4.4 串操作应用举例

4.4.1 文本编辑

文本编辑的实质是修改字符数据的形式或格式。

把文本看成是一个字符串,称为文本串。页是文本串的子串,行又是页的子串。

比如有下列一段源程序:

```
main(){
    float a, b, max;
    scanf ( "%f, %f" , &a, &b );
    if a > b max = a;
    else max = b;
};
    我们可以把此程序看成是一个文本串,输入到内存后如下图示.
图中 "↓" 换行符。
```

m	а	i	n	()	{	\downarrow			f	I	O	а	t		а	,	b	,
m	а	X	;	\	Ħ		S	С	а	n	f	("	%	f	,	%	f	"
,	&	а	,	&	b)	-,	\downarrow			i	f		а	>	b			m
а	Х	=	а	•	\downarrow			е	I	S	е			m	а	X	=	b	•
\downarrow	}	\downarrow																	

为了管理文本串的页和行,在进入文本编辑的时候,编辑程序

先为文本串建立相应的页表和行表,即建立各子串的存储映像。页

表的每一项给出了页号和该页的起始行号。而行表的每一项则指示

每一行的行号、起始地址和该行子串的长度。假设上图所示文本串

只占一页,且起始行号为100,则该文本串的行表如图:

行号	起始地址	长度
100	201	8
101	209	17
102	226	24
103	250	17
104	267	15
105	282	2

本章学习要点

- 熟悉串的七种基本操作的定义,并能利用这些基本操作来实现 串的其它各种操作的方法。
- 2. 熟练掌握在串的定长顺序存储结构上实现串的各种操作的方法。
- 3. 了解串的堆存储结构以及在其上实现串操作的基本方法。
- 4. 理解串匹配的KMP算法,熟悉NEXT函数的定义,学会手工计算 给定模式串的NEXT函数值和改进的NEXT函数值。
- 5. 了解串操作的应用方法和特点。