

山东大学 计算机科学与技术 学院

汇编语言 课程实验报告

学号：202200130053	姓名：陈红瑞	班级：3 班
实验题目：实验 11：实验 2.4		
实验学时：2	实验日期：20241223	
实验目的： 巩固子程序设计中所涉及的知识点。学会在自编程序中利用子程序设计的理论与技巧。学会在自编程序中使用系统调用和简单的表格数据结构。		
实验环境：Windows11、DOSBox-0.74、Masm64		
源程序清单： lab11.asm		
编译及运行结果： 下面是对数据段的定义。首先需要将所有的需要使用到的字符串存入到数据段，然后定义缓冲区变量，分别存储接收到的电话及姓名的输入。然后定义两个变量，分别用于存储电话和姓名的信息以及排序时临时存储的信息。最后，定义一个临时变量用于将特定的行进行填充。		

```

;PROGRAM TITLE GOES HERE--PHONE
dataarea segment
message1 db 'Input name:', '$'
message2 db 'Input a telephone number:', '$'
message3 db 'Do you want a telephone number?(Y/N)', '$'
message4 db 'name?', '$'
message5 db 'name', 16 dup(' '), 'tel.', 0ah, 0dh, '$'
message6 db 'Not Find!', 0ah, 0dh, '$'
num dw 0;用来标识电话表中数据个数
phonein label byte
    pmax db 9
    pact db ?
    phone db 9 dup(?)
namein label byte
    nmax db 21
    nact db ?
    nam db 21 dup(?)
temp_nam db 20 dup(?);临时单元
temp_tab db 20 dup(?), 8 dup(?), 13, 10, '$'
;排序用, 临时储存tel_tab中一项
tel_tab db 50 dup(20 dup(?), 8 dup(?), 13, 10, '$')
;每一项31 (20+8+4+3) 字节
tmp_line db 28 dup(' '), 13, 10, '$'

```

在代码段中，下面通过一个循环处理输入。这里使用 INT 21H 中的功能 01H 进行对单个字符的输入，用于表示输入的数据的数量，这里会先判断是否为换行符，如果是，就结束输入，并跳转到输入信息的部分，否则就将输入的 ASCII 值转换为数字，并将 CX 中原来的数字乘 10 后，将输入的数加到 CX 中。

```

cxin:
    mov ah, 01h
    int 21h
    cmp al, 0dh
    je begin
    xor al, 30h
    mov dl, al
    mov al, 10
    mul cl
    mov cl, al
    add cl, dl
    jmp cxin

```

下面执行对名字和号码的输入。首先调用功能 09H 实现将提示输入名字的字符串，然后将临时存储信息的部分清空。然后再调用过程实现对

名字的输入，调用的过程中实现了使用 INT 21H 中的功能 0AH 实现将字符串输入到缓冲区，并输出换行。

```
begin:
    push cx
    lea dx,message1;输入名字
    mov ah,09h
    int 21h
    call clear;清空临时单元temp_nam和tel_tab中的内容;
    call input_name
    cmp nact,0
    je search;如果没有输入名字，表明输入结束，开始排序、查询
    call stor_name
    lea dx,message2;输入电话号码
    mov ah,09h
    int 21h
    call inphone
    call name_sort
    pop cx
    loop begin;循环输入
```

```
;-----
input_name proc near
    lea dx,namein
    mov ah,0ah
    int 21h
    call crlf
    ret
input_name endp
```

下面再将输入的名字保存到表中，这里按照缓冲区的字符数量，使用 movsb 指令，将名字部分保存到临时变量中。

```
stor_name proc near;
    cmp nact,0
    je exit1
    lea si,nam
    lea di,temp_tab
    sub ch,ch
    mov cl,nact
    cld
    rep movsb
exit1:
    ret
stor_name endp
```

下面实现的是对电话进行输入的过程，这里同样需要将电话信息存储到临时变量中，但这里需要将临时存储区偏移 20 个字符，确保输出和比

较的时候格式正确。

```
inphone proc near
    lea dx,phonein
    mov ah,0ah
    int 21h
    cmp pact,0
    je exit2
    call crlf
    lea si,phone
    lea di,temp_tab
    add di,20
    sub ch,ch
    mov cl,pact
    cld
    rep movsb
exit2:
    ret
```

下面执行的是排序过程。如图，这里需要先判断表是否为空，如果是空的，可以直接将临时存储区的信息存入到表中，否则就继续执行排序过程，这里会使用 `cmpsb` 指令将名字部分按字符从左到右比较，如果临时区的字符大于表中当前行的字符，就将下一组内容进行比较，如果找到了插入位置，就调用过程实现插入操作。

```

name_sort proc near
    cmp num,0
    jnz sort
    lea si,temp_tab
    lea di,tel_tab
    mov cx,31
    cld
    repz movsb
    jmp exit3
sort:
    mov cx,num
    lea si,temp_tab
    lea di,tel_tab
loopsort:
    push di
    push cx
    mov cx,20
    repz cmpsb
    ja move
    pop cx
    pop di
    call insert
    jmp exit3
move:
    pop cx
    pop di
    add di,31
    lea si,temp_tab
    loop loopsort
    mov cx,31
    rep movsb
exit3:
    inc num
    ret
name_sort endp

```

下面是插入数据的实现。这里先从内存中获取表中的所有数据的个数，然后将其乘 31，表示表中已经存储的所有的字符数量。然后从最后一组数据开始，依次将数据向后移动 31 个字符，最后将临时变量中的数据插入到表中空出的位置。

```

insert proc near
    mov ax,num
loopinsert:
    push ax
    mov bx,31
    mul bx
    lea di,tel_tab
    add di,ax
    mov si,di
    sub si,31
    push cx
    mov cx,31
    rep movsb
    pop cx
    pop ax
    dec ax
    loop loopinsert

    lea si,temp_tab
    lea di,tel_tab
    mov bx,31
    mul bx
    add di,ax
    mov cx,31
    rep movsb
    ret
insert endp

```

下面进行查询，这里首先需要输入一个字符，如果是 N，就跳转到程序返回 DOS 的部分，否则调用输入名字的过程，通前面插入时输入名字部分的代码，然后调用过程实现对名字进行查找。

```

search:
    mov ah,09
    lea dx,message3
    int 21h
    mov ah,01h
    int 21h
    cmp al,'N'
    je exit
    call crlf
    mov ah,09
    lea dx,message4
    int 21h
    call input_name
    cmp nact,0
    je exit
    call name_search
    call printline
    jmp search

```

下面实现的是对名字进行查找的过程，这里先将缓冲区输入的名字存

入到临时变量中，然后执行循环，从表中依次比较这个字符串，如果找到，就按照当前的 CX，计算 num-CX, num 为存储数据组数的变量，并得到要输出的信息所在的位置。

```
name_search proc near
    call clear
    mov ch,0
    mov cl,nact
    lea si,nam
    lea di,temp_nam
    rep movsb
    mov cx,num
    lea di,tel_tab
    lea si,temp_nam
loopfind:
    push di
    push cx
    mov ch,0
    mov cl,20
    repz cmpsb
    je found
    pop cx
    pop di
    add di,31
    lea si,temp_nam
    loop loopfind
    mov cx,0
    jmp exit4
found:
    pop cx
    pop di
exit4:
    ret
```

找到要输出的位置后，就需要将信息进行输出。由于表中没有使用空格进行填充，这里实现的思路是将要输出的字符串存入到另一个临时变量中，然后使用空格填充空白位置，在将临时变量中的字符串输出。

```

lea dx,message5
mov ah,09
int 21h
mov ax,num
sub ax,cx
mov bx,31
mul bx           ;(n
lea dx,tel_tab
add dx,ax

mov si,dx
lea di,tmp_line
mov cx,31

cld
rep movsb

mov cx,28
mov bx,0
;用空格填充
t1:
cmp tmp_line[bx],0
jne to_loop
mov tmp_line[bx],'
to_loop:
inc bx
loop t1

lea dx,tmp_line
mov ah,09
int 21h

```

最后是退出程序时，需要先输出所有表中的信息，这里同样需要依次将每一组数据存入到一个临时变量中，然后使用空格填充空白位置，再使用功能 09H 进行输出。


```

put:
    mov si,dx
    lea di,tmp_line
    push cx
    mov cx,28

    cld
    rep movsb
    pop cx
    xor bx,bx
    ;用空格填充, 然后输出
cmp1:
    cmp tmp_line[bx],0
    jne next1
    mov tmp_line[bx],' '
next1:
    inc bx
    cmp bx,28
    jne cmp1
    push dx

    mov ah,09h

```

下面是 0J 的运行结果。

提交详情(汇编实验11)

×

✓ 代码

>

✓ 运行

>

3 总结

概览

评测信息

结论	测试通过
编号	91b96ca92c20e11
提交时间	2024-12-15 23:58:11
评测时间	2024-12-15 23:58:14
评测模板	汇编语言11

关联信息

提交用户	202200130053 (ID: 10996)
题目信息	汇编实验11(4)

关闭

问题及收获：

1. 这个实验掌握了通过多个过程来实现一个输入输出以及插入数据等复杂操作。这里需要注意的是输出数据的格式，这里还需要注意将信息存储到表中时，需要确保占用的字节数是正确的，这里设置为 31 个字符，直接输出会包含换行和结束符，可以确保结果正确。由于 OJ 中还需要用空格填充，因此这里需要一个额外的过程来实现将输出的部分单独放在一个变量中，并将空字符改为空格填充。