# 小组1

**成员：唐钰斌 苟明开 胡桂飞 任务：完成get\_opt文件夹下所有文件**

函数接口：

// 获取参数个数，不包含./a.out命令本身

int get\_opt\_size(int argc);

// 获取路径

char\* get\_path(int argc, char const \*argv[]);

// 判断是否有-l参数

int is\_l(int argc,char const \*argv[]);

代码编写：

## get\_opt\_size.c:

#include <stdio.h>

#include "get\_opt\_size.h"

int get\_opt\_size(int argc){

// 返回除ls命令本身参数个数

return argc - 1;

}

int main(int argc, char const \*argv[])

{

printf("%d\n", get\_opt\_size(argc));

return 0;

}

get\_path.c:

#include <stdio.h>

#include "get\_path.h"

char\* get\_path(int argc, char const \*argv[]){

return argv[argc - 1];

}

int main(int argc, char const \*argv[])

{

printf("the input path is %s\n", get\_path(argc, argv));

return 0;

}

is\_l.c:

#include <stdio.h>

#include <string.h>

#include "is\_l.h"

#include "get\_opt\_size.h"

// 判断是否有-l 参数 有返回1 无返回0

int is\_l(int argc, char const \*argv[]){

if(get\_opt\_size(argc) >= 1){

if(!strcmp("-l", argv[1])){

return 1;

}

}

return 0;

}

int main(int argc, char const \*argv[])

{

if(is\_l(argc, argv)){

printf("have -l parameter\n");

}else{

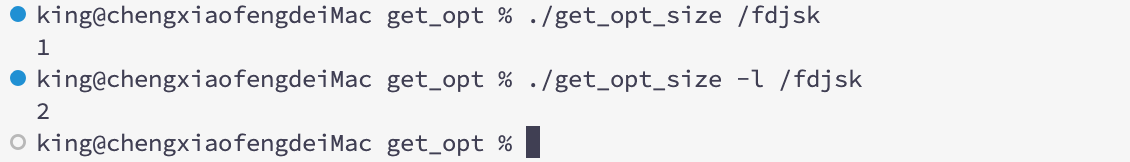
printf("no -l parameter\n");

}

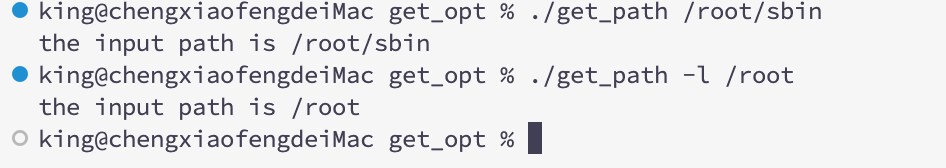
return 0;

}

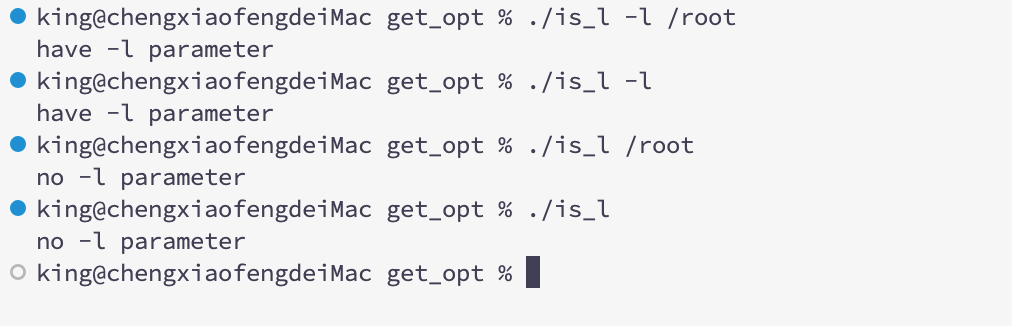
### 单元测试结果：

get\_opt\_size: 

get\_path:



is\_l:



# 小组2

**成员：罗淞 颜志豪 李思衡 任务：完成Main.c**

## Main.c

代码编写：#include<stdio.h>

#include<stdlib.h>

#include<get\_opt\_size.h>

#include<get\_path.h>

#include<is\_l.h>

#include<show\_file\_all\_info.h>

#include<show\_file\_names.h>

main(int argc, char\*argv[]){

char str = '.';

int rt = -1;

if(argc>3)

{

printf("閿欒");

exit(1);

}

if(argc == 1)

lsdir(&str);

else if(argc == 2)

{

if(strcmp(argv[1],"-1") == 0 )

lsdir(&str);

else

{

checkfiletype(argv[1]);

if(typeflag == 2)

lsdir(argv[1]);

else if(typeflag == 1)

printf("%s\n",argv[1]);

}

}

else if(argc == 3)

{

if(strcmp(argv[1],"-1") == 0)

{

checkfiletype(argv[2]);

if(typeflag == 2)

lsdir(argv[2]);

else if(typeflag == 1)

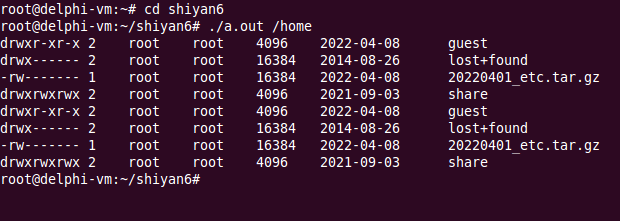
printlong(argv[2]);

}

}

}

### 单元测试结果：



# 小组3

**成员：李时进 施婷 申黄琪 董静 任务： 完成show\_file\_names.c**

函数接口：

// 显示路径path下所有文件的文件名

## void show\_file\_names(char\* path);

代码编写：

#include<stdio.h>

#include<sys/types.h>

#include<dirent.h>

#include<stdlib.h>

void show\_file\_names(char[]);

int main(int argc, char\*\* argv)

{

if (argc == 1) {//查看当前目录

show\_file\_names(".");

}

else {

while (--argc) {//查看指定的一个或多个目录

printf("%s : \n", \*++argv);//打印当前所查看的目录路径

show\_file\_names(\*argv);//print环节

}

}

}

void show\_file\_names(char dirname[])

{

DIR\* dir\_ptr;//文件夹指针

struct dirent\* direntp;//存放文件夹目录内容

if ((dir\_ptr = opendir(dirname)) == NULL) {

fprintf(stderr, "ls1: open error %s\n", dirname);//打开失败

}

else {

while ((direntp = readdir(dir\_ptr)) != NULL) {//while（readdir）

printf("%s\n", direntp->d\_name);//d\_name:

}

int flag;

if ((flag = closedir(dir\_ptr)) == -1) {

perror("ls1: close dir error");//关闭失败

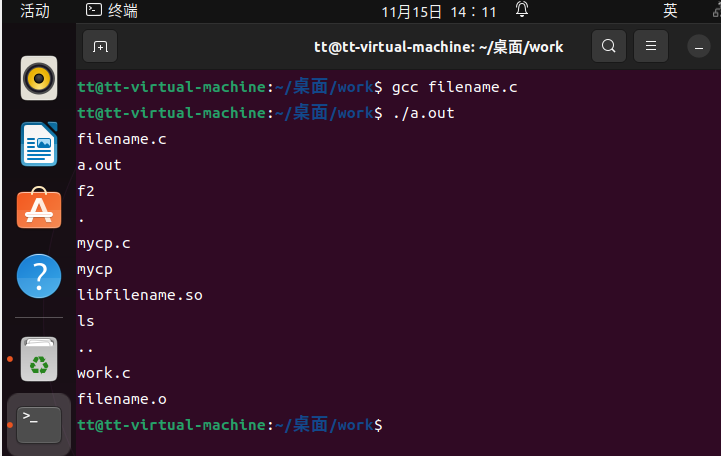
exit(0);

}

}

}

### 单元测试结果：



# 小组4

**成员：龙月 刘维 曾春琳 彭喆 任务：完成show\_file\_all\_info.c**

函数接口：

// 显示路径path下所有文件的长格式

void show\_file\_all\_info(char\* path)

代码编写：

## show\_file\_all\_info.c

void gettype(char \*name)//判断文件类型

{

struct stat buf;

lstat(name,&buf);

switch(buf.st\_mode&S\_IFMT)

{

case S\_IFREG:

typeflag=1;

break;

case S\_IFDIR:

typeflag=2;

break;

case S\_IFLNK:

typeflag=3;

break;

case S\_IFCHR:

typeflag=4;

break;

case S\_IFBLK:

typeflag=5;

break;

case S\_IFSOCK:

typeflag=6;

break;

case S\_IFIFO:

typeflag=7;

break;

default:

typeflag=0;

}

if(typeflag==1)

{

print\_long(name);

}

else if(typeflag==2)

{

lsdir(name);

}

else

printf("cannot print this type");

}

void lsdir(char \*name)//显示所有文件的文件名

{

DIR \*d1;

struct dirent \*dent1;

char namebuf[128];

char workdir[50];

d1=opendir(name);

if(d1==NULL)

{

perror("opendir");

exit(1);

}

getcwd(workdir,50);

chdir(name);

errno=0;

dent1=readdir(d1);

while(dent1!=NULL)

{

if(dent1->d\_name[0]!='.')

{

print\_long(dent1->d\_name);

}

dent1=readdir(d1);

}

if(errno!=0)

perror("readdir");

closedir(d1);

chdir(workdir);

}

void print\_size(struct stat \*statp)//打印文件大小

{

switch(statp->st\_mode & S\_IFMT)

{

case S\_IFCHR:

case S\_IFBLK:

printf("%u,%u",(unsigned)(statp->st\_rdev>>8),(unsigned)(statp->st\_rdev&0xFF));

break;

default:

printf("%lu",(unsigned long)(statp->st\_size));

}

}

void print\_long(char \*name)//得到长格式

{

struct stat buf;

struct passwd \*user;

struct group \*grp;

char linkname[64];

char dirfilename[64];

int rt;

rt=lstat(name,&buf);

if(-1==rt)

{

perror("in print\_long:lstat");

return;

}

switch(buf.st\_mode&S\_IFMT)

{

case S\_IFDIR:

printf("d");

break;

case S\_IFLNK:

printf("l");

break;

case S\_IFREG:

printf("-");

break;

case S\_IFBLK:

printf("b");

break;

case S\_IFCHR:

printf("c");

break;

case S\_IFSOCK:

printf("s");

break;

case S\_IFIFO:

printf("p");

break;

default:

printf("?");

break;

}

putchar((buf.st\_mode&S\_IRUSR)?'r':'-');

putchar((buf.st\_mode&S\_IWUSR)?'w':'-');

if(buf.st\_mode&S\_ISUID)

{

putchar((buf.st\_mode&S\_IXUSR)?'s':'S');

}

else

{

putchar((buf.st\_mode&S\_IXUSR)?'x':'-');

}

putchar((buf.st\_mode&S\_IRGRP)?'r':'-');

putchar((buf.st\_mode&S\_IWGRP)?'w':'-');

if(buf.st\_mode&S\_ISGID)

{

putchar((buf.st\_mode&S\_IXGRP)?'s':'S');

}

else

{

putchar((buf.st\_mode&S\_IXGRP)?'x':'-');

}

putchar((buf.st\_mode&S\_IROTH)?'r':'-');

putchar((buf.st\_mode&S\_IWOTH)?'w':'-');

if(buf.st\_mode&S\_ISVTX)

{

putchar((buf.st\_mode&S\_IXOTH)?'t':'T');

}

else

{

putchar((buf.st\_mode&S\_IXOTH)?'x':'-');

}

printf("%2u",buf.st\_nlink);

user=getpwuid(buf.st\_uid);

printf("\t%s",user->pw\_name);

grp=getgrgid(buf.st\_gid);

printf("\t%s\t",grp->gr\_name);

print\_size(&buf);

print\_date(&buf);

if(typeflag==3)

{

rt=readlink(name,linkname,sizeof(linkname));

linkname[rt]=0;

printf("\t%s->%s",name,linkname);

}

else

{

printf("\t%s",name);

}

printf("\n");

}

void print\_date(struct stat \*statp)//得到文件的时间

{

time\_t now;

double diff;

char buf[100],\*fmt;

if(time(&now)==-1)

{

printf("????????????");

return;

}

diff=difftime(now,statp->st\_mtime);

if(diff<0||diff>60\*60\*24\*128.5)

fmt="%Y-%m-%d";

else

fmt="%Y-%m-%d %H:%M";

strftime(buf,sizeof(buf),fmt,localtime(&statp->st\_mtime));

printf("\t%s",buf);

}

void show\_file\_all\_info(char\* path){

gettype(path);

}

### 单元测试结果：

