



本节主题:

串的顺序存储及其基本操作实现

串的顺序存储结构

- 在顺序串中，串中的字符被依次存放在一组连续的存储单元里。

```
#define MaxSize 100 //允许字符串的最大长度
typedef struct
{
    char data[MaxSize];
    int length;
} SqString;
```

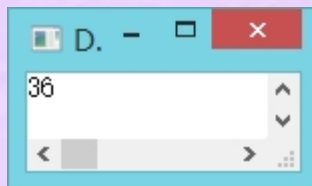
存储字符串

存储字符串的当前长度



```
1  #include <stdio.h>
2  #define MaxSize 30
3  typedef struct
4  {
5      char data[MaxSize];
6      int length;
7  } SqString;
8
9  int main()
10 {
11     SqString str;
12     printf("%d\n", sizeof(str));
13     return 0;
14 }
```

- SqString中的串用length控制
- C语言中的字符串以'\0'标志结束



串的顺序存储(以字长4字节为例)

1001	A			
1002	B			
1003	C			
1004	D			
1005	E			
1006	F			
1007	G			
1008	H			
1009	I			
100a	J			
100b	K			
100c	L			
100d	M			
100e	N			

要点

以**字**为单位的存储

不是以**字节**为单的存储

1001	A	B	C	D
1002	E	F	G	H
1003	I	J	K	L
1004	M	N		

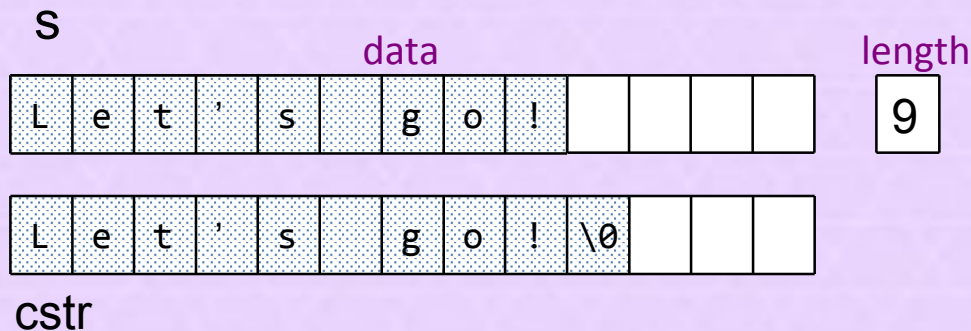
紧缩格式示例 (存储密度大)

非紧缩格式示例 (存储密度小)

串赋值: StrAssign(s,cstr)

将一个字符串常量cstr赋
给串s

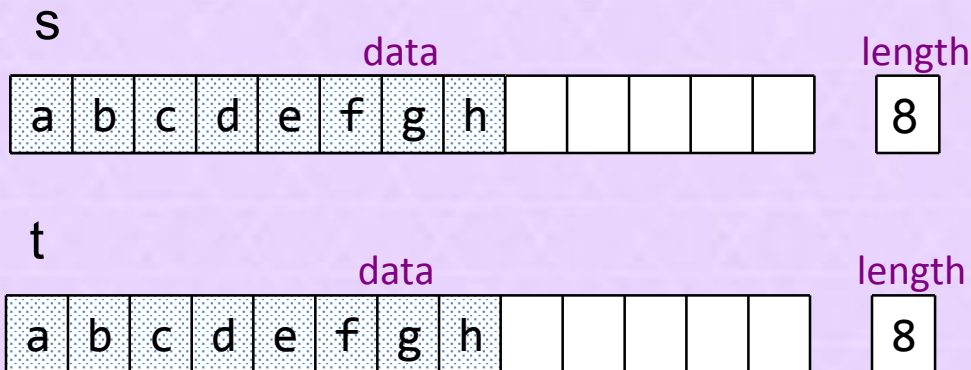
```
void StrAssign(SqString &s, char cstr[])
{
    int i;
    for (i=0; cstr[i]!='\0'; i++)
        s.data[i]=cstr[i];
    s.length=i;
}
```



串复制: StrCopy(s,t)

将串t复制给串s

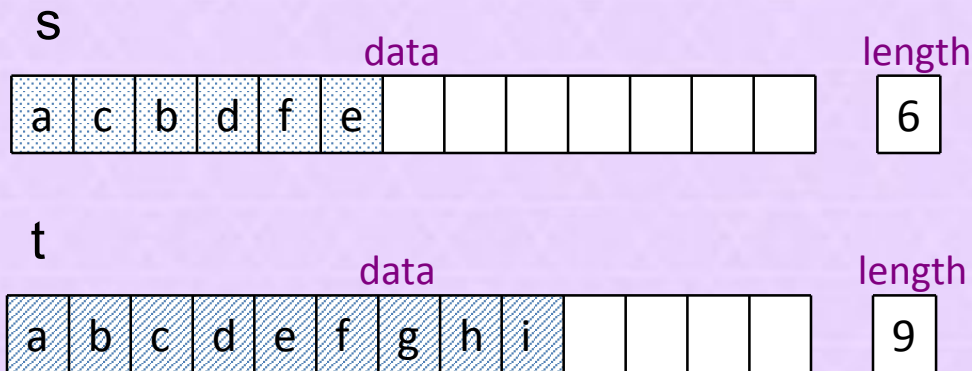
```
void StrCopy(SqString &s,SqString t)
{
    int i;
    for (i=0; i<t.length; i++)
        s.data[i]=t.data[i];
    s.length=t.length;
}
```



判串相等: StrEqual(s,t)

若两个串s与t相等返回真(1); 否则返回假(0)

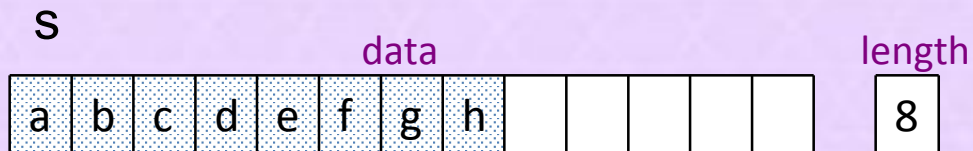
```
bool StrEqual(SqString s, SqString t)
{
    bool same=true;
    int i;
    if (s.length!=t.length)
        same=false;
    else
        for (i=0; i<s.length; i++)
            if (s.data[i]!=t.data[i])
            {
                same=false;
                break;
            }
    return same;
}
```



求串长: StrLength(s)

返回串的长度。

```
int StrLength(SqString s)
{
    return s.length;
}
```



串连接: Concat(s,t)

将两个串s和t连接形成新串，返回这个新串。

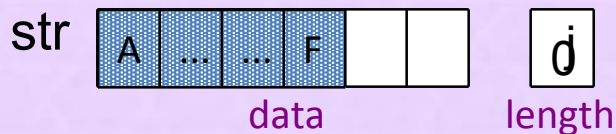
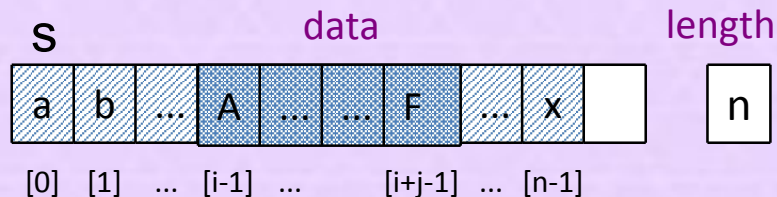
```
SqString Concat(SqString s, SqString t)
{
    SqString str;
    int i;
    str.length = s.length + t.length;
    for (i = 0; i < s.length; i++)
        str.data[i] = s.data[i];
    for (i = 0; i < t.length; i++)
        str.data[s.length + i] = t.data[i];
    return str;
}
```



求子串: SubStr(s,i,j)

```
SqString SubStr(SqString s,int i,int j)
{
    SqString str;
    int k;
    str.length=0;
    if (i<=0 || i>s.length || j<0 || i+j-1>s.length)
        return str;          //参数不正确时返回空串
    for (k=i-1; k<i+j-1; k++) //s.data[i..i+j]   str
        str.data[k-i+1]=s.data[k];
    str.length=j;
    return str;
}
```

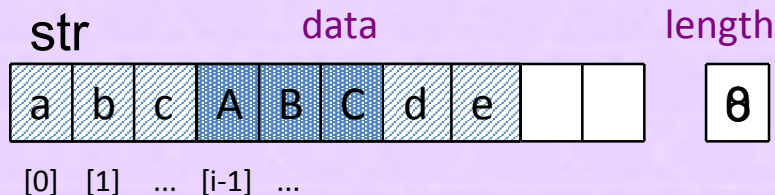
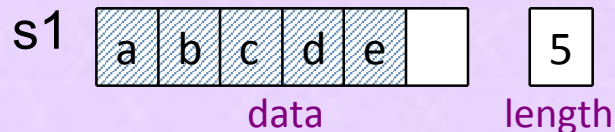
返回串 s 中从第 i 个字符开始的、由连续 j 个字符组成的子串。
参数不正确时返回一个空串。
($1 \leq i \leq \text{StrLength}(s)$)



插入串: InsStr(s1,i,s2)

```
SqString InsStr(SqString s1,int i,SqString s2)
{
    int j;
    SqString str;
    str.length=0;
    if (i<=0 || i>s1.length+1) //参数不正确时
        return str;
    for (j=0; j<i-1; j++)
        str.data[j]=s1.data[j];
    for (j=0; j<s2.length; j++)
        str.data[i+j-1]=s2.data[j];
    for (j=i-1; j<s1.length; j++)
        str.data[s2.length+j]=s1.data[j];
    str.length=s1.length+s2.length;
    return str;
}
```

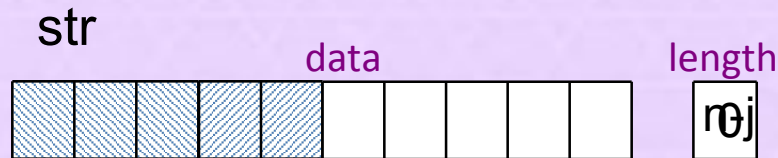
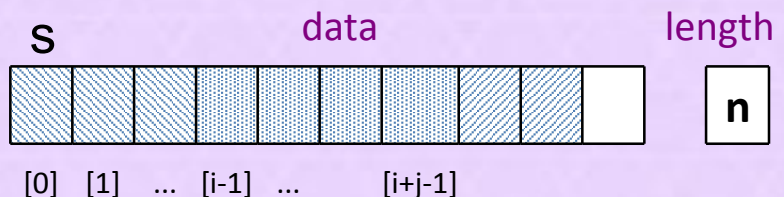
将串s2插入到串s1的第i个字符中，
即将s2的第一个字符作为s1的第i个
字符，并返回产生的新串。
参数不正确时返回一个空串。
($1 \leq i \leq \text{StrLength}(s1)+1$)



删除串中字符: DelStr(s,i,j)

```
SqString DelStr(SqString s,int i,int j)
{
    int k;
    SqString str;
    str.length=0;
    if (i<=0 || i>s.length || i+j>s.length+1)
        return str;    //参数不正确时返回空串
    for (k=0; k<i-1; k++)
        str.data[k]=s.data[k];
    for (k=i+j-1; k<s.length; k++)
        str.data[k-j]=s.data[k];
    str.length=s.length-j;
    return str;
}
```

从串s中删去第i个字符开始的长度为j的子串，并返回产生的新串。
参数不正确时返回一个空串，
($1 \leq i \leq \text{StrLength}(s)$)

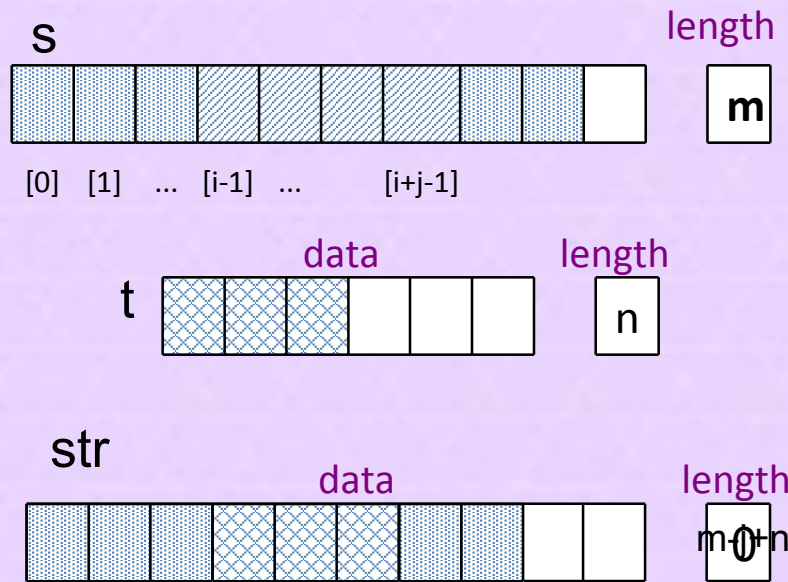


替换子串: RepStr(s,i,j,t)

SqString RepStr(SqString s,int i,int j,SqString t)

```
{
    int k;
    SqString str;
    str.length=0;
    if (i<=0 || i>s.length || i+j-1>s.length)
        return str; //参数不正确时返回空串
    for (k=0; k<i-1; k++)
        str.data[k]=s.data[k];
    for (k=0; k<t.length; k++)
        str.data[i+k-1]=t.data[k];
    for (k=i+j-1; k<s.length; k++)
        str.data[t.length+k-j]=s.data[k];
    str.length=s.length-j+t.length;
    return str;
}
```

在串s中，将第i个字符开始的j个字符构成的子串用串t替换，并返回产生的新串。参数不正确时返回一个空串。 $(1 \leq i \leq \text{StrLength}(s))$



输出串：DispStr(s)

显示串中的字符

```
void DispStr(SqString s)
{
    int i;
    if (s.length>0)
    {
        for (i=0; i<s.length; i++)
            printf("%c",s.data[i]);
        printf("\n");
    }
}
```

