

嘎啦

简单的幸福

博客园 首页 新随笔 管理

随笔-17 文章-6 评论-14

公告

昵称：Gala
园龄：5年8个月
粉丝：24
关注：53
+加关注

搜索

谷歌搜索

随笔分类

- C/C++(9)
- Design Pattern(1)
- GTK+(1)
- Linux系统应用(4)
- Python(1)

开发相关

- C++设计模式
- 嵌入式基础

积分与排名

积分 - 24447
排名 - 11840

推荐排行榜

Python使用struct处理二进制

有的时候需要用python处理二进制数据，比如，存取文件，socket操作时.这时候，可以使用python的struct模块来完成.可以用struct来处理c语言中的结构体。

struct模块中最重要的三个函数是pack(), unpack(), calcsiz ()

pack(fmt, v1, v2, ...) 按照给定的格式(fmt)，把数据封装成字符串(实际上是类似于c结构体的字节流)

unpack(fmt, string) 按照给定的格式(fmt)解析字节流string，返回解析出来的tuple

calcsiz (fmt) 计算给定的格式(fmt)占用多少字节的内存

struct中支持的格式如下表：

Format	C Type	Python	字节数
x	pad byte	no value	1
c	char	string of length 1	1
b	signed char	integer	1
B	unsigned char	integer	1
?	_Bool	bool	1
h	short	integer	2
H	unsigned short	integer	2

1. Python使用struct处理二进制(9)
2. 动态数组空间释放——指针误操作(3)
3. Fedora12配置OpenCV开发环境(2)
4. 指针与数组(2)
5. std::vector对象操作(1)

Format	C Type	Python	字节数
i	int	integer	4
I	unsigned int	integer or long	4
l	long	integer	4
L	unsigned long	long	4
q	long long	long	8
Q	unsigned long long	long	8
f	float	float	4
d	double	float	8
s	char[]	string	1
p	char[]	string	1
P	void *	long	

- 注1.q和Q只在机器支持64位操作时有意思
- 注2.每个格式前可以有一个数字，表示个数
- 注3.s格式表示一定长度的字符串，4s表示长度为4的字符串，但是p表示的是pascal字符串
- 注4.P用来转换一个指针，其长度和机器字长相关
- 注5.最后一个可以用来表示指针类型的，占4个字节

为了同c中的结构体交换数据，还要考虑有的c或c++编译器使用了字节对齐，通常是以4个字节为单位的32位系统，故而struct根据本地机器字节顺序转换.可以用格式中的第一个字符来改变对齐方式.定义如下：

Character	Byte order	Size and alignment
@	native	native 凑够4个字节

Character	Byte order	Size and alignment	
=	native	standard	按原字节数
<	little-endian	standard	按原字节数
>	big-endian	standard	按原字节数
!	network (= big-endian)	standard	按原字节数

使用方法是放在fmt的第一个位置，就像 '@5s6sif'

示例一：

比如有一个结构体

struct Header

```
{
    unsigned short id;
    char[4] tag;
    unsigned int version;
    unsigned int count;
}
```

通过socket.recv接收到了一个上面的结构体数据，存在字符串s中，现在需要把它解析出来，可以使用unpack()函数。

```
import struct
```

```
id, tag, version, count = struct.unpack("!H4s2I", s)
```

上面的格式字符串中，!表示我们要使用网络字节顺序解析，因为我们的数据是从网络中接收到的，在网络上传送的时候它是网络字节顺序的。后面的H表示 一个unsigned short的id,4s表示4字节长的字符串，2I表示有两个unsigned int类型的数据。

就通过一个unpack, 现在id, tag, version, count里已经保存好我们的信息了.

同样, 也可以很方便的把本地数据再pack成struct格式.

```
ss = struct.pack("!H4s2I", id, tag, version, count);
```

pack函数就把id, tag, version, count按照指定的格式转换成了结构体Header, ss现在是一个字符串(实际上是类似于c结构体的字节流), 可以通过 socket.send(ss)把这个字符串发送出去.

示例二:

```
import struct
```

```
a=12.34
```

```
#将a变为二进制
```

```
bytes=struct.pack('i',a)
```

此时bytes就是一个string字符串, 字符串按字节同a的二进制存储内容相同。

再进行反操作

现有二进制数据bytes, (其实就是字符串), 将它反过来转换成python的数据类型:

```
a,=struct.unpack('i',bytes)
```

注意, unpack返回的是tuple

所以如果只有一个变量的话:

```
bytes=struct.pack('i',a)
```

那么, 解码的时候需要这样

```
a,=struct.unpack('i',bytes) 或者 (a,)=struct.unpack('i',bytes)
```

如果直接用a=struct.unpack('i',bytes), 那么 a=(12.34,) , 是一个tuple而不是原来的浮点数了。

如果是由多个数据构成的，可以这样：

```
a='hello'
```

```
b='world!'
```

```
c=2
```

```
d=45.123
```

```
bytes=struct.pack('5s6sif,a,b,c,d')
```

此时的bytes就是二进制形式的数据了，可以直接写入文件比如 `binfile.write(bytes)`

然后，当我们需要时可以再读出来，`bytes=binfile.read()`

再通过`struct.unpack()`解码成python变量

```
a,b,c,d=struct.unpack('5s6sif,bytes')
```

'5s6sif'这个叫做fmt，就是格式化字符串，由数字加字符构成，5s表示占5个字符的字符串，2i，表示2个整数等等，下面是可用的字符及类型，ctype表示可以与python中的类型一一对应。

注意：二进制文件处理时会碰到的问题

我们使用处理二进制文件时，需要用如下方法

```
binfile=open(filepath,'rb') 读二进制文件
```

```
binfile=open(filepath,'wb') 写二进制文件
```

那么和`binfile=open(filepath,'r')`的结果到底有何不同呢？

不同之处有两个地方：

第一，使用'r'的时候如果碰到'0x1A'，就会视为文件结束，这就是EOF。使用'rb'则不存在这个问题。即，如果你用二进制写入再用文本读出的话，如果其中存在'0x1A'，就只会读出文件的一部分。使用'rb'的时候会一直读到文件末尾。

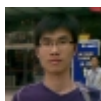
第二，对于字符串x='abc\ndef'，我们可用len(x)得到它的长度为7，\n我们称之为换行符，实际上是'0X0A'。当我们用'w'即文本方式写的时候，在windows平台上会自动将'0X0A'变成两个字符'0X0D', '0X0A'，即文件长度实际上变成8.。当用'r'文本方式读取时，又自动的转换成原来的换行符。如果换成'wb'二进制方式来写的话，则会保持一个字符不变，读取时也是原样读取。所以如果用文本方式写入，用二进制方式读取的话，就要考虑这多出的一个字节了。'0X0D'又称回车符。linux下不会变。因为linux只使用'0X0A'来表示换行。

分类: Python

好文要顶

关注我

收藏该文



Gala

关注 - 53

粉丝 - 24

+加关注

« 上一篇: 控制台密码输入控制

» 下一篇: typedef用法详解

9

0

posted @ 2011-09-22 10:36 Gala 阅读(85875) 评论(5) 编辑 收藏

评论列表

#1楼 2014-05-29 12:56 规格严格-功夫到家

写的不错啊。.....

支持(0) 反对(0)

#2楼 2014-08-20 16:16 niceheart

博主，请问一下类似于这个的数据ss = struct.pack("!H4s2I", id, tag, version, count);在java中又该如何处理才能和这种格式一样？

支持(0) 反对(0)

#3楼 2015-07-01 11:11 cool_navy