汇编语言与逆向技术

汇编语言基本概念

王志
zwang@nankai.edu.cn

updated on 2022-09-22

南开大学 网络空间安全学院

2021/2022

本章知识点

- 计算机编程语言
- 虚拟机的概念
- 数据的表示方法
- 字符集
- 字节序

计算机编程语言

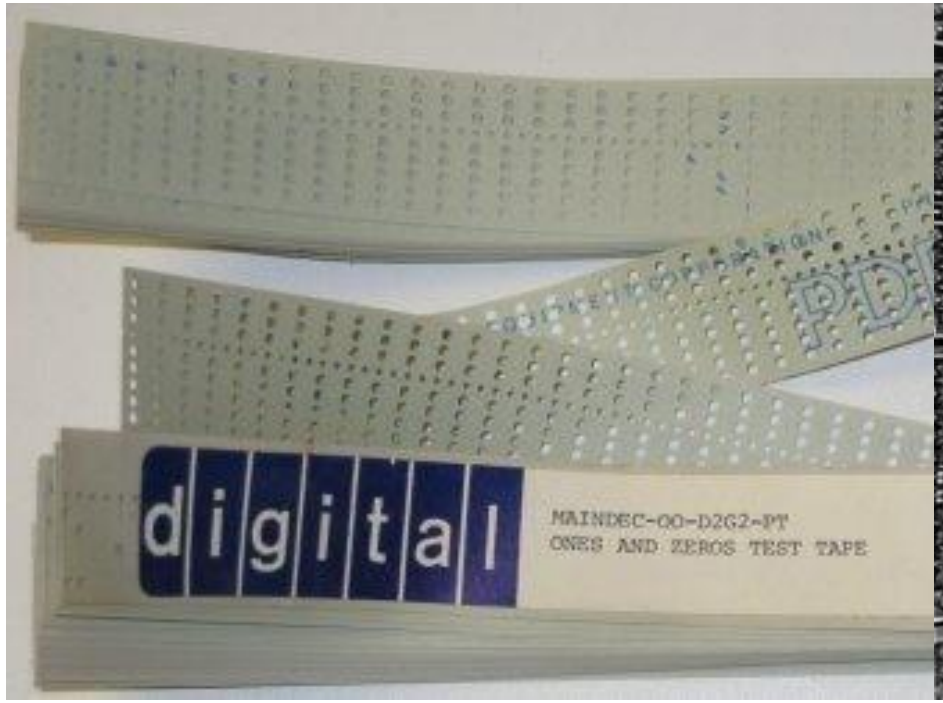
计算机编程语言

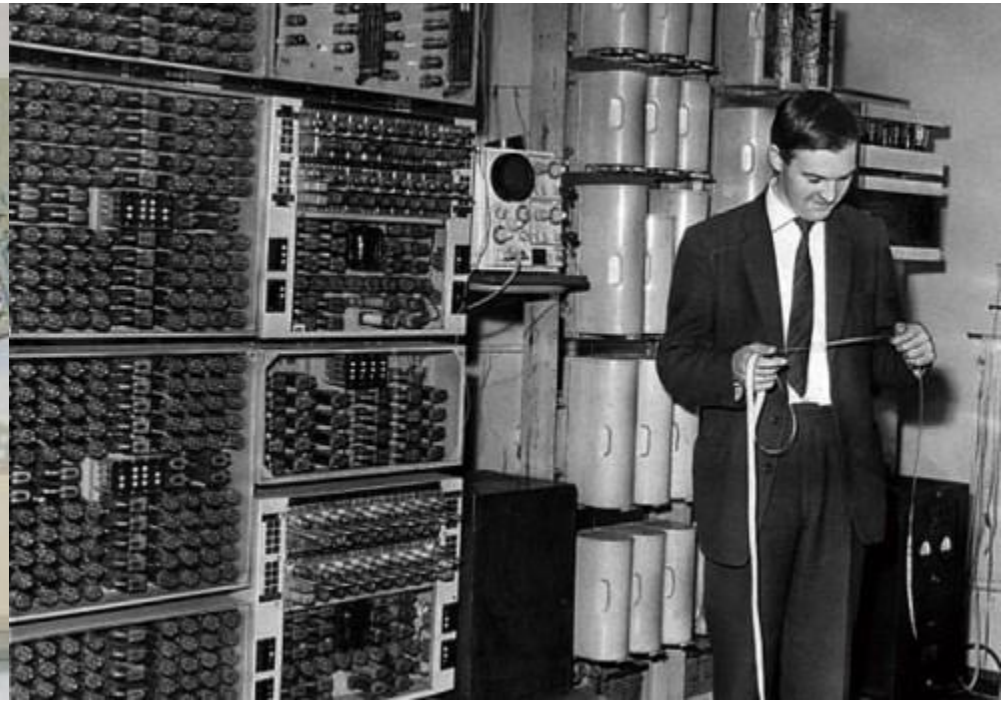
- 机器语言
- 汇编语言
- 高级语言

计算机编程语言

- 1946年第一台计算机问世
- 人机交互成为研究方向
- 更高效、更简便
  - 二进制机器语言
  - 助记符汇编语言（1949年）
  - 可以跨平台编译和执行的高级语言

机器语言





汇编语言

- **助记符**代替机器语言中的二进制**操作码**
- **地址符号**或**标号**，代替指令或操作数的二进制**地址**

汇编语言

```
00F8B8FD  .v 75 0C          jnz     short 00F8B90B
00F8B8FF  .  8BC6           mov     eax, esi
00F8B901  .  0D 11470000    or      eax, 0x4711
00F8B906  .  C1E0 10        shl     eax, 0x10
00F8B909  .  0BF0           or      esi, eax
00F8B90B  >  8935 B82EF900  mov     dword ptr ds:[0xF92EB8], esi
00F8B911  .  F7D6           not     esi
00F8B913  .  8935 BC2EF900  mov     dword ptr ds:[0xF92EBC], esi
00F8B919  .  5E             pop     esi
00F8B91A  >  5F             pop     edi
00F8B91B  .  5B             pop     ebx
00F8B91C  .  C9             leave
00F8B91D  L. C3             retn
00F8B91E     CC             int3
00F8B91F     CC             int3
00F8B920     CC             int3
00F8B921     CC             int3
00F8B922     CC             int3
00F8B923  >  8BFF           mov     edi, edi
00F8B925  r. 55             push    ebp
00F8B926  .  8BEC           mov     ebp, esp
00F8B928  .  81EC 28030000  sub     esp, 0x328
00F8B92E  .  A3 6090FA00    mov     dword ptr ds:[0xFA9060], eax
00F8B933  .  890D 5C90FA00  mov     dword ptr ds:[0xFA905C], ecx
00F8B939  .  8915 5890FA00  mov     dword ptr ds:[0xFA9058], edx
00F8B93F  .  891D 5490FA00  mov     dword ptr ds:[0xFA9054], ebx
00F8B945  .  8935 5090FA00  mov     dword ptr ds:[0xFA9050], esi
```

地址　机器码　　汇编指令

高级语言

- 高级语言更接近数学语言或者自然语言
  - 偏数学语言：Pascal、Ocaml、R
    - 函数式编程、多范式编程、基于规则的编程
  - 偏自然语言：C++、Java、Python
    - 面向对象（类、继承、多态）、面向过程

高级语言

- 高级语言与汇编语言和机器语言之间是一对多的关系。
  - 一条高级语言指令，编译之后，对应着多条机器码

可移植

- 如果一种语言的程序源代码可以在多种计算机系统上编译并运行，那么就说这种语言就是可移植的。

可移植

- C++中也可以使用汇编语言
  - 使用高级结构和访问底层细节之间提供了一种**折中方案**
  - 直接访问硬件，会使C++程序完全**丧失**可移植性

嵌入式系统

- 嵌入式系统的内存空间小
  - 电话、汽车、空调、打印机、摄像头、显卡、声卡、调制解调器等等
  - 汇编语言可以节省内存空间

实时系统

- 仿真、监控等实时系统要求精确计量时间和实时响应
  - 高级语言不能完全控制编译器生成的机器码
  - 汇编语言可以完全控制机器码，执行速度快

游戏机

- 游戏机有专用的系统，要求程序在**大小**和**运行速度**两方面都要做高度优化。
  - 充分利用目标系统的专用硬件特性
  - 手动进行游戏速度优化

驱动程序

- 硬件设备需要驱动程序
  - 驱动程序把操作系统上通用的命令转换为对特殊硬件的具体细节操作的程序。
  - 打印机需要编写Windows、macOS、Linux等平台的打印驱动。

加密算法

- 高级语言的种种限制会阻碍位操作、数据加密等底层操作的有效实现
- 汇编语言会加快数据加密速度

逆向分析

- 软件调试
- 软件漏洞挖掘
- 计算机病毒分析
- 软件知识产权保护

虚拟机的概念

虚拟机的概念

- 抽象出计算机软硬件之间的相互关系
- 虚拟机：对于每个语言层次，将其想象成一台假想的计算机（虚拟机）。
- 高层虚拟机的程序，通过解释或者翻译的方式，在底层虚拟机上执行

虚拟机的层次

- 第5层：高级语言
- 第4层：汇编语言
- 第3层：操作系统
- 第2层：指令集体系结构
- 第1层：微结构
- 第0层：数字逻辑

虚拟机的层次

- 1. 微结构：芯片上特殊的微结构指令
- 2. 指令集：固化在处理器内部的指令集
- 3. 操作系统：定义交互命令的虚拟机
- 4. 汇编语言：助记符
- 5. 高级语言：功能强大

执行方式

- 翻译方式：高层虚拟机的程序被整体翻译成底层虚拟机程序，然后在底层虚拟机上执行；
- 解释方式：低层虚拟机对高层虚拟机的程序，逐条指令进行解码并执行；

数据的表示方法

数据的表示方法

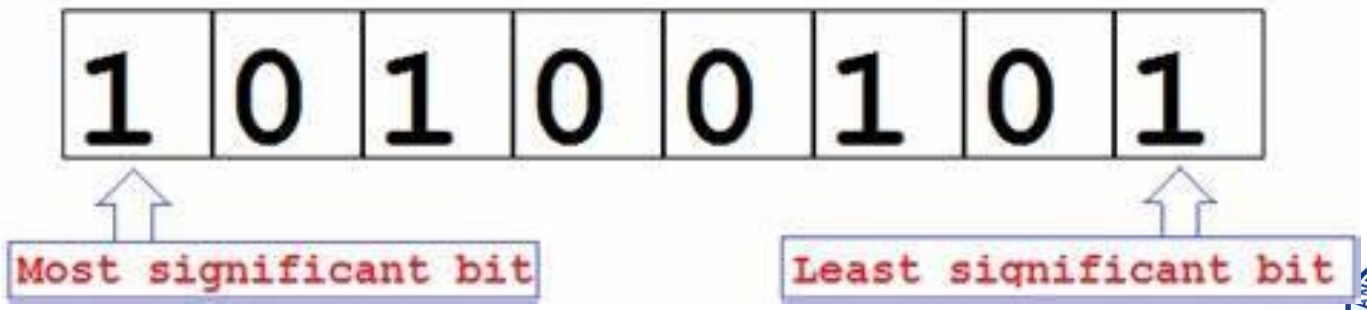
- 二进制
- 八进制
- 十进制
- 十六进制

二进制数

- 二进制数字以2为基数
- 每个二进制数字是一个比特位（bit）
- 位数从最右边的第0位开始计算，向左依次次递增

二进制数

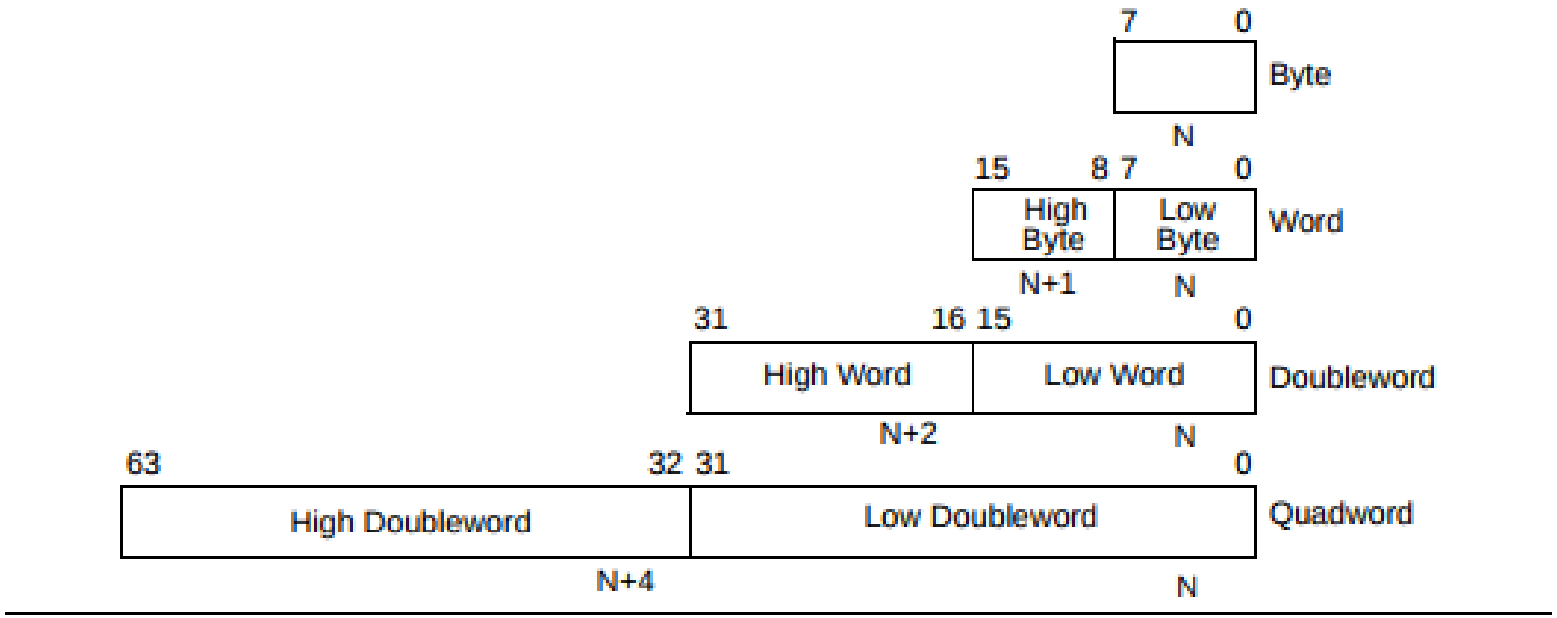
- 最左边的位称为最高有效位（**MSB，** Most Significant Bit）
- 最右边的位称为最低有效位（**LSB，** Least Significant Bit)



数据的存储单位

- 字节（byte）：包含8个bit位
- 字（word）：包含两个字节
- 双字（doubleword）：包含两个字
- 八字节（quadword）：包含两个双字

数据的存储单位


7
0
Byte
N
15
8
7
0
High Byte
Low Byte
Word
N+1
N
31
16
15
0
High Word
Low Word
Doubleword
N+2
N
63
32
31
0
High Doubleword
Low Doubleword
Quadword
N+4
N

数据的存储单位

- 1 kB = 1024 byte
- 1 MB = 1024 kB
- 1 GB = 1024 MB
- 1 TB = 1024 GB
- 1 PB = 1024 TB

十六进制

- 二进制数字的阅读不方便，十六进制使用更加方便
- 十六进制的每个数据可以表示4个二进制位
- 两个十六进制位就可以表示1个字节

十进制	十六进制	八进制	二进制
0	0	0	0000
1	1	1	0001
2	2	2	0010
3	3	3	0011
4	4	4	0100
5	5	5	0101
6	6	6	0110
7	7	7	0111
8	8	10	1000
9	9	11	1001
10	A	12	1010
11	B	13	1011
12	C	14	1100
13	D	15	1101
14	E	16	1110
15	F	17	1111
16	10	20	10000

字符集

字符集

- 字符集是一个系统支持的所有抽象字符的集和

文字

标点符号

图形符号

数字

ASCII字符集

- American Standard Code for Information Interchange
- 美国信息交换标准码
  - ASCII出现于20世纪50年代后期，于1967年定案

ASCII字符集

- ASCII是一个7位的编码标准，编码的取值范围实际上是00h-7Fh
  - 26个小写字母
  - 26个大写字母
  - 10个数字
  - 32个符号
  - 33个控制代码和空格

ASCII字符集

ASCII字符集

- 不同的计算机厂 商对ASCII进行了扩充，增加了128个附加字符，它们的值在**127以上的部分不是统一的**
  - ANSI字符集
  - Symbol字符集
  - OEM字符集

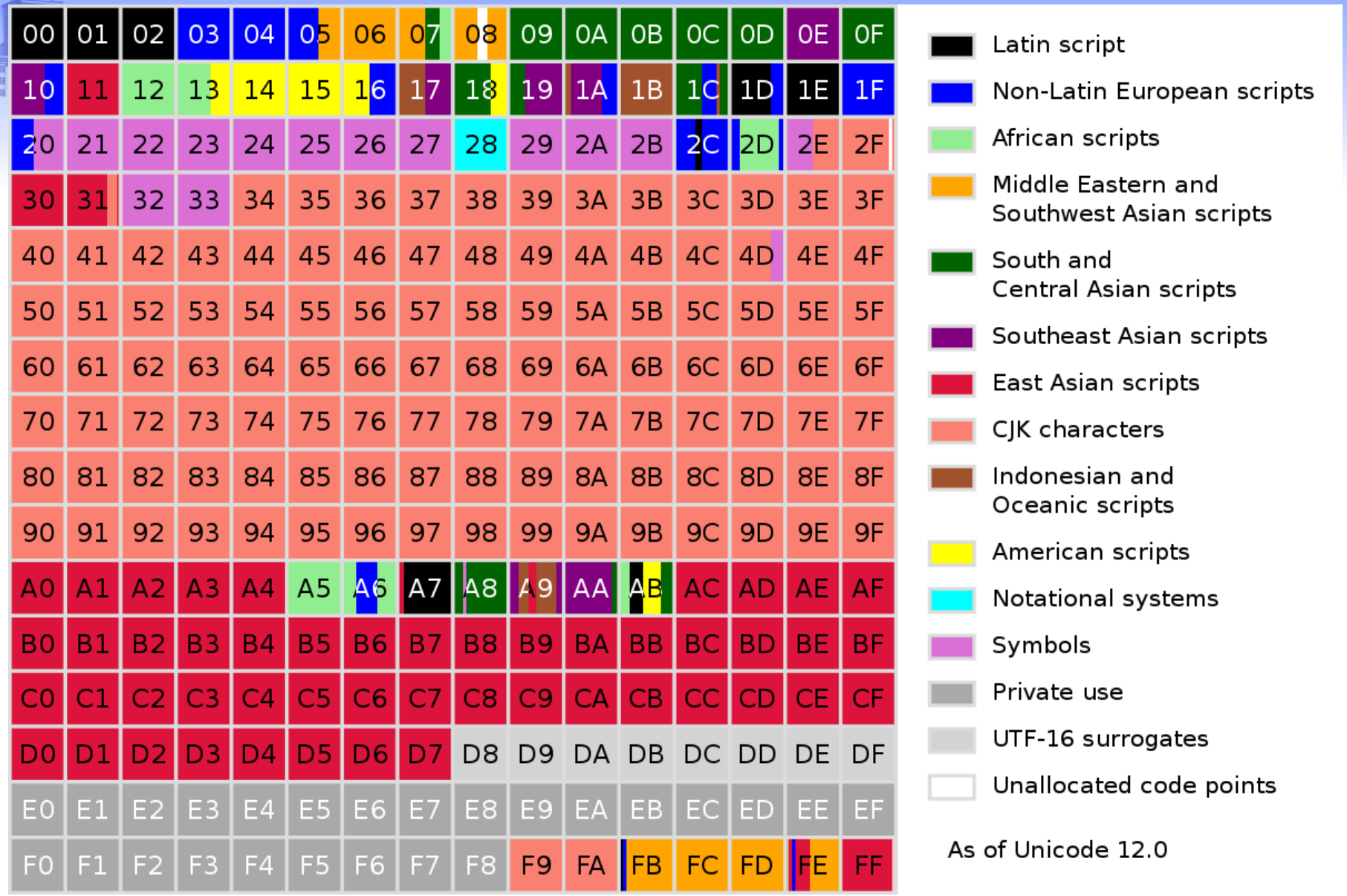
字符	ASCII值		字符	ASCII值		字符	ASCII值		字符	ASCII值		字符	ASCII值	
	DEC	HEX		DEC	HEX		DEC	HEX		DEC	HEX		DEC	HEX
Esc	27	1B	1	49	31	E	69	45	Y	89	59	m	109	6D
CR	13	0D	2	50	32	F	70	46	Z	90	5A	n	110	6E
LF	10	0A	3	51	33	G	71	47	[	91	5B	o	111	6F
Space	32	20	4	52	34	H	72	48	\	92	5C	p	112	70
!	33	21	5	53	35	I	73	49	]	93	5D	q	113	71
"	34	22	6	54	36	J	74	4A	^	94	5E	r	114	72
#	35	23	7	55	37	K	75	4B	_	95	5F	s	115	73
$	36	24	8	56	38	L	76	4C	`	96	60	t	116	74
&	37	25	9	57	39	M	77	4D	a	97	61	u	117	75
%	38	26	:	58	3A	N	78	4E	b	98	62	v	118	76
'	39	27	;	59	3B	O	79	4F	c	99	63	w	119	77
(	40	28	<	60	3C	P	80	50	d	100	64	x	120	78
)	41	29	=	61	3D	Q	81	51	e	101	65	y	121	79
*	42	2A	>	62	3E	R	82	52	f	102	66	z	122	7A
+	43	2B	?	63	3F	S	83	53	g	103	67	{	123	7B
,	44	2C	@	64	40	T	84	54	h	104	68	|	124	7C
-	45	2D	A	65	41	U	85	55	i	105	69	}	125	7D
.	46	2E	B	66	42	V	86	56	j	106	6A	~	126	7E
/	47	2F	C	67	43	W	87	57	k	107	6B	Del	127	7F
0	48	30	D	68	44	X	88	58	l	108	6C			

Unicode字符集

- 表示各种不同的国际语言
- Unicode为每种语言中的每个字符设定了统一并且唯一的二进制编码，以满足跨语言、跨平台进行文本转换、处理的要求。
- 1990年开始研发，1994年正式公布

Unicode字符集

- Unicode是ASCII字符编码的一个扩展
- 在Windows中用2字节对其进行编码，因此也被称为宽字符集(Widechars)。
- Unicode是一种双字节编码机制的字符集，使用0~65535的双字节无符号整数对每个字符进行编码。



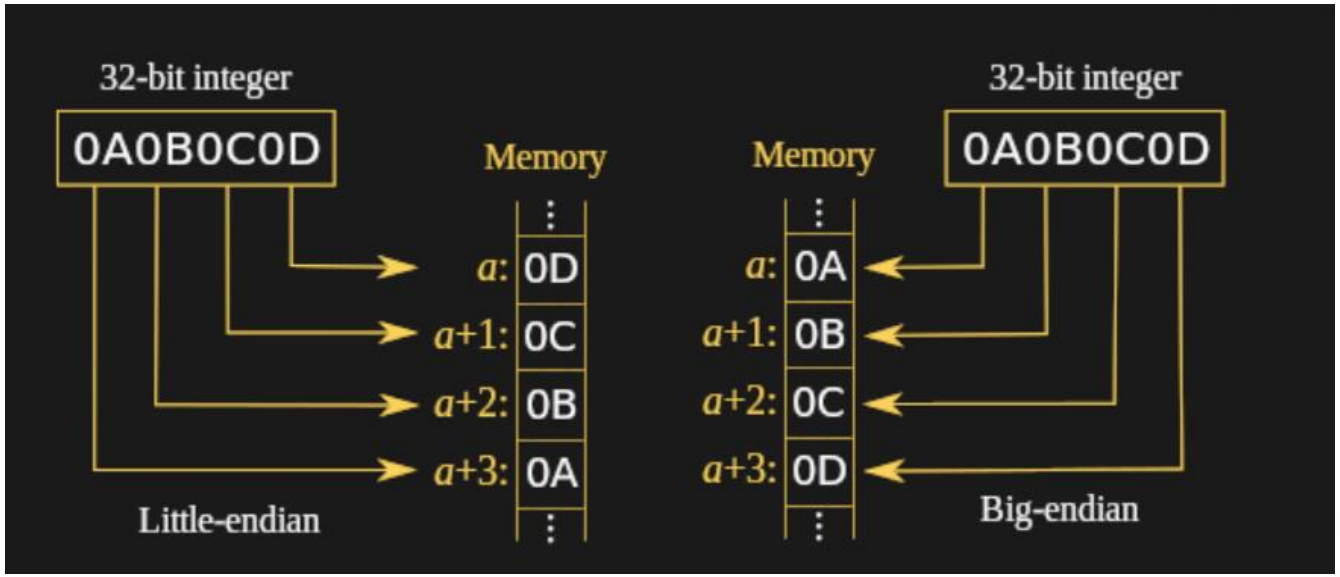
4DC0-4DFF：易经六十四卦符号 (Yijing Hexagrams Symbols)

南開大學 Nankai University

字节序

字节序

- 计算机领域在描述“关于字节该以什么样的顺序传送的争论”时引用了“endian”一词，翻译为“字节序”；
- 字节序表示数据在存储器中的存放顺序；



字节序

- 大端字节序(Big Endianness)；
- 小端字节序(Little Endianness)

大端字节序

- 一个多字节组成的数据，最高位被存储在内存的低地址上
  - 127.0.0.1，对应的正整数16进制表示0x7F000001
  - 大端字节序，表示为7F 00 00 01
  - 网络中的通信数据使用大端字节序,也称为网络字节序

小端字节序

- 一个多字节组成的数据，最低位被存储在内存的低地址上
  - 127.0.0.1，对应的正整数16进制表示0x7F000001
  - 小端字节序，表示为01 00 00 7F
  - Windows程序中使用小端字节序

字节序

- 将12345678h写入以1000h开始的内存中，以Big-endian和Little-endian模式存放

存放顺序	1000h	1001h	1002h	1003h
大端字节序	12	34	56	78
小端字节序	78	56	34	12

汇编语言与逆向技术

汇编语言基本概念

王志
zwang@nankai.edu.cn

南开大学 网络空间安全学院

2021/2022