

计算机科学基础II

实验十八 动态内存分配与深复制

曹鹏

Email: caopeng@seu.edu.cn

Tel: 13851945861

实验目的

实验十八

1. 理解运行时内存分配的概念，掌握自由存储区内存动态分配的方法。
2. 理解内部包含动态分配内存的类对象复制时的浅复制和深复制的概念，会编写深复制复制构造函数和赋值复制运算符。

实验十八 编程

hello
world
hello world

```
int main(){
    char arr1[] = "hello";
    char arr2[] = "world";
    char* str1, * str2, * str3;

    str1 = GenStr(arr1); //函数GenStr将arr1复制到str1, 注意要先动态分配空间
    str2 = GenStr(arr2); //注意arr1/2是定义的数组, str1/2指向动态分配空间
    *arr1 = '\0'; //arr1/2 清空
    *arr2 = '\0';
    cout << str1 << endl; //打印str1/2
    cout << str2 << endl;
    str3 = CombinStr(str1, str2); //函数CombinStr合并str1和str2
    cout << str3 << endl; //打印str3

    delete[] str1;
    delete[] str2;
    delete[] str3; //如果str3指向空间与str1/2重叠, 此处会重复释放

    return 0;
}
```

实验十八 编程

char* GenStr(**char***);

//功能：生成新字符串后返回，其内容从参数字符串复制

//注意：

//动态分配堆空间，而不是在栈空间定义局部变量

//堆空间的大小由参数字符串长度决定

char* CombinStr(**char***, **char***);

//功能：生成新字符串后返回，其内容是两个参数字符串的合并，中间加空格

//例如，根据" hello" 和" world" 生成" hello world"

//注意：

//动态分配堆空间，而不是在栈空间定义局部变量

//堆空间的大小由为两个参数字符串长度之和+1（中间的空格）



End