

2020 春第 2 章家庭作业

教材（第二版）P80-90，家庭作业：

2.58, 2.71, 2.84(有更正), 2.88, 2.92(有更正)。

2.58 编写过程 `is_little_endian`，当在小端格式机器上编译和运行时返回 1，在大端格式机器上编译运行时则返回 0。这个程序应该可以运行在任何机器上，无论机器的字长是多少。

2.71 你刚刚开始在一家公司工作，他们要实现一组过程来操作一个数据结构，要将 4 个有符号字节封装成一个 32 位 `unsigned`。一个字中的字节从 0（最低有效字节）编号到 3（最高有效字节）。分配给你的任务是：为一个使用补码运算和算术右移的机器编写一个具有如下原型的函数：

```
/* Declaration of data type where 4 bytes are packed into an unsigned */
typedef unsigned packed_t;
```

```
/* Extract byte from word. Return as signed integer */
int xbyte(packed_t word, int bytenum)
```

也就是说，函数会抽取出指定的字节，再把它符号扩展为一个 32 位 `int`。

你的前任（因为水平不够高而被解雇了）编写了下面的代码：

```
/* Failed attempt at xbyte */
int xbyte(packed_t word, int bytenum)
{
    return (word >> (bytenum << 3)) & 0xFF;
}
```

A. 这段代码错在哪里？

B. 给出函数的正确实现，只能使用左右移位和一个减法。

2.84 给定一个浮点格式，有 k 位指数和 n 位小数，对于下列数，写出阶码 E 、尾数 M 、小数 f 和值 V 的公式。另外，请描述其位表示。

A. 数 5.0。

B. 能够被准确描述的最大奇整数。

C. 最小的正规格化数的倒数。

2.88 我们在一个 `int` 类型为 32 位补码表示的机器上运行程序。`float` 类型的值使用 32 位 IEEE 格式，而 `double` 类型的值使用 64 位 IEEE 格式。

我们产生随机整数 `x`、`y` 和 `z`，并且把它们转换成 `double` 类型的值：

```
/* Create some arbitrary values */
int x = random();
int y = random();
int z = random();
/* Convert to double */
double dx = (double) x;
double dy = (double) y;
double dz = (double) z;
```

对于下列的每个 C 表达式，你要指出表达式是否总是为 1。如果它总是为 1，描述其中的数学原理。否则，列举出使它为 0 的参数的例子。请注意，不能使用 IA32 机器运行 GCC 来测试你的答案，因为对于 `float` 和 `double`，它使用的都是 80 位的扩展精度表示。

- A. `(double) (float) x == dx`
- B. `dx + dy == (double) (x+y)`
- C. `dx + dy + dz == dz + dy + dx`
- D. `dx * dy * dz == dz * dy * dx`
- E. `dx / dx == dy / dy`

2.92. 遵循位级浮点编码规则，实现具有如下原型的函数：

```
/* Compute -f. If f is NaN, then return f. */
float_bits float_negate(float_bits f);
```

对于浮点数 f ，这个函数计算 $-f$ 。如果 f 是 *NaN*，你的函数应该简单的返回 f 。测试你的函数，对参数 f 可以取的所有 2^{32} 个值求值，将结果与你使用机器的浮点运算得到的结果相比较。