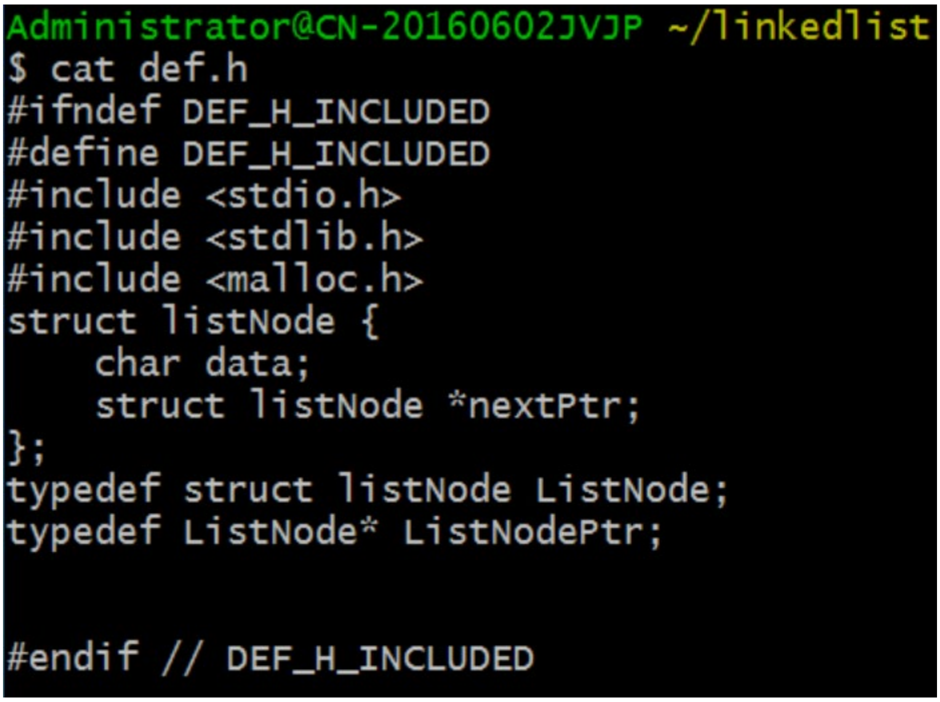
Linux 实验一指导书

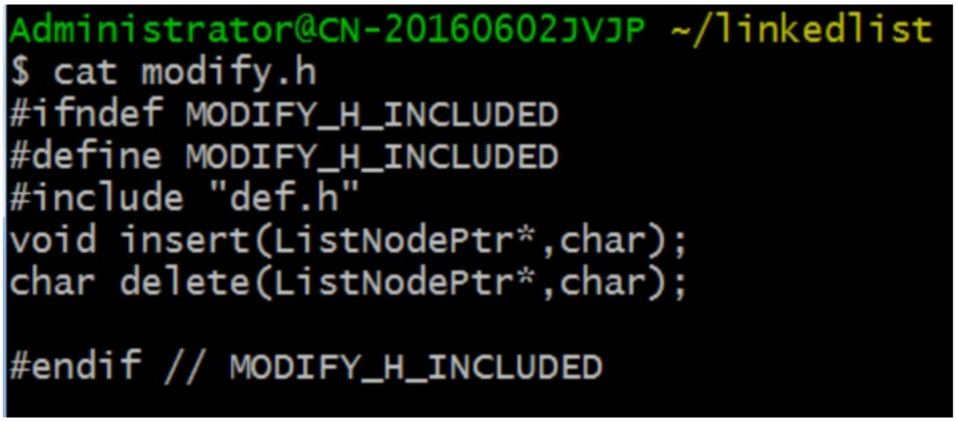
- 班级：软件 15 级，软件 zy 15 级，软件 sy 15 级。
- 指导老师：祁明龙。
- 地点：鉴主 10 楼计算机学院机房。
- 时间：第九周。
- 内容：利用 GNU Makefile C 语言项目管理机制实现链表。
- 示例程序：

1. 节点及类型定义模块：

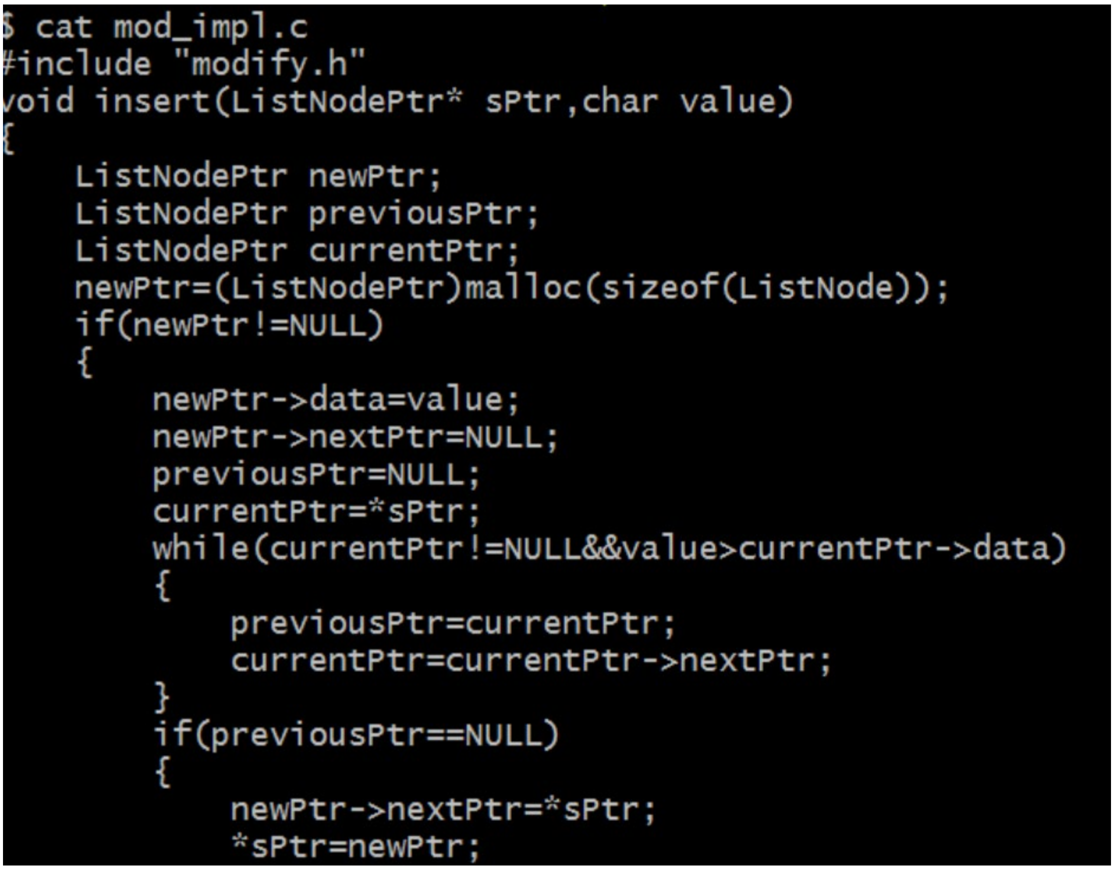


```
Administrator@CN-20160602JVJP ~/linkedlist
$ cat def.h
#ifndef DEF_H_INCLUDED
#define DEF_H_INCLUDED
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
struct listNode {
    char data;
    struct listNode *nextPtr;
};
typedef struct listNode ListNode;
typedef ListNode* ListNodePtr;


#endif // DEF_H_INCLUDED
```

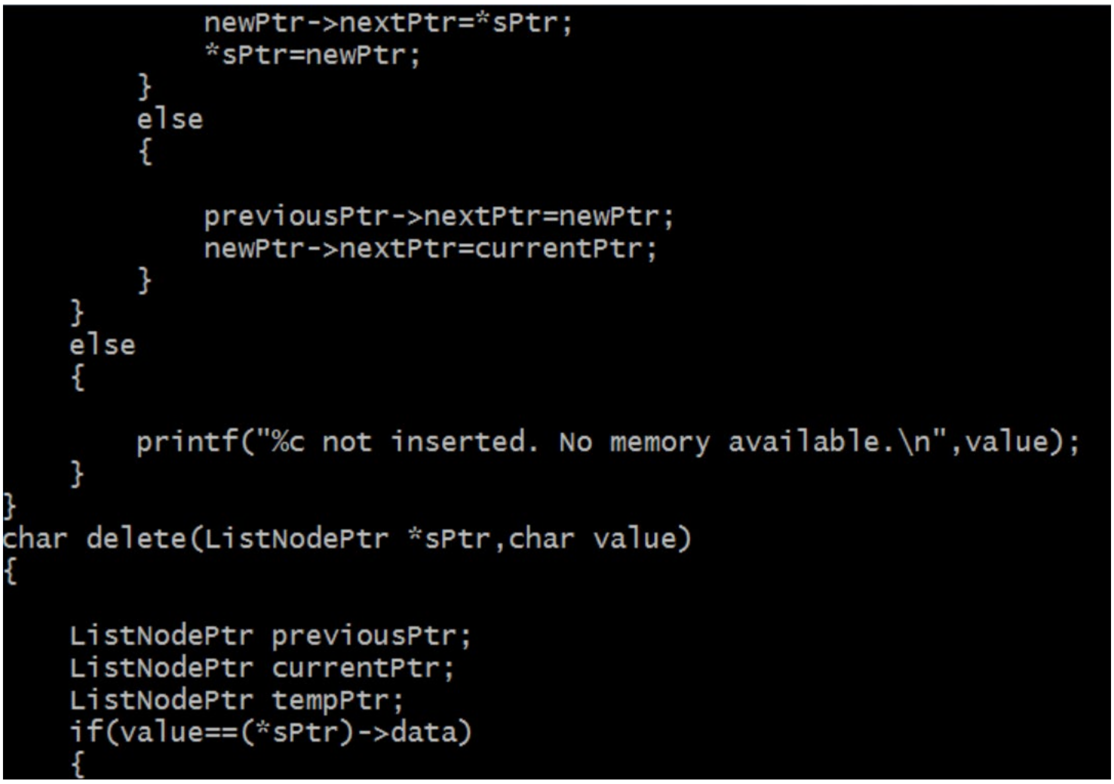
2. 插入、删除模块接口及实现：



```
Administrator@CN-20160602JVJP ~/linkedlist
$ cat modify.h
#ifndef MODIFY_H_INCLUDED
#define MODIFY_H_INCLUDED
#include "def.h"
void insert(ListNodePtr*,char);
char delete(ListNodePtr*,char);

#endif // MODIFY_H_INCLUDED
```

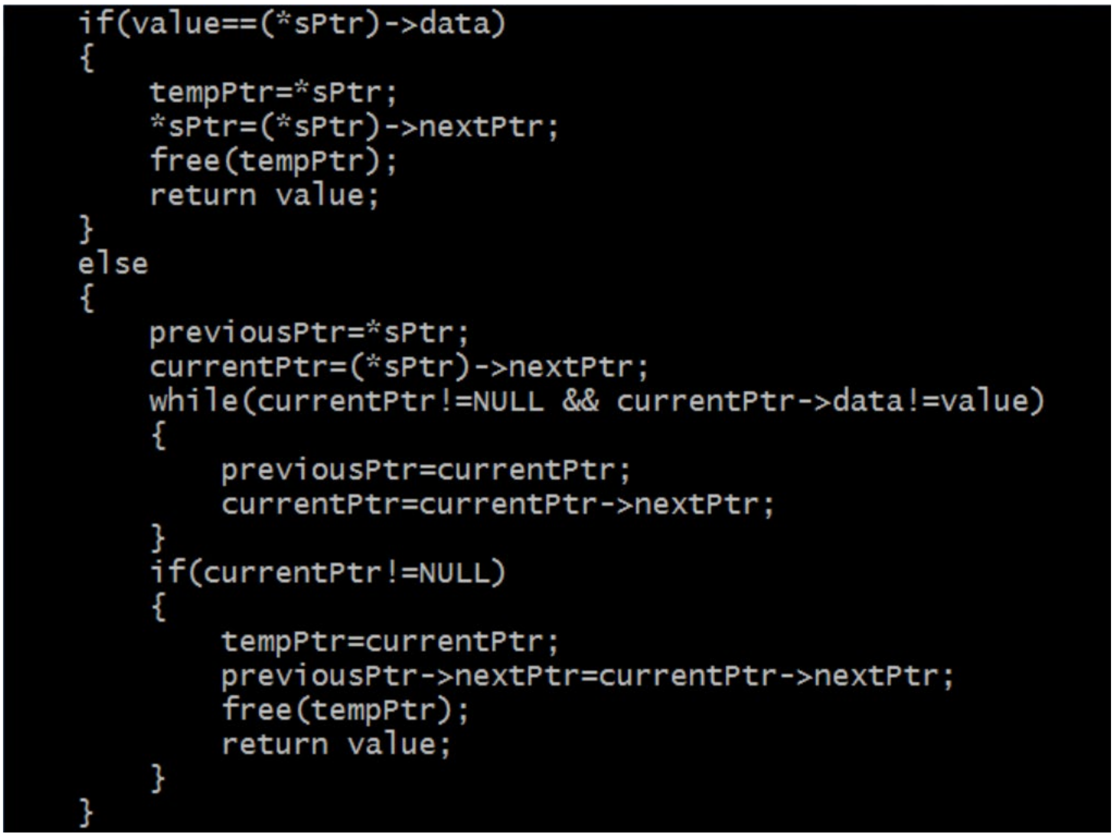


```
$ cat mod_impl.c
#include "modify.h"
void insert(ListNodePtr* sPtr,char value)
{
    ListNodePtr newPtr;
    ListNodePtr previousPtr;
    ListNodePtr currentPtr;
    newPtr=(ListNodePtr)malloc(sizeof(ListNode));
    if(newPtr!=NULL)
    {
        newPtr->data=value;
        newPtr->nextPtr=NULL;
        previousPtr=NULL;
        currentPtr=*sPtr;
        while(currentPtr!=NULL&&value>currentPtr->data)
        {
            previousPtr=currentPtr;
            currentPtr=currentPtr->nextPtr;
        }
        if(previousPtr==NULL)
        {
            newPtr->nextPtr=*sPtr;
            *sPtr=newPtr;
```

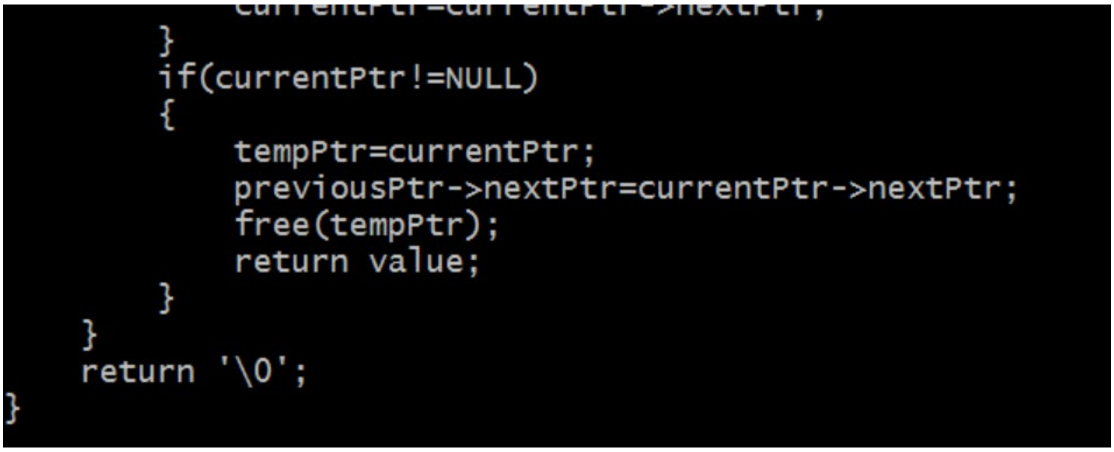


```
            newPtr->nextPtr=*sPtr;
            *sPtr=newPtr;
```

```
        }
        else
        {

            previousPtr->nextPtr=newPtr;
            newPtr->nextPtr=currentPtr;
        }
    }
    else
    {

        printf("%c not inserted. No memory available.\n",value);
    }
}
char delete(ListNodePtr *sPtr,char value)
{

    ListNodePtr previousPtr;
    ListNodePtr currentPtr;
    ListNodePtr tempPtr;
    if(value==(*sPtr)->data)
    {
```

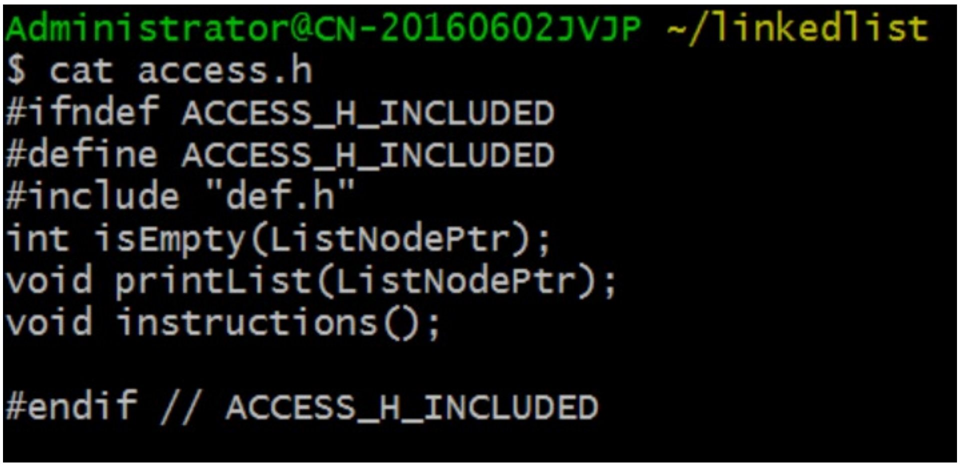


```
    if(value==(*sPtr)->data)
    {
        tempPtr=*sPtr;
        *sPtr=(*sPtr)->nextPtr;
        free(tempPtr);
        return value;
    }
    else
    {
        previousPtr=*sPtr;
        currentPtr=(*sPtr)->nextPtr;
        while(currentPtr!=NULL && currentPtr->data!=value)
        {
            previousPtr=currentPtr;
            currentPtr=currentPtr->nextPtr;
        }
        if(currentPtr!=NULL)
        {
            tempPtr=currentPtr;
            previousPtr->nextPtr=currentPtr->nextPtr;
            free(tempPtr);
            return value;
        }
    }
```

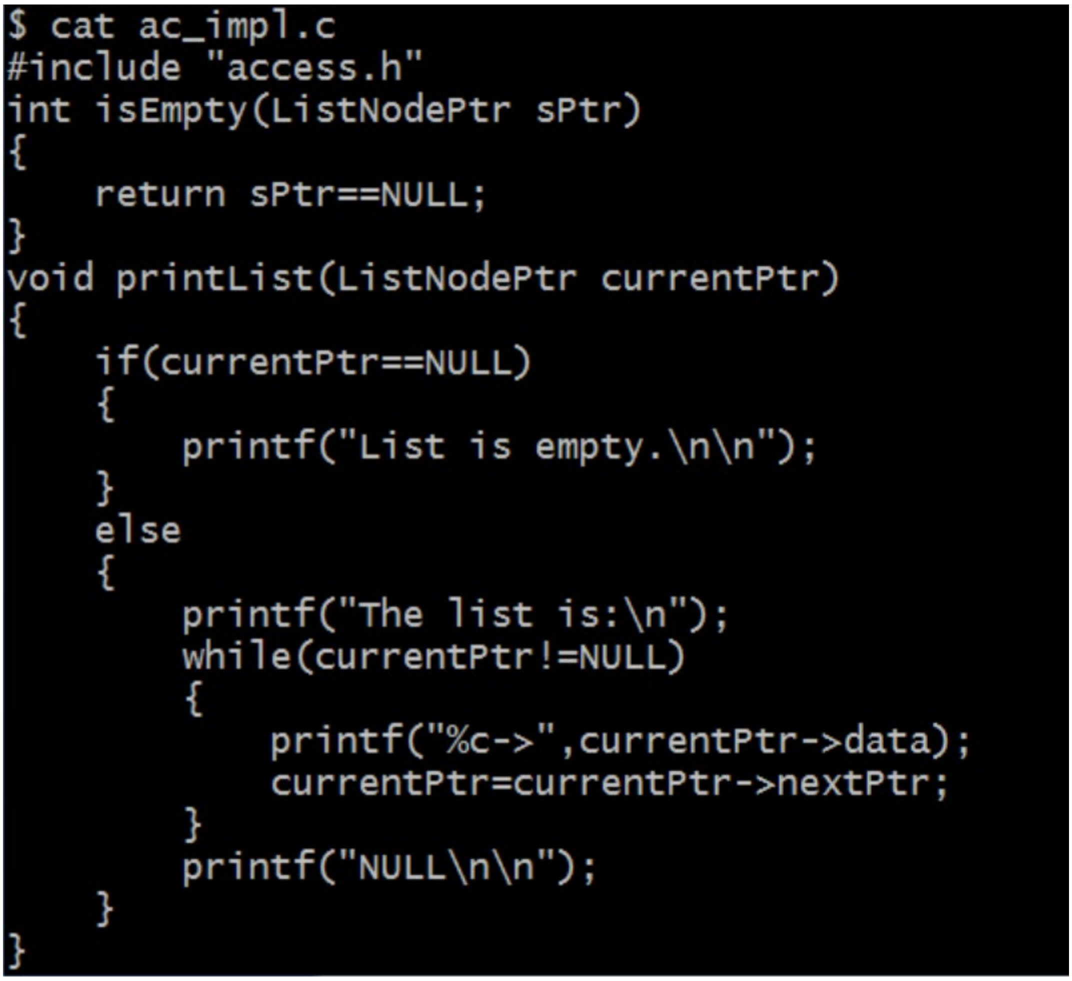


```
            currentPtr=currentPtr->nextPtr;
        }
        if(currentPtr!=NULL)
        {
            tempPtr=currentPtr;
            previousPtr->nextPtr=currentPtr->nextPtr;
            free(tempPtr);
            return value;
        }
    }
    return '\0';
}
```

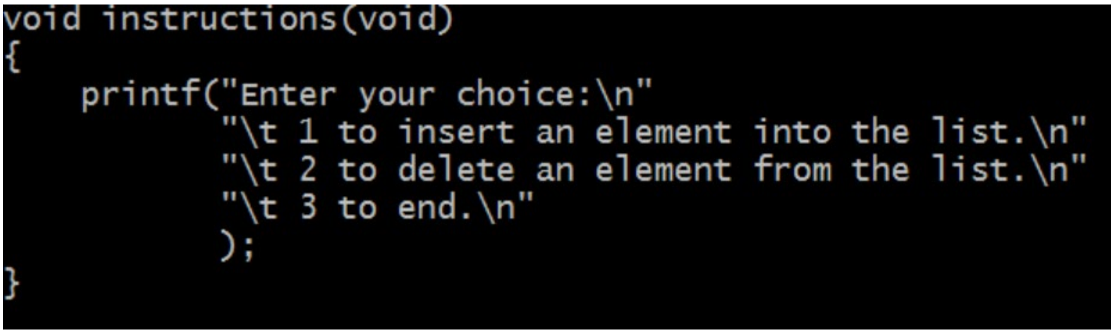
3. 链表访问模块接口文件及实现：



```
Administrator@CN-20160602JVJP ~/linkedlist
$ cat access.h
#ifndef ACCESS_H_INCLUDED
#define ACCESS_H_INCLUDED
#include "def.h"
int isEmpty(ListNodePtr);
void printList(ListNodePtr);
void instructions();

#endif // ACCESS_H_INCLUDED
```

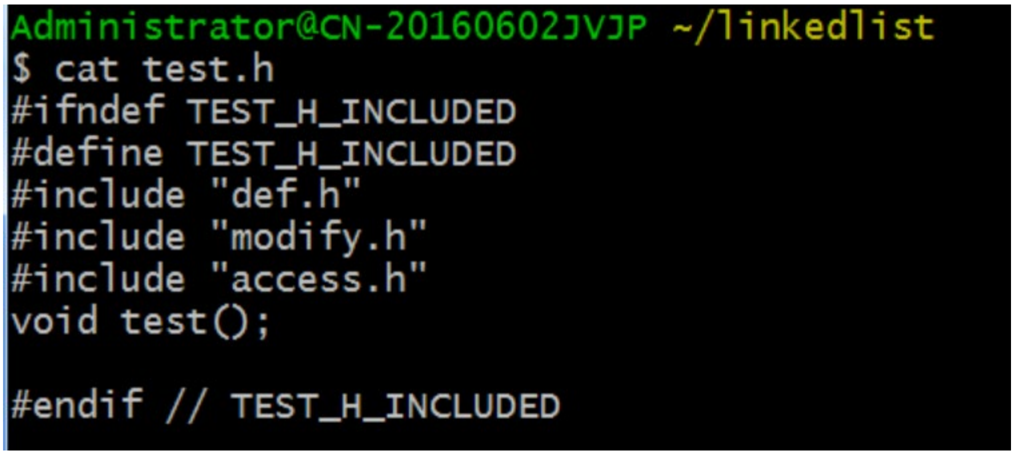


```
$ cat ac_impl.c
#include "access.h"
int isEmpty(ListNodePtr sPtr)
{
    return sPtr==NULL;
}
void printList(ListNodePtr currentPtr)
{
    if(currentPtr==NULL)
    {
        printf("List is empty.\n\n");
    }
    else
    {
        printf("The list is:\n");
        while(currentPtr!=NULL)
        {
            printf("%c->",currentPtr->data);
            currentPtr=currentPtr->nextPtr;
        }
        printf("NULL\n\n");
    }
}
```

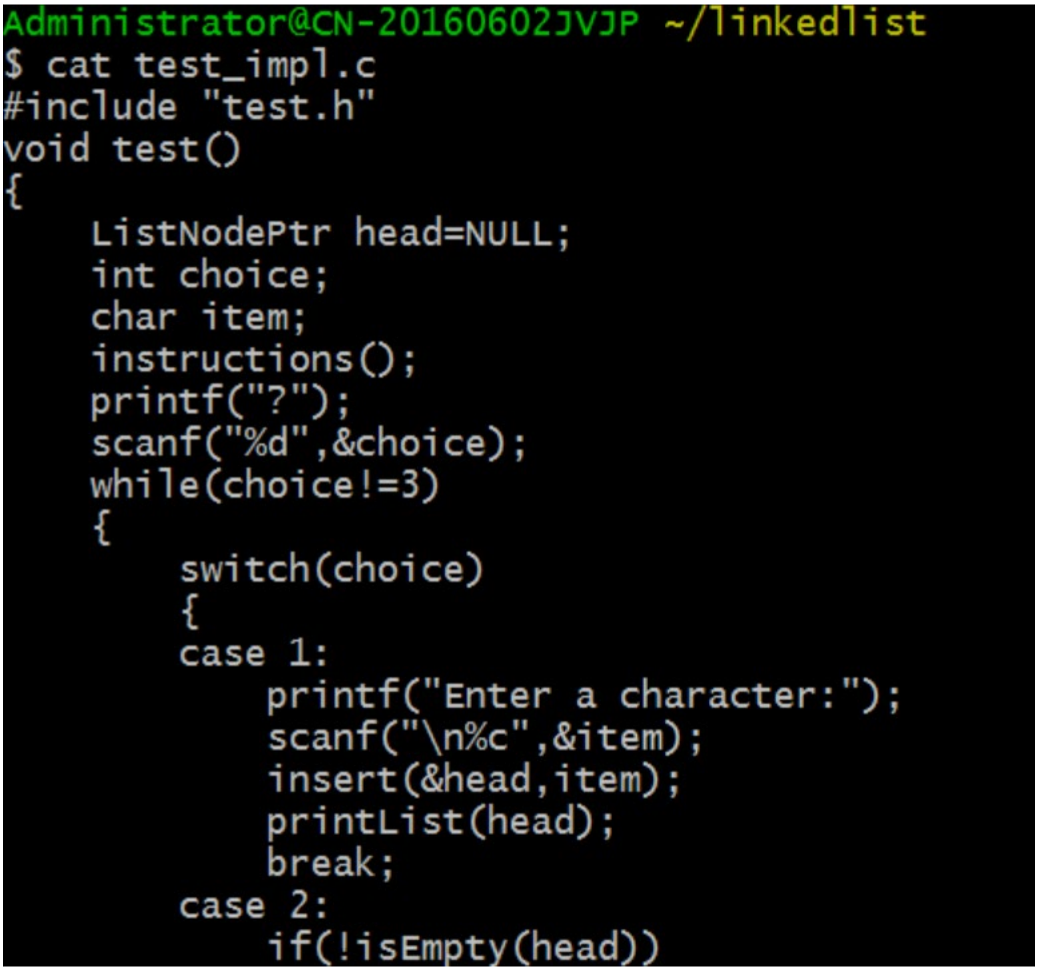


```
void instructions(void)
{
    printf("Enter your choice:\n"
           "\t 1 to insert an element into the list.\n"
           "\t 2 to delete an element from the list.\n"
           "\t 3 to end.\n"
           );
}
```

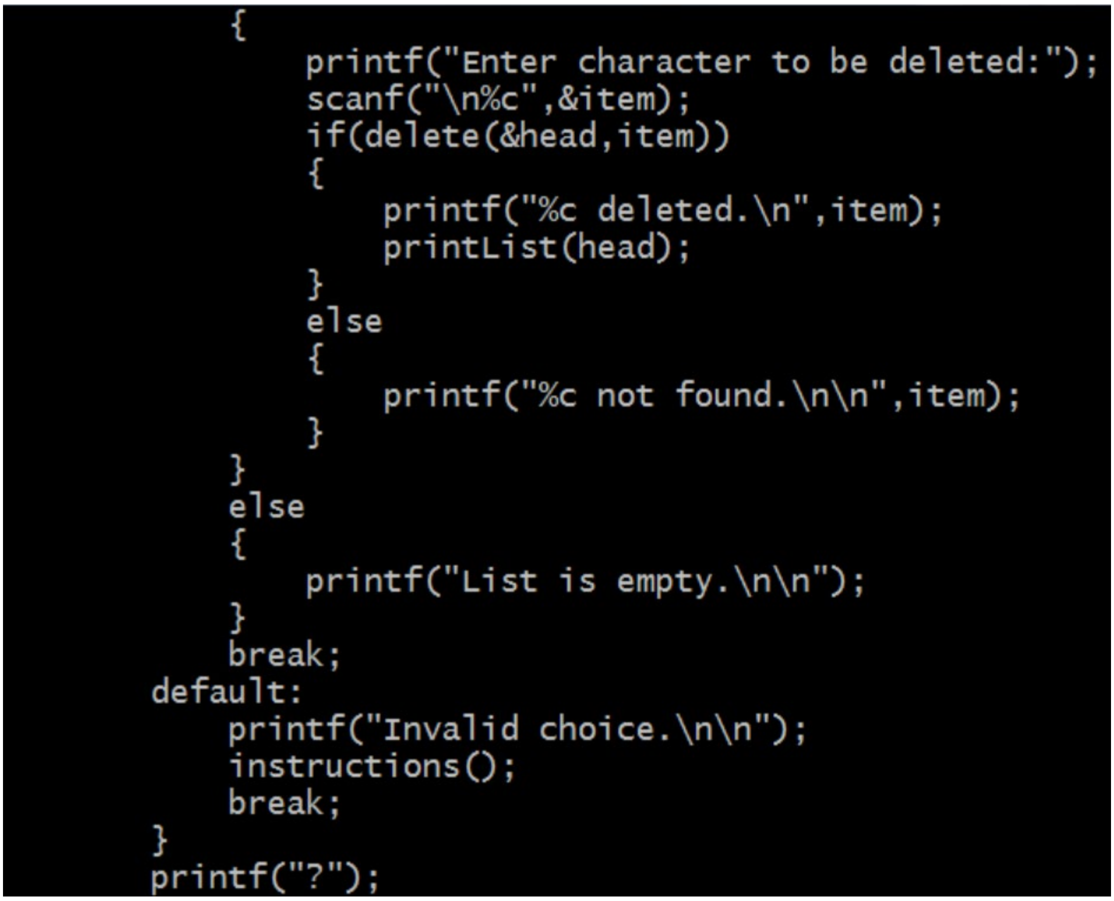
4.测试模块接口文件及实现



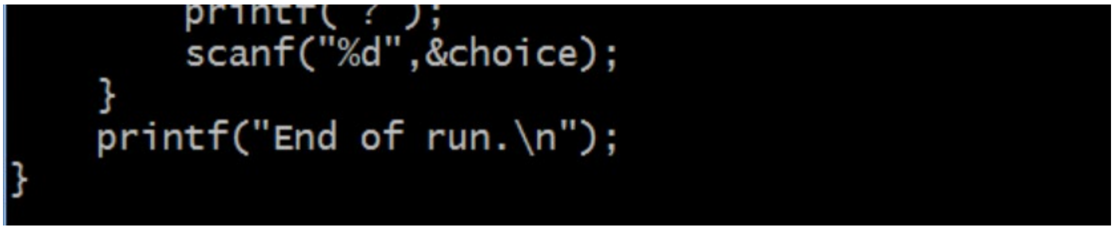
```
Administrator@CN-20160602JVJP ~/linkedlist
$ cat test.h
#ifndef TEST_H_INCLUDED
#define TEST_H_INCLUDED
#include "def.h"
#include "modify.h"
#include "access.h"
void test();

#endif // TEST_H_INCLUDED
```



```
Administrator@CN-20160602JVJP ~/linkedlist
$ cat test_impl.c
#include "test.h"
void test()
{
    ListNodePtr head=NULL;
    int choice;
    char item;
    instructions();
    printf("?");
    scanf("%d",&choice);
    while(choice!=3)
    {
        switch(choice)
        {
        case 1:
            printf("Enter a character:");
            scanf("\n%c",&item);
            insert(&head,item);
            printList(head);
            break;
        case 2:
            if(!isEmpty(head))
```

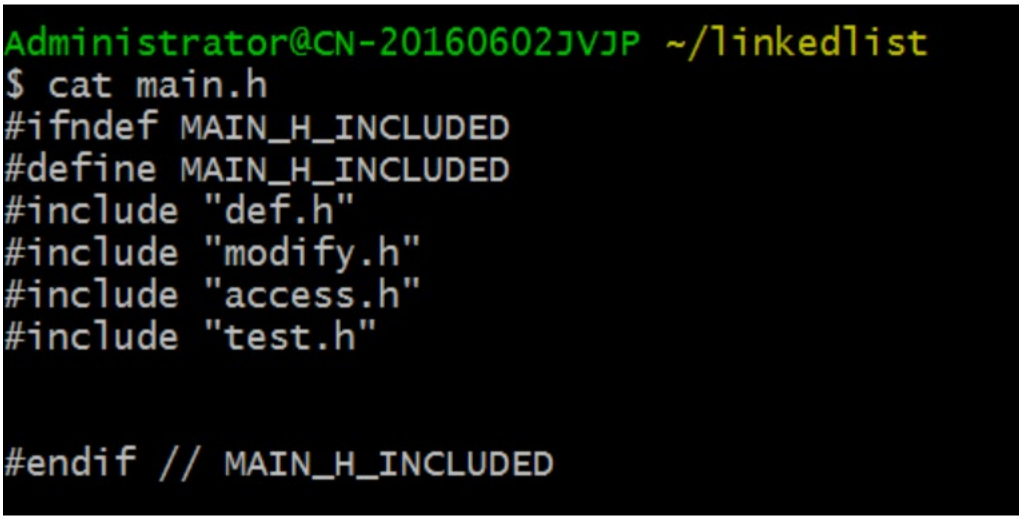


```
            {
                printf("Enter character to be deleted:");
                scanf("\n%c",&item);
                if(delete(&head,item))
                {
                    printf("%c deleted.\n",item);
                    printList(head);
                }
                else
                {
                    printf("%c not found.\n\n",item);
                }
            }
            else
            {
                printf("List is empty.\n\n");
            }
            break;
        default:
            printf("Invalid choice.\n\n");
            instructions();
            break;
        }
        printf("?");
```

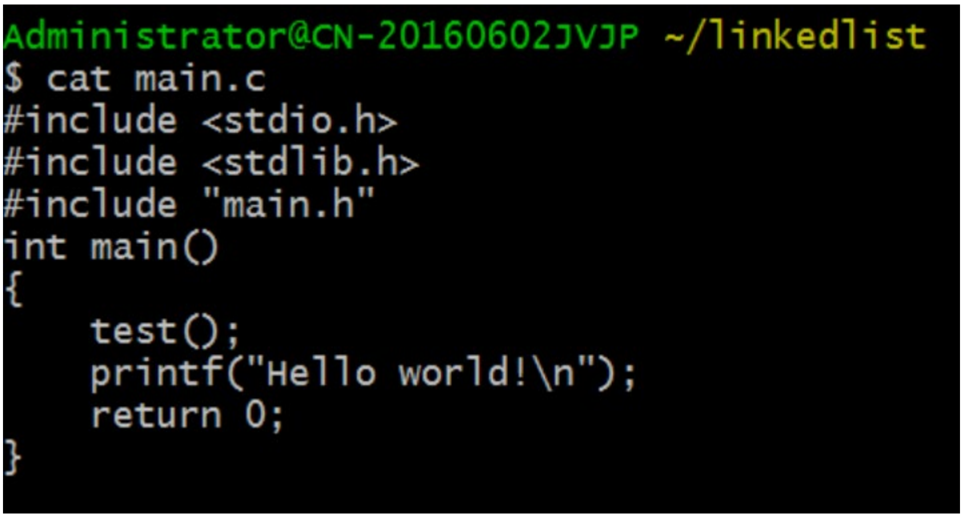


```
        printf("?");
        scanf("%d",&choice);
    }
    printf("End of run.\n");
}
```

5. 主函数模块接口文件及主函数

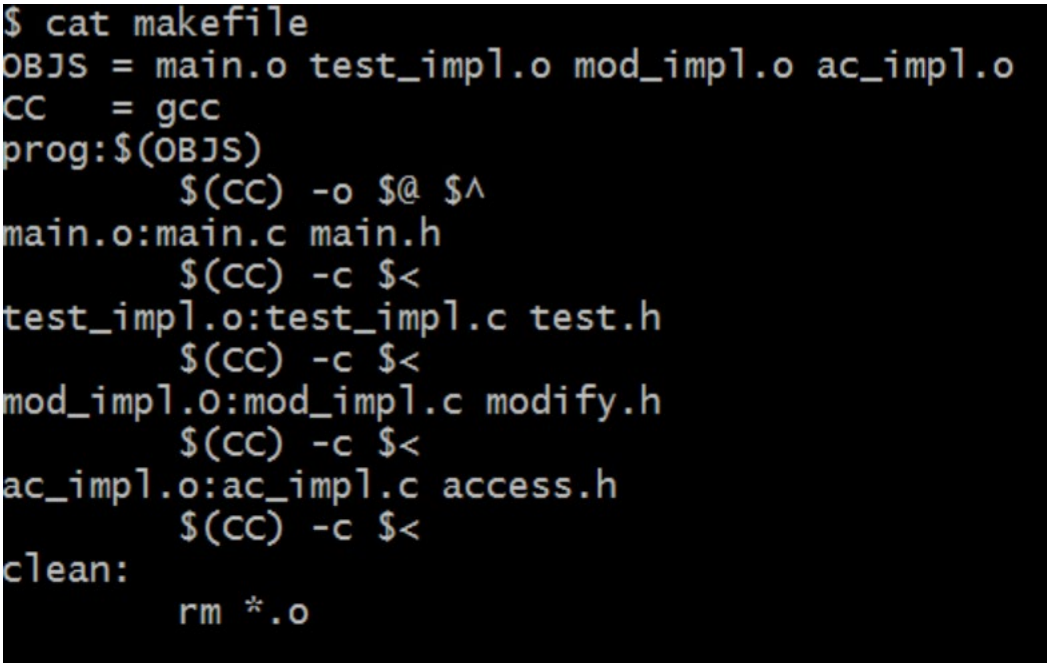


```
Administrator@CN-20160602JVJP ~/linkedlist
$ cat main.h
#ifndef MAIN_H_INCLUDED
#define MAIN_H_INCLUDED
#include "def.h"
#include "modify.h"
#include "access.h"
#include "test.h"


#endif // MAIN_H_INCLUDED
```



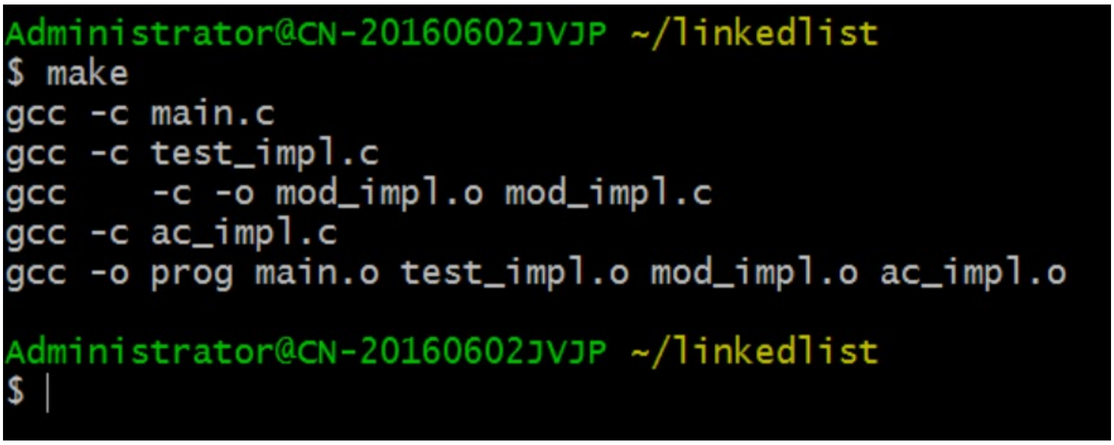
```
Administrator@CN-20160602JVJP ~/linkedlist
$ cat main.c
#include <stdio.h>
#include <stdlib.h>
#include "main.h"
int main()
{
    test();
    printf("Hello world!\n");
    return 0;
}
```

6.makefile 脚本程序



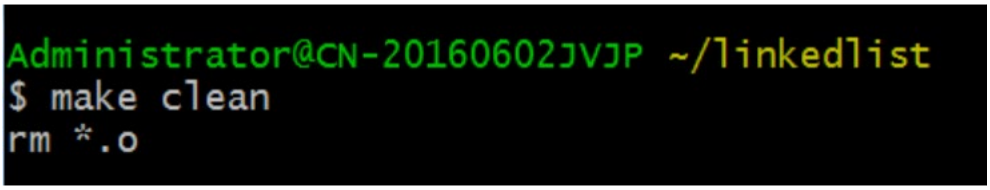
```
$ cat makefile
OBJS = main.o test_impl.o mod_impl.o ac_impl.o
CC   = gcc
prog:$(OBJS)
        $(CC) -o $@ $^
main.o:main.c main.h
        $(CC) -c $<
test_impl.o:test_impl.c test.h
        $(CC) -c $<
mod_impl.O:mod_impl.c modify.h
        $(CC) -c $<
ac_impl.o:ac_impl.c access.h
        $(CC) -c $<
clean:
        rm *.o
```

7.利用 make 命令编译程序



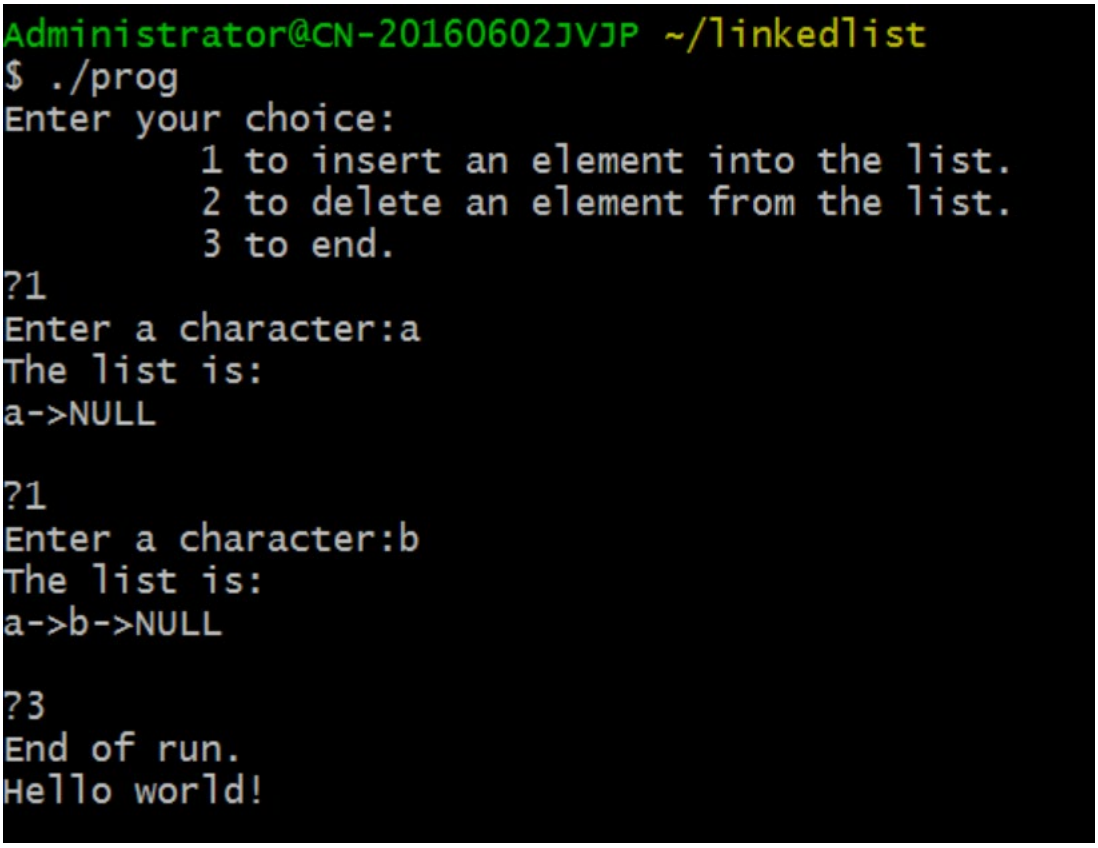
```
Administrator@CN-20160602JVJP ~/linkedlist
$ make
gcc -c main.c
gcc -c test_impl.c
gcc    -c -o mod_impl.o mod_impl.c
gcc -c ac_impl.c
gcc -o prog main.o test_impl.o mod_impl.o ac_impl.o

Administrator@CN-20160602JVJP ~/linkedlist
$ |
```

8.删除 C 目标文件



```
Administrator@CN-20160602JVJP ~/linkedlist
$ make clean
rm *.o
```

9.运行程序



```
Administrator@CN-20160602JVJP ~/linkedlist
$ ./prog
Enter your choice:
         1 to insert an element into the list.
         2 to delete an element from the list.
         3 to end.
?1
Enter a character:a
The list is:
a->NULL

?1
Enter a character:b
The list is:
a->b->NULL

?3
End of run.
Hello world!
```