汇编语言与逆向技术

第3章 汇编语言基础

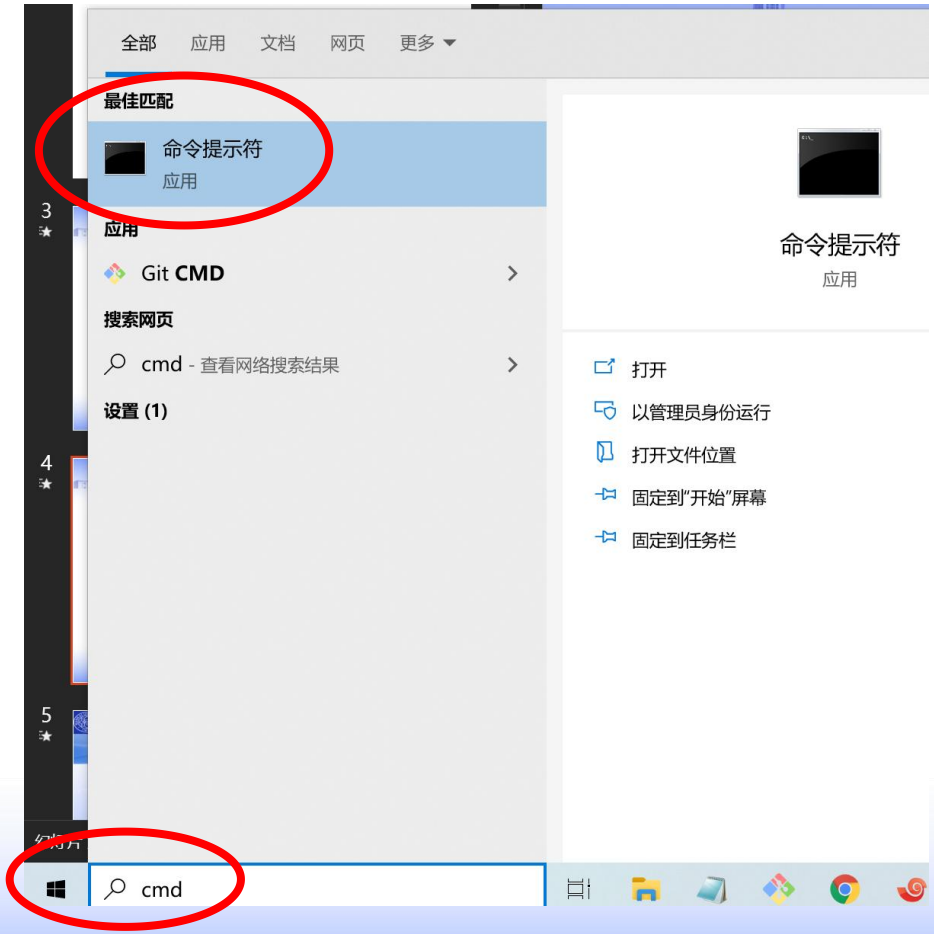
本章知识点

- 汇编语言的基本元素
- 定义数据
- 符号常量
- 汇编、链接和运行程序

Hello World实验的问题

- 打开cmd命令行窗口
- 当前路径、相对路径、绝对路径
- 路径切换
- 查看目录中的文件列表

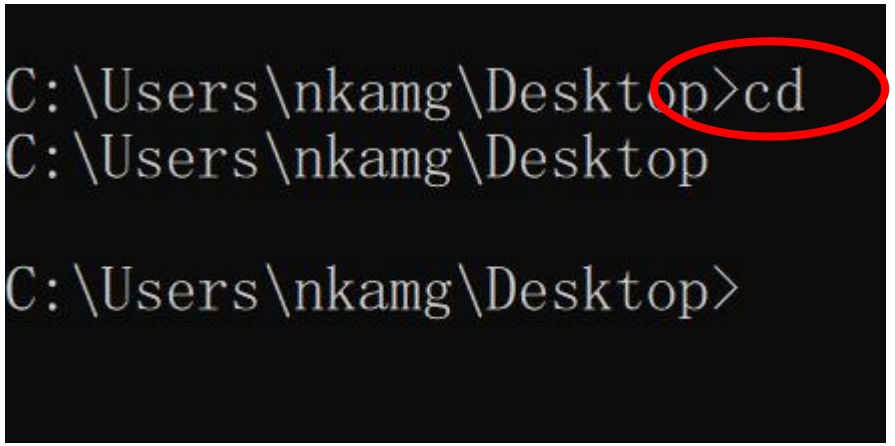
打开cmd命令行窗口



- 按Windows键
- 输入cmd
- 最佳匹配里面
  - “命令提示符”

当前路径

- 获得当前路径
  - cd命令

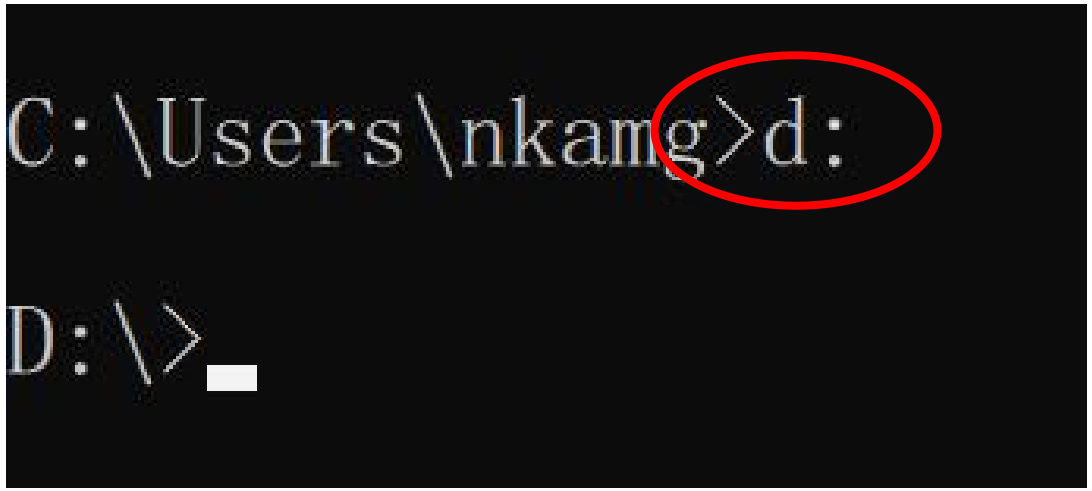


绝对路径

- 以盘符开始的路径

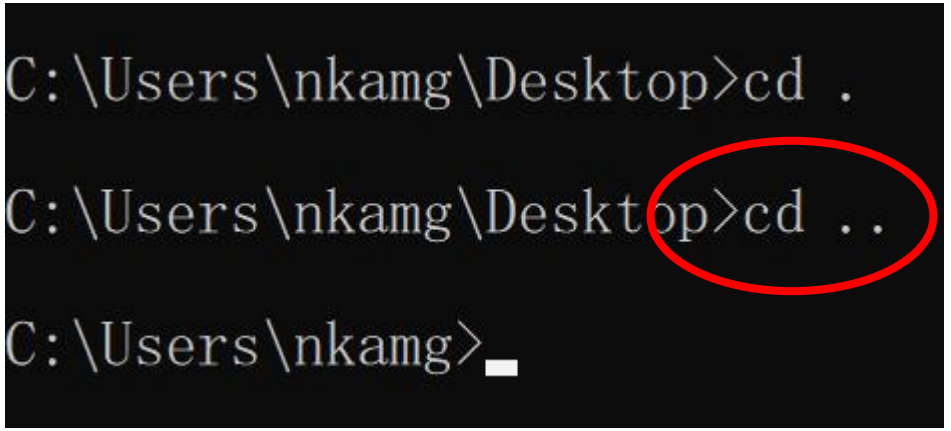
例如“C:\Users\nkamg\Desktop”

- 从C盘进入到D盘
  - d:



相对路径

- 相对于当前的路径
- “.” 表示的是当前路径
- “..”表示的是上一级路径



查看目录中的文件列表

- dir命令

```
C:\Users\nkamg>d:

D:\>dir
 驱动器 D 中的卷没有标签。
 卷的序列号是 1234-5678

 D:\ 的目录

2021/09/26  01:03    <DIR>          HPSCANS
2021/05/27  14:01        14,321,039 信息安全新技术研究室-20210528.pptx
2021/04/06  11:34         4,477,682 ch6-Recognizing C Constructs in Assembly.pptx
2021/06/23  12:29            39,028 PPT活动背景.pptx
2021/04/13  12:02       219,287,064 RainClassroom_Full_4.3.0.2006.exe
2021/04/20  09:40            20,113 第9章-动态调试.docx
2021/04/20  10:34         5,188,096 ch7-Analyzing Malicious Windows Programs.ppt
2021/04/25  10:43         3,161,088 ch8-Debugging.ppt
2021/04/25  11:30         3,263,488 ch9-OllyDbg.ppt
2021/04/27  09:33         2,338,730 教育.pptx
```

主观题 10分

设置

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

\masm32\bin\ml是一个相对地址还是绝对地址？

如何判断当前文件夹中是否有hello_console.asm文件？

正常使用主观题需2.0以上版本雨课堂

作答

遇到的路径问题

- 相对路径错误
  - \masm32\bin\ml /c /Zd /coff hello_console.asm
- 当前路径没有需要的文件
  - hello_console.asm
- asm代码中include、includelib路径问题
  - include \masm32\include\windows.inc

汇编语言的基本元素

汇编语言的基本元素

- 整数常量、整数表达式
- 实数常量
- 字符常量、字符串常量
- 保留字、标识符
- 指令、伪指令、NOP指令

整数常量

- [{+|-}]数字[基数]
- 基数后缀（Radix）
- h十六进制、q/o八进制、d十进制、b二进制
- r编码实数

整数常量

- 如果整数常量后面没有基数后缀，默认是十进制整数
- 10、10d、10o、10h、10q、0A0h，10b
- 以字母开头的十六进制常量前面必须加0

单选题 10分

设置

FFh是有效的整数常量吗？

A 是

B 不是

提交

南開大學 Nankai University

整数表达式

- 包含整数值和算数运算符的数学表达式
- 表达式的结果不能超过32bits的表示范围
- MOD：取余数运算

整数表达式

- 算术运算符的优先级
- （）优先级1
- *、/、MOD，乘、除、取余，优先级2
- +、-，加减，优先级3

填空题 10分

设置

表达式12 - 2 MOD 5 的计算结果是 [填空1]

正常使用填空题需3.0以上版本雨课堂

作答

南開大學 Nankai University

实数常量

- 十进制实数
- 编码（十六进制）实数

十进制实数常量

- -1.11E-5、2.、+3.0、2.E5
- 十进制实数常量由符号sign、整数、小数点、小数和指数组成
- [sign]integer.[integer][exponent]
- 至少要有一个数字和一个小数点

编码实数

- 编码实数是以十六进制数表示一个实数，遵循IEEE浮点数格式
- 《Intel汇编语言程序设计》第五版，第17章“浮点处理和指令编码”

字符常量

- 单引号或者双引号括起来的单个字符。
- 汇编器会将其转化为ASCII编码
- ‘A’、“B”

字符串常量

- 以单引号或者双引号括起来的一串字符
- ‘ABC’、“abc”
- 嵌套引号
- “print ‘Hello World’ on the terminal window”
- ‘print “Hello World” on the terminal window’

保留字

- 指令助记符：MOV、ADD
- 伪指令：INCLUDE、PROC
- 属性：BYTE、WORD
- 预定义符号：$、？
- 参考《Intel汇编语言程序设计》第五版 附录A

标识符

- 标识符是程序员选择用来标识变量、常量、过程、代码的标号
  - 包含1~247个字符
  - 大小写不敏感（MASM默认）
  - 第一个字符必须是字母、下划线、@、？或$
  - 第一个字符不能是数字（对比十六进制整数）

单选题 10分

设置

判断题：标识符可以用数字开头。

A 正确

B 错误

提交

指令

- 汇编语言中的指令是一条汇编语句
- 汇编器把汇编指令翻译成对应的机器指令
  - 标号
  - 指令助记符
  - 操作数
  - 注释

标号

- 标号是充当指令或数据位置标记的标识符
- 数据标号
  - 标识变量的地址
- 代码标号
  - 标识代码的地址

数据标号

- 标识变量的地址，方便变量的引用
- count DWORD 100
- array DWORD 100, 101, 102, 103
- 相对.data数据段在内存起始地址的偏移

OFFSET

- 获取数据标号的内存偏移地址

```
.data
    str_hello BYTE “Hello World！”，0
.code
    mov eax, OFFSET str_hello
```

代码标号

- 标识代码的地址，必须以冒号（:）结尾
- 通常作为跳转、循环指令的目标地址

```
target:
    mov eax，100h
    …
    jmp target
```

单选题 10分

设置

判断题：代码标号后面有冒号，数据标号后面没有冒号

A 正确

B 错误

提交

南開大學 Nankai University

指令助记符

- 指令助记符（instruction mnemonic）是一个简短的单词，用于表示一条指令。
  - mov、add、sub、mul、jmp、call

操作数

- 操作数是指令的操作对象
  - 寄存器
  - 内存
  - 常量
  - I/O端口

多选题 10分

设置

CPU指令可以直接访问的操作数类型有？

A 寄存器

B 内存

C I/O接口

D 硬盘

E 常量（立即数）

提交

操作数

- inc eax
  - eax寄存器的值加1
- mov count，ebx
  - mov指令有两个操作数：count、ebx
  - 第一个操作数是目的操作数
  - 第二个操作数是源操作数

注释

- 单行注释
  - mov count，ebx； save result to count
- 块注释: COMMENT伪指令和用户定义的符号

```
COMMENT !
        This is a comment
!
```

NOP指令

- NOP指令，空操作
  - 用于计时循环
- NOP指令占用1个字节的内存
  - 用于后继指令的对齐
  - IA-32处理器从偶数双字地址处加载代码和数据时更加快速

伪指令

- 伪指令内嵌在汇编语言源代码中，由汇编器识别、执行相应动作的命令
- 用于定义变量、段、过程、汇编器选项等
- 参考《Intel汇编语言程序设计》第五版，附录A， MASM的伪指令

伪指令

- 定义变量

  my_var DWORD 100h；DWORD伪指令

  mov eax, my_var ； mov指令

伪指令

- 定义段（Segment）
  - .data、.code、.stack
- 定义过程（Procedure）
  - PROC、ENDP
- 允许或禁止汇编器的某些特性
  - OPTION、.386、.MODEL

单选题 10分

设置

判断题：伪指令是在程序运行时执行的

A 正确

B 错误

提交

定义数据

内部数据类型

- MASM内部以数据位的个数定义了多种数据类型
  - BYTE，db，8位
  - WORD，dw， 16位
  - DWORD，dd， 32位
  - QWORD ，dq，64位

内部数据类型

- MASM汇编器默认情况下，大小写不敏感
- DWORD
  - Dword
  - dword
  - dWord

数据定义语句

- 为变量在内存中保留存储空间
- 为变量指定一个名字（数据标号）
- [变量名] 数据定义伪指令 初始值

数据定义伪指令

- BYTE，db，8 bits
- WORD，dw， 16 bits
- DWORD，dd， 32 bits
- QWORD ，dq，64 bits

初始值

- 数据定义语句中要指定初始值
- 多个初始值用逗号隔开
  - my_var DWORD 0, 1, 2, 3
- 0: 可以指定初始值为0
- ?: 表示在程序运行的时候初始化该变量

数据声明的位置

- .data段声明初始化的变量

```
.data
    dw_var1 DWORD 0
```

- .data?段声明未初始化的变量

```
.data?
    dw_var2 DWORD ?
```

定义字符串

```
str_hello BYTE “Hello World!”, 0Dh, 0Ah,
          BYTE “I love assembly language”,
          BYTE 0Dh, 0Ah, 0
```

- 0Dh和0Ah是CR/LF（回车、换行）的ASCII编码
- 字符串的结尾是0

DUP伪指令

- 为字符串或者数组分配内存空间
- BYTE 20 DUP（0）；20个字节的内存空间
- BYTE 4 DUP（“Hello”）：20个字节，连续的4个“Hello”，每个“Hello”5字节

填空题 10分

设置

声明一个包含单词“TEST”重复50次的字符串变量 [填空1]

正常使用填空题需3.0以上版本雨课堂

作答

符号常量

符号常量

- 符号常量（或符号定义），将标识符与整数表达式或文本联系起来
- 符号常量不占用存储空间
- 变量占用存储空间

等号伪指令

- 等号伪指令，将符号名和整数表达式联系起来

  COUNT = 500

  mov eax， COUNT

  - 易于阅读与维护，减少程序修改时的查找与替换次数

计算数组和字符串的大小

- MASM用$运算符存储当前语句的地址偏移值。
- $可以用来计算数组或字符串的大小

计算字符串大小

str_hello BYTE “Hello World!”, 0Dh, 0Ah,

BYTE “I love assembly language”,

BYTE 0Dh, 0Ah, 0

- str_size = ($ - str_hello)

计算数组大小

```
dw_array DWORD 0, 1, 2, 3, 4

array_size = ($ - dw_array)/4
```

EQU伪指令

- EQU伪指令将符号名与整数表达式或任意文本联系起来
  - name EQU expression
  - name EQU symbol
  - name EQU <text>

EQU 伪指令

PI EQU 3.1415926

press_key EQU <“Press any key to continue…”, 0>

.data

prompt BYTE pressKey ; 变量

EQU伪指令

- EQU伪指令的符号名，不能在程序中重定义
- “=”伪指令的符号名，可以在程序中重定义

汇编、链接和运行程序

汇编、链接和运行程序

- 源文件：用文本编辑器编写的asm文本文件
- 汇编：汇编器把汇编源文件翻译成机器语言，生成目标文件
- 链接：链接器从库中复制所需的过程，并将其同目标文件合并在一起生成可执行文件

hello.asm

```
.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
```

hello.asm

```
.data
HelloWorld db "Hello World!", 0
.code
start:
invoke StdOut, addr HelloWorld
invoke ExitProcess, 0
end start
```

hello.asm

- .386
  - 允许汇编80386处理器的非特权指令，禁用其后处理器引入的汇编指令
- .model 初始化程序的内存模式
  - flat：平坦模式，4GB内存空间
  - stdcall : 调用约定， stdcall是Win32 API函数的调用约定

hello.asm

- option casemap: none
  - 大小写敏感
- include …inc 函数的常量和声明
- includelib …lib 链接库

hello.asm

- .DATA
  - 定义已初始化数据段的开始
- .CODE
  - 定义代码段的开始
- start：，指令标号，标记指令地址

hello.asm

- StdOut, masm32.inc中定义的函数，将内存数据输出到命令行窗口上
- ExitProcess， Kernel32.inc中定义的函数，退出程序执行

hello.asm

- END start
  - 标记模块的结束
  - 指定程序的入口点

编译

- \masm\bin\ml /c /Zd /coff hello.asm
- **ml** 程序可以用来汇编并链接一个或多个汇编语言源文件
- ml的命令行选项是大小写敏感的

编译

- **/c** Assemble without linking
  - 只编译、不链接
- **/Zd** Add line number debug info
  - 在目标文件中生成行号信息
- **/coff** generate COFF format object file
  - 生成Microsoft公共目标文件格式（common object file format）的文件

链接

- \masm32\bin\Link /SUBSYSTEM:CONSOLE hello.obj
- Link.exe 链接器，将obj文件合并，生成可执行文件
- /SUBSYSTEM:CONSOLE，生成命令行程序

汇编与逆向技术基础

第3章 汇编语言基础

本章学习的知识点

- 汇编语言的基本元素
- 定义数据
- 符号常量
- 汇编、链接和运行程序