

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：202120130276	姓名：王云强	班级：21.2 班
实验题目：实验 2.1		
实验学时：2	实验日期：2023.12.12	
<p>实验目的：</p> <ol style="list-style-type: none">1、巩固循环与分支程序设计中涉及的知识点。2、学会在自编程序中利用循环程序设计的理论与技巧。3、学会在自编程序中使用系统调用。		
实验环境：Windows10、DOSBox-0.74、Masm64		
<p>源程序清单：</p> <ol style="list-style-type: none">1. EX10_1.ASM(实验 2.1 源程序)2. EX10_2.ASM(实验 2.1 修改后源程序)		
<p>编译及运行结果：</p> <p>EX10_1 编译结果：</p> <pre>C:\>masm EX10_1 Microsoft (R) Macro Assembler Version 5.00 Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved. Object filename [EX10_1.OBJ]: Source listing [NUL.LST]: Cross-reference [NUL.CRF]: Open procedures: MAIN 51746 + 464798 Bytes symbol space free 0 Warning Errors 0 Severe Errors C:\>link EX10_1 Microsoft (R) Overlay Linker Version 3.60 Copyright (C) Microsoft Corp 1983-1987. All rights reserved. Run File [EX10_1.EXE]: List File [NUL.MAP]: Libraries [LIB]: LINK : warning L4021: no stack segment</pre>		

运行结果：

[illegible]

EX10_2 编译结果:

```
C:\>masm EX10_2
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

Object filename [EX10_2.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:
Open procedures: MAIN

    51746 + 464798 Bytes symbol space free

    0 Warning Errors
    0 Severe Errors

C:\>link EX10_2

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

Run File [EX10_2.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment
```

运行结果：

```

C:\>EX10_2
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4
5 6 7 8 9 : ; < = > ? @ A B C D E F G H
I J K L M N O P Q R S T U V W X Y Z [ \
] ^ _ ` a b c d e f g h i j k l m n o p
q r s t u v w x y z { | } ~ ¢ £ ¤ ¥ ¦ § ¨
à á â ã ä å æ ç è é ê ë ì í î ï ð ñ
ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ
ı Ł ł Ɔ ɔ ɐ ɑ ɒ ɓ ɔ ɕ ɜ ʝ ʎ ɹ ʀ ʁ
ʂ ʃ ʄ ʅ ʆ ʇ ʈ ʉ ʊ ʋ ʌ ʍ ʎ ʏ ʐ ʑ
ʒ ʓ ʔ ʕ ʖ ʗ ʘ ʙ ʚ ʞ ʟ ʠ ʡ ʢ ʣ ʤ
ʥ ʦ ʧ ʨ ʩ ʪ ʫ ʬ ʭ ʮ ʯ ʰ ʱ ʲ ʳ ʴ
ʵ ʶ ʷ ʸ ʹ ʺ ʻ ʼ ʽ ʾ ʸ ʹ ʺ ʻ ʼ ʽ ʾ

```

问题及收获：

问题：

一、在本实验中如何实现的循环，在实现循环的时候需要注意到的哪些问题？

首先根据实验要求，设置外循环的次数是 15（这里使用一个变量 ROW 来实现，即给 ROW 初值设置为 15），内循环的次数是 16（使用 LOOP 指令来实现），然后用 DL 寄存器记录下起始值是 10H，之后每次循环我们输出 DL 的值即可。先是将 CX 赋值为 16，然后使用 LOOP 指令，循环 16 次，每次输出 DL 寄存器的值，同时还需将其的值加 1。当循环完 16 次以后，将 ROW 的值减 1，并判断是否为 0，如果不为 0 的话则重复上述过程，即可输出 15 行 16 列从 10H 开始的表格。在实现循环的时候要注意循环跳出条件，以及每次循环的时候，要进行相应的操作，汇编语言由于比较底层，所以在写循环的时候需要充分考虑清楚再写。

循环部分代码如下：

RESTART:	MOV	CX, 16	;每行循环次数
OUTPUT:	MOV	AH, 2	;输出字符
	INT	21H	
	PUSH	DX	
	MOV	DL, 0	;输出空格
	MOV	AH, 2	
	INT	21H	
	POP	DX	
	INC	DL	
	LOOP	OUTPUT	;循环输出每行的16个ASCII元素
NEXT:	PUSH	DX	;将下一行的首元素推入堆栈
	MOV	DL, 0DH	
	MOV	AH, 02H	
	INT	21H	
	MOV	AH, 02H	
	MOV	DL, 0AH	
	INT	21H	;输出换行
	POP	DX	;将栈顶元素推出
	DEC	ROW	
	CMP	ROW, 0	
	JNE	RESTART	
	JE	EXITING	

二、LOOP 循环和 CMP JMP 循环有什么相同之处和不同？

两种循环其实本质都是一样的，LOOP 内部实现就是每当执行到 LOOP 时，让 $CX = CX - 1$ ，如果 CX 为 0 了则继续向下执行，否则则跳转到 LOOP 后所跟着的程序段名称所在位置。

相同之处在于都是先判断再跳转，不同之处在于，LOOP 指定判断寄存器为 CX，而 CMP JMP 循环可以指定其他寄存器，而且 CMP JMP 的跳转可以有多种方式，比如 JA、JE 等等。但其实 LOOP 循环也可以根据 CX 的值，做到这样的。LOOP 循环的好处在于，代码更易懂，只需要根据 CX 的值进行判断就可以了，而且 CX 是多少，就会循环多少次。而 CMP JMP 循环的好处在于，可以使用其他寄存器，用起来可能更加灵活、多变（如

果对汇编语言熟练的话，用起来会更顺手，新手还是推荐用 LOOP 可能更好一点）

收获：

一、在本次实验中，练习了循环与分支程序，体会到在汇编程序中写循环与在 C 语言中的不同。在汇编程序中写循环，需要先考虑清楚循环结束条件，然后考虑使用哪种指令来判断循环的结束。

二、对 LOOP 指令和 CMP JMP 等指令有了更深刻地了解，通过在实际中使用这些指令，对实现循环以及考虑如何进行循环优化以提高程序的性能都有一定的帮助。