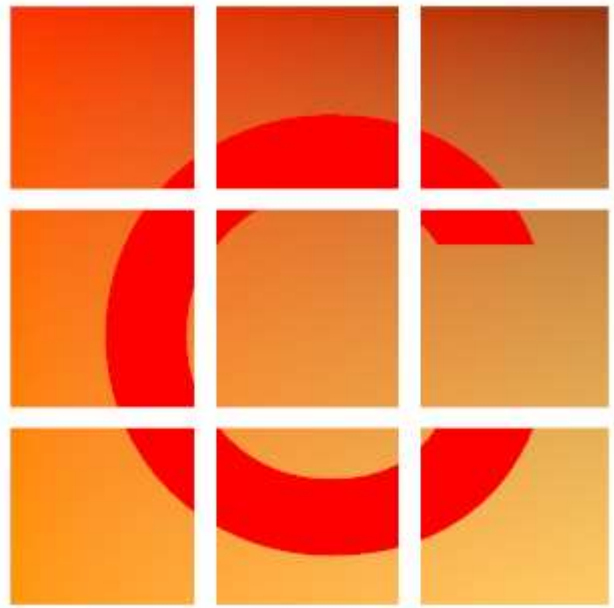
计算机与程序设计基础（C）
实验课堂

第四讲 指针1



任务一 再理解“地址”

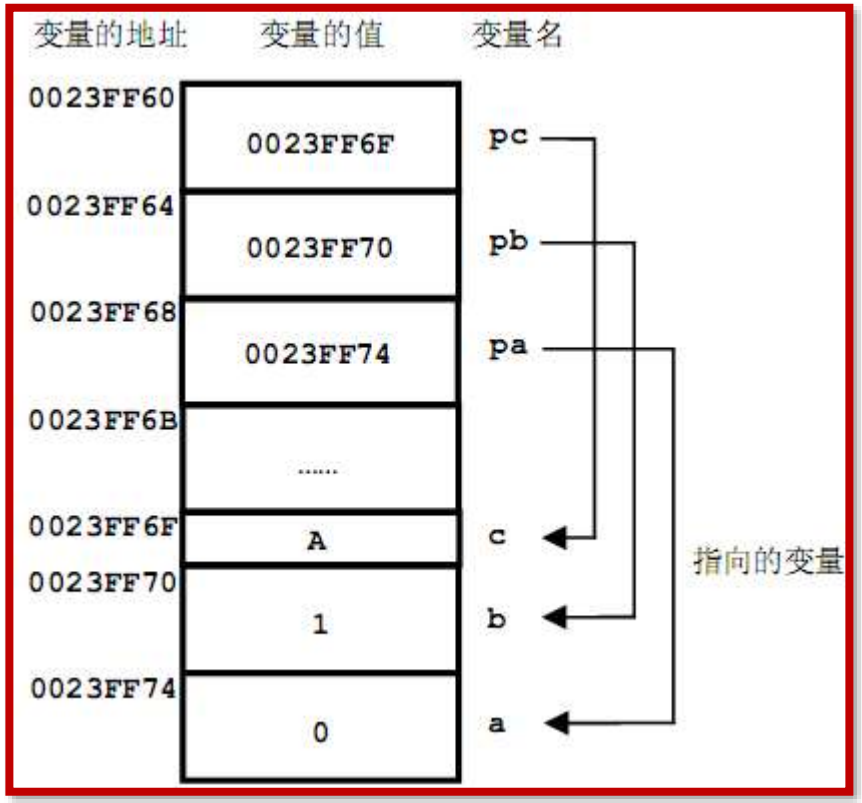
```c
#include<stdio.h>
void main()
{
    int martrix[2][2]={0};
    for(int i=0;i<2;i++)
    {
        for(int j=0;j<2;j++)
        {
            printf("%p,",martrix+2*i+j); //matrtrix视为一维数组的地址，加一就是偏移一行
            printf("%p,",martrix[0]+2*i+j);
            printf("%p,",&martrix[0][0]+2*i+j);
            printf("%p\n",&martrix[i][j]);
        }
        printf("\n");
    }
}
```

```
000000000061FE00,000000000061FE00,000000000061FE00,000000000061FE00
000000000061FE08,000000000061FE04,000000000061FE04,000000000061FE04

000000000061FE10,000000000061FE08,000000000061FE08,000000000061FE08
000000000061FE18,000000000061FE0C,000000000061FE0C,000000000061FE0C
```

任务二 **指针概念**

```c
#include <stdio.h>
int main()
{
    int  a = 0, b = 1;
    char c = 'A';
    int *pa, *pb;          /* 定义指针变量pa和pb */
    char *pc;              /* 定义指针变量pc */
    pa = &a;               /* 初始化指针变量pa使其指向a */
    pb = &b;               /* 初始化指针变量pb使其指向b */
    pc = &c;               /* 初始化指针变量pc使其指向c */
    printf("a is %d, &a is %p, pa is %p, &pa is %p\n", a, &a, pa, &pa);
    printf("b is %d, &b is %p, pb is %p, &pb is %p\n", b, &b, pb, &pb);
    printf("c is %c, &c is %p, pc is %p, &pc is %p\n", c, &c, pc, &pc);
    return 0;
}
```

```
D:\test-c\thefirstproject\bin\Debug\thefirstproject.exe
a is 0, &a is 000000000061FE1C, pa is 000000000061FE1C, &pa is 000000000061FE08
b is 1, &b is 000000000061FE18, pb is 000000000061FE18, &pb is 000000000061FE00
c is A, &c is 000000000061FE17, pc is 000000000061FE17, &pc is 000000000061FDF8

Process returned 0 (0x0)   execution time : 0.377 s
Press any key to continue.
```



任务三 **交换数据**

- 指针变量指向对象交换

```c
#include <stdio.h>
int main()
{
    int  *p1,*p2, a,b;
    int tmp;
    printf("please input tow numbers:");
    scanf("%d,%d",&a,&b);
    p1=&a;p2=&b;
    printf("p1指向(存放)的地址=%p, p2指向(存放)的地址=%p\n", p1, p2);
    printf("&p1=%p, *p1=%d, &p2=%p,*p2=%d\n", &p1, *p1, &p2, *p2);
    if(a<b)
    {
            tmp=*p1;*p1=*p2;*p2=tmp;
    }
    printf("a=%d,b=%d\n",a,b);
    printf("*p1=%d,*p2=%d\n",*p1,*p2);
    printf("p1指向的地址=%p, p2指向的地址=%p\n", p1, p2);
    printf("&p1=%p, &p2=%p\n", &p1, &p2);
    return 0;
}
```

please input tow numbers:**10,12**
p1指向(存放)的地址=000000000061FE04, p2指向(存放)的地址=000000000061FE00
&p1=000000000061FE10, ***p1=10**,
&p2=000000000061FE08,***p2=12**
a=12,b=10
*p1=12,*p2=10
p1指向的地址=000000000061FE04, p2指向的地址=000000000061FE00
&p1=000000000061FE10, &p2=000000000061FE08

任务三 **交换数据**

- 指针变量自身交换

```c
#include <stdio.h>
int main()
{
    int  *p1,*p2, *p, a,b;
    printf("please input tow numbers:");
    scanf("%d,%d",&a,&b);
    p1=&a; p2=&b;
    printf("p1指向(存放)的地址=%p, p2指向(存放)的地址=%p\n", p1, p2);
    printf("&p1=%p, *p1=%d, &p2=%p, *p2=%d\n", &p1, *p1, &p2,*p2);
    if(a<b)
    {
        p=p1; p1=p2; p2=p;
    }
    printf("a=%d,b=%d\n",a,b);
    printf("*p1=%d,*p2=%d\n",*p1,*p2);
    printf("p1指向的地址=%p, p2指向的地址=%p\n", p1, p2);
    printf("&p1=%p, &p2=%p\n", &p1, &p2);
    return 0;
}
```

please input tow numbers:**10,12**
p1指向(存放)的地址=000000000061FE04, p2指向(存放)的地址=000000000061FE00
&p1=000000000061FE10, ***p1=10**,
&p2=000000000061FE08, ***p2=12**
a=10,b=12
***p1=12,*p2=10**
p1指向的地址=000000000061FE00, p2指向的地址=000000000061FE04
&p1=000000000061FE10,
&p2=000000000061FE08

任务三 **交换数据（用函数）**·指针变量指向对象交换之函数版

```c
#include <stdio.h>
int main()
{
    void swap(int * p1, int * p2);
    int  *p1,*p2, a,b;
    printf("please input tow numbers:");
    scanf("%d,%d",&a,&b);
    p1=&a; p2=&b;
    printf("p1指向(存放)的地址=%p, p2指向(存放)的地址=%p\n", p1, p2);
    printf("&p1=%p, *p1=%d, &p2=%p, *p2=%d\n", &p1, *p1, &p2, *p2);
    if (a<b) swap(p1, p2);
    printf("a=%d,b=%d\n",a,b);
    printf("*p1=%d,*p2=%d\n",*p1,*p2);
    printf("p1指向的地址=%p, p2指向的地址=%p\n", p1, p2);
    printf("&p1=%p, &p2=%p\n", &p1, &p2);
    return 0;
}

void swap(int * p1, int * p2)
{
    int tmp;
    tmp=*p1;
    *p1=*p2;
    *p2=tmp;
}
```

please input tow numbers:**10,12**
p1指向(存放)的地址=000000000061FE0C, p2指向(存放)的地址=000000000061FE08
&p1=000000000061FE18,
***p1=10**,
&p2=000000000061FE10,
***p2=12**
a=12,b=10
***p1=12,*p2=10**
p1指向的地址=000000000061FE0C, p2指向的地址=000000000061FE08
&p1=000000000061FE18,
&p2=000000000061FE10

任务三 **交换数据（用函数）**·指针变量自身交换之函数版——错误!!!

```c
#include <stdio.h>
int main()
{
    void swap(int * p1, int * p2);
    int  *p1,*p2, a,b;
    printf("please input tow numbers:");
    scanf("%d,%d",&a,&b);
    p1=&a; p2=&b;
    printf("p1指向(存放)的地址=%p, p2指向(存放)的地址=%p\n", p1, p2);
    printf("&p1=%p, *p1=%d, &p2=%p, *p2=%d\n", &p1,*p1, &p2, *p2);
    if (a<b) swap(p1, p2);
    printf("a=%d,b=%d\n",a,b);
    printf("*p1=%d,*p2=%d\n",*p1,*p2);
    printf("p1指向的地址=%p, p2指向的地址=%p\n", p1, p2);
    printf("&p1=%p, &p2=%p\n", &p1, &p2);
    return 0;
}

void swap(int * p1, int * p2)
{
    int *tmp;
    tmp=p1;
    p1=p2;
    p2=tmp;
}
```

please input tow numbers:**10,12**
p1指向(存放)的地址=000000000061FE0C, p2指向(存放)的地址=000000000061FE08
&p1=000000000061FE18, ***p1=10**,
&p2=000000000061FE10, ***p2=12**
a=10,b=12
***p1=10,*p2=12**
p1指向的地址=000000000061FE0C,
p2指向的地址=000000000061FE08
&p1=000000000061FE18,
&p2=000000000061FE10

任务四 **想通过函数返回更多值?**

- 获取最高分和学号

```c
#include  <stdio.h>
#define  N  10
void FindMax(int score[],long num[],int n,int *pMaxScore,long *pMaxNum);
int main()
{
    int  score[N], maxScore;
    int    n, i;
    long   num[N], maxNum;
    printf("How many students (<=10)?");
    scanf("%d", &n);                                  /* 从键盘输入学生人数n */
    printf("Input student's ID and score:\n");
    for (i=0; i<n; i++)
    {
        scanf("%ld%d", &num[i], &score[i]);   /* 字母d前为字母l */
    }
    FindMax(score, num,n, &maxScore, &maxNum);/* 按地址调用函数 */
    printf("maxScore = %d, maxNum = %ld\n", maxScore, maxNum);
    return 0;
}
/* 函数功能：计算最高分及其相应学生的学号 */
void FindMax(int score[],long num[],int n,int *pMaxScore,long *pMaxNum)
{
    int    i;
    *pMaxScore = score[0];              /* 假设score[0]为当前最高分 */
    *pMaxNum = num[0];                  /* 记录score[0]的学号num[0] */
    for (i=1; i<n; i++)                 /* 对所有score[i]进行比较 */
    {
        if (score[i] > *pMaxScore)  /* 如果score[i]高于当前最高分 */
        {
            *pMaxScore = score[i];  /* 用score[i]修改当前最高分 */
            *pMaxNum = num[i];      /* 记录当前最高分学生的学号num[i] */
        }
    }
}
```

```
D:\test-c\thefirstproject\bin\Debug\thefirstproject.exe
Input student's ID and score:
123 50
124 51
125 90
126 91
127 88
maxScore = 91, maxNum = 126

Process returned 0 (0x0)   execution time : 49.238 s
Press any key to continue.
```

任务五 **野指针**

- 【错误案例】从键盘任意输入一串字符，然后在屏幕上输出。
- **warning**: function returns address of local variable [-Wreturn-local-addr]
- 野指针错误通常都发生在试图从一个函数返回指向局部变量的地址，因为系统给函数中声明的局部变量分配的内存，在函数调用结束后就被自动释放了。

```c
#include <stdio.h>
char * GetStr(void);
int main(void)
{
    char *ptr=NULL;
    ptr=GetStr();
    puts(ptr);
    return 0;
}

char* GetStr(void)
{
    char s[80];
    scanf("%s", s);
    return s;
}
```

```
asdf

Process returned 0 (0x0)   execution time : 4.982 s
Press any key to continue.
```

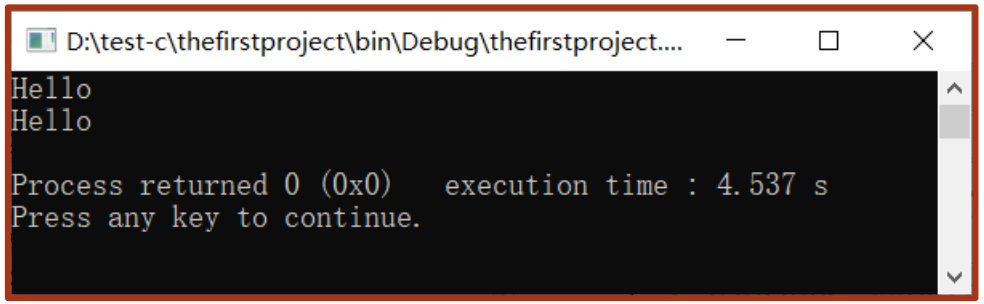
任务五 **野指针**

```c
#include <stdio.h>
void GetStr(char *);
int main(void)
{
    char s[80];
    char *ptr=s;
    GetStr(ptr);
    puts(ptr);
    return 0;
}

void GetStr(char *s)
{
    scanf("%s", s);
}
```

```
sdfdf
sdfdf

Process returned 0 (0x0)   execution time : 3.400 s
Press any key to continue.
```

任务五 **野指针**

```c
#include <stdio.h>
#include <stdlib.h>
char* GetStr(char *);
int main(void)
{
    char *ptr=NULL;
    ptr=GetStr(ptr);
    puts(ptr);
    free(ptr);
    return 0;
}

char* GetStr(char *s)
{
    s=(char*) malloc(100);
    scanf("%s", s);
    return s;
}
```



任务五 **野指针**

- 调用free释放内存之后

```c
#include <stdio.h>
#include <stdlib.h>
char* GetStr(char *);
int main(void)
{
    char *ptr=NULL;
    ptr=GetStr(ptr);
    puts(ptr);
    free(ptr);
    puts(ptr);
    return 0;
}

char* GetStr(char *s)
{
    s=(char*) malloc(100);
    scanf("%s", s);
    return s;
}
```

```
D:\tmp\c-coding\test\bin\Debug\test.exe
123qee
123qee
`gr

Process returned 0 (0x0)   execution time : 4.350 s
Press any key to continue.
```

任务六 **动态数组**

- 编程输入一个班的某课程的学生成绩，计算其平均分，然后输出。班级人数由键盘输入。

```
int *p=NULL,n,i,sum;
printf("Please enter array size:");
scanf("%d",&n);
p=(int *)malloc(n * sizeof (int));//申请n个sizeof(int)字节的内存
if (p==NULL) //确保指针使用前是非空指针，当p为空指针时结束程序运行
{
    printf("No enough memory!\n");
    exit(0);
}
printf("Please enter the score:");
for (i=0; i<n; i++) //输入n个学生的分数
{
    scanf("%d",p + i);
}
sum=0;
for (i=0; i<n; i++)
{
    sum=sum + *(p + i);
}
printf("aver=%d\n",sum/n);
free(p);
```

```
Please enter array size:4
Please enter the score:80 90 92 87
aver=87
```

任务六 **动态数组**

- 编程输入m个班（每班n个学生）全部学生的某门课的成绩，计算最高分，并指出具有该最高分成绩的学生是第几个班的第几个学生。

```
int *pScore=NULL,i,j,m,n,maxScore,row,col;
printf("Please enter array size m,n:");
scanf("%d,%d",&m,&n);//输入班级数m和学生数n
pScore=(int*)calloc(m*n,sizeof(int));//申请m*n个sizeof(int)字节的内存
if (pScore==NULL)
{
    printf("No enough memory!\n");
    exit(0);
}
printf("Please enter the score:\n");
for (i=0; i<m; i++)//输入m个班学生的某门课成绩
{
    for (j=0; j<n; j++)
    {
        scanf("%d",&pScore[i*n+j]);
    }
}
maxScore=FindMax(pScore,m,n,&row,&col);
printf("maxScore=%d,class=%d,number=%d\n",
       maxScore,row+1,col+1);//输出最高分及其所在的班级和班级内的序号
free(pScore);//释放用calloc()申请的内存
```

```
Please enter array size m,n:2,3
Please enter the score:
81 88 92
75 95 62
maxScore=95,class=2,number=2
```

```
35  //函数返回值：数组元素的最大值
36  int FindMax(int *p,int m,int n,int *pRow,int *pCol)
37  {
38      int i,j,max;
39      max=p[0];//置初值，假设第一个元素值最大
40      *pRow=0;//记录当前最大值所在的行下标
41      *pCol=0;//记录当前最大值所在的列下标
42      for (i=0; i<m; i++)
43      {
44          for (j=0; j<n; j++)
45          {
46              if (p[i*n+j] > max)
47              {
48                  max=p[i*n+j];//记录当前最大值
49                  *pRow=i;//记录当前最大值所在的行下标
50                  *pCol=j;//记录当前最大值所在的列下标
51              }
52          }
53      }
54      return max;//返回最大值
55  }
```

任务七 **指针自增**

```c
#include <stdio.h>

int main()
{
    int a=2,*p=&a;
    printf("1:%d\n",*p);
    printf("2:%p,%p,%p\n",p,&a,&p);
    printf("3:%d\n",*p++);
    printf("4:%p,%d\n",p,*p);
    printf("5:%p\n",p++);
    printf("6:%p,%d\n",p,*p);
    printf("7:%p,%p\n",p,p+1);
    return 0;
}
```

```
1:2
2:000000000061FE1C,000000000061FE1C,000000000061FE10
3:2
4:000000000061FE20,12260256
5:000000000061FE20
6:000000000061FE24,0
7:000000000061FE24,000000000061FE28
```

任务八 **围圈报数(使用指针)**

- N个人围成一圈，顺序排号。从第1个人开始报数（从1到3报数），凡报到3的人退出圈子，问最后留下的是原来第几号的人。

- 怎么表示序号?
- 怎么表示退出?
- 怎么记录1-2-3报数?
- 怎么实现围“圈”?
- 怎么判断循环停止?
- …

```
for(i=0; i<n; i++)
    *(p+i)=i+1; //以1至n为序给每个人编号
```