《Linux应用程序开发实验报告》

题目: C语言

班级:\_\_ \_\_ \_\_17计科3班 \_

学号: 10417307 \_\_\_

姓名: \_\_薛冬冬\_ \_\_\_

东南大学成贤学院计算机系

2020年3 月17 日

**实验题目**

1. **实验目的**

1.熟悉Linux下C语言编译调试环境。

2.学会使用gcc编译C程序。

1. **实验内容**

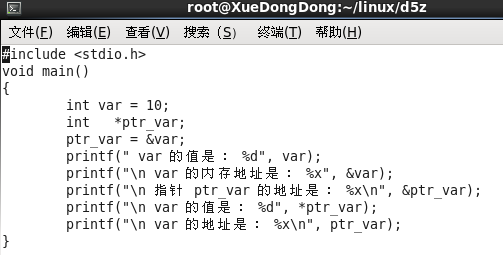
1. 在Linux下编译运行C语言程序，生成可执行文件。

1. **实验过程**

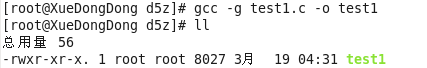
**1. 指针使用示例1**

**◆编译调试：**

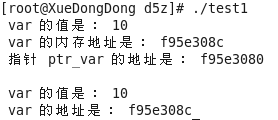
（1）在工作目录上新建文件test1.c，并用 vi 启动：vi test1.c。在 vi 中输入以上代码。在 vi 中保存并退出，使用命令“:wq”。



（2）用 gcc 编译：gcc -g test1.c -o test1。



（3）运行 test1，使用命令“./ test1”，结果正确，程序无误。



**◆编译调试后正确代码：**

#include <stdio.h>

void main()

{ int var = 10;

int \*ptr\_var;

ptr\_var = &var;

printf(" var 的值是： %d", var);

printf("\n var 的内存地址是： %x", &var);

printf("\n 指针 ptr\_var 的地址是： %x\n", &ptr\_var);

printf("\n var 的值是： %d", \*ptr\_var);

printf("\n var 的地址是： %x\n", ptr\_var);

}

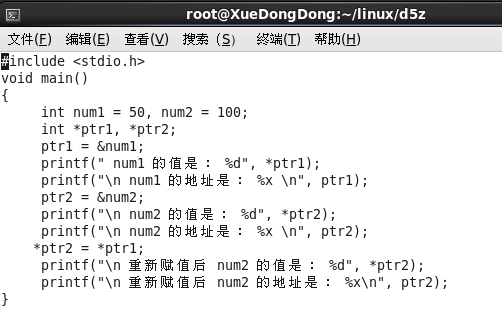
**◆代码完成功能、效果：**

1. 通过var直接获取var的值:10
2. 通过&var直接获取var的内存地址。
3. 通过&ptr\_var直接获取指针ptr\_var的地址。
4. 获取指针ptr\_var所指的存储单元var的值:10
5. 通过指针ptr\_var获取var的地址。

**2.指针使用示例2**

**◆编译调试：**

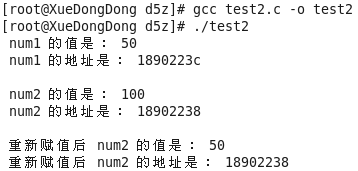
（1）在工作目录上新建文件test2.c，并用 vi 启动：vi test2.c。在 vi 中输入以上代码。在 vi 中保存并退出，使用命令“:wq”。



（2）用 gcc 编译：gcc -g test2.c -o test2。



（3）运行 test2，使用命令“./ test2”，结果正确，程序无误。



**◆编译调试后正确代码：**

#include <stdio.h>

void main()

{ int num1 = 50, num2 = 100;

int \*ptr1, \*ptr2;

ptr1 = &num1;

printf(" num1 的值是： %d", \*ptr1);

printf("\n num1 的地址是： %x \n", ptr1);

ptr2 = &num2;

printf("\n num2 的值是： %d", \*ptr2);

printf("\n num2 的地址是： %x \n", ptr2);

\*ptr2 = \*ptr1;

printf("\n 重新赋值后 num2 的值是： %d", \*ptr2);

printf("\n 重新赋值后 num2 的地址是： %x\n", ptr2);

}

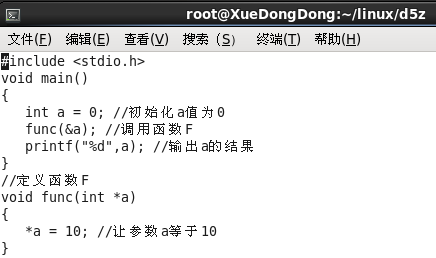
**◆代码完成功能、效果：**

1. 通过指针获取num1的值:50
2. 通过指针获取num1的地址:1890223c
3. 通过指针获取num2的值:100
4. 通过指针获取num2的地址:18902238
5. 将指针ptr1所指向的存储单元的值赋值给ptr2所指向的存储单元的值。
6. 通过指针获取重新赋值后num2的值:50
7. 通过指针获取重新赋值后num2的地址:18902238

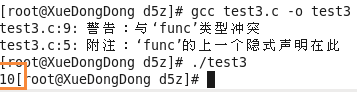
**3.引用调用**

**◆编译调试：**

（1）在工作目录上新建文件test3.c，并用 vi 启动：vi test3.c。在 vi 中输入以上代码。在 vi 中保存并退出，使用命令“:wq”。



（2）用 gcc 编译：gcc -g test3.c -o test3。运行 test3，使用命令“./ test”3，结果正确，程序无误。



**◆编译调试后正确代码：**

#include <stdio.h>

void main()

{ int a = 0; //初始化a值为0

func(&a); //调用函数F

printf("%d",a); //输出a的结果

}

//定义函数F

void func(int \*a)

{\*a = 10; //让参数a等于10

}

**◆代码完成功能、效果：**

（1）初始化a，调用函数func，然后输出a的结果；

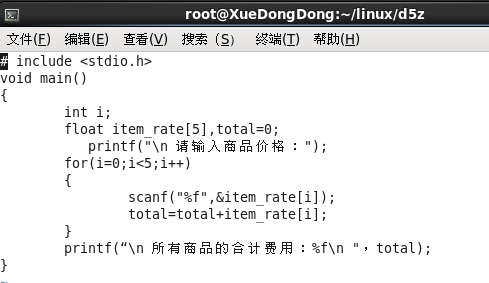
（2）函数func的功能是取出a对应存储单元地址，将10放入a所指向的存储单元中；

（3）由（1）（2）得知，a的值最终为10。

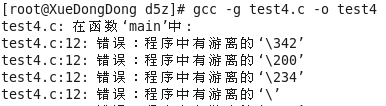
**4.一维数组示例**

**◆编译调试：**

（1）在工作目录上新建文件test4.c，并用 vi 启动：vi test4.c。在 vi 中输入以上代码。在 vi 中保存并退出，使用命令“:wq”。



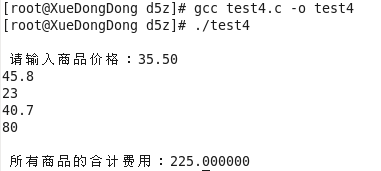
（2）用 gcc 编译：gcc -g test4.c -o test4。



（3）由上一步提示得知程序代码有误，使用vi重新编辑 test4.c,注意大小写和符号的中英化。

中文上引号改为英文上引号，中文“，”改为英文“,”。

（4）使用 gcc 重新编译：gcc -g test4.c -o test4。查看运行结果：./ test4



**◆编译调试后正确代码：**

# include <stdio.h>

void main()

{

int i;

float item\_rate[5],total=0;

printf("\n 请输入商品价格：");

for(i=0;i<5;i++)

{

scanf("%f",&item\_rate[i]);

total=total+item\_rate[i];

}

printf("\n 所有商品的合计费用：%f\n ",total);

}

**◆代码完成功能、效果：**

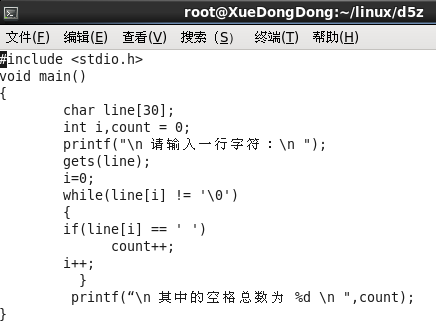
计算5个商品的总价。

1. 输入5个商品的价格，将其存放在一维数组中；
2. 通过for循环得到数组中5个元素的累加和即5个商品的总费用。
3. 35.5+45.8+23+40.7+80=225

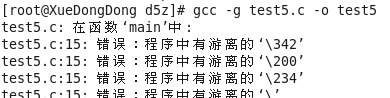
**5.字符串示例**

**◆编译调试：**

（1）在工作目录上新建文件test5.c，并用 vi 启动：vi test5.c。在 vi 中输入以上代码。在 vi 中保存并退出，使用命令“:wq”。



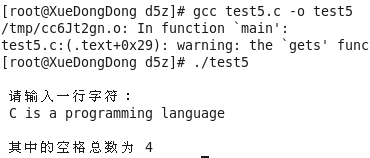
（2）用 gcc 编译：gcc -g test5.c -o test5。



（3）由上一步提示得知程序代码有误，使用vi重新编辑 test5.c,注意大小写和符号的中英化。

中文上引号改为英文上引号。

（4）使用 gcc 重新编译：gcc -g test5.c -o test5。查看运行结果：./ test5



**◆编译调试后正确代码：**

#include <stdio.h>

void main()

{

char line[30];

int i,count = 0;

printf("\n 请输入一行字符：\n ");

gets(line);

i=0;

while(line[i] != '\0')

{

if(line[i] == ' ')

count++;

i++;

}

printf("\n 其中的空格总数为 %d \n ",count);

}

**◆代码完成功能、效果：**

统计一行字符串中的空格数。

1. 通过键盘键入一行字符串；
2. 当字符串未结束时，统计字符串中空格的个数，最后输出在屏幕上。
3. C is a programming language中共4个空格，所以应该输出4。

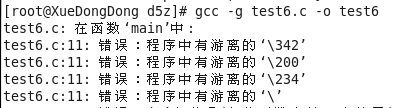
**6.字符指针数组示例**

**◆编译调试：**

（1）在工作目录上新建文件test6.c，并用 vi 启动：vi test6.c。在 vi 中输入以上代码。在 vi 中保存并退出，使用命令“:wq”。



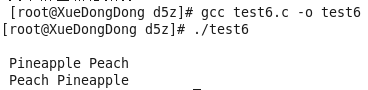
（2）用 gcc 编译：gcc -g test6.c -o test6。



（3）由上一步提示得知程序代码有误，使用vi重新编辑 test6.c,注意大小写和符号的中英化。

中文上引号改为英文上引号。

（4）使用 gcc 重新编译：gcc -g test6.c -o test6。查看运行结果：./ test6。



**◆编译调试后正确代码：**

#include <stdio.h>

void main()

{

char \*names[] = { "Apple", "Banana","Pineapple",

"Peach",

"Strawberry",

"Grapes" };

char \*temp;

printf("\n %s %s",names[2],names[3]);

temp = names[2];

names[2] = names[3];

names[3] = temp;

printf("\n %s %s",names[2],names[3]);

printf("\n");

}

**◆代码完成功能、效果：**

输出字符指针数组中names[2]和names[3]的值。

1. 向字符指针数组中赋5个初值，分别是“Apple”、“Banana”、“Pineapple”、“Peach”、“Strawberry”、“Grapes”；
2. 首先输出names[2]和names[3]分别对应着Pineapple和Peach；
3. 然后names[2]和names[3]通过中间变量temp交换地址，使得交换后输出为Peach和Pineapple。

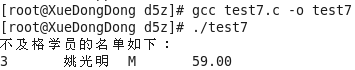
**7.使用结构示例**

**◆编译调试：**

（1）在工作目录上新建文件test7.c，并用 vi 启动：vi test7.c。在 vi 中输入以上代码。在 vi 中保存并退出，使用命令“:wq”。



（2）用 gcc 编译：gcc -g test7.c -o test7。运行 test7，使用命令“./ test7”，结果正确，程序无误。



**◆编译调试后正确代码：**

#include<stdio.h>

struct student

{

int num;//学号

char \*name;//姓名

char sex;//性别

float score;//成绩

};

void main()

{

static struct student stu1={1,"李亚鹏",'M',61};

static struct student stu2={2,"周晶晶",'F',92.5};

static struct student stu3={3,"姚光明",'M',59};

printf("不及格学员的名单如下：\n");

if(stu1.score<60) printf("%d\t%s\t%c\t%5.2f\n",stu1.num,stu1.name,stu1.sex,stu1.score);

if(stu2.score<60) printf("%d\t%s\t%c\t%5.2f\n",stu2.num,stu2.name,stu2.sex,stu2.score);

if(stu3.score<60) printf("%d\t%s\t%c\t%5.2f\n",stu3.num,stu3.name,stu3.sex,stu3.score);

if(stu1.score>=60 && stu2.score>=60 && stu3.score>=60)

printf("没有不及格的学员。\n");

}

**◆代码完成功能、效果：**

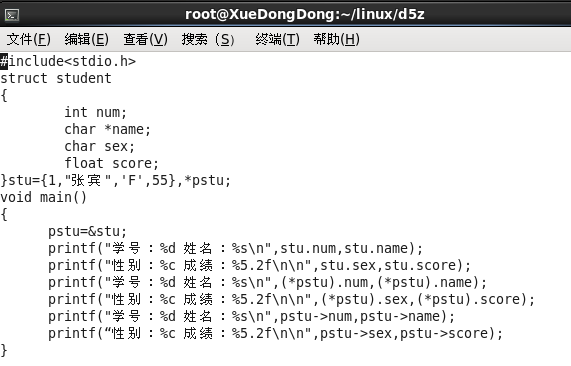
输出成绩不及格学生的信息。

1. 判断结构变量成员score是否大于60即判断学生成绩是否及格；
2. 若学生成绩不及格，将其学号、姓名、性别、成绩信息输出在屏幕上；
3. 因姚光明同学成绩59<60，所以将姚光明同学所有信息都输出。

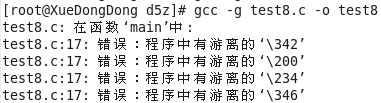
**8.结构指针变量示例**

**◆编译调试：**

（1）在工作目录上新建文件test8.c，并用 vi 启动：vi test8.c。在 vi 中输入以上代码。在 vi 中保存并退出，使用命令“:wq”。



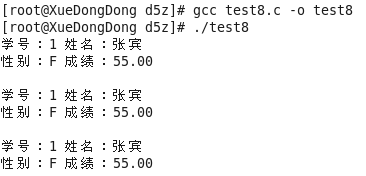
（2）用 gcc 编译：gcc -g test8.c -o test8。



（3）由上一步提示得知程序代码有误，使用vi重新编辑 test8.c,注意大小写和符号的中英化。

中文上引号改为英文上引号。

（4）使用 gcc 重新编译：gcc -g test8.c -o test8。查看运行结果：./ test8。



**◆编译调试后正确代码：**

#include<stdio.h>

struct student

{

int num;

char \*name;

char sex;

float score;

}stu={1,"张宾",'F',55},\*pstu;

void main()

{

pstu=&stu;

printf("学号：%d 姓名：%s\n",stu.num,stu.name);

printf("性别：%c 成绩：%5.2f\n\n",stu.sex,stu.score);

printf("学号：%d 姓名：%s\n",(\*pstu).num,(\*pstu).name);

printf("性别：%c 成绩：%5.2f\n\n",(\*pstu).sex,(\*pstu).score);

printf("学号：%d 姓名：%s\n",pstu->num,pstu->name);

printf("性别：%c 成绩：%5.2f\n\n",pstu->sex,pstu->score);

}

**◆代码完成功能、效果：**

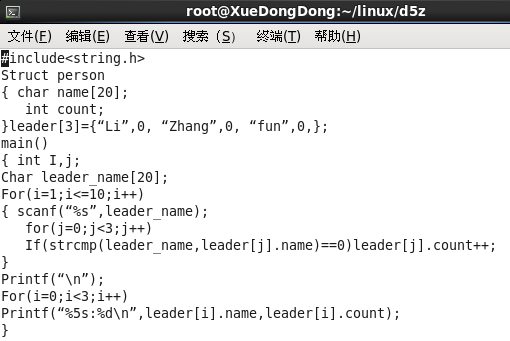
通过3种方法获取某个学生信息，并将其显示在屏幕上。

1. 使用表示结构变量的一般格式，获取结构成员变量num、name、sex和score的值并输出；
2. 使用结构指针访问结构变量成员：（\*结构指针变量）.成员名、结构指针变量->成员名，输出获取的值；
3. 3种方法获取结构变量成员的值相等，因此将张宾同学的学号、姓名、性别和成绩分别输出了3遍。

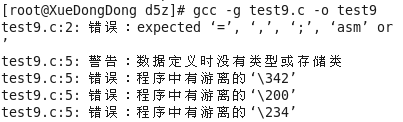
**9.结构体数组应用举例**

**◆编译调试：**

（1）在工作目录上新建文件test9.c，并用 vi 启动：vi test9.c。在 vi 中输入以上代码。在 vi 中保存并退出，使用命令“:wq”。



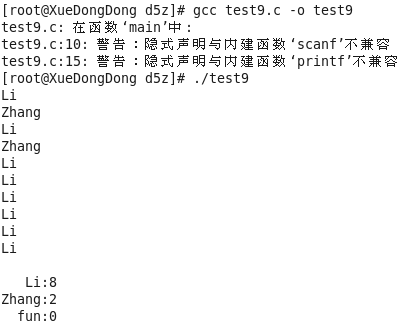
（2）用 gcc 编译：gcc -g test9.c -o test9。



（3）由上一步提示得知程序代码有误，使用vi重新编辑 test9.c,注意大小写和符号的中英化。

中文上引号改为英文上引号，英文大小写更改。

（4）使用 gcc 重新编译：gcc -g test9.c -o test9。查看运行结果：./ test9。



**◆编译调试后正确代码：**

#include<string.h>

struct person

{ char name[20];

int count;

}leader[3]={"Li",0, "Zhang",0, "fun",0,};

main()

{ int i,j;

char leader\_name[20];

for(i=1;i<=10;i++)

{ scanf("%s",leader\_name);

for(j=0;j<3;j++)

if(strcmp(leader\_name,leader[j].name)==0)leader[j].count++;

}

printf("\n");

for(i=0;i<3;i++)

printf("%5s:%d\n",leader[i].name,leader[i].count);

}

**◆代码完成功能、效果：**

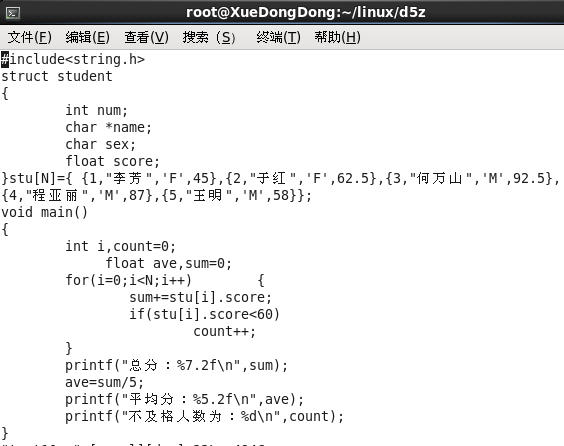
给3个候选人投票，每个投票人有10票，投完票后显示3个候选人票数。

1. 候选人得票的统计程序，给予每个人10次投票机会，输入10次要投票人的姓名，即可完成10次投票，最后显示每个人的得票结果。
2. 利用结构体数组实现for循环遍历，实现每个人得票数的统计和每个人得票数的输出。
3. 10次中我给Li投了8票，给Zhang 投了两票，最后输出结果应为Li有8票，Zhang有2票，fun0票。

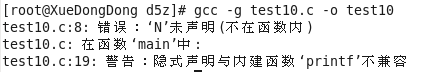
**10.结构指针变量示例**

**◆编译调试：**

（1）在工作目录上新建文件test10.c，并用 vi 启动：vi test10.c。在 vi 中输入以上代码。在 vi 中保存并退出，使用命令“:wq”。



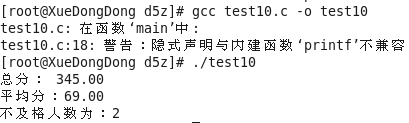
（2）用 gcc 编译：gcc -g test10.c -o test10。



（3）由上一步提示得知程序代码有误，使用vi重新编辑 test10.c,注意大小写和符号的中英化。

将程序代码中“N”改为5。

（4）使用 gcc 重新编译：gcc -g test10.c -o test10。查看运行结果：./ test10。



**◆编译调试后正确代码：**

#include<string.h>

struct student

{

int num;

char \*name;

char sex;

float score;

}stu[5]={ {1,"李芳",'F',45},{2,"于红",'F',62.5},{3,"何万山",'M',92.5},

{4,"程亚丽",'M',87},{5,"王明",'M',58}};

void main()

{

int i,count=0;

float ave,sum=0;

for(i=0;i<5;i++) {

sum+=stu[i].score;

if(stu[i].score<60)

count++;

}

printf("总分：%7.2f\n",sum)

ave=sum/5;

printf("平均分：%5.2f\n",ave);

printf("不及格人数为：%d\n",count);

}

**◆代码完成功能、效果：**

求出学员的总成绩和平均成绩，并统计不及格人数。

1. 总分=45+62.5+92.5+87+58=345

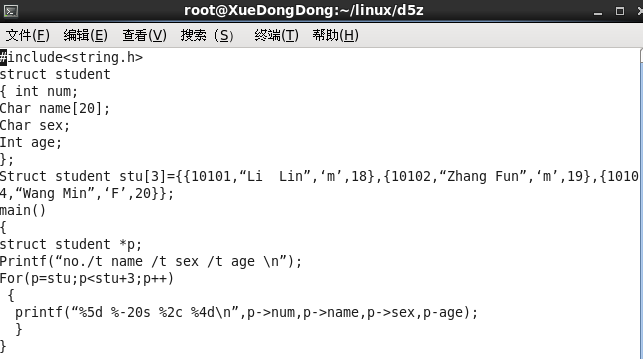
平均分=总分/5=69

不及格人数2个（45，58）

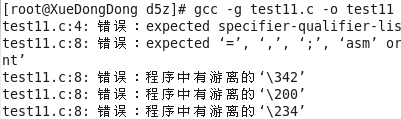
**11. 指向结构体数组的指针**

**◆编译调试：**

（1）在工作目录上新建文件test11.c，并用 vi 启动：vi test11.c。在 vi 中输入以上代码。在 vi 中保存并退出，使用命令“:wq”。



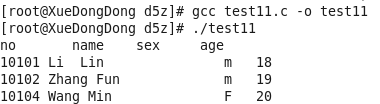
（2）用 gcc 编译：gcc -g test11.c -o test11。



（3）由上一步提示得知程序代码有误，使用vi重新编辑 test11.c,注意大小写和符号的中英化。

中文引号改为英文引号，英文大小写更改，补上遗漏的“->”，“/t”改为“\t”。

（4）使用 gcc 重新编译：gcc -g test11.c -o test11。查看运行结果：./ test11。



**◆编译调试后正确代码：**

#include<string.h>

struct student

{ int num;

char name[20];

char sex;

int age;

};

struct student stu[3]={{10101,"Li Lin",'m',18},

{10102,"Zhang Fun",'m',19},{10104,"Wang Min",'F',20}};

main()

{

struct student \*p;

printf("no.\t name \t sex \t age \n");

for(p=stu;p<stu+3;p++)

{

printf("%5d %-20s %2c %4d\n",p->num,p->name,p->sex,p->age);

}

}

**◆代码完成功能、效果：**

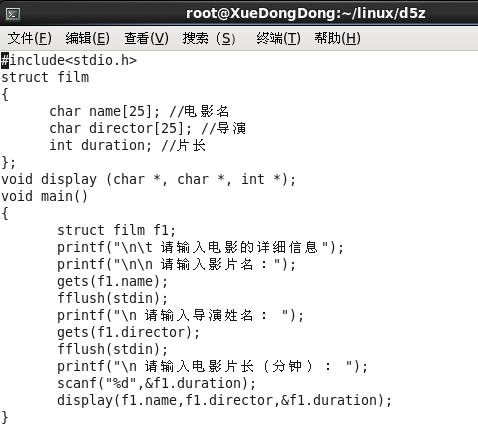
输出学生的学号、姓名、性别、年龄信息。

1. 通过指向结构体数组的指针和for循环，输出结构体数组成员的结构变量成员即所有学生的信息；
2. 屏幕上应该显示3个学生的所有信息。

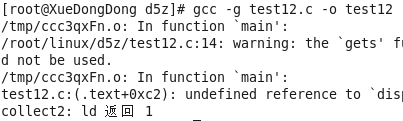
**12.结构成员作为函数参数**

**◆编译调试：**

（1）在工作目录上新建文件test12.c，并用 vi 启动：vi test12.c。在 vi 中输入以上代码。在 vi 中保存并退出，使用命令“:wq”。



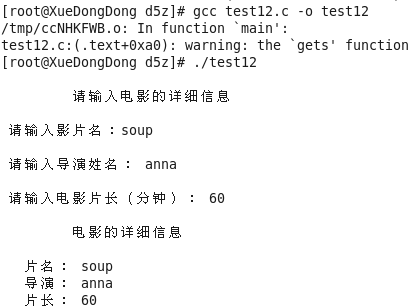
（2）用 gcc 编译：gcc -g test12.c -o test12。



（3）由上一步提示得知程序代码有误，使用vi重新编辑 test12.c,注意大小写和符号的中英化。

添加函数display方法的内容，使其能够实现。

（4）使用 gcc 重新编译：gcc -g test12.c -o test12。查看运行结果：./ test12。



**◆编译调试后正确代码：**

#include<stdio.h>

struct film

{

char name[25]; //电影名

char director[25]; //导演

int duration; //片长

};

void display(char \*n, char \*d, int \*m)

{

printf("\n\t 电影的详细信息\n");

printf("\n 片名： %s",n);

printf("\n 导演： %s",d);

printf("\n 片长： %d\n",\*m);

}

void main()

{

struct film f1;

printf("\n\t 请输入电影的详细信息");

printf("\n\n 请输入影片名：");

gets(f1.name);

fflush(stdin);

printf("\n 请输入导演姓名： ");

gets(f1.director);

fflush(stdin);

printf("\n 请输入电影片长（分钟）： ");

scanf("%d",&f1.duration);

display(f1.name,f1.director,&f1.duration);

}

**◆代码完成功能、效果：**

输入电影的详细信息，并将其输出。

1. 将结构成员作为函数display的参数，并将其输出；
2. 通过gets获取键盘输入的信息，通过display方法将其输出在屏幕。
3. **实验体会**

本次实验，我复习了前几次学习的在linux环境下进行C语言编译和调试的方法，还重点学习了C语言的一些基本知识，其中最重要的是指针方面的知识。

我在linux环境中使用gcc编译C语言程序时，出现了好几次编译不成功，然后我通过gdb进行调试发现，直接复制的老师PPT中的代码有几个常见的问题，例如：字母大小写混乱、中英文标点符号有误、符号的遗漏等问题，然后我就先通过vi修改检查.c文件中常见错误，确认代码拼写无误后再进行编译，若还有问题再通过gdb调试寻找。因为有C语言的知识基础，所以整个实验完成的还是很快的。

本次实验，老师在视频中将C语言PPT从头到尾都详细且仔细地讲解了一遍，听完后使我对C语言知识理解地更加透彻了，尤其指针方面，老师讲的很清晰易懂。