**汇编语言程序设计课程实验报告**

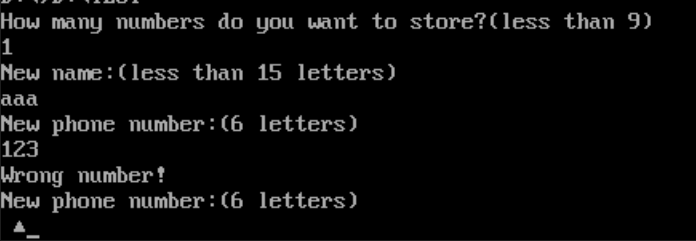
**实验6**

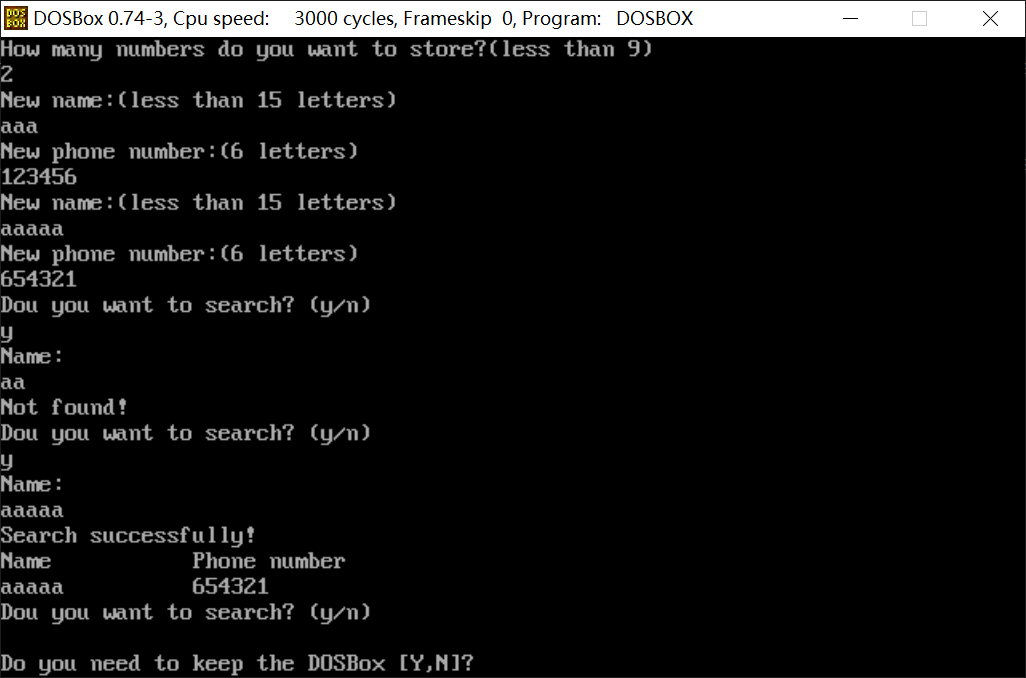
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 姓名 |  | 院系 | 计算机工程与科学学院 | 学号 |  |
| 实验目的： | | | | | |
| 建立并查找电话号码表 | | | | | |
| 实验要求： | | | | | |
| 通过子程序设计要求编程建立一个电话号码表（含人名与电话号码两部分），并进行查找。 | | | | | |

1. 程序分析

程序的基本思路是分为 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。

1. 程序运行结果





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。

1. 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。

1. 程序流程图



1. 程序代码

|  |  |
| --- | --- |
|  | data segment |
|  | mess1          db 'New name:(less than 15 letters)','$' |
|  | mess2          db 'New phone number:(6 letters)','$' |
|  | mess3          db 'Dou you want to search? (y/n)','$' |
|  | mess4          db 0dh,0ah,'Name:','$' |
|  | mess5          db 'Not found!','$' |
|  | mess6          db 'How many numbers do you want to store?(less than 9)','$' |
|  | mess7          db 'Wrong number!','$' |
|  | mess8          db 'Wrong name','$' |
|  | mess9          db 'Search successfully!','$' |
|  | show           db 'Name           Phone number','$' |
|  | changerow      db 0dh,0ah,'$' |
|  | ;存储需要计入电话簿的姓名和电话号码 |
|  | max1           db 15 |
|  | act1           db ? |
|  | name1          db 15 dup(?) |
|  | max2           db 8 |
|  | act2           db ? |
|  | phone\_number   db 8 dup(?) |
|  | ;开辟空间存储电话簿 |
|  | table          db 20 dup(21 dup(?)) |
|  | ;记录电话簿总数 |
|  | totnum         db ? |
|  | ;存储输入需要查找的名字 |
|  | max3           db 15 |
|  | act3           db ? |
|  | find\_name      db 15 dup(?) |
|  | ;查找成功时，存储查找到的信息 |
|  | found          db 21 dup(?),'$' |
|  | now\_address    dw 0 |
|  | ;记录当前存储位置 |
|  | search\_address dw 0 |
|  | ;记录查找到的位置 |
|  | search\_tot     db 0 |
|  | ;记录查找到的总数 |
|  | data ends |
|  |  |
|  | data\_seg segment |
|  | db 20 dup(21 dup(?)) |
|  | data\_seg ends |
|  |  |
|  | stack segment |
|  | dw 128 dup(?) |
|  | stack ends |
|  | code segment |
|  | assume cs:code,ds:data,es:data\_seg,ss:stack |
|  | start: |
|  | ;输入需要存储的电话总数 |
|  | mov    ax,data |
|  | mov    ds,ax |
|  |  |
|  | lea    dx,mess6 |
|  | mov    ah,9 |
|  | int    21h |
|  | call   change |
|  |  |
|  | mov    ah,1 |
|  | int    21h |
|  | call   change |
|  |  |
|  | sub    cx,cx |
|  | sub    al,30h |
|  | mov    cl,al |
|  | mov    totnum,cl |
|  | ;循环获取输入的姓名和电话 |
|  | input: |
|  | push   cx |
|  | call   input\_name |
|  | call   input\_number |
|  | pop    cx |
|  | loop   input |
|  | ;查找循环体 |
|  | search: |
|  | sub    ax,ax |
|  | sub    bx,bx |
|  | sub    cx,cx |
|  | sub    dx,dx |
|  | mov    search\_tot,0 |
|  | lea    dx,mess3 |
|  | mov    ah,9 |
|  | int    21h |
|  | call   change |
|  |  |
|  | mov    ah,1 |
|  | int    21h |
|  |  |
|  | cmp    al,'y' |
|  | je     search1 |
|  | cmp    al,'n' |
|  | je     exit |
|  | exit: |
|  | mov    ah,4CH |
|  | int    21H |
|  | search1: |
|  | lea    dx,mess4 |
|  | mov    ah,9 |
|  | int    21h |
|  | call   change |
|  |  |
|  | lea    dx,max3 |
|  | mov    ah,0ah |
|  | int    21h |
|  | call   change |
|  |  |
|  | call   search\_name |
|  | ;查找子程序 |
|  | search\_name proc near |
|  | mov    search\_address,bx |
|  | lea    bx,table |
|  | ;新一轮查找，初始化各个参数 |
|  | search2: |
|  | sub    cx,cx |
|  | mov    cl,act3 |
|  | mov    di,0 |
|  | mov    si,0 |
|  | mov    bx,search\_address |
|  | ;逐位比对 |
|  | search3: |
|  | mov    al,find\_name[di] |
|  | cmp    al,table[bx+di] |
|  | jne    notfound |
|  | inc    di |
|  | inc    si |
|  | loop   search3 |
|  | cmp    table[bx+di],' ' |
|  | je     succ |
|  | jmp    notfound |
|  | ;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 |
|  | notfound: |
|  | add    search\_address,21 |
|  | inc    search\_tot |
|  | mov    dl,search\_tot |
|  | cmp    dl,totnum |
|  | jl     search2 |
|  |  |
|  | lea    dx,mess5 |
|  | mov    ah,9 |
|  | int    21h |
|  | call   change |
|  | jmp    search |
|  | ;查找成功，输出信息，并回到查找循环体 |
|  | succ: |
|  | lea    dx,mess9 |
|  | mov    ah,9 |
|  | int    21h |
|  | call   change |
|  |  |
|  | lea    dx,show |
|  | mov    ah,9 |
|  | int    21h |
|  | call   change |
|  |  |
|  | mov    bx,search\_address |
|  | mov    cx,21 |
|  | mov    di,0 |
|  |  |
|  | storefound: |
|  | mov    al,table[bx+di] |
|  | mov    found[di],al |
|  | inc    di |
|  | loop   storefound |
|  |  |
|  | lea    dx,found |
|  | mov    ah,9 |
|  | int    21h |
|  | call   change |
|  | jmp    search |
|  | search\_name endp |
|  | ;存储信息子程序 |
|  | input\_name proc near |
|  | lea    dx,mess1 |
|  | mov    ah,9 |
|  | int    21h |
|  | call   change |
|  |  |
|  | lea    dx,max1 |
|  | mov    ah,0ah |
|  | int    21h |
|  | call   change |
|  |  |
|  | sub    bx,bx |
|  | mov    bl,act1 |
|  | mov    cx,15 |
|  | sub    cx,bx |
|  | ;名字存储不够填充空格 |
|  | pushspace: |
|  | mov    name1[bx],' ' |
|  | inc    bx |
|  | loop   pushspace |
|  |  |
|  | mov    cx,15 |
|  | mov    bx,now\_address |
|  | mov    di,0 |
|  |  |
|  | storename: |
|  | mov    al,name1[di] |
|  | mov    table[bx+di],al |
|  | inc    di |
|  | loop   storename |
|  |  |
|  | add    bx,15 |
|  | mov    now\_address,bx |
|  | ret |
|  | input\_name endp |
|  | ;存储电话号码子程序 |
|  | input\_number proc near |
|  | input\_number1: |
|  | lea    dx,mess2 |
|  | mov    ah,9 |
|  | int    21h |
|  | call   change |
|  |  |
|  | lea    dx,max2 |
|  | mov    ah,0ah |
|  | int    21h |
|  | call   change |
|  |  |
|  | cmp    act2,6 |
|  | jne    deal1 |
|  |  |
|  | sub    cx,cx |
|  | mov    cx,6 |
|  | mov    bx,now\_address |
|  | mov    si,0 |
|  | storenumber: |
|  | mov    al,phone\_number[si] |
|  | mov    table[bx+si],al |
|  | inc    si |
|  | loop   storenumber |
|  | add    bx,6 |
|  | mov    now\_address,bx |
|  | ret |
|  | ;电话号码位数不对，重新输入 |
|  | deal1: |
|  | lea    dx,mess7 |
|  | mov    ah,9 |
|  | int    21h |
|  | call   change |
|  | jmp    input\_number1 |
|  | input\_number endp |
|  | change PROC |
|  | mov    dx,OFFSET changerow |
|  | mov    ah,09h |
|  | int    21H |
|  | RET |
|  | change ENDP |
|  |  |
|  | code ends |
|  | end start |