5.7 读写压缩文件¶

问题¶

你想读写一个gzip或bz2格式的压缩文件。

解决方案¶

gzip 和 bz2 模块可以很容易的处理这些文件。 两个模块都为 open() 函数提供了另外的实现来解决这个问题。 比如，为了以文本形式读取压缩文件，可以这样做：

```
# gzip compression
import gzip
with gzip.open('somefile.gz', 'rt') as f:
    text = f.read()

# bz2 compression
import bz2
with bz2.open('somefile.bz2', 'rt') as f:
    text = f.read()
```

类似的，为了写入压缩数据，可以这样做：

```
# gzip compression
import gzip
with gzip.open('somefile.gz', 'wt') as f:
    f.write(text)

# bz2 compression
import bz2
with bz2.open('somefile.bz2', 'wt') as f:
    f.write(text)
```

如上，所有的I/O操作都使用文本模式并执行Unicode的编码/解码。 类似的，如果你想操作二进制数据，使用 rb 或者 wb 文件模式即可。

讨论¶

大部分情况下读写压缩数据都是很简单的。但是要注意的是选择一个正确的文件模式是非常重要的。 如果你不指定模式，那么默认的就是二进制模式，如果这时候程序想要接受的是文本数据，那么就会出错。 gzip.open() 和 bz2.open() 接受跟内置的 open() 函数一样的参数， 包括 encoding，errors，newline 等等。

当写入压缩数据时，可以使用 compresslevel 这个可选的关键字参数来指定一个压缩级别。比如：

```
with gzip.open('somefile.gz', 'wt', compresslevel=5) as f:
    f.write(text)
```

默认的等级是9，也是最高的压缩等级。等级越低性能越好，但是数据压缩程度也越低。

最后一点， gzip.open() 和 bz2.open() 还有一个很少被知道的特性， 它们可以作用在一个已存在并以二进制模式打开的文件上。比如，下面代码是可行的：

```
import gzip
f = open('somefile.gz', 'rb')
with gzip.open(f, 'rt') as g:
    text = g.read()
```

这样就允许 gzip 和 bz2 模块可以工作在许多类文件对象上，比如套接字，管道和内存中文件等。