

**课程设计(综合实验)报告**

( 2019-- 2020年度第 2 学期)

名 称： 汇编语言课程设计

题 目：显示ASCII码、查找字符等

院 系： 计算机系

班 级： 软件1802

学 号： 201809000121

学生姓名： 王毅涛

指导教师： 刘书刚

设计周数： 一 周（分散进行）

成 绩：

日期：2020年 7月 5 日

**一、课程设计(综合实验)的目的与要求**

通过本门课程的学习与实践，学生可以加深对汇编语言程序设计课程的理解与掌握，有助于提高学生的汇编语言程序设计能力，同时可以加深对计算机工作原理的理解，有助于促进后续课程的学习。

本课程要求学生针对课本所学知识从调试程序、循环程序设计、分支程序设计、子程序设计、键盘输入程序设计和中断练习程序等部分学习实践，使学生对汇编语言有更深入细致和全面的了解，并能用所学知识解决实际问题，每个实验的具体要求如下：

各个实验要求有设计说明，对有实现要求的程序画出程序流程图，然后根据流程图编写程序，并上机调试写出运行结果。

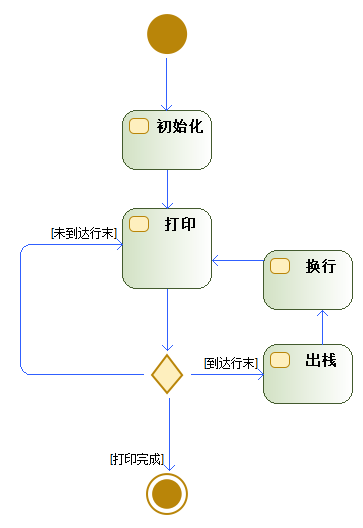
**二、设计（实验）正文**

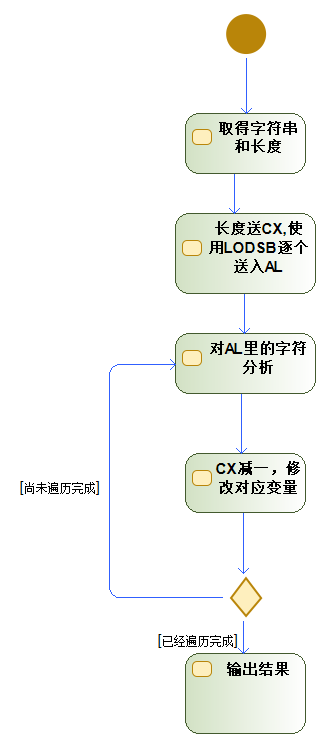
（1）设计思路

用表格形式显示ASCII字符SMASCII

按15行X16列的表格显示表示ASCII码为10H—100H的所有字符，即以行为主的顺序及ASCII码递增的次序一次显示对应的字符。每16个字符为一行，每行中的相邻的字符之间用空白（ASCII为0）隔开。显示空白符时，用其ASCII码置入dl寄存器。每行结束时，用显示回车（ASCII为0dh）和换行符（ASCII为0ah）来结束本行并开始下一行。

（2）程序框图（画出程序主要部分流程图）





（3）程序实现（注意：不用写出整体程序，写出部分关键代码即可，代码后要求用文字描述代码完成的功能）

第一部分：

1. data segment
2. ; add your data here!
3. pkey db "press any key...$"
4. VAR1 DB 00H
5. ends
7. stack segment
8. dw   128  dup(0)
9. ends

12. code segment
13. start:
14. MOV DX,0d0aH
15. PUSH DX ;PUSH IN
16. MOV DL,10H
17. MOV BX,15;行数
18. OUTER:
19. MOV CX,16; 列数
20. ;MOV DL,10H
21. A:
22. MOV AH,02H
23. **INT** 21H
24. INC DL
25. LOOP A
26. ;换行
27. MOV VAR1,DL ;暂存DL
28. pop dx
29. **int** 21h
30. push cx
31. mov cl,8
32. ror dx,cl
33. **int** 21h
34. ror dx,cl
35. pop cx
36. push dx
37. MOV DL,VAR1;暂存DL
38. ;换行
39. DEC BX
40. CMP BX,0
41. JNE OUTER
43. ends
44. end start ; set entry point and stop the assembler.

第二部分：

1. ; multi-segment executable file **template**.
3. data segment
4. ; add your data here!
5. buf 80,?,80 DUP(?)
6. letter db 0
7. digit db 0
8. other db 0
9. ten db 10
10. lstr db "char number==$"
11. dstr db "digit number==$"
12. ostr db "others number==$"
13. pkey db "press any key...$"
14. ends
16. stack segment
17. dw   128  dup(0)
18. ends
20. code segment
21. start:
22. ; set segment registers:
23. mov ax, data
24. mov ds, ax
25. mov es, ax
27. mov cx,0
29. mov dx,offset buf
30. mov ah,0ah
31. **int** 21h
33. mov cl,buf[1]
34. mov si,(offset buf)+2
35. loops:
36. lodsb
37. CMP AL,30H
38. JB COTHER
39. CMP AL,3AH
40. JB CDIGIT
41. CMP AL,41H
42. JB COTHER
43. CMP AL,5AH
44. JBE CLETTER
45. CMP AL,61H
46. JB COTHER
47. CMP AL,7AH
48. JBE CLETTER
49. CLETTER:
50. inc letter
51. jmp HUB
52. COTHER:
53. INC other
54. jmp HUB
55. CDIGIT:
56. inc digit
57. jmp HUB
58. HUB:
59. DEC CX
60. JZ OUTPUT
61. jmp loops

64. OUTPUT:
65. ;NOW HOW TO PRINT THESE CHARS?
66. CALL RELINE
67. lea dx, lstr
68. mov ah,9
69. **int** 21h;×ÖÄ¸Ç°ÖÃ½éÉÜ
70. mov dl,letter
71. MOV AL,DL
72. MOV AH,0
73. CALL PRINT10;´òÓ¡×ÖÄ¸¸öÊý
74. call RELINE
75. lea dx,dstr
76. mov ah,9
77. **int** 21h;´òÓ¡Êý×Ö½éÉÜ
78. mov dl,digit
79. MOV AL,DL
80. MOV AH,0
81. CALL PRINT10;´òÓ¡Êý×Ö¸öÊý
82. call RELINE
83. lea dx,ostr
84. mov ah,9
85. **int** 21h;´òÓ¡ÆäËû½éÉÜ
86. mov dl,other
87. MOV AL,DL
88. MOV AH,0
89. call PRINT10;´òÓ¡ÆäËû¸öÊý
90. CALL RELINE
91. CALL TERMINATE

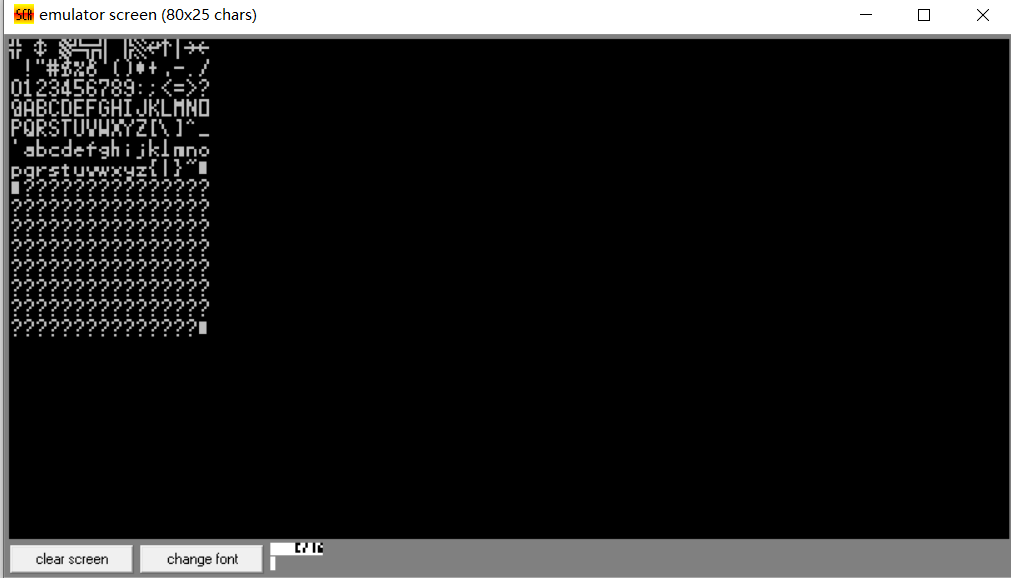
94. RELINE:
95. mov ah,02h
96. mov dl,0dh
97. **int** 21h
98. mov ah,02h
99. mov dl,0ah
100. **int** 21h
101. RET
103. PRINT10:
104. DIV ten
105. CALL PRINTAH
106. CMP AL,0
107. Jne PRINT10
108. RET
110. PRINTAH:
111. PUSH AX
112. PUSH DX
113. MOV DL,AH
114. add dl,48
115. MOV AH,2
116. **INT** 21H
117. POP DX
118. POP AX
119. RET

122. TERMINATE:
123. lea dx, pkey
124. mov ah, 9
125. **int** 21h        ; output string at ds:dx
127. ; wait **for** any key....
128. mov ah, 1
129. **int** 21h
131. mov ax, 4c00h ; exit to operating system.
132. **int** 21h
133. RET

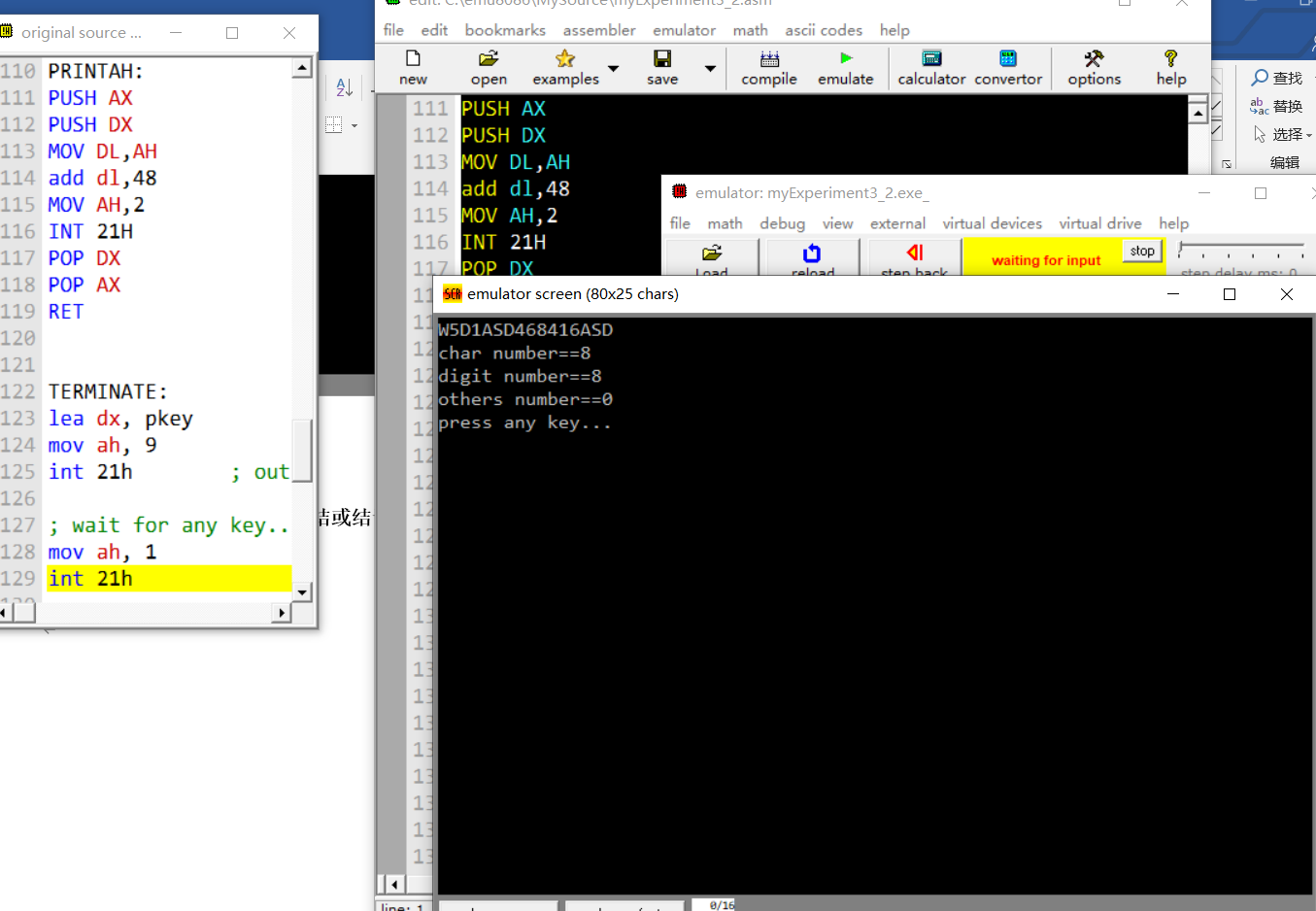
136. ends
138. end start ; set entry point and stop the assembler.

（4）实验结果（实验的运行结果）

第一部分：



第二部分：



**三、课程设计（综合实验）总结或结论**

本次实验是考察具体的串和ASCII码相关操作，当然涉及到了条件跳转等知识。第一个实验部分比较简单，但是由于字体原因，码表没有显示完全，第二个实验部分比较困难，困难在于如何处理串，如何分割串，如何统计，如何输出，以及把放在寄存器里的数字用ASCII转换，还有最重要的，十进制数字输出。我采用的是比较通用的，也就是引入除法，逐个送打印。

**四、参考文献**

《IBM-PC汇编语言程序设计（沉美明第二版）》