**第7次作业**

**1.** 解释中断和异常。异常有哪几种类型？它们有什么不同？

**2.** 什么是中断矢量表？什么是中断描述符表？如何利用中断描述符表找到中断服务程序的地址？

**3.** 假定PROGN为某一中断源的中断处理程序的入口地址（第一条指令的标号），该中断源的类型号为**n**，试编写实方式下的程序段，将该入口地址填入中断向量表中的相应位置。

**4.** 在实模式下，如何使用CALL指令去实现int 21h的功能。

**5.** Windows的异常处理机制有哪些？各有什么特点？

**6.** 简述在Windows SEH链表的表头插入一个新的结点（新的异常处理函数）的实现步骤。

**7.** 一个Windows异常可以被多个处理程序处理吗？为什么？

**8.** 程序填空

在下面的程序段，测试字符串STR中是否存在’+’或者’-’字符。如存在，则把DL的最高位为1，否则置为0；DL其他位内容保持不变。

.686P

.model flat, stdcall

ExitProcess proto stdcall :dword

includelib libcmt.lib

includelib legacy\_stdio\_definitions.lib

.data

str1 db "ahhjhjsa + bbs -", 0

len equ $ - str1

.stack 200

.code

main proc c

mov ecx, **\_\_\_\_\_\_\_**

mov esi, **\_\_\_\_\_\_\_**

L0: mov al, [esi]

cmp al, '+'

jz **\_\_\_\_\_**

cmp al, **\_\_\_\_**

jnz **\_\_\_\_**

L1: or dl, **\_\_\_\_**

jmp **\_\_\_\_**

L2: **\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_**

and **\_\_\_\_\_\_\_**

exit: invoke **\_\_\_\_\_\_\_**, 0

main endp

end

**9.** 程序填空。变量char中定义了一个字符，将该字符的ASCII码转换成16进制字符串，然后调用C函数printf显示出来。

.686P

.model flat, stdcall

ExitProcess proto stdcall :dword

printf proto C :vararg

includelib libcmt.lib

includelib legacy\_stdio\_definitions.lib

.data

char db 'A'

msg db 0, 0, 0ah, 0dh, **\_\_\_\_**

.code

main proc c

mov al, **\_\_\_\_\_\_\_**

mov ebx, 1

L1: and al, 0fH

cmp al, 10

jb L2

sub **\_\_\_\_\_\_\_**

add al, 'A'

jmp **\_\_\_\_\_\_\_**

L2: **\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

L3: mov **\_\_\_\_\_\_\_** , al

cmp ebx, **\_\_\_\_\_\_\_**

jz L4

mov al, **\_\_\_\_\_\_\_\_**

shr **\_\_\_\_\_\_\_\_\_\_\_**

dec ebx

jmp L1

L4: invoke printf, offset msg

**\_\_\_\_\_** ExitProcess, 0

main endp

**\_\_\_\_\_\_**

**10.** 阅读程序，回答问题。

.686P

.model flat, stdcall

ExitProcess proto stdcall :dword

includelib kernel32.lib ;; ExitProcess()

includelib libcmt.lib ;; mainCRTStartup => main

.data

string db “12ABCKJHaaabjufdsalb47”

len equ $ - string ;;$表示当前位置偏移地址

.stack 200

.code

main proc c

lea esi, string

mov ecx, len **;** **①**

next: mov al, [esi] **; ②**

cmp al, ‘A’

jb L1

cmp al, ‘Z’

ja L1

sub al, ‘A’ - ‘a’

mov [esi], al

L1: inc esi **;** **③**

loop next

invoke ExitProcess, 0 **;返回Windows操作系统**

main endp

end

1. 上述程序的功能是什么？
2. 若将语句②处的标号next上移一行，误写到语句①处，则程序执行结果会怎样？
3. 若漏写了语句③，程序功能会发生什么变化？