转 汇编语言基本指令

2014年12月10日 20:05:14 苍痕 阅读数：6720

一.机械码,又称机器码.
ultraedit打开,编辑exe文件时你会看到
许许多多的由0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F组成的数码,这些数码
就是机器码.
修改程序时必须通过修改机器码来修改exe文件.

二.需要熟练掌握的全部汇编知识(只有这么多)
不大容易理解,可先强行背住,混个脸儿熟,以后慢慢的就理解了
cmp a,b 比较a与b
mov a,b 把b的值送给a
ret 返回主程序
nop 无作用,英文“no operation”的简写，意思是“do nothing”(机器码90)***机器码的含义参看上面
(解释:ultraedit打开编辑exe文件时你看到90,等同于汇编语句nop)
call 调用子程序
je 或jz 若相等则跳(机器码74 或0F84)
jne或jnz 若不相等则跳(机器码75或0F85)
jmp 无条件跳(机器码EB)
jb 若小于则跳
ja 若大于则跳
jg 若大于则跳
jge 若大于等于则跳
jl 若小于则跳
jle 若小于等于则跳
pop 出栈
push 压栈

三.常见修改(机器码)
74=>75 74=>90 74=>EB
75=>74 75=>90 75=>EB

jnz->nop
75->90(相应的机器码修改)

jnz -> jmp
75 -> EB(相应的机器码修改)

jnz -> jz
75->74 (正常) 0F 85 -> 0F 84(特殊情况下,有时,相应的机器码修改)

四.两种不同情况的不同修改方法
1.修改为jmp
je(jne,jz,jnz) =>jmp相应的机器码EB （出错信息向上找到的第一个跳转）jmp的作用是绝对跳，无条件跳，从而跳过下面的出错信息

xxxxxxxxxxx 出错信息，例如：注册码不对，sorry,未注册版不能…，”Function Not Avaible in Demo” 或 ”Command Not Avaible” 或 ”Can’t save in Sharewar
（我们希望把它跳过，不让它出现）
。。。
。。。
xxxxxxxxxxx 正确路线所在

2.修改为nop
je(jne,jz,jnz) =>nop相应的机器码90 （正确信息向上找到的第一个跳转） nop的作用是抹掉这个跳转，使这个跳转无效，失去作用，从而使程序顺利来到紧跟其后的正确

xxxxxxxxxxx 正确信息，例如：注册成功，谢谢您的支持等（我们希望它不被跳过，让它出现，程序一定要顺利来到这里）
。。。
。。。
xxxxxxxxxxx 出错信息（我们希望不要跳到这里，不让它出现）它们在存贮器和寄存器、寄存器和输入输出端口之间传送数据.
1. 通用数据传送指令.
MOV 传送字或字节.
MOVSX 先符号扩展,再传送.
MOVZX 先零扩展,再传送.
PUSH 把字压入堆栈.
POP 把字弹出堆栈.
PUSHA 把AX,CX,DX,BX,SP,BP,SI,DI依次压入堆栈.
POPA 把DI,SI,BP,SP,BX,DX,CX,AX依次弹出堆栈.

PUSHAD 把EAX,ECX,EDX,EBX,ESP,EBP,ESI,EDI依次压入堆栈.
POPAD 把EDI,ESI,EBP,ESP,EBX,EDX,ECX,EAX依次弹出堆栈.
BSWAP 交换32位寄存器里字节的顺序
XCHG 交换字或字节.(至少有一个操作数为寄存器,段寄存器不可作为操作数)
CMPXCHG 比较并交换操作数.(第二个操作数必须为累加器AL/AX/EAX)
XADD 先交换再累加.(结果在第一个操作数里)
XLAT 字节查表转换.
—— BX 指向一张 256 字节的表的起点, AL 为表的索引值 (0-255,即
0-FFH); 返回 AL 为查表结果. ([BX+AL]->AL)
2. 输入输出端口传送指令.
IN I/O端口输入. (语法: IN 累加器, {端口号 | DX})
OUT I/O端口输出. (语法: OUT {端口号 | DX},累加器)
输入输出端口由立即方式指定时, 其范围是 0-255; 由寄存器 DX 指定时,
其范围是 0-65535.
3. 目的地址传送指令.
LEA 装入有效地址.
例: LEA DX,string ;把偏移地址存到DX.
LDS 传送目标指针,把指针内容装入DS.
例: LDS SI,string ;把段地址:偏移地址存到DS:SI.
LES 传送目标指针,把指针内容装入ES.
例: LES DI,string ;把段地址:偏移地址存到ESI.
LFS 传送目标指针,把指针内容装入FS.
例: LFS DI,string ;把段地址:偏移地址存到FSI.
LGS 传送目标指针,把指针内容装入GS.
例: LGS DI,string ;把段地址:偏移地址存到GSI.
LSS 传送目标指针,把指针内容装入SS.
例: LSS DI,string ;把段地址:偏移地址存到SSI.
4. 标志传送指令.
LAHF 标志寄存器传送,把标志装入AH.
SAHF 标志寄存器传送,把AH内容装入标志寄存器.
PUSHF 标志入栈.
POPF 标志出栈.
PUSHD 32位标志入栈.
POPD 32位标志出栈.

二、算术运算指令
ADD 加法.
ADC 带进位加法.
INC 加 1.
AAA 加法的ASCII码调整.
DAA 加法的十进制调整.
SUB 减法.
SBB 带借位减法.
DEC 减 1.
NEC 求反(以 0 减之).
CMP 比较.(两操作数作减法,仅修改标志位,不回送结果).
AAS 减法的ASCII码调整.
DAS 减法的十进制调整.
MUL 无符号乘法.
IMUL 整数乘法.
以上两条,结果回送AH和AL(字节运算),或DX和AX(字运算),
AAM 乘法的ASCII码调整.
DIV 无符号除法.
IDIV 整数除法.
以上两条,结果回送:
商回送AL,余数回送AH, (字节运算);
或 商回送AX,余数回送DX, (字运算).
AAD 除法的ASCII码调整.
CBW 字节转换为字. (把AL中字节的符号扩展到AH中去)
CWD 字转换为双字. (把AX中的字的符号扩展到DX中去)
CWDE 字转换为双字. (把AX中的字符号扩展到EAX中去)
CDQ 双字扩展. (把EAX中的字的符号扩展到EDX中去)

三、逻辑运算指令
AND 与运算.
OR 或运算.
XOR 异或运算.
NOT 取反.
TEST 测试.(两操作数作与运算,仅修改标志位,不回送结果).
SHL 逻辑左移.
SAL 算术左移.(=SHL)

SHR 逻辑右移.
SAR 算术右移.(=SHR)
ROL 循环左移.
ROR 循环右移.
RCL 通过进位的循环左移.
RCR 通过进位的循环右移.
以上八种移位指令,其移位次数可达255次.
移位一次时, 可直接用操作码. 如 SHL AX,1.
移位>1次时, 则由寄存器CL给出移位次数.
如 MOV CL,04
SHL AX,CL

四、串指令
DS:SI 源串段寄存器 :源串变址.
ESI 目标串段寄存器:目标串变址.
CX 重复次数计数器.
AL/AX 扫描值.
D标志 0表示重复操作中SI和DI应自动增量; 1表示应自动减量.
Z标志 用来控制扫描或比较操作的结束.
MOVS 串传送.
(MOVSB 传送字符. MOVSW 传送字. MOVSD 传送双字.)
CMPS 串比较.
(CMPSB 比较字符. CMPSW 比较字.)
SCAS 串扫描.
把AL或AX的内容与目标串作比较,比较结果反映在标志位.
LODS 装入串.
把源串中的元素(字或字节)逐一装入AL或AX中.
(LODSB 传送字符. LODSW 传送字. LODSD 传送双字.)
STOS 保存串.
是LODS的逆过程.
REP 当CX/ECX0时重复.
REPE/REPZ 当ZF=1或比较结果相等,且CX/ECX0时重复.
REPNE/REPNZ 当ZF=0或比较结果不相等,且CX/ECX0时重复.
REPC 当CF=1且CX/ECX0时重复.
REPNC 当CF=0且CX/ECX0时重复.

五、程序转移指令
1>无条件转移指令 (长转移)
JMP 无条件转移指令
CALL 过程调用
RET/RETF过程返回.
2>条件转移指令 (短转移,-128到+127的距离内)
(当且仅当(SF XOR OF)=1时,OP1循环控制指令(短转移)
LOOP CX不为零时循环.
LOOPE/LOOPZ CX不为零且标志Z=1时循环.
LOOPNE/LOOPNZ CX不为零且标志Z=0时循环.
JCXZ CX为零时转移.
JECXZ ECX为零时转移.
4>中断指令
INT 中断指令
INTO 溢出中断
IRET 中断返回
5>处理器控制指令
HLT 处理器暂停, 直到出现中断或复位信号才继续.
WAIT 当芯片引线TEST为高电平时使CPU进入等待状态.
ESC 转换到外处理器.
LOCK 封锁总线.
NOP 空操作.
STC 置进位标志位.
CLC 清进位标志位.
CMC 进位标志取反.
STD 置方向标志位.
CLD 清方向标志位.
STI 置中断允许位.
CLI 清中断允许位.

六、伪指令
DW 定义字(2字节).
PROC 定义过程.
ENDP 过程结束.
SEGMENT 定义段.