



数据结构与算法（四）

张铭 主讲

采用教材：张铭，王腾蛟，赵海燕 编写
高等教育出版社，2008.6（“十一五”国家级规划教材）

<https://pkumoooc.coursera.org/bdsalgo-001>



主要内容

- 字符串基本概念
- 字符串的存储结构
 - 字符串的顺序存储
 - 字符串类 `class String` 的存储结构
- 字符串运算的算法实现
 - 字符串运算的实现
 - `String` 类的实现
- 字符串的模式匹配
 - 朴素算法
 - KMP 快速模式匹配

字符串的顺序存储

- 对串长变化不大的字符串，有三种处理方案
 1. 用 $S[0]$ 作为记录串长的存储单元 (Pascal)
 - 缺点：限制了串的最大长度不能超过256
 2. 为存储串的长度，另辟一个存储的地方
 - 缺点：串的最大长度一般是静态给定的，不是动态申请数组空间
 3. 用一个特殊的末尾标记 ‘\0’ (C/C++)
 - 例如：C/C++ 语言的 string 函数库 (`#include <string.h>`) 采用这一存储结构
 - ‘\0’ 的 ASCII 字符表中编号为 0，等价于常量 `NULL`、数字 `0`、常量 `false`

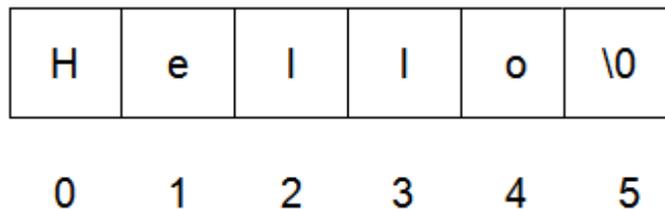
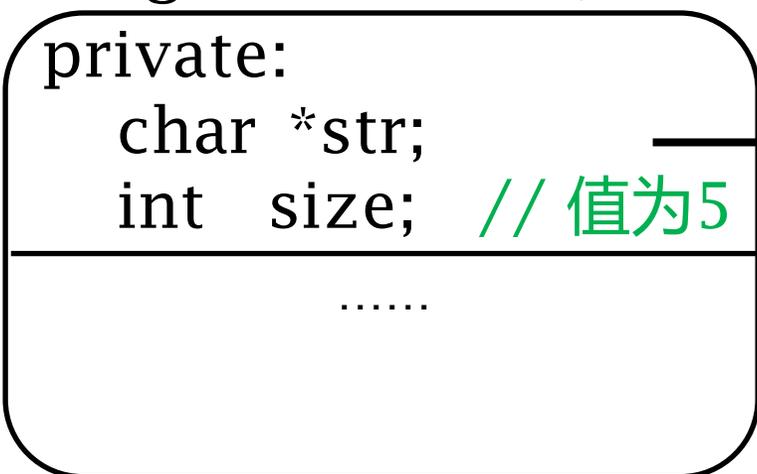


字符串类的存储结构

```
private: // 具体实现的字符串存储结构
char *str; // 字符串的数据表示
int size; // 串在当前长度
```

例如，

```
String s1 = "Hello";
```





字符串运算的算法实现

- 串长函数
 - `int strlen(char *s);`
- 串复制
 - `char *strcpy(char *s1, char*s2);`
- 串拼接
 - `char *strcat(char *s1, char *s2);`
- 串比较
 - `int strcmp(char *s1, char *s2);`

串运算的实现

```
// 求字符串的长度
int strlen(char d[ ]) {
    int i = 0;
    while (d[i] != '\0')
        i++;
    return i;
}
```

串运算的实现

```
// 字符串的复制
char *strcpy(char *d, char *s) {
    int i = 0;
    while (s[i] != '\0') {
        d[i] = s[i];  i++;
    }
    d[i] = '\0';
    return d;
}
```

串运算的实现

```
// 字符串的比较
int strcmp(const char *s1, const char *s2) {
    int i = 0;
    while (s2[i] != '\0' && s1[i] != '\0') {
        if (s1[i] > s2[i])
            return 1;
        else if (s1[i] < s2[i])
            return -1;
        i++;
    }
    if (s1[i] == '\0' && s2[i] != '\0')
        return -1;
    else if (s2[i] == '\0' && s1[i] != '\0')
        return 1;
    return 0;
}
```

更简便的算法

```
int strcmp_1(char *s1, char *s2) {  
    int i;  
    for (i = 0; s1[i] == s2[i]; ++i) {  
        if(s1[i] == '\0' && s2[i] == '\0')  
            return 0;           // 两个字符串相等  
    }  
    // 不等, 比较第一个不同的字符  
    return (s1[i]-s2[i]) / abs(s1[i]-s2[i]);  
}
```

串运算的实现

```
// 构造函数(constructor)
String::String(char *s) {
    // 先要确定新创字符串实际需要的存储空间，s的类型为(char *)，
    // 作为新创字符串的初值。确定s的长度，用标准字符串函数
    // strlen(s)计算长度
    size = strlen(s);

    // 然后，在动态存储区域开辟一块空间，用于存储初值s，把结束
    // 字符也包括进来
    str = new char [size + 1];
    // 开辟空间不成功时，运行异常，退出
    assert(str != NULL);

    // 用标准字符串函数strcpy，将s完全复制到指针str所指的存储空间
    strcpy(str, s);
}
```



String 串运算的实现

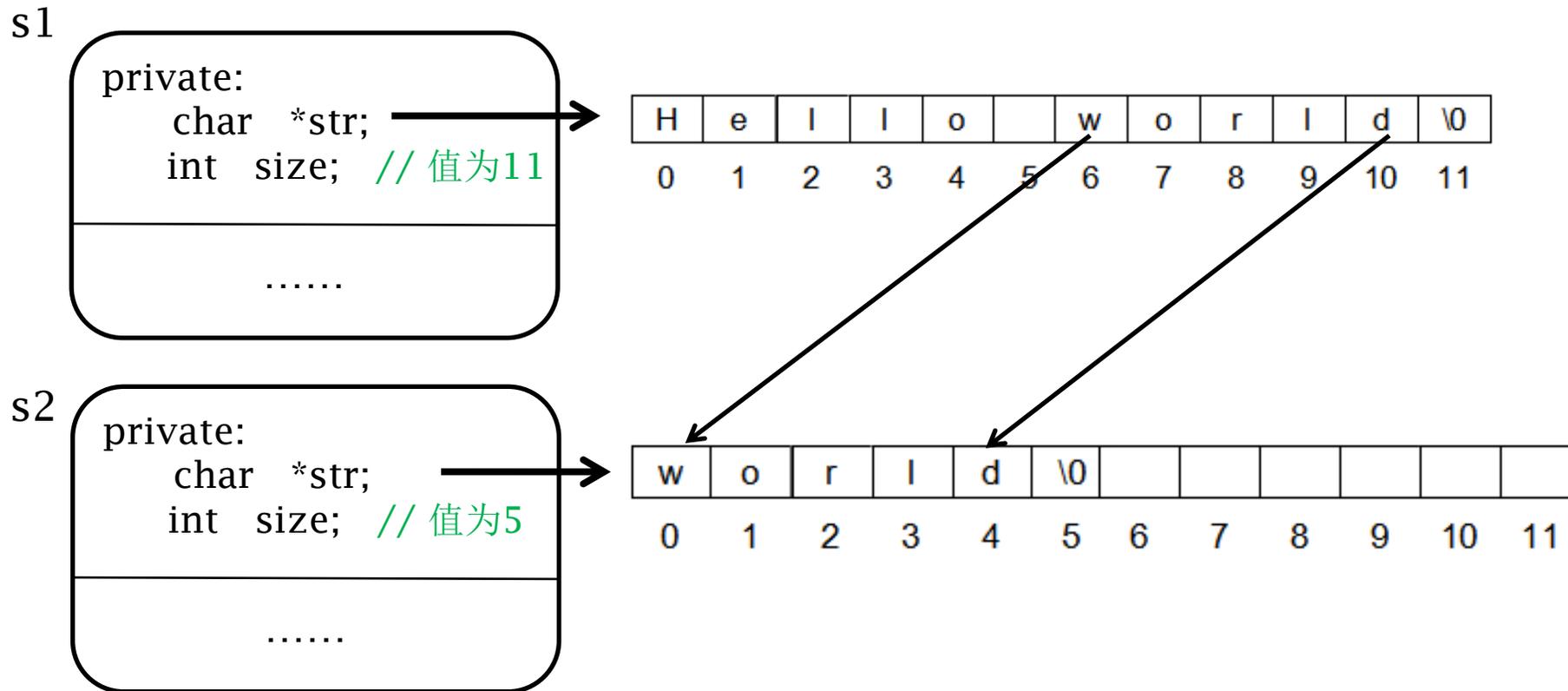
```
// 析构函数  
String::~~String() {  
    // 必须释放动态存储空间  
    delete [] str;  
}
```

String串运算的实现

```
// 赋值函数
String String::operator= (String& s) {
    // 参数 s 将被赋值到本串。
    // 若本串的串长和s的串长不同，则应该释放本串的
    // str存储空间，并开辟新的空间
    if (size != s.size) {
        delete [] str;      // 释放原存储空间
        str = new char [s.size+1];
        // 若开辟动态存储空间失败，则退出正常运行
        assert(str != NULL);
        size = s.size;
    }
    strcpy(str, s.str );
    // 返回本实例，作为String类的一个实例
    return *this;
}
```

思考：String抽取子串

- `s2 = s1.Substr(6, 5);`



思考

- 设 S_1, S_2 为串，请给出使 $S_1+S_2 == S_2+S_1$ 成立的所有可能的条件（其中 $+$ 为连接运算）
- 设计一个算法来实现字符串逆序存储，要求不另设串存储空间



数据结构与算法

谢谢聆听

国家精品课“数据结构与算法”

<https://pkumoooc.coursera.org/bdsalgo-001>

张铭, 王腾蛟, 赵海燕

高等教育出版社, 2008. 6. “十一五”国家级规划教材