

汇编语言程序设计课程实验报告

实验 6

姓名		院系	计算机工程与科学学院	学号	
实验目的:					
建立并查找电话号码表					
实验要求:					
通过子程序设计要求编程建立一个电话号码表（含人名与电话号码两部分），并进行查找。					

1. 程序分析

程序的基本思路是分为一个子程序模块进行设计，第一个是存储姓名，电话模块，因为电话簿的存储空间是连续开辟的，需要记录每一次写入后当前存储到了哪个单元，并继续存储下一个，这里需要注意的是，对于每个姓名规定一个上限，未达到上限长度的姓名填充空格，方便后面查找。第二个是判错的子程序，存储的过程中如若输入的名字是否超过规定长度，输入的电话号码是否为规定长度等，不符合要求则立马提示重新输入。第三个是查找的子程序，这个子程序又分为两个部分，存储用户输入的姓名和查找过程，查找时采用逐位判断的方法，如果每一位都符合且下一个存储单元不是空格，说明该姓名存在，否则该姓名只是某个姓名的前缀，查找失败。

2. 程序运行结果

```
How many numbers do you want to store?(less than 9)
1
New name:(less than 15 letters)
aaa
New phone number:(6 letters)
123
Wrong number!
New phone number:(6 letters)
_
```

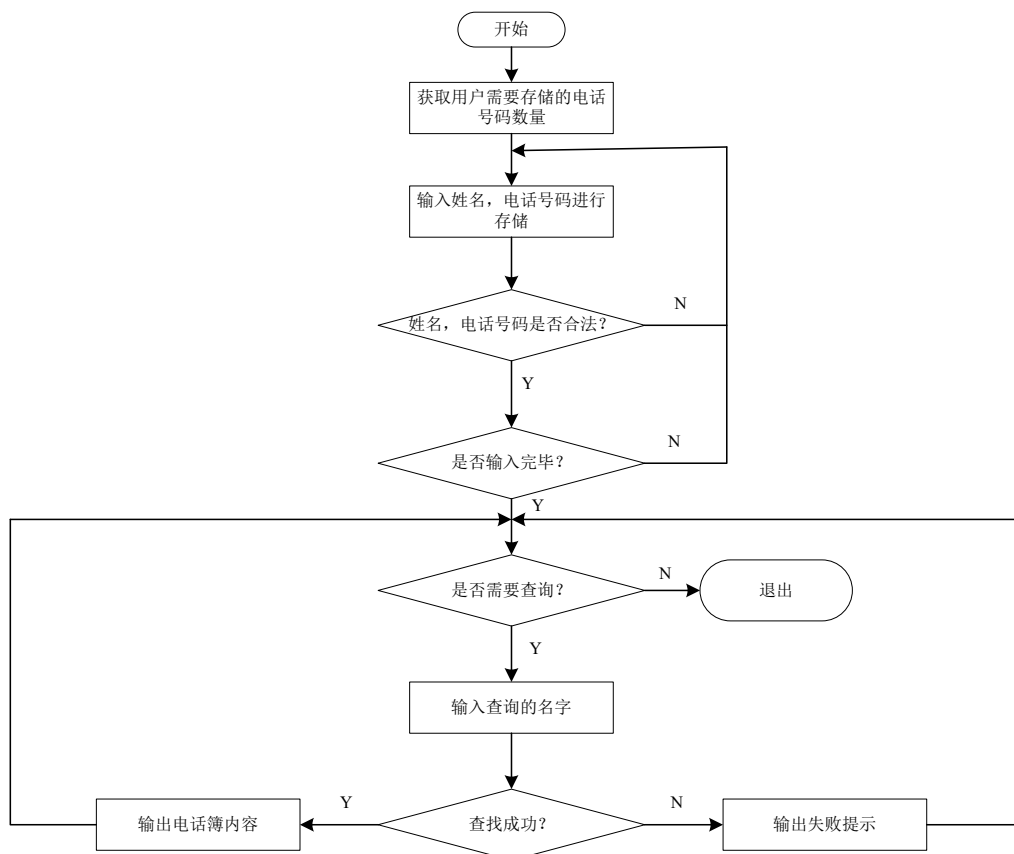
```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
How many numbers do you want to store?(less than 9)
2
New name:(less than 15 letters)
aaa
New phone number:(6 letters)
123456
New name:(less than 15 letters)
aaaaa
New phone number:(6 letters)
654321
Dou you want to search? (y/n)
y
Name:
aa
Not found!
Dou you want to search? (y/n)
y
Name:
aaaaa
Search successfully!
Name          Phone number
aaaaa        654321
Dou you want to search? (y/n)
Do you need to keep the DOSBox [Y,N]?
```

由运行结果可知，当输入的电话号码不符合要求时，会提示重新输入；当输入的姓名为前缀时，查找结果显示正确。

3. 实验体会

这次实验的难度较大，是在上个实验字符串匹配的基础上增加了搜索，存储，子程序设计的内容，关键点在于如何在连续开辟的空间内多次正确的存储离散输入的信息，以及在查找过程中匹配失败，如何跳到下一个查找位置的设计，模块化的设计使得程序的主要部分变得更加的简洁与直观，但同时也出现了使用到的寄存器的保存问题，在实验课中我就因为这个问题 debug 了非常长的时间，更加提醒我保存现场以及使用寄存器前清零的重要性。

4. 程序流程图



5. 程序代码

1. data segment

2. mess1 db 'New name:(less than 15 letters)', '\$'

3. mess2 db 'New phone number:(6 letters)', '\$'

4. mess3 db 'Dou you want to search? (y/n)', '\$'

5. mess4 db 0dh, 0ah, 'Name:', '\$'

6. mess5 db 'Not found!', '\$'

7. mess6 db 'How many numbers do you want to store?(less than 9)', '\$'

8. mess7 db 'Wrong number!', '\$'

9. mess8 db 'Wrong name', '\$'

10. mess9 db 'Search successfully!', '\$'

11. show db 'Name Phone number', '\$'

12. changerow db 0dh, 0ah, '\$'

13. ;存储需要计入电话簿的姓名和电话号码

14. max1 db 15

15. act1 db ?

16. name1 db 15 dup(?)

17. max2 db 8

18. act2 db ?

19. phone_number db 8 dup(?)

20. ;开辟空间存储电话簿

21. table db 20 dup(21 dup(?))

22. ;记录电话簿总数

23. totnum db ?

24. ;存储输入需要查找的名字

25. max3 db 15

26. act3 db ?

27. find_name db 15 dup(?)

28. ;查找成功时，存储查找到的信息

29. found db 21 dup(?), '\$'

30. now_address dw 0

31. ;记录当前存储位置

32. search_address dw 0

33. ;记录查找到的位置

34. search_tot db 0

35. ;记录查找到的总数

36. data ends

37.

38. data_seg segment

39. db 20 dup(21 dup(?))

40. data_seg ends

41.

42. stack segment

43. dw 128 dup(?)

44.	stack ends	
45.	code segment	
46.		assume cs:code,ds:data,es:data_seg,ss:stack
47.	start:	
48.	;输入需要存储的电话总数	
49.	mov	ax,data
50.	mov	ds,ax
51.		
52.	lea	dx,mess6
53.	mov	ah,9
54.	int	21h
55.	call	change
56.		
57.	mov	ah,1
58.	int	21h
59.	call	change
60.		
61.	sub	cx,cx
62.	sub	al,30h
63.	mov	cl,al
64.	mov	totnum,cl
65.	;循环获取输入的姓名和电话	
66.	input:	
67.	push	cx
68.	call	input_name
69.	call	input_number
70.	pop	cx
71.	loop	input
72.	;查找循环体	
73.	search:	
74.	sub	ax,ax
75.	sub	bx,bx
76.	sub	cx,cx
77.	sub	dx,dx
78.	mov	search_tot,0
79.	lea	dx,mess3
80.	mov	ah,9
81.	int	21h
82.	call	change
83.		
84.	mov	ah,1
85.	int	21h
86.		
87.	cmp	al,'y'

88.		je	search1
89.		cmp	al,'n'
90.		je	exit
91.	exit:		
92.		mov	ah,4CH
93.		int	21H
94.	search1:		
95.		lea	dx,mess4
96.		mov	ah,9
97.		int	21h
98.		call	change
99.			
100.		lea	dx,max3
101.		mov	ah,0ah
102.		int	21h
103.		call	change
104.			
105.		call	search_name
106.	;查找子程序		
107.	search_name proc near		
108.		mov	search_address,bx
109.		lea	bx,table
110.	;新一轮查找，初始化各个参数		
111.	search2:		
112.		sub	cx,cx
113.		mov	cl,act3
114.		mov	di,0
115.		mov	si,0
116.		mov	bx,search_address
117.	;逐位比对		
118.	search3:		
119.		mov	al,find_name[di]
120.		cmp	al,table[bx+di]
121.		jne	notfound
122.		inc	di
123.		inc	si
124.		loop	search3
125.		cmp	table[bx+di],' '
126.		je	succ
127.		jmp	notfound
128.	;比对出错或者名字是别的名字的一部分，如果没找完电话簿，跳到下一条信息开始查找		
129.	notfound:		
130.		add	search_address,21

```

131.          inc    search_tot
132.          mov     dl,search_tot
133.          cmp     dl,totnum
134.          jl      search2
135.
136.          lea     dx,mess5
137.          mov     ah,9
138.          int     21h
139.          call    change
140.          jmp     search
141.      ;查找成功，输出信息，并回到查找循环体
142.      succ:
143.          lea     dx,mess9
144.          mov     ah,9
145.          int     21h
146.          call    change
147.
148.          lea     dx,show
149.          mov     ah,9
150.          int     21h
151.          call    change
152.
153.          mov     bx,search_address
154.          mov     cx,21
155.          mov     di,0
156.
157.      storefound:
158.          mov     al,table[bx+di]
159.          mov     found[di],al
160.          inc     di
161.          loop    storefound
162.
163.          lea     dx,found
164.          mov     ah,9
165.          int     21h
166.          call    change
167.          jmp     search
168.      search_name endp
169.      ;存储信息子程序
170.      input_name proc near
171.          lea     dx,mess1
172.          mov     ah,9
173.          int     21h
174.          call    change

```

175.		
176.	lea	dx,max1
177.	mov	ah,0ah
178.	int	21h
179.	call	change
180.		
181.	sub	bx,bx
182.	mov	bl,act1
183.	mov	cx,15
184.	sub	cx,bx
185.	;名字存储不够填充空格	
186.	pushspace:	
187.	mov	name1[bx],' '
188.	inc	bx
189.	loop	pushspace
190.		
191.	mov	cx,15
192.	mov	bx,now_address
193.	mov	di,0
194.		
195.	storename:	
196.	mov	al,name1[di]
197.	mov	table[bx+di],al
198.	inc	di
199.	loop	storename
200.		
201.	add	bx,15
202.	mov	now_address,bx
203.	ret	
204.	input_name endp	
205.	;存储电话号码子程序	
206.	input_number proc near	
207.	input_number1:	
208.	lea	dx,mess2
209.	mov	ah,9
210.	int	21h
211.	call	change
212.		
213.	lea	dx,max2
214.	mov	ah,0ah
215.	int	21h
216.	call	change
217.		
218.	cmp	act2,6

```

219.             jne    deal1
220.
221.             sub     cx,cx
222.             mov     cx,6
223.             mov     bx,now_address
224.             mov     si,0
225. storenumber:
226.             mov     al,phone_number[si]
227.             mov     table[bx+si],al
228.             inc     si
229.             loop    storenumber
230.             add     bx,6
231.             mov     now_address,bx
232.             ret
233. ;电话号码位数不对，重新输入
234. deal1:
235.             lea     dx,mess7
236.             mov     ah,9
237.             int     21h
238.             call    change
239.             jmp     input_number1
240. input_number endp
241. change PROC
242.             mov     dx,OFFSET changerow
243.             mov     ah,09h
244.             int     21H
245.             RET
246. change ENDP
247.
248. code ends
249. end start

```