P31-35

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. 字节顺利非常重要，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，字节顺序决定了转换的正确性。

C 语言 Tips

强制访问例程：

```
#include <stdio.h>

typedef unsigned char *byte_pointer;

void show_bytes(byte_pointer start, size_t len) {
    size_t i;
    for (i = 0; i < len; i++)
        printf(" %.2x", start[i]);
    printf("\n");
}

void show_int(int x) {
    show_bytes((byte_pointer) &x, sizeof(int));
}

void show_float(float x) {
    show_bytes((byte_pointer) &x, sizeof(float));
}

void show_pointer(void *x) {
    show_bytes((byte_pointer) &x, sizeof(void *));
}
```

- 使用 **typedef** 来给数据类型命名。改善代码可读性

```C
typedef int *int_pointer;
int_pointer ip;
//等同于：
int *ip;
```

- 使用 **printf** 格式化输出。第一个参数是格式串，其余都是要打印的值。

```C
//格式串中每个%开头的字符序列都表示如何格式化下一个打印参数。比如：
1. %d 十进制整数
2. %f 浮点数
3. %c 字符
对于需要指定大小的数据类型如 int32_t 可以参见 2.2.3
//同一个数值，用不同类型表示时，有截然不同的字节模式
```

- 指针和数组，指针的创建和直接引用

```C
//可以通过数组表示法来引用指针，也能用指针来引用数组元素
```

- ascII 字符码表：

```C
man ascii
```

1. 字符串表示：
   a. c 语言中字符串被编码为一个以 null（0）结尾的字符数组
2. 代码表示：

a. 不同机器类型使用不同的指令编码，并不兼容。因此二进制代码是基本无法移植的。

3. 布尔代数简介