

1. 代码实践

```
#include <iostream>
using namespace std;
void bubblesort(int* arr,int len)
{
    for(int i=0; i<len; i++)
        for(int j=0;j<len-i-1;j++)
        {
            if(arr[j]>arr[j+1])
            {
                int temp=arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=temp;
            }
        }
}
void print(int *arr,int len)
{
    for(int i=0;i<len;i++)
        cout << arr[i] <<endl;
}

int main()
{
    int arr[10]={4,3,6,9,1,2,10,8,7,5};
    int len = sizeof(arr)/sizeof(arr[0]);
    bubblesort(arr,len);
    print(arr,len);
    system("pause");
    return 0;
}
```

结构体指针

```
#include <iostream>
using namespace std;
struct Stu
{
    string name;
    int age;
    int score;
};
int main()
{
```

```
    Stu stuArray[3]=
    {
        {"zhangsan",18,100},
        {"zhangsi",20,90},
        {"zhangwu",30,80}
    };
    for(int i=0;i<3;i++)
        cout << "name is " << stuArray[i].name << " age is " << stuArray
[i].age << " score is " << stuArray[i].score << endl;
        system("pause");
        return 0;
}

//结构体案例
#include <iostream>
#include <string>
using namespace std;
//学生的结构体定义
struct Student
{
    string sName;
    int score;
};
//老师的结构体定义
struct Teacher
{
    string tName;
    struct Student stu[5];
};
void allocateSpace(struct Teacher tArray[],int len)
{
    string nameSeed="ABCDE";
    for(int i=0;i<3;i++)
    {
        tArray[i].tName="Teacher_";
        tArray[i].tName+=nameSeed[i];
        for(int j=0;j<5;j++)
        {
            tArray[i].stu[j].sName="Student_";
            tArray[i].stu[j].sName+=nameSeed[j];
            tArray[i].stu[j].score=60+j;
        }
    }
}
```

```
}  
void printInfo(struct Teacher tarray[],int len)  
{  
    for(int i=0;i<3;i++)  
    {  
        cout << "老师姓名: " << tarray[i].tName << endl;  
        for(int j=0;j<5;j++)  
        {  
            cout << "学生姓名: " << tarray[i].stu[j].sName <<endl;  
            cout << "学生姓名: " << tarray[i].stu[j].score <<endl;  
        }  
    }  
}  
int main()  
{  
    Teacher tch[3];  
    int len = sizeof(tch)/sizeof(tch[1]);  
    allocateSpace(tch,len);  
    printInfo(tch,len);  
    system("pause");  
    return 0;  
}
```

2. 计算机基础知识整理

2.1 ARP 地址解析协议工作原理

每台主机都有一个 ARP 列表，存放 IP 地址和 MAC 地址的对应关系。

当源主机向目标主机发送数据时，首先查看 ARP 列表中 IP 地址对应的目标主机的 MAC 地址，如果找到则直接发送数据；如果找不到，就向该网段中的所有主机发送 ARP 请求包，里面存放源 IP 地址，源 MAC 地址，目标 IP 地址。

当该网段中的所有主机收到该 ARP 响应包之后，首先查看目标 ip 地址是否与自己相匹配，如果不是则忽略，如果是，就将源 ip 地址和源 MAC 地址存放到自己的 ARP 列表中，然后将自己的 MAC 地址存放到 ARP 响应包中发送给源主机；

目标主机收到 ARP 响应包，则取出对应的 IP 和 MAC 地址存放到 ARP 列表中，并发送数据。若未收到则 ARP 查询失败。

广播 ARP 请求，单播 ARP 响应。

2.2 何为死锁？何为系统调用？

答：死锁：指多个有关进程由于争夺资源而造成的一种僵局,在无外力的情况下这些进程都将无法再向前推进的状态。

系统调用：系统调用是 OS 与应用程序之间的接口,它是用户程序取得 OS 服务的唯一途径。

3. 当日工作总结

- ① 建立练习号，尝试使用 issue 和 pull request 功能。
- ② 上传学习笔记