

# 上海大学

Shanghai University

## 实验报告

### 用表格形式显示字符

课程名称： 汇编语言程序设计

姓名： 王小明

学院： 计算机工程与科学学院

专业： 计算机科学与技术

学号： 123456789

指导老师： 杨洪斌

2021 年 9 月 23 日

## 一、 实验目的

通过循环控制编程方式用表格形式显示 ASCII 字符表。

## 二、 实验要求

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。

## 三、 实验内容及思路

### 1. 实验内容

本次实验的主要内容为按 15 行每列 16 个 ASCII 码以表格形式显示这些 ASCII 码。

### 2. 实验思路

对于这个实验，我们可以将实验过程分为如下两大步骤：

- (1) 计算出 10H-100H 的值，并将其置入寄存器中
- (2) 将上述寄存器的值输出到屏幕上

## 四、 实验步骤

### 1. 编写汇编代码

#### (1) 计算 ASCII 码的值

由于我们最终要以 15 行每列 16 个 ASCII 码的表格形式程序实验结果，所以我们很容易就想到了用双重循环来计算 ASCII 码的值。外层循环为 15 次，用于计算行值；内层循环为 16 次，用于计算列值。需要注意的是，在内层循环中，每输出一个 ASCII 码值，就要输出一个空格（ASCII 码值为 0），在两个外层循环间还要输出一个换行和一个回车。

#### (2) 输出部分

DOS 系统 21 号中断的 2 号功能（ $AH = 02$ ）可以将一个字符显示输出，因此，我们只需将 21 号中断的 2 号功能封装成一个函数，每次输出字符时调用该函数即可。

### 2. 编译链接并执行

- (1) 打开 DOSBox，挂载相应目录
- (2) 键入 `masm ascii.asm`, 回车
- (3) 键入 `link assii.obj`, 回车
- (4) 键入 `assii.exe`, 回车

## 五、 实验代码及结果

### 1. 实验代码

实验代码如下：

```
1  codesg segment
2  main proc far
3      assume cs:codesg
4
5  start:
6      mov al, 10h
7      mov cx, 15
8  s1: mov bx, cx
9      mov cx, 16
10 s2:
11     mov dl, al
12     call prints
13     inc al
14     mov dl, 0
15     call prints
16     loop s2
17
18     mov dl, 0Ah
19     call prints
20     mov dl, 13
21     call prints
22     mov cx, bx
23     loop s1
24
25     mov ah, 4ch
26     int 21h
27
28 main endp
29
30 prints proc near
31     push ax
32     push bx
33     push cx
34     push dx
35     mov ah, 2
36     int 21h
37     pop dx
38     pop cx
39     pop bx
40     pop ax
41     ret
42 prints endp
43 codesg ends
44
```

45 end start

## 2. 实验结果

实验结果如图 1 所示。



图 1: ASCII 码为 10H-100H 的所有字符

## 六、 实验体会

本次实验最大的收获是对 call 和 ret 的原理有了更深入的理解。开始时我认为 call 和 ret 会自动对所有寄存器的值做压栈和出栈操作。但在本次实验中，我发现，如果在 call 一个函数之后系统不会对通用寄存器的值进行压栈操作，通用寄存器的值将在函数调用过程中将发生改变。进过查阅课本我发现，call 和 ret 在调用函数过程中，仅对程序在调用函数时的下一地址进行压栈出栈操作，这纠正了我之前的错误认识。