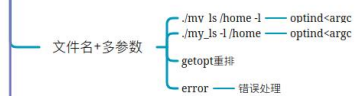
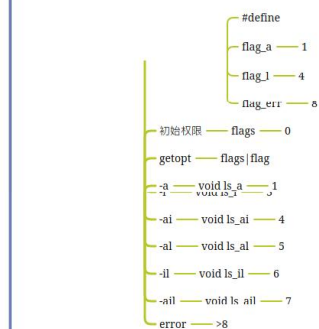
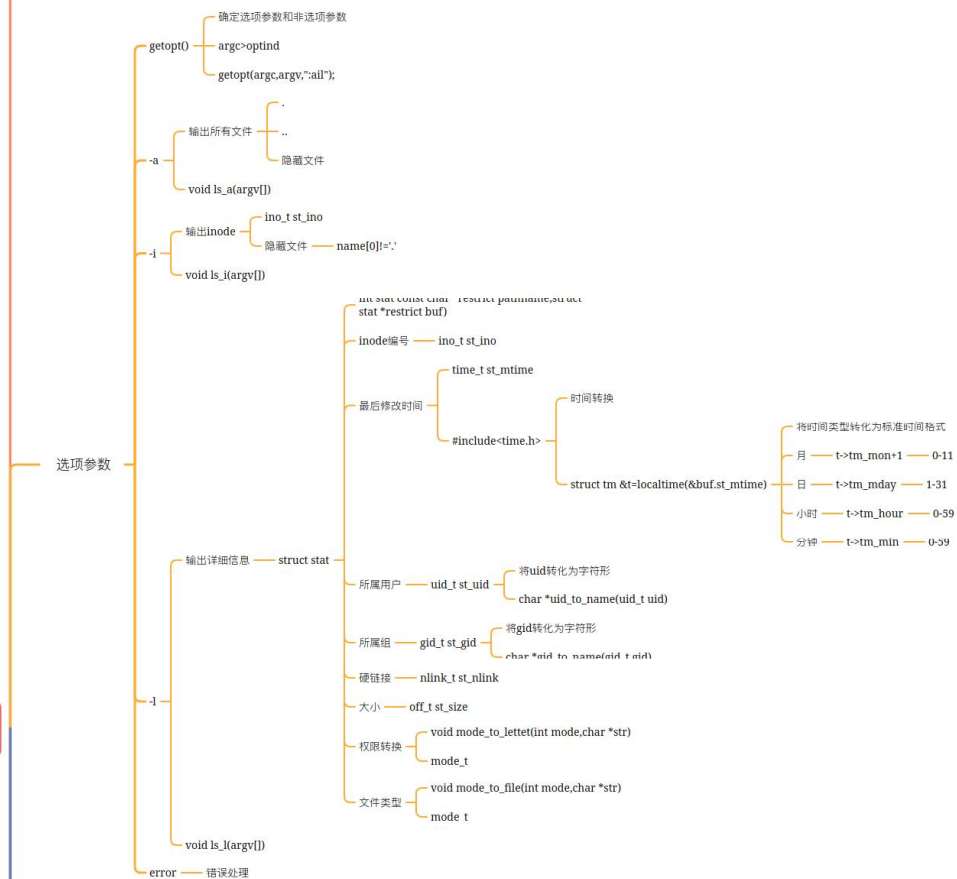
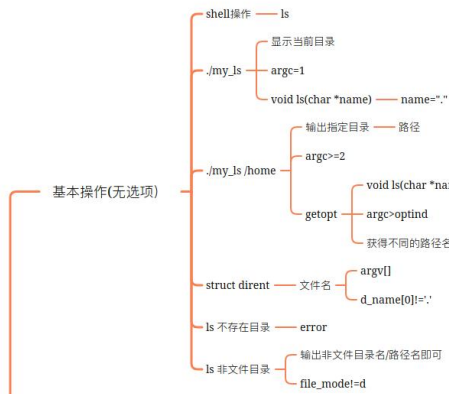


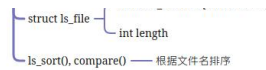
3.系统环境

本项目基于manjaro linux操作系统

4.思维导图

ls





5.功能概述

通过编译代码生成可执行文件my_ls，使用shell命令行调用my_ls，显示当前目录下的文件名。之后通过调用类似ls相关的指令来进行文件信息的查看。

调用./my_ls -a filename 来进行输出filename目录下的所有的文件名（包括隐藏文件）若是普通文件则触发异常。

调用./my_ls -i filename 来进行输出filename目录下的文件名及其inode号

调用./my_ls -l filename 来进行输出filename目录下的文件详细信息

以上操作可组合进行，且文件输出按文件名进行排序

6.程序运行说明

1)编译程序

```
fish /root/ls — Konsole
root ~ > ls
root ~ > gcc test.c -o my_ls
```

2) ./my_ls

```
Fish /root/ls — Konsole
root ~ > ls ./my_ls
a.out ch04.h err_exit.h hello.c ls.xmind my_ls实验报告.md my_ls temp test.c tips words
root ~ > ls
a.out ch04.h err_exit.h hello.c ls.xmind my_ls my_ls实验报告.md temp test.c tips words
root ~ > ls
```

3) ./my_ls -a

```
Fish /root/ls — Konsole
root ~ > ls ./my_ls -a
...hello a.out ch04.h err_exit.h hello.c ls.xmind my_ls temp test.c tips words
root ~ > ls -a
... a.out ch04.h err_exit.h .hello hello.c ls.xmind my_ls temp test.c tips words
root ~ > ls
```

4) ./my_ls -i

```
Fish /root/ls — Konsole
root ~ > ls ./my_ls -l
5257533 a.out 5260901 ch04.h 5266490 err_exit.h 5263695 hello.c 5266492 ls.xmind
5257183 my_ls 3166179 temp 5267013 test.c 5256978 tips 5266933 words
root ~ > ls -l
5257533 a.out 5266490 err_exit.h 5266492 ls.xmind 3166179 temp 5256978 tips
5260901 ch04.h 5263695 hello.c 5257183 my_ls 5267013 test.c 5266933 words
root ~ > ls
```

5) ./my_ls -l

```
Fish /root/ls — Konsole
root ~ > ls ./my_ls -l
-rw-r--r-- 1 root root 116 12月 7 10:18 err_exit.h
-rwxr-xr-x 1 root root 16088 12月 20 22:41 a.out
-rw-r--r-- 1 root root 302 12月 20 09:13 ch04.h
-rw-r--r-- 1 root root 116 12月 7 10:18 err_exit.h
-rw-r--r-- 1 root root 222 12月 20 22:42 hello.c
-rw-r--r-- 1 root root 171343 12月 21 23:06 ls.xmind
-rwxr-xr-x 1 root root 21480 12月 29 17:39 my_ls
drwxr-xr-x 2 root root 4096 12月 29 17:43 temp
-rw-r--r-- 1 root root 8804 12月 28 10:21 test.c
-rw-r--r-- 1 root root 2875 12月 15 16:17 tips
-rw-r--r-- 1 root root 90 12月 15 23:04 words
root ~ > ls -l
总用量 244
-rwxr-xr-x 1 root root 16088 12月 20 22:41 a.out
-rw-r--r-- 1 root root 302 12月 20 09:13 ch04.h
-rw-r--r-- 1 root root 116 12月 7 10:18 err_exit.h
-rw-r--r-- 1 root root 222 12月 20 22:42 hello.c
-rw-r--r-- 1 root root 171343 12月 21 23:06 ls.xmind
-rwxr-xr-x 1 root root 21480 12月 29 17:39 my_ls
drwxr-xr-x 2 root root 4096 12月 29 17:43 temp
-rw-r--r-- 1 root root 8804 12月 28 10:21 test.c
-rw-r--r-- 1 root root 2875 12月 15 16:17 tips
-rw-r--r-- 1 root root 90 12月 15 23:04 words
root ~ > ls
```

6) ./my_ls -a -i

```
Fish /root/ls — Konsole
root ~ > ls ./my_ls -al
5276365 . 5256903 .. 5276366 .hello 5257533 a.out 5260901 ch04.h 5266490 err_exit.h 5263695 hello.c 5266492 ls.xmind
5257183 my_ls 3166179 temp 5267013 test.c 5256978 tips 5266933 words
root ~ > ls -al
5276365 . 5257533 a.out 5266490 err_exit.h 5263695 hello.c 5257183 my_ls 5267013 test.c 5266933 words
5256903 .. 5260901 ch04.h 5276366 .hello 5266492 ls.xmind 3166179 temp 5256978 tips
root ~ > ls
```

7) ./my_ls -a -l

```
Fish /root/ls — Konsole
root ~ > ls ./my_ls -al
drwxr-xr-x 4 root root 4096 12月 29 17:42 .
drwxr-x-- 46 root root 4096 12月 29 16:26 ..
drwxr-xr-x 2 root root 4096 12月 14 18:57 .hello
-rwxr-xr-x 1 root root 16088 12月 20 22:41 a.out
-rw-r--r-- 1 root root 302 12月 20 09:13 ch04.h
-rw-r--r-- 1 root root 116 12月 7 10:18 err_exit.h
-rw-r--r-- 1 root root 222 12月 20 22:42 hello.c
-rw-r--r-- 1 root root 171343 12月 21 23:06 ls.xmind
-rwxr-xr-x 1 root root 21480 12月 29 17:39 my_ls
drwxr-xr-x 2 root root 4096 12月 29 17:49 temp
-rw-r--r-- 1 root root 8804 12月 28 10:21 test.c
-rw-r--r-- 1 root root 2875 12月 15 16:17 tips
-rw-r--r-- 1 root root 90 12月 15 23:04 words
root ~ > ls -al
总用量 256
drwxr-xr-x 4 root root 4096 12月 29 17:42 .
drwxr-x-- 46 root root 4096 12月 29 16:26 ..
drwxr-xr-x 1 root root 16088 12月 20 22:41 a.out
-rw-r--r-- 1 root root 302 12月 20 09:13 ch04.h
-rw-r--r-- 1 root root 116 12月 7 10:18 err_exit.h
drwxr-xr-x 2 root root 4096 12月 14 18:57 .hello
-rw-r--r-- 1 root root 222 12月 20 22:42 hello.c
-rw-r--r-- 1 root root 171343 12月 21 23:06 ls.xmind
-rwxr-xr-x 1 root root 21480 12月 29 17:39 my_ls
drwxr-xr-x 2 root root 4096 12月 29 17:49 temp
-rw-r--r-- 1 root root 8804 12月 28 10:21 test.c
-rw-r--r-- 1 root root 2875 12月 15 16:17 tips
-rw-r--r-- 1 root root 90 12月 15 23:04 words
root ~ > ls
```

8) ./my_ls -i -l

```
Fish /root/ls — Konsole
root ~ > ls ./my_ls -il
5257533 -rwxr-xr-x 1 root root 16088 12月 20 22:41 a.out
5260901 -rw-r--r-- 1 root root 302 12月 20 09:13 ch04.h
5266490 -rw-r--r-- 1 root root 116 12月 7 10:18 err_exit.h
5263695 -rw-r--r-- 1 root root 222 12月 20 22:42 hello.c
5266492 -rw-r--r-- 1 root root 171343 12月 21 23:06 ls.xmind
5257183 -rwxr-xr-x 1 root root 21480 12月 29 17:39 my_ls
3166179 drwxr-xr-x 2 root root 4096 12月 29 17:49 temp
5267013 -rw-r--r-- 1 root root 8804 12月 28 10:21 test.c
5256978 -rw-r--r-- 1 root root 2875 12月 15 16:17 tips
5266933 -rw-r--r-- 1 root root 90 12月 15 23:04 words
root ~ > ls -il
总用量 244
5257533 -rwxr-xr-x 1 root root 16088 12月 20 22:41 a.out
5260901 -rw-r--r-- 1 root root 302 12月 20 09:13 ch04.h
5266490 -rw-r--r-- 1 root root 116 12月 7 10:18 err_exit.h
5263695 -rw-r--r-- 1 root root 222 12月 20 22:42 hello.c
5266492 -rw-r--r-- 1 root root 171343 12月 21 23:06 ls.xmind
5257183 -rwxr-xr-x 1 root root 21480 12月 29 17:39 my_ls
3166179 drwxr-xr-x 2 root root 4096 12月 29 17:49 temp
5267013 -rw-r--r-- 1 root root 8804 12月 28 10:21 test.c
5256978 -rw-r--r-- 1 root root 2875 12月 15 16:17 tips
5266933 -rw-r--r-- 1 root root 90 12月 15 23:04 words
root ~ > ls
```

9) ./my_ls -a -i -l

```
Fish /root/ls — Konsole
root ~ > ls ./my_ls -ail
5276365 drwxr-xr-x 4 root root 4096 12月 29 17:42 .
5256903 drwxr-xr-x 46 root root 4096 12月 29 16:26 ..
5276366 drwxr-xr-x 2 root root 4096 12月 14 18:57 .hello
5257533 -rwxr-xr-x 1 root root 16088 12月 20 22:41 a.out
5260901 -rw-r--r-- 1 root root 302 12月 20 09:13 ch04.h
5266490 -rw-r--r-- 1 root root 116 12月 7 10:18 err_exit.h
5263695 -rw-r--r-- 1 root root 222 12月 20 22:42 hello.c
5266492 -rw-r--r-- 1 root root 171343 12月 21 23:06 ls.xmind
5257183 -rwxr-xr-x 1 root root 21480 12月 29 17:39 my_ls
3166179 drwxr-xr-x 2 root root 4096 12月 29 17:50 temp
5267013 -rw-r--r-- 1 root root 8804 12月 28 10:21 test.c
5256978 -rw-r--r-- 1 root root 2875 12月 15 16:17 tips
5266933 -rw-r--r-- 1 root root 90 12月 15 23:04 words
root ~ > ls -ail
总用量 256
5276365 drwxr-xr-x 4 root root 4096 12月 29 17:42 .
5256903 drwxr-xr-x 46 root root 4096 12月 29 16:26 ..
5257533 -rwxr-xr-x 1 root root 16088 12月 20 22:41 a.out
5260901 -rw-r--r-- 1 root root 302 12月 20 09:13 ch04.h
5266490 -rw-r--r-- 1 root root 116 12月 7 10:18 err_exit.h
5276366 drwxr-xr-x 2 root root 4096 12月 14 18:57 .hello
5263695 -rw-r--r-- 1 root root 222 12月 20 22:42 hello.c
5266492 -rw-r--r-- 1 root root 171343 12月 21 23:06 ls.xmind
5257183 -rwxr-xr-x 1 root root 21480 12月 29 17:39 my_ls
3166179 drwxr-xr-x 2 root root 4096 12月 29 17:50 temp
5267013 -rw-r--r-- 1 root root 8804 12月 28 10:21 test.c
5256978 -rw-r--r-- 1 root root 2875 12月 15 16:17 tips
5266933 -rw-r--r-- 1 root root 90 12月 15 23:04 words
root ~ > ls
```

10) ./my_ls filename

```
Fish /root/ls — Konsole
root ~ - ls ./my_ls ./SQL/
GDB(程序调试工具) .xmind NoSQL.xmind SQLserver.xmind mysql.xmind postgresql.xmind
root ~ - ls ./SQL/
GDB(程序调试工具) .xmind' mysql.xmind NoSQL.xmind postgresql.xmind SQLserver.xmind
root ~ - ls
```

7.代码分析

选项选择

为-a定义一个宏参量#define flag_a 1 为-i定义一个宏参量 #define flag_i 2

为-l定义一个宏参量#define flag_l 4 为错误选项定义一个宏参量#define flag_err 8

通过while(getopt)来判断是否拥有该参数若有进行运算flags|flag_x得到该权限，若是错误参数则进行flags|flag_err

文件地址

本题中的所有使用到的选项均不要求带有参数，所以所有的非选项都可以认为是文件地址，即当while（getopt）循环结束时

若optind==argc,可说明未带有地址参数，输出当前目录中的文件(".")

若optind<argc,可说明带有了地址参数，输出optind++目录中的文件信息（argv[index]）

无选项输出

dp=opendir（filename）打开filename目录，通过while(dir=readdir(dp))!=NULL来读取filename中的文件信息

若文件开头dir->d_name[0]=='.'即为隐藏文件，则不输出,其他直接输出目录下的文件名

-a输出

输出所有的文件（包括隐藏文件），即去掉判断dir->d_name[0]=='.'即可

-i输出

在while（dir=readdir(dp))!=NULL中先判断读取到的文件非隐藏文件

再读取文件的dir->d_ino与其dir->d_name来进行输出对应的文件名与inode

-l输出

在while (dir=readdir(dp)!=NULL)中先判断读取到的文件非隐藏文件

通过dir来输出文件的具体信息d_name,d_ino.....

文件权限装换

在得到文件的mode即stat(dir->d_name,&buf)后, 通过buf访问文件的st_mode,得到文件的权限。

通过自定义函数mode_to_letter将其转化为一般计算机显示形式(-rwxr-xr-x)的形式

文件时间转换

在得到文件的时间buf.st_mtime后, 通过调用<time.h>中的函数localtime(&buf.st_mtime)将其转化为struct tm的类似,

并将其赋值给struct tm *t;即可通过t进行调用结构体中的分(tm_min), 小时(tm_hour), 日(tm_mday), 月(tm_mon+1)变量

进行时间的输出

```
//struct tm *localtime(const time_t timer);

struct tm { int tm_sec;    /* 秒, 范围从 0 到 59 */
int tm_min;    /* 分, 范围从 0 到 59 */
int tm_hour;    /* 小时, 范围从 0 到 23 */
int tm_mday;    /* 一月中的第几天, 范围从 1 到 31 */
int tm_mon;    /* 月份, 范围从 0 到 11 */
int tm_year;    /* 自 1900 起的年数 */
int tm_wday;    /* 一周中的第几天, 范围从 0 到 6 */
int tm_yday;    /* 一年中的第几天, 范围从 0 到 365 */
int tm_isdst;    /* 夏令时 */
};
```

文件所属用户转换

调用自定义函数uid_to_name(buf.st_uid),通过 getpwuid (uid) ->pw_name得到用户信息。

文件所属组转换

调用自定义函数gid_to_name(buf.st_gid),通过getgrgid(gid)->gr_name得到组信息

文件输出排序

自定义一个结构体struct include_file进行存储文件的信息,再自定义一个结构体

```
typedef struct {
    include_file files[MAXSIZE+1];
    int length;
}ls_file;
```

来进行存储所有文件的信息。

自定义一个函数ls_sort()和compare(char *s1,char *s2)来对文件进行排序compare文件排序关键字比较

8.实验缺陷

1).ls -i输出的顺序为前列后行，而本实验中-i输出为先行后列

2).对隐藏目录中的比较不同，ls默认忽略'.'进行排序

3).输出时可以对不同文件以不同颜色输出