1. 传递一个路径名，深度优先遍历该路径并打印

#include <func.h>

int print\_dir(char \*path,int width)

{

    DIR \*dir;

    dir=opendir(path);

    ERROR\_CHECK(dir,NULL,"opendir")

    struct dirent \*p;

    char buf[1024];

    while((p=readdir(dir)))

    {

        if(!strcmp(".",p->d\_name)||!strcmp("..",p->d\_name))

        {

        }else{

            printf("%\*s%s\n",width,"",p->d\_name);

            if(p->d\_type==4)

            {

                memset(buf,0,sizeof(buf));

                sprintf(buf,"%s%s%s",path,"/",p->d\_name);

                print\_dir(buf,width+4);

            }

        }

    }

}

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

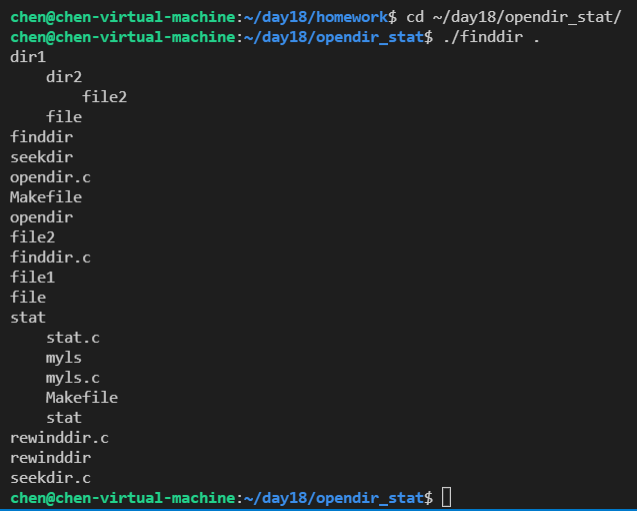
{

    ARGS\_CHECK(argc,2)

    print\_dir(argv[1],0);

    return 0;

}



1. 传递任意一个目录路径，能够显示该目录的ls -l的效果

#include <func.h>

int getmode(mode\_t x,char\* mode)

{

    int i,j,count=0;

    char tmp;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

            if(x&(1<<(8-count)))

            {

                switch(j)

                {

                    case 0:

                        tmp='r';

                        break;

                    case 1:

                        tmp='w';

                        break;

                    case 2:

                        tmp='x';

                        break;

                }

            }

            else {

                tmp='-';

            }

            mode[count]=tmp;

            count++;

        }

    }

    mode[count]=0;

    return 0;

}

char gettype(unsigned char x)

{

    switch(x)

    {

        case 1:

            return 'p';

        case 2:

            return 'c';

        case 4:

            return 'd';

        case 6:

            return 'b';

        case 8:

            return '-';

    }

}

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

{

    ARGS\_CHECK(argc,2)

    DIR\* dir;

    dir=opendir(argv[1]);

    if(NULL==dir)

    {

        perror("opendir");

        return -1;

    }

    struct dirent \*p;

    struct stat buf;

    char time[13];

    char mode[10];

    int ret;

    while((p=readdir(dir)))

    {

        if(strcmp(p->d\_name,".")==0||strcmp(p->d\_name,"..")==0)

        {

            continue;

        }

        ret=stat(p->d\_name,&buf);

        getmode(buf.st\_mode,mode);

        strncpy(time,ctime(&buf.st\_mtime)+4,12);

        printf("%c%s %3ld %s %s %6ld %s %s\n",\

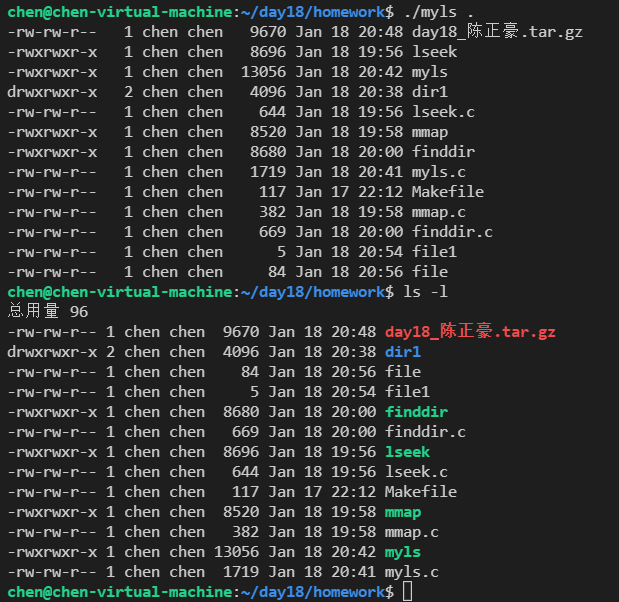
        gettype(p->d\_type),mode,buf.st\_nlink,getpwuid(buf.st\_uid)->pw\_name,getgrgid(buf.st\_gid)->gr\_name,buf.st\_size,time,p->d\_name);

    }

    closedir(dir);

    return 0;

}



3、定义一个学生结构体类型struct student，里边含有学号，姓名，分数，定义结构体数组struct student s[3],给数组赋初值后，写入文件，然后通过lseek偏移到开头，然后再读取进行打印输出（注意，是直接把结构体数组写入文件，双击打开文件看不到学号，分数这些的，也不需要双击打开看）

#include <func.h>

typedef struct student

{

    int num;

    char name[20];

    float score;

}stu,\*pstu;

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

{

    ARGS\_CHECK(argc,2)

    stu buf[3]={{1001,"lele",92.30},{1002,"xiongda",95.60},{1003,"hanmeimei",93.50}};

    int fd;

    fd=open(argv[1],O\_RDWR);

    ERROR\_CHECK(fd,-1,"open")

    write(fd,buf,sizeof(buf));

    int ret;

    ret=lseek(fd,0,SEEK\_SET);

    ERROR\_CHECK(ret,-1,"lseek")

    memset(buf,0,sizeof(buf));

    read(fd,buf,sizeof(buf));

    int i;

    for(i=0;i<3;i++)

    {

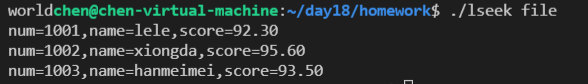
        printf("num=%d,name=%s,score=%.2f\n",buf[i].num,buf[i].name,buf[i].score);

    }

    close(fd);

    return 0;

}



4、新建一个文件，里边内容为hello，通过mmap映射该文件后，修改hello为world，然后解除映射

#include <func.h>

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

{

    ARGS\_CHECK(argc,2)

    int fd;

    fd=open(argv[1],O\_RDWR);

    ERROR\_CHECK(fd,-1,"open")

    int ret=ftruncate(fd,5);

    char \*p;

    p=(char\*)mmap(NULL,5,PROT\_READ|PROT\_WRITE,MAP\_SHARED,fd,0);

    ERROR\_CHECK(p,(char\*)-1,"mmap")

    strcpy(p,"world");

    ret=munmap(p,5);

    ERROR\_CHECK(ret,-1,"munmap")

    return 0;

}

