

ICS 笔记（以章节划分）

第二章

一 • 基本概念

2.1.1

1. 字节：8 位的块，最小的可寻址的内存单位
2. 虚拟：机器级程序将内存视为一个非常大的字节数组
3. 地址：内存的每个字节都由 一个唯一的数字来标识
4. 虚拟地址空间：所有可能地址的集合
5. 程序对象：即程序数据、指令和控制信息
6. C 语言中一个指针的值都是某个存储块的第一个字节的虚拟地址。
7. 十六进制表示法优点：二进制表示法太冗长，而十进制表示法与位模式的互相转化很麻烦
8. C 语言中，以 0x 或 0X 开头的数字常量被认为是十六进制的值

2.1.2

9. 字长：处理器一次能处理的字节数，指明指针数据的标称大小，决定虚拟地址空间的最大大小（字长 $w \sim$ 虚拟地址 $2^w - 1$ ）
10. 32 位机器：字长为 4 字节，虚拟地址空间为 $2^{32} \sim 4\text{GB}$ ，经过伪指令 `linux> gcc -m32 prog.c` 编译后，就可在 32/64 位机器上运行
- 64 位机器：字长为 8 字节，虚拟地址空间为 $2^{64} \sim 16\text{EB}$ ，经过伪指令 `linux> gcc -m64 prog.c` 编译后，就只能在 64 位机器上运行

C 声明		字节数	
有符号	无符号	32位	64位
[signed] char	unsigned char	1	1
short	unsigned short	2	2
int	unsigned	4	4
long	unsigned long	4	8
int32_t	uint32_t	4	4
int64_t	uint64_t	8	8
char *		4	8
float		4	4
double		8	8

图 2-3 基本 C 数据类型的典型大小(以字节为单位)。分配的字节数受程序是如何编译的影响而变化。本图给出的是 32 位和 64 位程序的典型值

11. 有符号：可以表示为 R；

无符号：只能为非负数

大部分数据类型都编码为有符号数值，除非有 `unsigned` 或 `u-int32_t`；`char` 是一个例外，尽管大多数编译器和机器将它们视为有符号数，但 C 标准不保证这一点

12. `unsigned long/unsigned long int/long unsigned/long unsigned int` 都是一个意思

13. 可移植性的一个方面就是使程序对不同数据类型的确切大小不敏感。假设一个声明为 `int` 类型的程序对象能被用来存储一个指针。这在大多数 32 位的机器上能正常工作，但是在一台 64 位的机器上却会导致问题。

2.1.3

14. 多字节对象被存储为连续的字节序列，对象的地址为所使用字节中最小的地址

15.

二 • 公式定理

三 • 精选例题

1. 进制转换

(1) 将 `0xD5E4C` 转换为二进制

(2) 将二进制 `1001101110011110110101` 转换为十六进制

(3) 将十进制 `2048` 转换为十六进制

解：

(1) 查表法，`11010101111001001100`

(2) 从右至左，每四位一组，不足四位的高位补 0，得到 `0010 0110 1110 0111 1011 0101`，对应的十六进制数为 `0X26E7B`

(3) $2048 = 2^{11}$, $11 = 3 + 4 * 2$ ，故十六进制为 `0x800`

模块 2：图论

一 • 基本概念

二 • 公式定理

三 • 精选例题