汇编语言课程设计攻略

1. 代码部分

写在开篇：

1. 攻略的目的是为更好地帮助大家开展课程设计，在一定程度上辅助老师开展课设实验，非老师、非官方资料，请勿滥用，请勿肆意传播。一切以老师的教学要求和实验任务为准。
2. 详见个人博客，网址如下。有代码、有思路、有流程图，不建议拷贝，可以根据思路自学或者按需求借鉴。尤其不建议写实验报告的时候照搬我的流程图，建议自己画一遍流程图，对写程序大有裨益，而且可以加深印象，我们考试的时候考过画流程图
3. 如有疏漏在所难免，各位的批评指正我都接受，但是改不改随缘。
4. ASCII字符按表输出<https://blog.csdn.net/DafengYY/article/details/106626455?spm=1001.2014.3001.5501>
5. 分类统计字符个数：

<https://blog.csdn.net/DafengYY/article/details/106626620>

1. 句子中关键字查找与匹配：

<https://blog.csdn.net/DafengYY/article/details/106627329>

1. 电话本（借鉴自17级某凡神）：

DATAS SEGMENT

;此处输入数据段代码

MESSAGE1 DB 'Please input name:$'

MESSAGE2 DB 'Please input telephone:$'

MESSAGE3 DB 'Check or Search?(c/s):$'

MESSAGE4 DB 'Not found!$'

MESSAGE5 DB 'Found TEL:$'

INFO1 DB 'Name:$'

INFO2 DB 'TEL :$'

NAME1 DB 20 DUP(?)

TEL1 DB 20 DUP(?)

NAME2 DB 20 DUP(?)

TEL2 DB 20 DUP(?)

NAME3 DB 20 DUP(?)

TEL3 DB 20 DUP(?)

NAME4 DB 20 DUP(?)

STRING DB 20 DUP(?)

FLAG DW 0

DATAS ENDS

STACKS SEGMENT

;此处输入堆栈段代码

STACKS ENDS

CODES SEGMENT

ASSUME CS:CODES,DS:DATAS,SS:STACKS

START:

MOV AX,DATAS

MOV DS,AX

;此处输入代码段代码

CALL INPUT\_NAME1

CALL INPUT\_TEL1

CALL INPUT\_NAME2

CALL INPUT\_TEL2

CALL INPUT\_NAME3

CALL INPUT\_TEL3

SELECT:

LEA DX,MESSAGE3

MOV AH,09H

INT 21H

MOV AH,01H

INT 21H

CMP AL,'s'

JE SEARCH

JMP CHECK

SEARCH:

MOV FLAG,0

CALL ENDLINE

LEA DX,MESSAGE1

MOV AH,09H

INT 21H

MOV BX,0

REINPUT\_STRING:

MOV AH,01H

INT 21H

CMP AL,0DH

JE INPUT\_STRING\_END

MOV STRING[BX],AL

INC BX

JMP REINPUT\_STRING

INPUT\_STRING\_END:

MOV STRING[BX+1],'$'

LEA BX,NAME1

CALL FUNC\_COMPARE

CMP FLAG,1

JE S\_1

LEA BX,NAME2

CALL FUNC\_COMPARE

CMP FLAG,1

JE S\_2

LEA BX,NAME3

CALL FUNC\_COMPARE

CMP FLAG,1

JE S\_3

LEA DX,MESSAGE4

MOV AH,09H

INT 21H

CALL ENDLINE

JMP SELECT

S\_1:

CALL OUTPUT\_MESSAGE\_5

LEA DX,TEL1

MOV AH,09H

INT 21H

CALL ENDLINE

JMP SELECT

S\_2:

CALL OUTPUT\_MESSAGE\_5

LEA DX,TEL2

MOV AH,09H

INT 21H

CALL ENDLINE

JMP SELECT

S\_3:

CALL OUTPUT\_MESSAGE\_5

LEA DX,TEL3

MOV AH,09H

INT 21H

CALL ENDLINE

JMP SELECT

CHECK:

CALL ENDLINE

CALL FUNC\_CHECK

CALL ENDLINE

JMP SELECT

MOV AH,4CH

INT 21H

FUNC\_COMPARE PROC NEAR

MOV DI,0

C\_START:

MOV AL,[BX+DI]

CMP STRING[DI],AL

JE C\_CHECK

RET

C\_CHECK:

CMP AL,'$'

JNE C\_CONTINUE

MOV FLAG,1

RET

C\_CONTINUE:

INC DI

JMP C\_START

RET

FUNC\_COMPARE ENDP

FUNC\_CHECK PROC NEAR

CALL OUTPUT\_1

CALL OUTPUT\_2

CALL OUTPUT\_3

RET

FUNC\_CHECK ENDP

INPUT\_NAME1 PROC NEAR

LEA DX,MESSAGE1

MOV AH,09H

INT 21H

MOV BX,0

REINPUT\_NAME1:

MOV AH,01H

INT 21H

CMP AL,0DH

JE INPUT\_NAME1\_END

MOV NAME1[BX],AL

INC BX

JMP REINPUT\_NAME1

INPUT\_NAME1\_END:

MOV NAME1[BX+1],'$'

RET

INPUT\_NAME1 ENDP

INPUT\_TEL1 PROC NEAR

LEA DX,MESSAGE2

MOV AH,09H

INT 21H

MOV BX,0

REINPUT\_TEL1:

MOV AH,01H

INT 21H

CMP AL,0DH

JE INPUT\_TEL1\_END

MOV TEL1[BX],AL

INC BX

JMP REINPUT\_TEL1

INPUT\_TEL1\_END:

MOV TEL1[BX+1],'$'

RET

INPUT\_TEL1 ENDP

INPUT\_NAME2 PROC NEAR

LEA DX,MESSAGE1

MOV AH,09H

INT 21H

MOV BX,0

REINPUT\_NAME2:

MOV AH,01H

INT 21H

CMP AL,0DH

JE INPUT\_NAME2\_END

MOV NAME2[BX],AL

INC BX

JMP REINPUT\_NAME2

INPUT\_NAME2\_END:

MOV NAME2[BX+1],'$'

RET

INPUT\_NAME2 ENDP

INPUT\_TEL2 PROC NEAR

LEA DX,MESSAGE2

MOV AH,09H

INT 21H

MOV BX,0

REINPUT\_TEL2:

MOV AH,01H

INT 21H

CMP AL,0DH

JE INPUT\_TEL2\_END

MOV TEL2[BX],AL

INC BX

JMP REINPUT\_TEL2

INPUT\_TEL2\_END:

MOV TEL2[BX+1],'$'

RET

INPUT\_TEL2 ENDP

INPUT\_NAME3 PROC NEAR

LEA DX,MESSAGE1

MOV AH,09H

INT 21H

MOV BX,0

REINPUT\_NAME3:

MOV AH,01H

INT 21H

CMP AL,0DH

JE INPUT\_NAME3\_END

MOV NAME3[BX],AL

INC BX

JMP REINPUT\_NAME3

INPUT\_NAME3\_END:

MOV NAME3[BX+1],'$'

RET

INPUT\_NAME3 ENDP

INPUT\_TEL3 PROC NEAR

LEA DX,MESSAGE2

MOV AH,09H

INT 21H

MOV BX,0

REINPUT\_TEL3:

MOV AH,01H

INT 21H

CMP AL,0DH

JE INPUT\_TEL3\_END

MOV TEL3[BX],AL

INC BX

JMP REINPUT\_TEL3

INPUT\_TEL3\_END:

MOV TEL3[BX+1],'$'

RET

INPUT\_TEL3 ENDP

OUTPUT\_1 PROC NEAR

LEA DX,INFO1

MOV AH,09H

INT 21H

MOV BX,0

LEA DX,NAME1

MOV AH,09H

INT 21H

MOV BX,0

CALL ENDLINE

LEA DX,INFO2

MOV AH,09H

INT 21H

MOV BX,0

LEA DX,TEL1

MOV AH,09H

INT 21H

MOV BX,0

CALL ENDLINE

RET

OUTPUT\_1 ENDP

OUTPUT\_2 PROC NEAR

LEA DX,INFO1

MOV AH,09H

INT 21H

MOV BX,0

LEA DX,NAME2

MOV AH,09H

INT 21H

MOV BX,0

CALL ENDLINE

LEA DX,INFO2

MOV AH,09H

INT 21H

MOV BX,0

LEA DX,TEL2

MOV AH,09H

INT 21H

MOV BX,0

CALL ENDLINE

RET

OUTPUT\_2 ENDP

OUTPUT\_3 PROC NEAR

LEA DX,INFO1

MOV AH,09H

INT 21H

MOV BX,0

LEA DX,NAME3

MOV AH,09H

INT 21H

MOV BX,0

CALL ENDLINE

LEA DX,INFO2

MOV AH,09H

INT 21H

MOV BX,0

LEA DX,TEL3

MOV AH,09H

INT 21H

MOV BX,0

CALL ENDLINE

RET

OUTPUT\_3 ENDP

OUTPUT\_MESSAGE\_5 PROC NEAR

LEA DX,MESSAGE5

MOV AH,09H

INT 21H

RET

OUTPUT\_MESSAGE\_5 ENDP

ENDLINE PROC NEAR ;输出回车换行

MOV AH,02H

MOV DL,0AH

INT 21H

MOV AH,02H

MOV DL,0DH

INT 21H

RET

ENDLINE ENDP

CODES ENDS

END START

1. 经验部分
2. 验收实验主要考察对代码的熟悉程度，要能清楚每一行代码的目的，尤其要清楚各种指令的功能和对标志位的影响；
3. 老师会提问某一段代码功能，会问执行逻辑，会让把涉及到十六进制的数字输出改为二进制或者十进制的输出（比如在字符分类统计中，数字个数10个，十六进制显示000Ah,要是想让统计结果二进制输出或者十进制输出，应该怎么改，类似这种有点难度的改动一般只讲思路不动手操作）；
4. 如果现场让改代码，不会很复杂，基本很简单的改动。
5. 电话本属于扩展性实验，一般来讲很少人做出来，诚实点，告诉老师借鉴别人的，别硬装逼非说自己写的，老师们都任教多年，一眼就能看穿各位，欺骗老师不仅愚蠢而且无效；
6. 老师们验收实验是为了更加公平地考核，帮助大家回忆指令、更好地掌握知识，态度要谦虚有礼，不要硬怼老师；
7. 想拿优的同学要在基础实验上有所创新，让老师看到你的独创性工作。

最终解释权归本人所有，本人是谁不重要，一个菜鸡而已，祝大家实验顺利，学习快乐，汇编虽然枯燥但是很有用！！！