# 1 Linux下查看版本方法：

cat /proc/version

Linux version 3.10.0-123.4.4.el7.x86\_64 (builder@kbuilder.dev.centos.org) (gcc version 4.8.2 20140120 (Red Hat 4.8.2-16) (GCC) ) #1 SMP Fri Jul 25 05:07:12 UTC 2014

x86\_64架构，centos7，gcc版本为4.8

python安装三方包时选择对应架构的三方包即可。

# 2 cp27-cp27m与cp27-cp27mu

在https://pypi.python.org/pypi下载时候

看到cp27除了none外，还有2种包

一个是cp27m

一个是cp27mu

搜了一下区别

ucs2和ucs4

https://github.com/pypa/manylinux

Note that prior to CPython 3.3, there were two ABI-incompatible waysof building CPython: --enable-unicode=ucs2 and--enable-unicode=ucs4. We provide both versions(e.g. /opt/python/cp27-cp27m for narrow-unicode,/opt/python/cp27-cp27mu for wide-unicode). NB: essentially allLinux distributions configure CPython in mu(--enable-unicode=ucs4) mode, but --enable-unicode=ucs2 buildsare also encountered in the wild. Other less common or virtuallyunheard of flag combinations (such as --with-pydebug (d) and--without-pymalloc (absence of m)) are not provided.

注意，本文内容基于python 2。python 3的处理方式可能有所不同，有需要的读者可以自己了解一下。

# 3 wheel 包的命名规定

wheel 包的命名格式为 {distribution}-{version}(-{build tag})?-{python tag}-{abi tag}-{platform tag}.whl。  
其中各个 tag 的意义和取值在 PEP425 中有规定：[https://www.python.org/dev/pe...](https://www.python.org/dev/peps/pep-0425/)

python tag 标记了具体的 python 实现。其中：

* py 无实现特定的拓展
* cp CPython，也就是通常使用的 Python 实现
* ip IronPython，跑在 Windows CLI 平台上的 Python 实现
* pp PyPy，带 JIT 的 Python 实现
* jy Jython，跑在 JVM 上的 Python 实现

举个例子，如果 wheel 包里面包含了 C 拓展，那么打包出来的 python tag 就是 cpxx，其中 xx 是具体的版本号，如 cp27.

platform tag 也好理解，就是系统 \_ 架构。比如 linux\_x86\_64。

最陌生的恐怕是之间的 abi tag，这正是本文讨论的主题。

abi 这东西，看不见摸不着。系统上的东西嘛，敲下个命令就知道是什么操作系统；架构虽然玄乎点，不过也就是那么几种；然而有多少人知道自己当前使用的平台遵循着怎样的 abi 标准？什么时候 abi 可以兼容，什么时候又不可以？

**pip wheel 打包时，abi tag是怎么敲定的**

Python 对此有另外一个 PEP：[https://www.python.org/dev/pe...](https://www.python.org/dev/peps/pep-3149/)

如果 sysconfig 定义了 SOABI，那么就用 SOABI 的值。当然这是 Python 3 的事务，这里我自然不用管。如果没定义 SOABI，比如 Python 2，wheel 会生成一个类似的 abi tag。在 Python 的标准里，这个 abi 取决于打包时使用的 Python 实现。举 CPython 为例，首先必须包含的是实现名和版本号，比如 CPython2.7.9 对应的是 cp27.其次，需要包含构建 CPython 时特定的选项。具体来说，打包时会依次判断当前的 CPython 是否有下列的功能，如果有，加上对应的 flag：

1. --with-pydebug (flag: d )
2. --with-pymalloc (flag: m )
3. --with-wide-unicode (flag: u )

通常，我们看到的 abi tag 会是这样的 cp27mu，这是因为 --with-pymalloc 是默认开启的，而包管理中分发的 CPython 会加上 --with-wide-unicode 选项。

有趣的是，如果打包时没办法判断 abi 类型，生成的 abi tag 会是 none。而如果 Python 包是不依赖特定的 abi 的纯 Python 实现，生成的 abi tag 也是 none。在安装时，值为 none 的 abi tag 会享受特殊待遇。这个下文再说。

另外同样的 Python 代码打出来 abi tag 相同的包，不一定完全一样。以我的亲身经历举例，pycrypto 这个库，在打包的时候会判断 libgmp 是否存在，如果存在，就构建 \_fastmath 这个库。如果打包平台上存在 libgmp，打出来的包就会包含 \_fastmath。反之，则不存在。而这两种情形下打出来的包，名字是一模一样的。

**如何判断给定 wheel 包是否能够安装**

通常判断依赖的时候，需要看下是否符合最低版本。不过 pip 判断给定 wheel 包的 abi 兼容的做法与此有些许差异。pip 的做法是，计算出一个支持的 abi tag 集合，然后判断目标 abi tag 是否在这个集合里。这个计算过程跟在打包时是一样的。这意味着，打包拓展的 CPython 需要跟安装的机器上的 CPython 版本是一致的，否则就装不了。对于“永远的2.7”来说，这不是什么问题；不过如果用的是 Python 3，又不能控制具体的 CPython 版本，对于 C 拓展还是现场编译安装比较靠谱。

其实说了这么多，还不如跑一个脚本：  
/usr/local/lib/python2.7/site-packages/pip/pep425tags.py

pip 会运行这个脚本来判断 wheel 包。所以你只要

from pip import pep425tags

print(pep425tags.get\_supported())

就能报出该系统上支持的 wheel 包名字。

当然对于 abi tag 为 none 的包，它可以在任何一个 abi 版本上安装。因为所有的平台都至少支持 none abi。但如果一个平台上的 Python 如此古怪，以致于没办法确定它的 abi 类型，那么也就只能装上 abi tag 为 none 的包，即纯 Python 实现抑或同样古怪得无法判断打包时的 abi 的包。真是同类相聚啊。顺便一提，在 Ubuntu 14.04 和同期的系统，通过包管理安装的 pip 有一个已知 bug，有些时候无法正确判断 abi 类型，所以打出来的包的 tag 是 none，且任何带特定 abi 的包都装不上去。我们曾经遇到这个 bug，最后通过升级 pip 解决了。感兴趣的读者可以比较这包管理版本和最新版本，两个 pip 的 pep425tags.py 有什么异同。

# 4 python报错PyUnicodeUCS2\_AsUTF8String或者PyUnicodeUCS4\_AsUTF8String的问题

如果自己单独升级了python,或者有多个版本的python时,便有可能出现此问题.

问题表象为:报错:undefined symbol: PyUnicodeUCS2\_AsUTF8String或者undefined symbol: PyUnicodeUCS4\_AsUTF8String.

根本原因时python和某个你用的库编译时指定的UCS编码方式不对.

编译python时,可以通过指定--enable-unicode[=ucs[24]]来选择使用UCS2或者UCS4.

如果你的错误是undefined symbol: PyUnicodeUCS2\_AsUTF8String,说明你的python编译时使用的是UCS4,反之依然.

解决方案两个:1,重新编译python,2,重新编译库.

选择一般是重新编译库.

如果使用的是./configure这样的编译方式,如果你的./configure --help没有unicode的选择,那么在./configure之后,

在config.h内增加#define Py\_UNICODE\_SIZE 4,然后再make,make install之类的,就搞定拉.

原理如下:

在头文件/usr/include/python2.7/unicodeobject.h内:

#ifndef Py\_UNICODE\_WIDE

# define PyUnicode\_AsUTF8String PyUnicodeUCS2\_AsUTF8String

python2.7.11默认是使用UCS2.