

struct

630次阅读

---

准确地讲，Python没有专门处理字节的数据类型。但由于str既是字符串，又可以表示字节，所以，字节数组=str。而在C语言中，我们可以很方便地用struct、union来处理字节，以及字节和int，float的转换。

在Python中，比方说要把一个32位无符号整数变成字节，也就是4个长度的str，你得配合位运算符这么写：

```
>>> n = 10240099
>>> b1 = chr((n & 0xff000000) >> 24)
>>> b2 = chr((n & 0xff0000) >> 16)
>>> b3 = chr((n & 0xff00) >> 8)
>>> b4 = chr(n & 0xff)
>>> s = b1 + b2 + b3 + b4
>>> s
'\x00\x9c@c'
```

非常麻烦。如果换成浮点数就无能为力了。

好在Python提供了一个struct模块来解决str和其他二进制数据类型的转换。

struct的pack函数把任意数据类型变成字符串：

```
>>> import struct
>>> struct.pack('>I', 10240099)
'\x00\x9c@c'
```

pack的第一个参数是处理指令，'>I'的意思是：

>表示字节顺序是big-endian，也就是网络序，I表示4字节无符号整数。

后面的参数个数要和处理指令一致。

unpack把str变成相应的数据类型：

```
>>> struct.unpack('>IH', '\xf0\xf0\xf0\xf0\x80\x80')
(4042322160, 32896)
```

根据>IH的说明，后面的str依次变为I：4字节无符号整数和H：2字节无符号整数。

所以，尽管Python不适合编写底层操作字节流的代码，但在对性能要求不高的地方，利用struct就方便多了。

struct模块定义的数据类型可以参考Python官方文档：

<https://docs.python.org/2/library/struct.html#format-characters>

Windows的位图文件（.bmp）是一种非常简单的文件格式，我们来用struct分析一下。

首先找一个bmp文件，没有的话用“画图”画一个。

读入前30个字节来分析：

```
>>> s = '\x42\x4d\x38\x8c\x0a\x00\x00\x00\x00\x00\x36\x00\x00\x00\x28\x00\x00\x00\x80\x02\x00\x00\x68\x01\x00\x00\x01\x00\x18\x00'
```

BMP格式采用小端方式存储数据，文件头的结构按顺序如下：

两个字节：'BM'表示Windows位图，'BA'表示OS/2位图； 一个4字节整数：表示位图大小； 一个4字节整数：保留位，始终为0； 一个4字节整数：实际图像的偏移量； 一个4字节整数：Header的字节数； 一个4字节整数：图像宽度； 一个4字节整数：图像高度； 一个2字节整数：始终为1； 一个2字节整数：颜色数。

所以，组合起来用unpack读取：

```
>>> struct.unpack('<ccIIIIIIHH', s)
('B', 'M', 691256, 0, 54, 40, 640, 360, 1, 24)
```

结果显示，'B'、'M'说明是Windows位图，位图大小为640x360，颜色数为24。

请编写一个bmpinfo.py，可以检查任意文件是否是位图文件，如果是，打印出图片大小和颜色数。