

# 上海大学



SHANGHAI UNIVERSITY

2021-2022 学年秋季学期

上海大学 计算机学院

《汇编语言程序设计》

实验 4

实验名称 : 建立并查找电话号码表

专业 : 19 级直招计科 2 班

姓名 : 毕欣怡

学号 : 19120194

实验名称： 建立并查找电话号码表

## 一、实验目的

通过子程序设计，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。

## 二、实验内容

### (1) 实验原理

#### ①建立电话号码表

建立电话号码表，即将从键盘输入的人名与号码一一对应保存在一串连续的存储空间里。同时，为了让之后的查找和显示更加方便，设置了一个记录人名号码数量的变量 count，每次输入一组人名和号码之后，count 加一。

#### ②查找电话号码表

查找分按姓名查找和按号码查找，二者具体操作相似。

先将键盘输入的姓名或号码保存在寄存器中，将保存总体号码表的 Telbook 的偏移地址保存在变址寄存器中。从第一条记录开始，按字节进行比较，如果该条记录比较不匹配，则加上 32 跳转到下一个姓名或号码开始的地方，重新开始比较。若是在 count 为 0 前（比较结束前）找到，则输出该条人名与号码的记录，否则输出没有记录。

#### ③显示电话号码表

显示电话号码表主要需要保存 count 到 cx，再通过循环 loop 操作，将 Telbook 中所有的记录一条条输出即可。

另外，本实验功能比较复杂，所以设置了多个菜单进行指引跳转。设置子程序与宏操作，将操作简化。

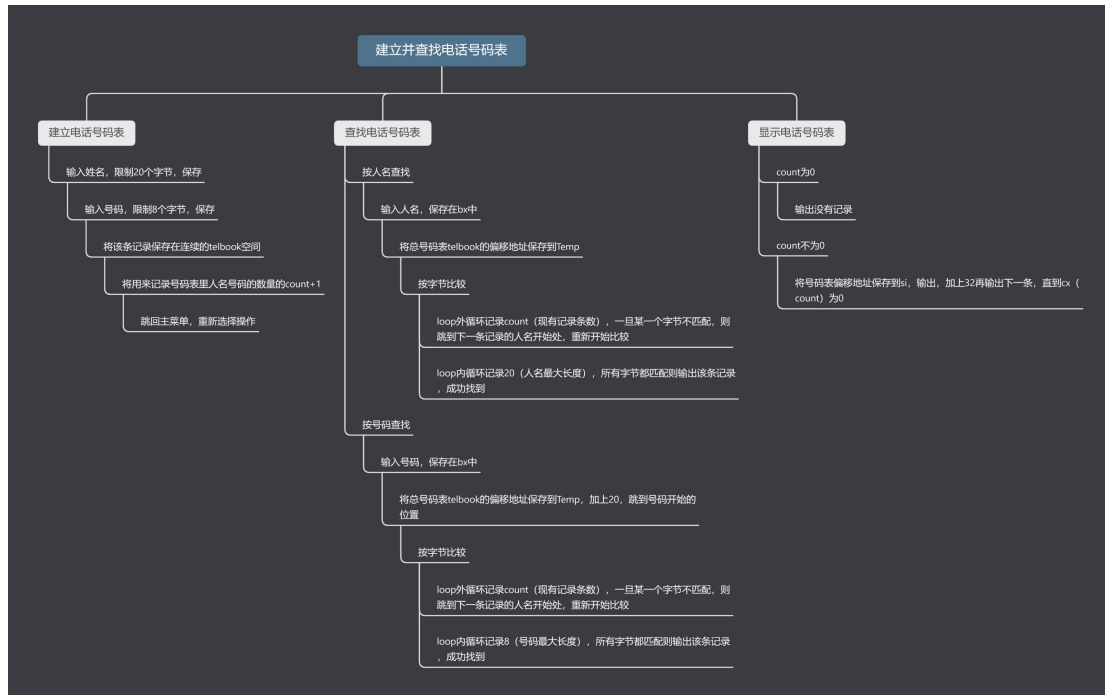
### (2) 实验步骤

- (1) 启动 MASM 6.0 或 MASM for Windows 集成编程环境。
- (2) 分支指令形式编写 .ASM 源程序。
- (3) 对其进行汇编及连接，产生 .EXE 文件。

(4) 作必要的调试。

### (3) 实验记录

实验思路如下图所示。



① “建立电话号码表”功能设计为 Insert 子程序。

```
Insert proc;输入子程序
    renew N_T,20 ;还原缓冲区内容
    renew T_T,8
    print str1
    input N_Temp ;输入姓名存在暂存缓冲区，带回车符结束
    quhc N_Temp,N_T ;去除回车符保存
    print str2
    input T_Temp ;输入电话号码暂存缓冲区，带回车符结束
    quhc T_Temp,T_T ;去除回车符保存
    lea si,N_T ;将姓名的偏移地址保存在si
    mov cx,20
    cld
    rep movsb ;将si中的20个字节的姓名内容按字节传送到di，即50*32的号码簿中
    lea si,T_T
    mov cx,12
    cld
    rep movsb ;将si中的12个字节的号码内容按字节传送到di，即50*32的号码簿中
    inc count ;count记录有几条记录
    ret ;返回主程序
Insert endp
```

当主菜单选择“Insert”的时候，表示跳转到插入电话号码表一条新记录子程序。此时按指示输入相应的姓名和号码，程序会自动将键盘输入到缓存区的姓名和号码依次保存在 Telbook 这个连续的 50\*32 的地址空间中。

每次输入一条姓名号码记录完毕后，系统会自动跳回主菜单，重新选择操作。

## ② “查找电话号码表”功能设计为 Search 子程序。

```

Search proc;查找子程序
;-----按姓名查找-----;
T0:print window2
mov ah,1
int 21h
cmp al,'1' ;按姓名查找
jne T1 ;否则进入下一个判断
print str1
renew N_Temp,20 ;清空姓名缓冲区
input N_Temp ;输入姓名,保存在缓冲区
quhc N_Temp,N_T ;将输入的姓名去除回车符,保存
;*****按姓名查找*****;
mov cx,count ;将号码簿中存在的记录数保存到cx中
mov Temp,offset telbook ;用Temp记录号码簿的偏移地址
F3:push cx ;cx入栈保存,即所有的记录数量
mov si,Temp ;si记录号码簿的偏移地址
lea bx,N_T ;将刚刚输入的姓名的偏移地址保存在bx中
mov cx,20 ;cx中保存姓名的最大值20
F2:mov dl,[si] ;号码簿记录保存在dl
cmp dl,[bx] ;号码簿中的记录与刚刚输入的姓名作比较
jne F1 ;不相等跳到F1,移动号码簿的开始位置
inc si ;否则都向后移一位继续比较下一个字节
inc bx ;cx初始值为20,每次loop操作减一,到0结束,表示该姓名结束比较
loop F2 ;该姓名20个字节全部比较完成,且没有跳到JNE,所以匹配,输出找到的记录*****;
print str3 ;找到,输出记录
mov dx,Temp ;将此条记录输出
mov ah,9
int 21h
print str5
pop cx ;cx出栈,变回号码簿的记录条数
jmp F0 ;查找结束,找到,跳回主程序
;*****该名字不匹配,跳到下一个名字开始的地方。直到号码簿中所有记录比完,输出没找到*****;

F1:add Temp,32 ;跳32个字节到下一个姓名开始的位置
pop cx
loop F3 ;重复查找操作,直到cx为0时结束,注意此时的cx是出栈后的cx,即号码簿记录的条数
print str4 ;两个loop操作都结束时,表示号码簿中的记录都比较完成,没有找到
jmp F0 ;查找结束,没找到,跳到主程序

;-----按号码查找-----;
T1:cmp al,'2'
jne T2 ;否则进入下一个判断
print str2
renew T_Temp,8 ;清空号码缓冲区
input T_Temp
quhc T_Temp,T_T ;将输入到缓冲区的号码去除回车符保存
;-----按号码查找(具体操作与按姓名相同)-----;
mov cx,count
mov Temp,offset telbook
K3:push cx
mov si,Temp
add si,20 ;将si的位置移到号码开始的位置,因为名字占20个字节,所以加上20
lea bx,T_T
mov cx,8 ;号码占8个字节
K2:mov dl,[si]
cmp dl,[bx]
jne K1 ;不相等就跳到下一个号码开始的位置,重新开始比较
inc si
inc bx
loop K2

;-----找到号码,匹配,输出该条记录-----;
print str3
mov dx,Temp
mov ah,9
int 21h
print str5

pop cx
jmp F0

;-----跳到下一个号码,直到所有记录都被比过-----;
K1:add Temp,32
pop cx
loop K3
print str4 ;所有记录没有找到,输出不存在
jmp F0
T2:cmp al,'3' ;返回主程序重新选择功能
je F0
print error1 ;以上都不符合,则输入的数字有问题,输出提示语句并重新跳回查找菜单
jmp T0
F0:ret
Search endp

```

当主菜单选择“Search”的时候，表示跳转到查找电话号码表的子程序。该子程序也包含一个子菜单：可以选择姓名查找或者号码查找，二者操作类似。

按姓名/号码查找，首先需要将从键盘读入的姓名（最多 20 个字符）/号码（最多 8 个字符）保存在寄存器中，再将保存所有记录的电话号码表的偏移地址也保存在 si 中。之后从第一条记录的相应位置开始，按字节进行比较，若是全部相等（比较完全部 20/8 个字节），则表示查找成功，输出该条记录。在比较过程中只要有一个字节不匹配，就加上 32 个字节跳到下一条记录的对应开始位置重新开始比较。直到 count 为 0，表示所有记录都已经比较过了，查找失败，输出没有查找到

每次查找一次完毕后，系统会自动跳回主菜单，重新选择操作。

③ “显示电话号码表”功能设计为 List 子程序。

```
List proc;显示号码簿子程序
    print str3
    cmp count,0
    je Z2          ;如果count为0，则表示号码簿里没有记录
    mov cx,count   ;否则将要输入的记录的条数保存到cx，控制loop循环次数
    lea si,telbook ;将号码簿偏移地址保存到si
Z1:mov dx,si
    mov ah,9
    int 21h
    add si,32      ;加上32，跳到下一条记录重复输出操作
    loop Z1
    jmp Z3
Z2:print str6
Z3:print str5
    ret
List endp
```

显示电话号码表，即将保存在 Telbook 中的所有记录全部输出，直到 count 为 0。所以先将 count 的值保存在 cx 中，用作控制 loop 循环的条件。再调用 9 号功能输出一串记录，再加上 32 表示跳到下一条记录的开头，继续输出。

#### (4) 数据处理

按照实验流程，首先建立一个电话号码表。

按指示，分别输入信息如下：linda 12345

bob 2222

alice 2323

simon 12345678

<pre> *****!Menu!***** 1--Insert 2--Search 3--List 4--Quit ***** Select:1  1--Input name:linda 2--Input telephone:12345  *****!Menu!***** 1--Insert 2--Search 3--List 4--Quit ***** Select:1  1--Input name:bob 2--Input telephone:2222 </pre>	<pre> *****!Menu!***** 1--Insert 2--Search 3--List 4--Quit ***** Select:1  1--Input name:alice 2--Input telephone:2323  *****!Menu!***** 1--Insert 2--Search 3--List 4--Quit ***** Select:1  1--Input name:simon 2--Input telephone:12345678 </pre>
--	---

显示当前电话号码表如下：

```

--Name-----Telephone--
linda          12345
bob            2222
alice          2323
simon          12345678
-----

```

接着在刚才建立好的电话号码表中进行查找。

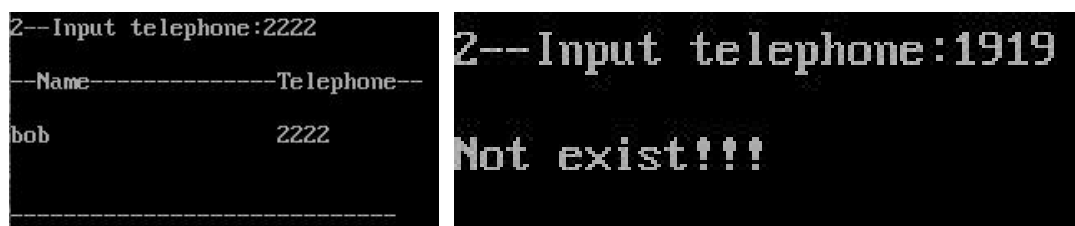
按姓名查找：输入 linda，输出显示找到该记录。

按姓名查找：输入 lili，没有该条记录，输出不存在。

<pre> 1--Input name:linda  --Name-----Telephone-- linda          12345 ----- </pre>	<pre> 1--Input name:lili  Not exist!!! </pre>
---	---

按号码查找：输入 2222，输出显示找到该记录。

按号码查找：输入 1919，没有该条记录，输出不存在。



```
2--Input telephone:2222
--Name-----Telephone--
bob                2222
-----

2--Input telephone:1919
Not exist!!!
```

### 三、实验体会

如果说前几次的实验只是实践部分的知识，那这次的实验可以算是将前面学到的各个知识点贯穿起来的结合实验。

在这次的实验中，我第一次知道了什么是宏操作，也见识到了宏操作的方便性，能够将冗余的重复多次的相同操作写在最前面，之后再代码段使用该功能的时候就可以直接用该操作，只要 1 行就可以避免每次需要使用该操作的时候重复书写相同的代码段，非常快捷便利。

除此之外，我也学会了如何用汇编语言写菜单这种可视化界面。通过菜单这种操作，能够将有很多功能的程序按功能分解，不仅便于书写代码、检错纠错，也更加方便了用户。通过很多的比较指令和条件跳转指令就可以达成这一点。

这次实验让我收获颇丰，但是也让我大受挫败。我尝试了很多我能想到的方法去保存输入的姓名和号码，通过将串结束符保留以调用串输出指令。但是这些都出现了或大或小的问题。最后我求助于网上的代码，才顿悟可以设置一个固定的存储空间，然后通过指针的指向的改变，来完成不同条数据的记录和输出。因而才顺利完成了本次实验。

但是这个代码仍然存在着很多我还尚不能解决的问题，比如：程序限制输入的号码位数为 8 位，但是每一行的限制是 32 个字节，所以在号码的 8 个字节的后面其实还有 4 个空字节。所以在输入测试的时候，我可以将号码输入超过 8 位，但是此时不能回车结束操作。此时删除多出 8 位的数字可以再回车，但是留下的问题是之前超过的位数仍然留下了几位，与符合规定的电话号码直接以空格相隔开。令人欣慰的是，这个问题并不影响查找，即使只输入符合规定的号码位数，也能输出正确的结果。我觉得是号码后面的空白字节保留了缓冲区的内容没有清空（代码段定义的是清空 20 和 8 个字节），所以才会有显示。但是查找的时



候设计的代码是 8 位结束比较，所以查找并不会出错。

实验代码如下：

```
001 data segment
002     count dw 0;记录有几行人名-号码
003     temp dw 0;输入寄存
004     telbook db 50 dup(32 dup(' '));设置一个50*32的号码簿
005     window1 db 13,10,13,10, '*****|Menu|*****',13,10;菜单提示
006             db '1--Insert' , 13,10
007             db '2--Search' , 13,10
008             db '3--List' ,13,10
009             db '4--Quit' ,13,10
010             db '*****' ,13,10
011             db 'Select:', '$'
012     window2 db 13,10,13,10, '*****|Search|*****',13,10;查找菜单提示
013             db '1--Name' , 13,10
014             db '2--Telephone' , 13,10
015             db '3--Quit' ,13,10
016             db '*****' ,13,10
017             db 'Select:', '$'
018     str1 db 13,10,13,10, '1--Input name:', '$'
019     str2 db 13,10, '2--Input telephone:', '$'
020     str3 db 13,10,13,10, '1--Name-----Telephone--',13,10,13,10, '$'
021     str4 db 13,10,13,10, 'Not exist!!!' , 13,10, '$'
022     str5 db 13,10,13,10, '-----',13,10,13,10, '$'
023     str6 db 13,10,13,10, 'No record!!!' , 13,10, '$'
024     error db 13,10,13,10, 'Error!!!You should input 1-4 !',13,10, '$'
025     error1 db 13,10,13,10, 'Error!!!You should input 1-3 !',13,10, '$'
026     N_Temp db 21,?;缓冲区最大容量21
027     N_T db 20 dup (' ');姓名, 20个可输入空间, 以及最后的分隔空间
028     T_Temp db 9,?;缓冲区最大容量9
029     T_T db 8 dup (' '), '$';电话号码, 8个可输入空间, 最后分隔
030 data ends
031
032 ;宏操作--操作名 macro 操作数, 调用时--操作名 操作数
033 print macro x1;输出字符串
034     lea dx,x1
035     mov ah,9
036     int 21h
037 endm
038
039 input macro x1;键盘输入到缓冲区
040     lea dx,x1
041     mov ah,0ah
042     int 21h
043 endm
044
045 quhc macro Temp,T ;去除回车符, 即把Temp的内容复制一份到T的时候, 不复制Temp的最后的回车符
046     sub dx,dx
047     mov di,Temp+1 ;Temp中的字符数
048     lea si,T ;T的偏移地址
049     add si,dx ;T转移到Temp结束位置
050     mov byte ptr[si], ' ';把本来Temp的回车位置的回车替换成空格
051 endm
052
053 renew macro nn,cnt;还原缓冲区内内容, 重新置为空白内容再输入
054     local next
055     mov cx,cnt ;置为空格的位数, inc命令自动与cx作比较, 到了自动结束loop
056     lea si,nn ;还原nn的内容, 将其置为空白内容
057     next:mov byte ptr[si], ' '
058     inc si
059     loop next
060 endm
061
062 code segment
063     assume cs:code,ds:data
064 start:
065     mov ax,data
066     mov ds,ax
```



```

067     mov es,ax
068     lea di,telbook;di设为号码簿的偏移地址
069
070     ;主菜单功能选择
071     L1:print window1
072     mov ah,1
073     int 21h
074     cmp al,'1';如果选择1号功能,则跳转到输入Insert子程序
075     jne L2     ;否则进行下一个判断
076     call Insert
077     jmp L1
078     L2:cmp al,'2';如果选择2号功能,则跳转到查找Search子程序
079     jne L3     ;否则进行下一个判断
080     call Search
081     jmp L1
082     L3:cmp al,'3';如果选择3号功能,则跳转到显示List子程序
083     jne L4     ;否则进行下一个判断
084     call List
085     jmp L1
086     L4:cmp al,'4';如果选择4号功能,则跳出DOS, 程序结束
087     je L5
088     print error;否则选择有误, 输出提示语句再重新输入
089     jmp L1
090
091     Insert proc;输入子程序
092     renew N_T,20     ;还原缓冲区内容
093     renew T_T,8
094     print str1
095     input N_Temp     ;输入姓名存在暂存缓冲区, 带回回车符结束
096     quhc N_Temp,N_T  ;去除回车符保存
097     print str2
098     input T_Temp     ;输入电话号码暂存缓冲区, 带回回车符结束
099     quhc T_Temp,T_T  ;去除回车符保存
100
100     lea si,N_T       ;将姓名的偏移地址保存在si
101     mov cx,20
102     cld
103     rep movsb        ;将si中的20个字节的姓名内容按字节传送到di, 即50*32的号码簿中
104     lea si,T_T
105     mov cx,12
106     cld
107     rep movsb        ;将si中的12个字节的号码内容按字节传送到di, 即50*32的号码簿中
108     inc count        ;count记录有几条记录
109     ret              ;返回主程序
110 Insert endp
111
112     Search proc;查找子程序
113     ;-----按姓名查找-----;
114     T0:print window2
115     mov ah,1
116     int 21h
117     cmp al,'1'       ;按姓名查找
118     jne T1           ;否则进入下一个判断
119     print str1
120     renew N_Temp,20  ;清空姓名缓冲区
121     input N_Temp     ;输入姓名, 保存在缓冲区
122     quhc N_Temp,N_T  ;将输入的姓名去除回车符, 保存
123     ;*****按姓名查找*****;
124     mov cx,count     ;将号码簿中存在的记录数保存到cx中
125     mov Temp,offset telbook ;用Temp记录号码簿的偏移地址
126     F3:push cx       ;cx入栈保存, 即所有的记录数量
127     mov si,Temp      ;si记录号码簿的偏移地址
128     lea bx,N_T       ;将刚刚输入的姓名的偏移地址保存在bx中
129     mov cx,20        ;cx中保存姓名的最大值20
130     F2:mov d1,[si]   ;号码簿记录保存在d1
131     cmp d1,[bx]      ;号码簿中的记录与刚刚输入的姓名作比较
132     jne F1           ;不相等跳到F1, 移动号码簿的开始位置

```

```

133     inc si
134     inc bx           ;否则都向后移一位继续比较下一个字节
135     loop F2          ;cx初始值为20, 每次loop操作减一, 到0结束, 表示该姓名结束比较
136 ;*****cx为0, 该姓名20个字节全部比较完成, 且没有跳到JNE, 所以匹配, 输出找到的记录*****;
137     print str3       ;找到, 输出记录
138     mov dx,Temp      ;将此条记录输出
139     mov ah,9
140     int 21h
141     print str5
142     pop cx           ;cx出栈, 变回号码簿的记录条数
143     jmp F0           ;查找结束, 找到, 跳回主程序
144 ;*****该名字不匹配, 跳到下一个名字开始的地方。直到号码簿中所有记录比完, 输出没找到*****;
145     F1:add Temp,32   ;跳32个字节到下一个姓名开始的位置
146     pop cx
147     loop F3          ;重复查找操作, 直到cx为0时结束, 注意此时的cx是出栈后的cx, 即号码簿记录的条数
148     print str4       ;两个loop操作都结束时, 表示号码簿中的记录都比较完成, 没有找到
149     jmp F0           ;查找结束, 没找到, 跳到主程序
150
151 ;-----按号码查找-----;
152     T1:cmp al,'2'
153     jne T2           ;否则进入下一个判断
154     print str2
155     renew T_Temp,8   ;清空号码缓冲区
156     input T_Temp
157     quhc T_Temp,T_T ;将输入到缓冲区的号码去除回车符保存
158 ;-----按号码查找(具体操作与按姓名相同)-----;
159     mov cx,count
160     mov Temp,offset telbook
161     K3:push cx
162     mov si,Temp
163     add si,20        ;将si的位置移到号码开始的位置, 因为名字占20个字节, 所以加上20
164     lea bx,T_T
165     mov cx,8         ;号码占8个字节
166     K2:mov dl,[si]
167     cmp dl,[bx]
168     jne K1           ;不相等就跳到下一个号码开始的位置, 重新开始比较
169     inc si
170     inc bx
171     loop K2
172 ;-----找到号码, 匹配, 输出该条记录-----;
173     print str3
174     mov dx,Temp
175     mov ah,9
176     int 21h
177     print str5
178     pop cx
179     jmp F0
180 ;-----跳到下一个号码, 直到所有记录都被比过-----;
181     K1:add Temp,32
182     pop cx
183     loop K3
184     print str4       ;所有记录没有找到, 输出不存在
185     jmp F0
186     T2:cmp al,'3'   ;返回主程序重新选择功能
187     je F0
188     print error1    ;以上都不符合, 则输入的数字有问题, 输出提示语句并重新跳回查找菜单
189     jmp T0
190     F0:ret
191 Search endp
192
193 List proc;显示号码簿子程序
194     print str3
195     cmp count,0
196     je Z2           ;如果count为0, 则表示号码簿里没有记录
197     mov cx,count     ;否则将要输入的记录的条数保存到cx, 控制loop循环次数
198     lea si,telbook   ;将号码簿偏移地址保存到si
199     Z1:mov dx,si
200     mov ah,9
201     int 21h
202     add si,32        ;加上32, 跳到下一条记录重复输出操作
203     loop Z1
204     jmp Z3
205     Z2:print str6
206     Z3:print str5
207     ret
208 List endp
209
210 L5:mov ah,4ch       ;跳出DOS, 程序结束
211     int 21h
212
213 code ends
214 end start

```