**常用linux命令**

---->在我们平时对CentOS的软件安装中,yum命令在查找安装盘中的rpm比较快，现将收集的一些常用命令整理如下：

1、列出任何可更新的软件清单

yum check-update

2、安装任何更新软件

yum update

3、仅安装指定的软件

yum install \*\*\*

4、仅更新指定的软件

yum update \*\*\*

5、列出任何可安裝的软件清单

yum list

--->安装的命令是，yum install xxx，yum会查询数据库，有无这一软件包，假如有，则检查其依赖冲突关系，假如没有依赖冲突，那么最好，下载安装;假如有，则会给出提示，询问是否要同时安装依赖，或删除冲突的包，您能够自己作出判断。

删除的命令是，yum remove xxx，同安装相同，yum也会查询数据库，给出解决依赖关系的提示。

1、用YUM安装软件包

yum install \*\*\*

2、用YUM删除（卸载）软件包

yum remove \*\*\*

--->用YUM查询软件信息

我们常会碰到这样的情况，想要安装一个软件，只知道他和某方面有关，但又不能确切知道他的名字。这时yum的查询功能就起作用了。您能够用 yum search keyword这样的命令来进行搜索，比如我们要则安装一个Instant Messenger，但又不知到底有哪些，这时不妨用 yum search messenger这样的指令进行搜索，yum会搜索任何可用rpm的描述，列出任何描述中和messeger有关的rpm包，于 是我们可能得到gaim，kopete等等，并从中选择。

有时我们还会碰到安装了一个包，但又不知道其用途，我们能够用yum info packagename这个指令来获取信息。

1、使用YUM查找软件包

yum search \*\*\*

2、列出任何可安装的软件包

yum list

3、列出任何可更新的软件包

yum list updates

4、列出任何已安装的软件包

yum list installed

5、列出任何已安装但不在 Yum Repository 內的软件包

yum list extras

6、列出所指定的软件包

yum list \*\*\*

7、使用YUM获取软件包信息

yum info \*\*\*

8、列出任何软件包的信息

yum info

9、列出任何可更新的软件包信息

yum info updates

10、列出任何已安裝的软件包信息

yum info installed

11、列出任何已安裝但不在 Yum Repository 內的软件包信息

yum info extras

12、列出软件包提供哪些文档

yum provides

--->清除YUM缓存

　　yum 会把下载的软件包和header存储在cache中，而不会自动删除。假如我们觉得他们占用了磁盘空间，能够使用yum clean指令进行清除，更精确 的用法是yum clean headers清除header，yum clean packages清除下载的rpm包，yum clean all一 股脑儿端 .

1、清除缓存目录(/var/cache/yum)下的软件包

yum clean packages

2、清除缓存目录(/var/cache/yum)下的 headers

yum clean headers

3、清除缓存目录(/var/cache/yum)下旧的 headers

yum clean oldheaders

4、清除缓存目录(/var/cache/yum)下的软件包及旧的headers

yum clean, yum clean all (= yum clean packages; yum clean oldheaders)

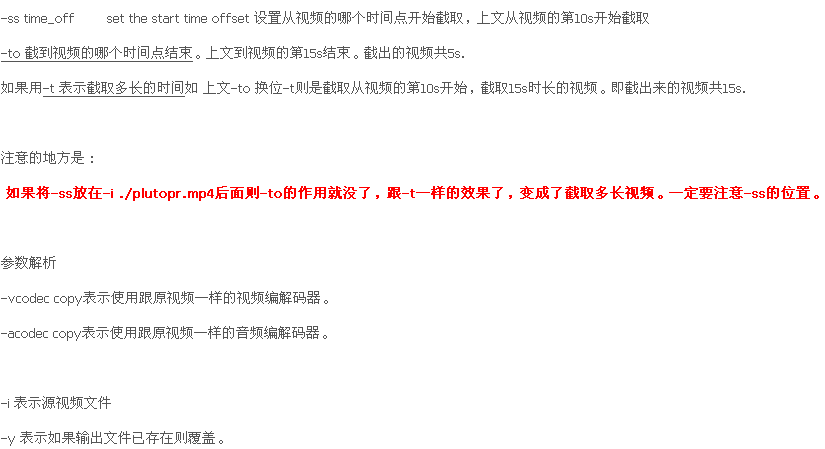
--->查看ffmpeg版本

ffmpeg -version

****

--->使用ffmpeg截取视频

ffmpeg -i aaa.avi -vcodec copy -acodec copy -ss 00:00:00 -to 00:00:59 bbb.avi -y



--->protobuf

google protobuf是一个灵活的、高效的用于序列化数据的协议。相比较XML和JSON格式，protobuf更小、更快、更便捷。google protobuf是跨语言的，并且自带了一个编译器(protoc)，只需要用它进行编译，可以编译成Java、python、C++、C#、Go等代码，然后就可以直接使用，不需要再写其他代码，自带有解析的代码。

--->git 上传步骤

（1）git status 查看状态

（2）git add <文件>

（3）git commit <文件>

（4）git push -u origin master

--->查看可用的GPU

nvidia-smi -L

--->如何查看tensorflow跑的是gpu版本还是cpu版本？

import tensorflow as tf

a = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[2, 3], name='a')

b = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[3, 2], name='b')

c = tf.matmul(a, b)

tf.Session(config=tf.ConfigProto(log\_device\_placement=True))

print(sess.run(c))

结果如图：



可见我的是跑CPU版本的

--->在github上添加密钥

生成 SSH 公钥

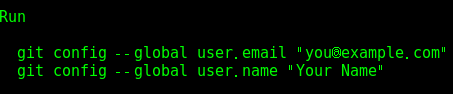
大多数 Git 服务器都会选择使用 SSH 公钥来进行授权。系统中的每个用户都必须提供一个公钥用于授权，没有的话就要生成一个。生成公钥的过程在所有操作系统上都差不多。 首先先确认一下是否已经有一个公钥了。SSH 公钥默认储存在账户的主目录下的 ~/.ssh 目录。进去看看：



（1）安装git

Yum install git

（2）初始化



这应该就可以上传了

--->查看GPU信息

lspci | grep -i vga

--->压缩本地文件命令

tar -cf bbb.tar bbb.avi

--->make -j8

-j8表示并行计算，根据自己电脑的配置进行设置，配置比较低的电脑可以将数字改小或不使用，直接输make。

--->查看网卡速度

Ethtool eno1