


附录A 浮点指令

 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY



声明：本电子文档是《加密与解密(第三版)》的配套辅助电子教程！电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396；(010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

说明：本篇文档是对《加密与解密》第三版“附录 A 浮点指令”补充，为了降低书的价格，直接以电子文档提供给图书购买者。

看雪软件安全网站

<http://www.pediy.com>

kanxue

2008-6-1

附录A 浮点指令补充

浮点数不是一种保护方式，但在序列号里采用浮点数编程可以增加代码复杂程度。建议在必要时采用这种方法。

实例 Fpu.exe 是一个计算 $ax^2+bx+c=0$ 方程的程序，其中用到了浮点指令。代码如下：

```
BOOL ABC( HWND hwnd)
{
    TCHAR szBuffer[100], a_str[100], b_str[100], c_str[100];
    double x1,x2,a,b,c,d;
    .....           // 省去了一段代码，其作用是取得 a_str, b_str, c_str 字符串
    a=atof(a_str);   // 将字符串转换成浮点类型
    b=atof(b_str);
    c=atof(c_str);

    d=b*b-4*a*c;
    if (d>=0)
    {
        x1=(-b+sqrt(d))/(2*a);
        x2=(-b-sqrt(d))/(2*a);
        sprintf(szBuffer,TEXT("%lf"),x1);
        SetDlgItemText(hwnd,IDC_TXTX1,szBuffer);
        sprintf(szBuffer,TEXT("%lf"),x2);
        SetDlgItemText(hwnd,IDC_TXTX2,szBuffer);
    }
    else
        return FALSE;
    return TRUE;
}
```

用 SoftICE 调试时, 请用 WF 命令打开浮点窗口, OllyDbg 在寄存器窗口中直接显示了浮点寄存器。另外, 建议再用 IDA 反汇编一下, 其 FLIRT 技术可以识别 C 函数, 如_atof, _sprintf 等函数。用“bpx GetDlgItemTextA”设断可来到如下代码处:

```

; a=atof(a_str)
0040121E    PUSH EAX                ; 系数 a
0040121F    CALL 0040142A           ; _atof, 将字符串转换成浮点数, 放到 st(0)
00401224    FSTP QWORD PTR SS:[EBP-10] ; 将 a 的浮点格式存放到[EBP-10]处并出栈
00401227    LEA EAX,DWORD PTR SS:[EBP-90]
; b=atof(b_str)
0040122D    PUSH EAX                ; 系数 b
0040122E    CALL 0040142A(_atof)    ; st(0)=b
00401233    FST QWORD PTR SS:[EBP-8] ; [EBP-8] = st(0)
00401236    FLD ST      ; 取 st(0), 压入栈顶, 原来的 st(0)变成 st(1), st(0)=st(1)=b
00401238    FMUL ST,ST(1)           ; st(0)=st(0)×st(1)=b2
0040123A    LEA EAX,DWORD PTR SS:[EBP-F4]
; c=atof(c_str)
00401240    PUSH EAX                ; 系数 c 入栈
00401241    FSTP QWORD PTR SS:[EBP-18] ; [EBP-18]=st(0), 并出栈, 即 st(0)←st(1)
00401244    FSTP ST                 ; st(0)= st(0), 并出栈, 即清空了浮点栈
00401246    CALL 0040142A(_atof)    ; st(0)=c
; d=b2-4ac
0040124B    FMUL QWORD PTR SS:[EBP-10] ; st(0)= st(0)×[EBP-10]=ac
0040124E    ADD ESP,0C              ; 调整堆栈平衡 (C 调用约定)
00401251    FMUL QWORD PTR DS:[4080F8] ; st(0)= st(0)×4=4ac
00401257    FSUBR QWORD PTR SS:[EBP-18] ; st(0)= [EBP-18] - st(0)= b2-4ac
; if (d >= 0)
0040125A    FST QWORD PTR SS:[EBP-18] ; [EBP-18]=st(0), 即将 b2-4ac 存入
0040125D    FCOMP QWORD PTR DS:[4080F0] ; 即 b2-4ac 与 0.0 比较
00401263    FSTSW AX                ; 浮点状态寄存器送 AX
00401265    SAHF                    ; AX 高字节转送到整数标志寄存器的低字节
00401266    JB 004012FB             ; 小于 0, 无实根
; x1=(-b+sqrt(d))/(2a)
0040126C    FLD QWORD PTR SS:[EBP-10] ; st(0)= [EBP-10], 即 st(0)=a
0040126F    FADD ST,ST              ; st(0)= st(0)+ st(0), 即 st(0)=2a
00401271    PUSH EDI
00401272    PUSH ECX
00401273    PUSH ECX
00401274    FSTP QWORD PTR SS:[EBP-10] ; [EBP-10]= st(0)并出栈, 即[EBP-10]=2a
00401277    FLD QWORD PTR SS:[EBP-18] ; st(0)= [EBP-18], 即 st(0)=b2-4ac

```

0040127A	FSTP QWORD PTR SS:[ESP]	; SS:[ESP]= st(0), 即将 b^2-4ac 入栈
0040127D	CALL 00401384	; _sqrt 函数, 求平方根, 即 $st(0)=\sqrt{b^2-4ac}$
00401282	FSUB QWORD PTR SS:[EBP-8]	; $st(0)=st(0)-[EBP-8]=-b+\sqrt{b^2-4ac}$
00401285	FDIV QWORD PTR SS:[EBP-10]	; $st(0)=st(0)/[EBP-10]=(-b+\sqrt{b^2-4ac})/2a$
00401288	FSTP QWORD PTR SS:[EBP-20]	; [EBP-20]=st(0), 即将结果 x1 放到[EBP-20]
; $x2=(-b-\sqrt{b^2-4ac})/(2a)$		
0040128B	FLD QWORD PTR SS:[EBP-8]	; $st(0)=[EBP-8]$, 即 $st(0)=b$
0040128E	FCHS	; 求负数, $st(0)=-st(0)$, 即 $st(0)=-b$
00401290	FSTP QWORD PTR SS:[EBP-8]	; [EBP-8]= $st(0)=-b$
00401293	FLD QWORD PTR SS:[EBP-18]	; $st(0)=b^2-4ac$
00401296	FSTP QWORD PTR SS:[ESP]	; b^2-4ac 入栈
00401299	CALL 00401384	; _sqrt 函数, 求平方根, 即 $st(0)=\sqrt{b^2-4ac}$
0040129E	FSUBR QWORD PTR SS:[EBP-8]	; $st(0)=-b-\sqrt{b^2-4ac}$
004012A1	MOV EDI,00409058	; ASCII "%lf"
004012A6	LEA EAX,DWORD PTR SS:[EBP-2C]	
004012A9	FDIV QWORD PTR SS:[EBP-10]	; $st(0)=(-b-\sqrt{b^2-4ac})/2a$
004012AC	FSTP QWORD PTR SS:[EBP-8]	; 将 x2 保存到[EBP-8]
004012AF	FLD QWORD PTR SS:[EBP-20]	; 将 x1 转到 st(0)
; sprintf(szBuffer,TEXT("%lf"),x1)		
004012B2	FSTP QWORD PTR SS:[ESP]	; 将浮点数 x1 入栈
004012B5	PUSH EDI	; 参数"%lf"
004012B6	PUSH EAX	; 缓存地址
004012B7	CALL 00401315	; _sprintf 函数将浮点数转换成字符串