**《软件逆向工程》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | | **2022级信息安全** | | | **姓名** |  |
| **实验题目** | **X86汇编程序设计与分析** | | | | | |
| **实验时间** | **2024.09.27** | | **实验地点** | **DS3401** | | |
| **实验成绩** | **4** | | **实验性质** | **☑验证性 □设计性 □综合性** | | |
| 教师评价：  ☑算法/实验过程正确； ☑源程序/实验内容提交； ☑程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； ☑报告规范；  其他：  评价教师签名： 杨吉云 | | | | | | |
| 1. 实验目的 2. 验证X86移位指令 3. 验证X86运算指令 4. 验证X86分支及循环指令 5. 软件逆向工具的使用 | | | | | | |
| 1. 实验项目内容   1. 写程序在每条指令执行后，观察并记录CF、SF、ZF、OF的值  mov ax, 7FF0h  add al, 10h  add ah,1  add ax, 2  2. 编写程序实现表达式求值：((3450h+76h)\*1234h)/2345h，记录运行结果；  3. 定义数组A为：567Fh, 78FFh, 4C6Ah, 90A4h, 6ABCh, F56Ah，编写程序：  (1) 将该数组数据复制到数组B中；  (2) 将数组A中每个数据算术左移2位，数组B中每个数据循环右移两位； (3) 将两个数组对应数据进行异或运算，结果保存到A数组中；  4. 定义为“asdterDXRT345\*!@as4”字符串，编写程序：  (1) 将其中字母大小写反转；  (2) 将其中的数字转换为二进制存储到数组中；   1. 实验要求 2. 搭建实验所需的环境； 3. 能正确使用编译工具和逆向分析工具； | | | | | | |
| 1. 实验过程或算法（源程序）   **1.写程序在每条指令执行后，观察并记录CF、SF、ZF、OF的值**  .386  .model flat, stdcall  .stack  include windows.inc  include user32.inc  include kernel32.inc  includelib user32.lib  includelib kernel32.lib  .code  main proc    mov ax,7ff0h  add al,10h  add ah,1  add ax,2  main endp  end main  **2.编写程序实现表达式求值：((3450h+76h)\*1234h)/2345h，记录运行结果**  .386  .model flat,stdcall  .stack  include windows.inc  include user32.inc  include kernel32.inc  includelib user32.lib  includelib kernel32.lib  .data  val1 WORD 3450h  val2 WORD 0076h  val3 WORD 1234h  val4 WORD 2345h  .code  main proc    mov ax,0  add ax,val1  add ax,val2  mul val3  div val4  main endp  end main  **3.处理数组实现复制和运算**  .386  .model flat,stdcall  option casemap:none  .stack 4096  includelib user32.lib  includelib kernel32.lib  includelib masm32.lib  ExitProcess PROTO,dwExitCode:DWORD  .data  A dd 567Fh, 78FFh, 4C6Ah, 90A4h, 6ABCh, 0F56Ah  B dd 6 DUP(0)  .code  main proc  mov ecx, LENGTHOF A  lea esi, A  lea edi, B  L1:  ;把数组A中的数据复制到数组B中  mov ax, [esi]  mov [edi], ax  add esi, TYPE A  add edi, TYPE B  loop L1    mov ecx, LENGTHOF A  lea esi, A  lea edi, B  L2:  ;A算数左移2位，B循环右移2位，A、B按位异或并将结果存入A数组  mov eax, [esi]  sal eax, 2  mov ebx, [edi]  ror ebx, 2  xor eax, ebx  mov [esi], eax  add esi, TYPE A  add edi, TYPE B  loop L2  invoke ExitProcess,0;  main endp  End main  **4.处理字符串实现大小写转换和保存二进制数字到新数组中**  .386  .model flat,stdcall  option casemap:none  .stack 4096  includelib user32.lib  includelib kernel32.lib  includelib masm32.lib  ExitProcess PROTO,dwExitCode:DWORD  .data  strA byte "asdterDXRT345\*!@as4",0  arrayB dword 4 DUP(0)  .code  main proc  lea esi, strA  lea edi, arrayB  L1:  mov al,[esi]  cmp al,'0'  jb next2  cmp al,':'  jb next1  cmp al,'A'  jb next2  cmp al,'['  jb ChangeToLower  cmp al,'{'  jb ChangeToCapital  next1:  ;用来将数字存入新的数组  sub al,'0'  mov [edi],al  add edi,TYPE arrayB  jmp next2  next2:  ;用来处理非数字非字母字符，直接保留在原字符串中  inc esi  jmp L1  ChangeToLower:  ;用来处理字母大写向小写  or al,00100000b  mov [esi],al  jmp next2  ChangeToCapital:  ;用来处理字母小写向大写  and al,11011111b  mov [esi],al  jmp next2  main endp  End main | | | | | | |
| 1. 实验结果及分析和（或）源程序调试过程 2. **写程序在每条指令执行后，观察并记录CF、SF、ZF、OF的值**   如下图所示,寄存器EAX中初始值为0019FFCCh  此时，CF=0、SF=0、ZF=1、OF=0  执行mov ax,7ff0h    如下图所示,EAX寄存器被赋值变为00197FF0h  此时，CF=0、SF=0、ZF=1、OF=0    执行add al,10h    如下图所示,EAX寄存器低8位变为00h  此时，CF=1、SF=0、ZF=1、OF=0    执行add ah,1    如下图所示,EAX寄存器高8位变为80h  此时，CF=0、SF=1、ZF=0、OF=1  执行add ax,2  如下图所示,EAX寄存器中的数据变为00198002h  此时，CF=0、SF=1、ZF=0、OF=0  **2.编写程序实现表达式求值：((3450h+76h)\*1234h)/2345h，记录运行结果**      如下图所示，EAX寄存器初始值为FFCCh    执行指令mov ax,0  如下图， EAX寄存器的值变为0000h  执行指令add ax,val1    如下图，EAX寄存器中的值变为3450h    执行add ax,val2    如下图，EAX寄存器中的值变为34C6h    执行mul val3    如下图，EAX寄存器中的值变为A438h    执行div val4    如下图，EAX寄存器中的值变为1B3Ch，最终结果为1B3Ch    **3.处理数组实现复制和运算**    如下图，地址为00403000~00403017显示的是数组A中存储的6个数据    执行mov ecx, LENGTHOF A  将A数组的长度存到ecx寄存器中，方便loop指令判断何时结束循环    如下图，ECX的值变为00000006h    执行lea esi, A和lea edi,B  将数组A和数组B的首地址分别存到esi和edi寄存器中    进入L1，执行mov ax, [esi]  将数组A的第一个数据读入ax寄存器    执行指令mov [edi]  ax将数组A的第一个数据存入数组B的第一项    如下图，数组A的第一个值被复制进数组B中    执行add esi, TYPE A和add edi, TYPE B  如下图,移动esi和edi寄存器到数组A和数组B的第二个数据首地址处    多次循环，如下图，数组A中的值已经全部复制到数组B中    恢复ecx寄存器的值为6，esi寄存器指向A的首地址，edi寄存器指向B的首地址，进入L2    数组A的第一个数据0000567Fh算数左移两位后变为000159FCh      数组B的第一个数据0000567Fh循环右移两位后变为C000159Fh  两个数据进行异或运算，更新EAX中的数据为C0014C63h    将异或过的结果存到数组A中    多次循环，得到最终结果，存放在00403000~00403017中    **4.处理字符串实现大小写转换和保存二进制数字到新数组中**    如下图，这是修改前的字符串    对第一个字符’a’,其asii码比较,小于’{‘,  跳转到ChangeToCapital:  and al,11011111b//按位与运算将al字符转为大写(通过清除第五位)  mov [esi],al//转换后的字符存回原字符串  jmp next2//处理下一个字符  如下图所示,原字符a转换为A,并在原字符串中更新为A:    剩下的小写字母字符进行同样的操作,多次循环后得到,如下图:  小写字符asdter转换为ASDTER:    对第七个字符’D’,其asii码比较,小于’[‘,,  跳转到ChangeToLower:  or al,00100000b//使用按位或运算将al中字符转为小写(通过设置第五位,asii码中大写与小写之间差32,大写+32=小写)  mov [esi],al//转换后的字符存回原字符串  jmp next2//跳next2,处理下一个字符  如下图所示,原字符D转换为d,并在原字符串中更新为d:    剩下的小写字母字符进行同样的操作,多次循环后得到,如下图:  小写字符dxrt转换为DXRT:    对第十一个字符’3’,其asii码比较,小于’:’,  跳转到next1:  sub al,'0'//将al中字符0-9转为对应整数0-9  mov [edi],al//将转换后的数字存在edi指向的数组arrayB当前位置  add edi,TYPE arrayB//移动edi到下一个dword位置,准备存储下一个数字  jmp next2//跳next2,处理下一个字符  如下图所示,原字符3转换为数字3,并存在新数组中:    剩下的数字字符进行同样的操作,多次循环后得到,如下图:  数字字符345以及最后的4被存放在新数组中:    对于非字母且非数字的字符串,将其保留在原字符串中,如下图:    最后,结果多次循环,得到转变后的字符串以及新数组,字符串中的数字被依次存入00402014~00402023中,如下图所示: | | | | | | |