

# 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 1

实验名称	: _用表格形式显示字符
专业: _	计算机科学与技术专业
姓名:_	<b>汪雨卿</b>
<b>学무</b> .	10120101

## 

### 一、实验目的

通过循环控制编程方式用表格和不同形状的形式显示 ASCII 字符表。

## 二、实验内容

#### (1) 实验原理

按 15 行×16 列的表格形式显示 ASCII 码为 10H—100H 的所有字符,即以行为主的顺序。及 ASCII 码递增的次序显示对应的字符。每 16 个字符为一行,每行中的相邻两个字符之间。用空白符(ASCII 为 0)隔开。

1. ASCII 字符输出:

ASCII 字符的输出主要调用 dos 的 02 回显功能。通过将被输出字符的 ASCII 码传给 d1,将 ah 赋值为 2,并且利用 int 21h 调用 dos 命令窗口,实现字符的输出。

2. 循环控制:

汇编指令的循环控制是利用标志位的状态字结合跳转指令,以及对于循环计数器的加减进行控制的。首先设置,一个循环计数器为它赋值,之后每次对它进行减数操作,判断进位标志,为0时跳转,不为0时继续执行。对于每行的结尾,输出回车,实现循环换行的工能。

- 3. 对于特殊形状的表单输出:
- I. 三角形的输出:

控制每行输出的个数是递增的。设置两个循环标志位,对于控制每行输出个数的标志位,在每次进入循环的时候,进行赋值,且赋值内容是上一次加1.

II. 平行四边形的输出:

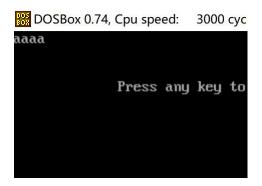
控制每次换行之后输出的空格数依次为 1, 2, 3, ······以此类推。依采用两个循环标志位,类似于三角形的输出,控制空格数的输出。

4. 利用了 cmp 指令以及 ja 进行跳转判断

#### (2) 实验步骤

- 1. 文件执行
  - 1.1 打开 masm 程序窗口,新建代码文件。
  - 1.2 输入需要执行的代码。
  - 1.3 保存所需要的代码。
  - 1.4 运行执行代码文件。
- 2. 代码编写的步骤
  - 2.1 尝试利用循环输出连续的 ASCII 码字符
  - 2.2 尝试单行输出整张 ASCII 码。
  - 2.3 尝试输出 ASCII 码表格
  - 2.4 尝试输出三角形的 ASCII 码表格
  - 2.5 尝试输出平行四边形的 ASCII 码

- (3) 实验记录
- 1. 输出连续 4 个 a



2. 输出单行的 ASCII 对应表



3. 输出 ASCII 表格



4. 输出 ASCII 码的三角形表格

```
;输出的循环
     mov d1, b1
mov ah,2
                                   DOSBox 0.74, Cpu spe
     int 21h
                                   4‡
     add bl,1
                                   !!¶§
■±↑↓
     sub bh,1
     jnz skip
                                   →+*
▼ !"#$
     mov d1,10
     mov ah,2
                                   /&' ()*+
     int 21h
                                    ,-./0123
                                   456789:;<
=>?@ABCDEF
     add dh.1
                                   GHIJKLMNOPQ
     mov bh,dh
                                   RSTUWXYZI\1
^_`abcdefghij
klmnopqrstuwx
yz{|}~oGüéâäåå$
skip:
     sub ch.1
     jnz next
     MOV AH,4CH
     INT 21H
CODES ENDS
                                                    Press
     END START
```

5. 输出 ASCII 码的平行四边形表单

```
mov ch. 128
mov bl. 10h; ascII的起始值
mov bl. 10h :每行输出的个数 16个,用十六进制表示
mov dl. 1 : 空格数的基数
mov cl. 1
next:
;输出的循环
mov d1, b1
mov ah,2
int 21h
                                                                                                                                                                                    DOSBox 0.74, Cpu speed: 3000 c
                                                                                                                                                                                                | DOSBOW 0.74, Cpu speed: 30
| ITTS_ITI_NAT
| "ITS_XE ( ) **+, -, -,
| 01.23456789; :<->?
| 04.52456789; :<->?
| 04.52456789; :<->?
| 04.52456789; :<->?
| 04.52456789; :<->?
| 04.52456789; :<->?
| 04.5245679; :<->?
| 04.5245679; :<->?
| 04.5245679; :</>| 04.5245679; :</>| 04.5245679; :</>| 04.5245679; :</>| 04.5245679; :</>| 04.5245679; :</>| 04.5245679; :</>| 04.5245679; :</>| 04.5245679; :</>| 04.5245679; :</>| 04.5245679; :</>| 04.5245679; :</>| 04.5245679; :</>| 04.5245679; :</a>
                 add b1,1
sub bh,1
jnz skip;
mov d1,10
mov ah,2
int 21h
                                                                                                                                                                                                                                                   Press anu keu
                  ;
mov d1,13
mov ah,2
int 21h
int 27h
;
add dh,1
mov bh,10h
mov cl,dh
cnp cl,0h; 如果大于0, 则跳转
blank:
sub cl,1
mov dl, 20h
mov ah,2
int 27h
jnz blank
jz ext
skip:
```

(4) 数据处理 通过对于 128 个 ASCII 码对应字符的行列控制,实现了不同形状的表格 输出。

# 三、源程序清单

1. 输出4个a

- 2. 输出单行 ASCII 码表 3. 输出 ASCII 码表

```
START:

;push ds
;push ds
;vub ax, ax
;push ax de
;pu
CODES SEGMENT
ASSUME CS:CODES
START:
mov ch, 4
exit:
:输出的循环
mov dh,2
int 21h
Sub cb.1
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 ;<-----;
next:
;输出的循环
mov d1, b1
mov ah,2
int 21h
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     real enclosy

rintit;

mod di, bl. (输出bl

mod di, 20 (输出ch

fat 2th

mod di, 20 (输出ch

min 2th

ad bl. 1 (bl.1

sub ci. 1

sub ci. 1

call neuline 循环输出16个、输出操行

enclosy:

mod ah, 4ch: 以下两行类似于return

cooks fies

teb STRBT
                                                                              int 21n
sub ch,1
jnz exit
; 结束语句
MOV AH,4CH
INT 21H
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           ;<---add-
add bl,1
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             ado ba,
;<-----
sub ch,1
jnz next
MOU AH,4CH
INT 21H
CODES ENDS
END START
        CODES ENDS
END START
```

#### 4. 输出平行四边形时

```
CODES SEGMENT
   ASSUME CS:CODES
START:
   mov ch. 128
   mov bl, 10h ;ASCII的起始值
   mov bh, 10h ;每行输出的个数 16个,用十六进制表示
   mov dh, 1 ;空格数的基数
   mov cl, 1
next:
   ;输出的循环
   mov dl. bl
   mov ah,2
   int 21h
   add bl,1
   sub bh,1
   jnz skip
   mov dl,10
   mov ah,2
   int 21h
   mov ah,2
   int 21h
   add dh.1
   mov bh,10h
   mov cl,dh
   cmp cl,0h; 如果大于0, 则跳转
```

```
ja blank
blank:

sub cl,1

mov dl, 20h

mov ah,2

int 21h

jnz blank

jz next

skip:

sub ch,1

jnz next

MOV AH,4CH

INT 21H

CODES ENDS

END START
```

## 四、心得体会

通过这次汇编程序的上机实验,让我很好的熟悉了汇编语言程序编写的结构,通过实践更好了理解了各个寄存器的使用规范,以及不同指令的作用。从最开始的从书本上复刻代码进行运行,我对于汇编程序编写只是一个小白。不明白代码段落的作用,也不知道它的 jmp 指令跳转的发则,对于不同寄存器的位数和使用场景更是一头雾水。随着老师通过实验,对于代码每一步过程的讲解,我开始明白每一步指令的用途。从最开始的尝试输出一位 ASCII 码,到利用 ch 实现汇编的循环指令。再举一反三,实现控制每行输出,实现表格的打印。到最后,输出平行四边形时,我已经可以通过查看命令文档,自主学习心得命令和使用的方法,通过不断的错误调试,最终实现了自己的程序。整个循序渐进的学习过程,让我体验到了在实践中学习代码的乐趣,也让我打下了相对牢靠的汇编基础。

此外,在本次代码的编写过程中,我注意到了两方面的内容,尤其值得注意。其一, al 寄存器在某些情况下并不能作为普通的数据寄存器。由于误将 al 存入数据进行判断,导 致程序进入了死循环。其二,汇编程序的执行循序是顺序执行,如果没有跳转指令的帮助, 会一直执行到代码运行完成。因此每个段的函数顺序也会影响整个程序的运行结果。