

**汇编语言与逆向技术**

**课程实验报告**

**实验二：dex2hex**

****

学 院 网络空间安全

专 业 信息安全

学 号 2212998

姓 名 胡博浩

班 级­­­­ 2022级信息安全

**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；

**二、实验内容**

编写汇编程序dec2hex.asm，编译成dec2hex.exe。dec2hex.exe的功能是将Windows命令行输入的十进制无符号整数，转换成对应的十六进制整数，输出在Windows命令行中，如图1所示。

输入的十进制无符号整数的范围是**0到4294967295（232-1）。**

输出对应的十六进制整数，对应的范围是**00000000h到FFFFFFFFh。**

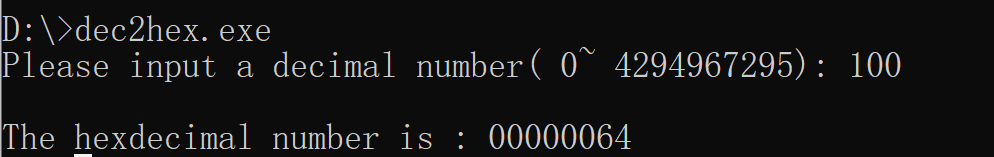


图1. dec2hex.exe将十进制100转换成十六进制00000064

3.1 使用StdIn函数获得用户输入的十进制整数。StdIn函数的定义在\masm32\include\masm32.inc，库文件是\masm32\lib\masm32.lib。StdIn函数的定义“StdIn PROTO :DWORD,:DWORD”，有两个参数，第一个是内存存储空间的起始地址，第二个是内存存储空间的大小。函数的例子：

.data

buf BYTE 20 DUP(0)

.code

invoke StdIn, addr buf, 20

invoke StdOut, addr buf

3.2 用户输入的十进制数对应的ASCII编码字符串存储在内存中，编写过程dec2dw，将ASCII字符串转换成DWORD数据。例如，将字符串“100”转换成DWORD数据00000064h。

3.3 编写过程dw2hex，将DWORD数据转换成十六进制数的ASCII字符串。例如，将DWORD数据00000064h转换成ASCII字符串“00000064”

3.4 使用StdOut函数在Windows命令函中输出十六进制整数的ASCII字符串。StdOut函数的定义在\masm32\include\masm32.inc，库文件是\masm32\lib\masm32.lib。StdOut函数的定义“StdOut PROTO :DWORD”，只有一个参数，是内存存储空间的起始地址。函数使用的例子同StdIn函数的例子。

3.5 使用ml将dec2hex.asm文件汇编到dec2hex.obj目标文件，编译命令：“\masm32\bin\ml /c /coff dec2hex.asm”

3.6 使用link将目标文件dec2hex.obj链接成dec2hex.exe可执行文件，链接命令：“\masm32\bin\link /SUBSYSTEM: CONSOLE dec2hex.obj”

**三、实验步骤**

1. dec2hex.asm源代码，其中包括dex2dw和dw2hex过程的定义。

**.386**

**.model flat, stdcall**

**option casemap:none**

**include \masm32\include\windows.inc**

**include \masm32\include\kernel32.inc**

**include \masm32\include\masm32.inc**

**includelib \masm32\lib\kernel32.lib**

**includelib \masm32\lib\masm32.lib**

**.data**

**message1 BYTE "Please input a decimal number:",0**

**message2 BYTE "The hexdecimal number is :",0**

**buf BYTE 20 DUP (0), 0 ; 输入的十进制字符串**

**decnum DWORD 0 ; 转换后的十进制值，以DWORD形式存储**

**const10 DWORD 10 ; 辅助转换的常数10**

**result BYTE 20 DUP (0), 0 ; 最终结果的字符串**

**.code**

**start:**

**invoke StdOut, addr message1 ; 提示用户输入**

**invoke StdIn, addr buf, 20 ; 接收输入**

**call mydec2dw**

**call mydw2hex**

**invoke StdOut, addr message2**

**invoke StdOut, addr result ; 输出转换后的结果**

**invoke ExitProcess, 0**

**mydec2dw PROC**

**mov esi, 0**

**xor ebx, ebx ; 清零ebx寄存器**

**L1:**

**mov bl, [buf + esi] ; 将位于esi位置的字符加载到bl寄存器**

**sub bl, 48 ; 将字符转换为数字**

**mov eax, decnum ; 加载decnum（最初为0）**

**mul const10 ; 乘以10以进行位移**

**mov decnum, eax ; 将结果存储在decnum中**

**add decnum, ebx ; 添加当前数字**

**inc esi ; 移至下一个字符**

**mov bl, [buf + esi] ; 加载下一个字符**

**cmp bl, 0 ; 检查字符串是否结束**

**jnz L1 ; 如果不是零，跳转（还有更多字符）**

**ret**

**mydec2dw ENDP**

**mydw2hex PROC**

**mov eax, decnum ; 将十进制值加载到eax寄存器**

**mov esi, 7 ; esi作为十六进制字符串的索引**

**mov ecx, 8 ; 共8个十六进制数字**

**L2:**

**mov edx, eax**

**and edx, 0Fh ; 提取最后4位**

**add dl, '0' ; 转换为ASCII字符**

**cmp dl, '9' ; 检查是否为数字字符**

**jbe @F ; 跳转到@F标签（继续）**

**add dl, 7 ; 如果不是数字，转换为字母字符**

**@@:**

**mov [result + esi], dl ; 存储到结果字符串**

**dec esi ; 移至下一个位置**

**shr eax, 4 ; 右移4位以处理下一个四位**

**loop L2**

**mov byte ptr [result + 8], 0 ; 在结果字符串末尾添加null字符**

**ret**

**mydw2hex ENDP**

**END start**

汇编代码主要由两个过程组成，mydec2dw和mydw2hex，分别用于将十进制数转换为双字（DWORD）表示，并将双字表示转换为十六进制字符串。

1.mydec2dw过程：

mydec2dw过程的主要功能是将存储在buf中的十进制字符串转换为 DWORD 格式的数值，存储在decnum中。

它通过循环遍历字符串中的字符，将每个字符转换为数字，并根据位数将其相应地加到decnum中。

最后，它返回转换后的decnum值。

2. mydw2hex过程：

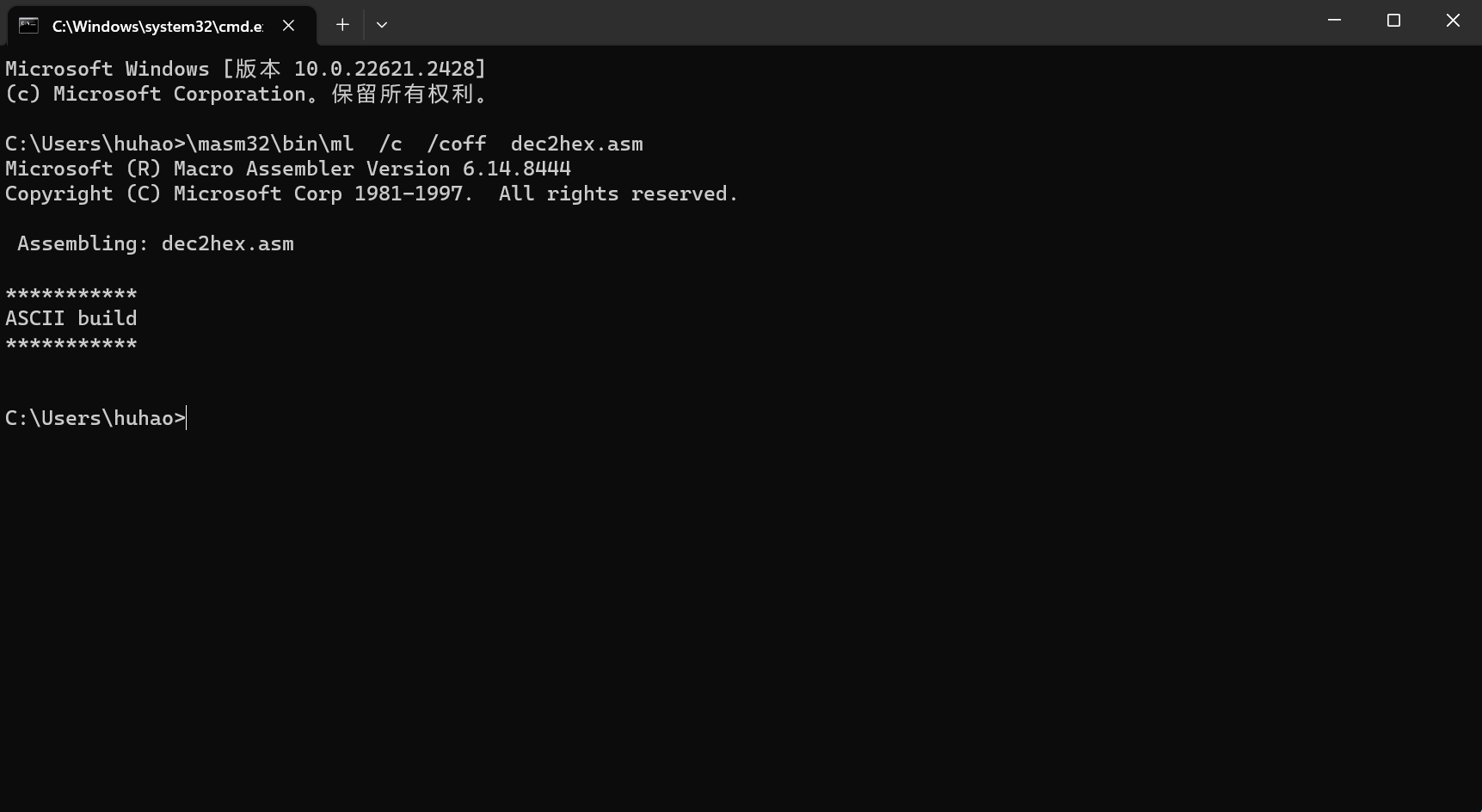
mydw2hex过程的主要功能是将 DWORD 格式的十进制数decnum转换为十六进制字符串，并将结果存储在result中。

它通过循环将decnum中的每四位转换为相应的十六进制字符，然后存储在result中。

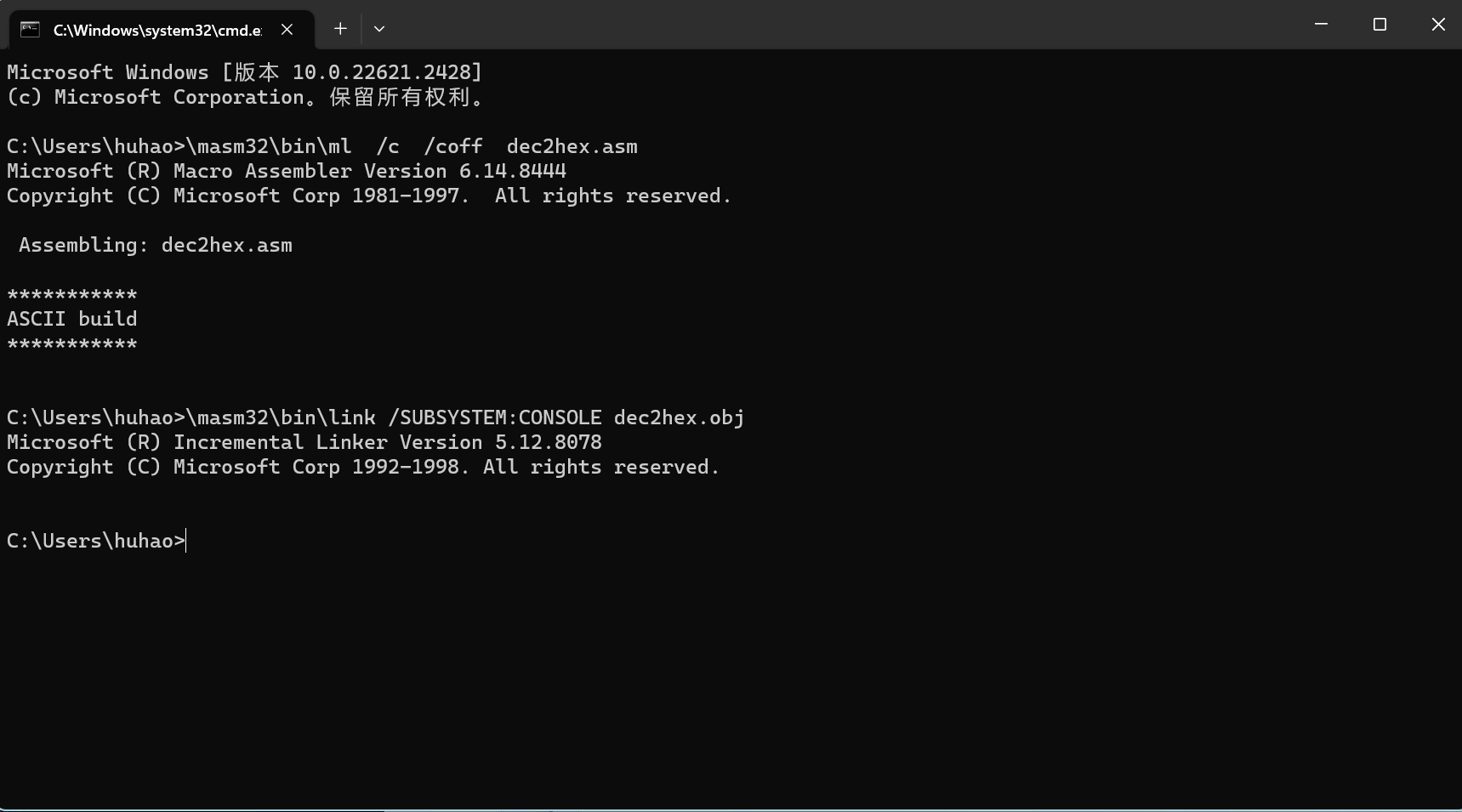
最后，它添加了一个 null 字符来终止结果字符串，并返回结果。

这两个过程协同工作，实现了将输入的十进制数字转换为十六进制字符串的功能。整个程序在start标签下开始，接受用户输入，执行转换，然后输出结果并退出。

1. dec2hex.asm源代码的编译和链接过程说明。

1）使用ml将dec2hex.asm文件汇编到dec2hex.obj目标文件，编译命令：“\masm32\bin\ml /c /coff dec2hex.asm”

2）使用link将目标文件dec2hex.obj链接成dec2hex.exe可执行文件，链接命令：“\masm32\bin\link /SUBSYSTEM:CONSOLE dec2hex.obj”



1. dec2hex.exe的测试说明。

执行生成的 dec2hex.exe 文件。输入三个不同的案例，程序能够正确输出3个十六进制的数，说明程序正确，实验二成功。

