

## 程序结构和执行

### P21-30

## 信息的处理与表示

数字表示：

- 无符号(unsigned)编码
- 补码(two's-complement)编码
- 浮点数编码

用有限数字位对应一个数字编码，所以可能会溢出。

#### 1. 信息存储

- 大多数计算机使用字节作为最小寻址内存单位。
- 机器程序将内存视为一个非常大的字节数组，即虚拟内存。内存每个字节都有自己的地址标识。
- 即将随机动态访问存储器、闪存、磁盘存储器、硬件和操作系统软件结合起来提供的字节数组
- c 语言指针的作用：<第三章重点介绍> 值都是某个存储块的第一个字节虚拟地址。

#### 2. 16 进制表示法：

将二进制数转换为十六进制数：

|      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|
| 十六进制 | 1    | 7    | 3    | A    | 4    | C    |
| 二进制  | 0001 | 0111 | 0011 | 1010 | 0100 | 1100 |

这样就得到了二进制表示 000101110011101001001100。

反过来，如果给定一个二进制数字 1111001010110110011，可以通过首先把它分为 4 位一组来转换为十六进制。不过要注意，如果位总数不是 4 的倍数，最左边的一组可少于 4 位，前面用 0 补足。然后将每个 4 位组转换为相应的十六进制数字：

|      |    |      |      |      |      |      |
|------|----|------|------|------|------|------|
| 二进制  | 11 | 1100 | 1010 | 1101 | 1011 | 0011 |
| 十六进制 | 3  | C    | A    | D    | B    | 3    |

练习题：

- 完成数字转换
- 0x39A7F8 转二进制：

001110011010011111111000

· 二进制 1100100101111011 转 16 进制:

0xc97b

### 3. 字数据大小

a. 每台计算机的位即为字长，决定虚拟地址空间的最大大小。字长为  $w$ ，则最多访问  $2$  的  $w$  次方字节

b. 64 位计算机可以运行 32 位程序，向下兼容

`gcc -m32 prog.c`

c. 32 位字长限制虚拟空间大小为 4GB，64 位则为 16GB

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

Plain Text

对于任何数据类型  $T$ ，声明

`T *p`

表示 `p` 是指针变量，指向类型为 `T` 的对象，比如

```
char *p
```

## 1. 寻址和字节顺序

- a. 几乎在所有机器上，多字节对象均存储为连续的字节序列，其中最小地址的字节为对象地址；
- b. 地址表达式：`&x`
- c. 字节顺序非常重要，比如如下几种情况：
  - i. 大小端机器在互相直接传递数据时，会反序。为了避免这类问题，必须按网络标准进行转换。11 章会接触到。
  - ii. 阅读指令中的整数数据时；
  - iii. 系统级编程往往需要使用强制类型转换来允许一种数据类型来引用另一种类型的对象；