

南開大學

## 汇编语言与逆向技术课程实验报告

### 实验二：dec2hex



学 院 网络空间安全学院  
专 业 信息安全  
学 号 2313546  
姓 名 蒋枘言  
班 级 信息安全班

## 一、实验目的

- 1.熟悉汇编语言的数据传送、寻址和算术运算；
- 2.熟悉汇编语言过程的定义和使用；
- 3.熟悉十进制和十六进制的数制转换。

## 二、实验环境

Windows 操作系统，MASM32 编译环境。

## 三、实验原理

使用除基取余法，十进制数不断除以 16 取余数得到十六进制数。

## 四、实验过程

- 1.编辑：用编辑软件（记事本）形成源程序。

### ●代码解析

```
.386
.model flat, stdcall
option casemap :none
include \masm32\include\windows.inc
include \masm32\include\kernel32.inc
include \masm32\include\user32.inc
include \masm32\include\masm32.inc

includelib \masm32\lib\kernel32.lib
includelib \masm32\lib\user32.lib
includelib \masm32\lib\masm32.lib

.data
decstr BYTE 20 DUP(0)
; 定义了一个 20 字节的缓冲区 decstr，用于存储用户输入的十进制字符串。
hexstr BYTE 9 DUP(0)
; 定义了一个 9 字节的缓冲区 hexstr，
; 用于存储转换后的十六进制字符串（8 个十六进制字符+1 个字符串结束符）。
decnim DWORD 0
; 定义了一个 32 位整数 decnim，用于存储转换后的十进制数值。
const10 DWORD 10
const16 DWORD 16
; 定义了两个 32 位常数，分别用于十进制和十六进制的转换。
strInput BYTE "Please input a decimal number(0~4294967295):",0
strOutput BYTE "The hexdecimal number is:",0
; 定义了字符串 strInput 和 strOutput，用于提示用户输入十进制数和输出十六进制数。

.code
; 定义了一个过程 dec2dw，用于将十进制字符串转换为十进制整数。
dec2dw PROC
    mov esi, 0
    ; esi 清零，esi 将用作字符串索引。
    mov ebx, 0
    ; ebx 清零，ebx 将临时存储当前字符转换后的十进制值。
L1:
    mov bl, [decstr+esi]
    ; 从 decstr 字符串中取出当前字符存入 bl。
    sub bl, 48
    ; 将字符 ('0'-'9') 转换为对应的十进制数（0-9），ASCII 码中 '0' 的值为 48。
    mov eax, decnim
```

```

; 将当前的 decnum 值存入 eax。
mul const10
; eax = decnum * 10, 进行乘 10 操作。
mov decnum, eax
; 将乘 10 后的结果存回 decnum。
add decnum, ebx
; decnum = decnum * 10 + ebx, 加上当前字符代表的十进制数。
inc esi
; esi++, 移动到字符串的下一个字符。
mov bl, [decstr+esi]
; 再次取出下一个字符存入 bl, 为下一次循环做准备。
cmp bl, 0
; 比较 bl 是否为 0 (字符串结束标志)。
jnz L1
; 如果 bl 不为 0, 则跳转并继续循环。
ret
dec2dw ENDP

; 定义了一个过程 dw2hex, 用于将十进制整数转换为十六进制字符串。
dw2hex PROC
    mov ecx, 8
    ; 设置循环计数器 ecx 为 8, 因为 DWORD (32 位) 需要 8 个十六进制字符来表示。
    mov edi, OFFSET hexstr + 8
    ; edi 指向 hexstr 的末尾, 用于逆向存储字符。
    mov eax, [decnum]
    ; 将 decnum 的值加载到 eax 中。
L2:
    xor edx, edx
    ; 清零 edx 寄存器, 用于存储除法余数。
    div DWORD PTR const16
    ; 用 eax 除以 16, 商存储在 eax 中, 余数存储在 edx 中。
    add dl, '0'
    ; 将余数转换为 ASCII 字符 (如果余数小于 10)。
    cmp dl, '9'
    jbe short L3
    ; 如果余数小于等于 9, 直接跳转到 L2。
    add dl, 7
    ; 否则, 将余数转换为'A'-'F' (ASCII 码偏移)。
L3:
    mov [edi-1], dl
    ; 存储当前位的十六进制字符。
    dec edi
    ; 向前移动指针。
    loop L2
    ; 循环, 直到处理完所有位。
    mov BYTE PTR [edi-1], 0
    ; 添加字符串结束符。
    ret
dw2hex ENDP

start:
    invoke StdOut, addr strInput
    invoke StdIn, addr decstr, 20

```

```
invoke StdOut, addr strOutput  
call dec2dw  
call dw2hex  
invoke StdOut, addr hexstr  
invoke ExitProcess, 0  
END start
```

**2.编译:** 用汇编程序（\masm32\bin\ml.exe）对源程序进行汇编，形成目标文件（.obj），格式如下：

“\masm32\bin\ml /c /Zd /coff dec2hex.asm”

●\masm32\bin\ml: 这是 MASM32 汇编器的完整路径。它告诉操作系统在哪里找到 ml.exe 程序以执行汇编任务。\\masm32\\bin\\ 是 MASM32 安装目录中包含 ml.exe 的文件夹。

●/c: 这个选项指示汇编器仅进行编译，不进行链接。也就是说，它会生成目标文件（.obj），但不会尝试创建可执行文件（.exe）。

●/Zd: 这个选项用于生成调试信息，并包含源代码行号。它使得生成的目标文件可以用于调试器，以便在调试时能够关联源代码行和生成的机器代码。

●/coff: 这个选项指示汇编器生成 COFF (Common Object File Format) 格式的目标文件。COFF 是一种用于存储程序对象代码的文件格式，它支持调试信息、重定位信息等。这是 Windows 平台上常用的目标文件格式之一。

●dec2hex.asm: 这是要编译的汇编源代码文件的名称。它应该位于当前工作目录或指定的路径中，以便汇编器可以找到并读取它。

**3.链接:** 用连接程序（\masm32\bin\link.exe）对目标程序进行连接，形成可执行文件（.exe），格式如下：

“\masm32\bin\Link /SUBSYSTEM:CONSOLE dec2hex.obj”

●\masm32\bin\Link: 这是链接器的完整路径，它告诉操作系统在哪里找到 link.exe 程序以执行链接任务。

●/SUBSYSTEM:CONSOLE: 生成命令行程序。这个选项指定了生成的可执行文件的子系统类型。CONSOLE 表明生成的是一个控制台应用程序，这意味着当程序运行时，它会打开一个命令行窗口（如果尚未打开），并且所有的输入和输出都会通过这个窗口进行。

●dec2hex.obj: 这是要链接的目标文件的名称。它是之前通过汇编器编译 dec2hex.asm 文件生成的。链接器会读取这个文件，并将其中的代码与其他必要的库代码和资源链接在一起，以生成最终的可执行文件。

**4.执行:** 如果结果在屏幕显示，则直接执行可执行文件。



The screenshot shows a Windows Command Prompt window. The title bar says "命令提示符". The window content is as follows:

```
Microsoft Windows [版本 10.0.22631.4169]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Lenovo>"E:\Assembly Language Code\Sep29\dec2hex.exe"
Please input a decimal number(0~4294967295):1919
The hexadecimal number is:0000077F
C:\Users\Lenovo>
```