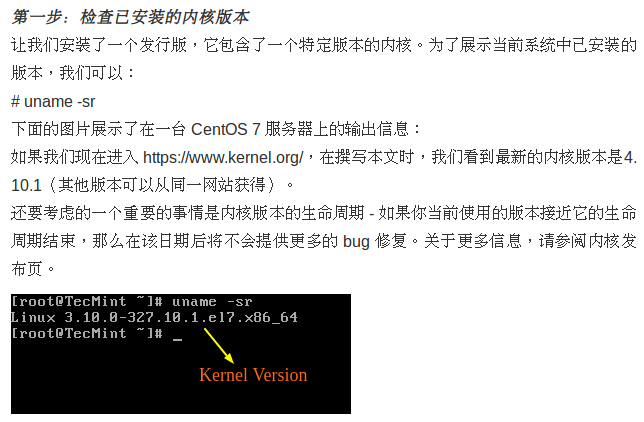
**0. Centos 7 内核升级**

升级办法参考

<http://jingyan.baidu.com/article/0a52e3f4ddc749bf62ed7221.html>

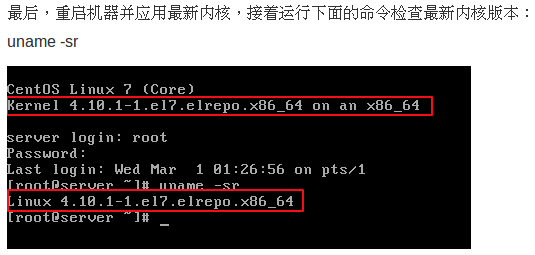




说明：上面的rpm命令是2条，需要分开执行。







**1. 安装gcc-4.9.4**

|  |
| --- |
| 先安装编译工具  yum install gcc  yum install gcc-c++ |
| 下载gcc-4.9.4.tar.gz  http://ftp.gnu.org/gnu/gcc/gcc-4.9.4/ |
| 解压  tar -zxvf gcc-4.9.4.tar.gz |
| 编译/安装  cd gcc-4.9.4  ./contrib/download\_prerequisties  mkdir build  cd build  ../configure --prefix=/usr/local/gcc-4.9.4 --enable-checking=release --enable-languages=c,c++ --disable-multilib  make -j40  make install  在/etc/profile里面加入: export PATH=“/usr/local/gcc-4.9.4/bin:$PATH” |

**2. 安装NVIDIA驱动**

|  |
| --- |
| 下载NVIDIA驱动  192.168.8.89服务器的驱动是：NVIDIA-Linux-x86\_64-384.66.run |
| 下载和kernel版本配套的kernel-ml-devel和kernel-ml-header文件  以服务器（192.168.8.89）为例，uname -r的结果是4.13.8-1.el7.elrepo.x86\_64。即要下载，  kernel-ml-devel-4.13.8-1.el7.elrepo.x86\_64.rpm  kernel-ml-headers-4.13.8-1.el7.elrepo.x86\_64.rpm  下载网址：<http://ftp.osuosl.org/pub/elrepo/kernel/el7/x86_64/RPMS/>  安装kernel-ml-devel，kernel-ml-header  rpm -ivh --force --nodeps kernel-ml-devel-4.13.8-1.el7.elrepo.x86\_64.rpm  rpm -ivh --force --nodeps kernel-ml-headers-4.13.8-1.el7.elrepo.x86\_64.rpm  说明：安装完毕后，在/usr/src/kernels下可以看到安装后的结果。 |
| 顺序执行以下命令：  echo “blacklist nouveau” >> /etc/modprobe.d/blacklist.conf  mv /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r).img.bak  dracut /boot/initramfs-$(uname -r).img $(uname -r)  init 3  sh NVIDIA-Linux-x86\_64-384.66.run --kernel-source-path=/usr/src/kernels/4.13.8-1.el7.elrepo.x86\_64 -k $(uname -r)  说明：按照界面上的一路accept就好  init 5  说明：使用nvidia-smi检查驱动是否安装成功。 |

3. 安装CUDA

sh cuda\_8.0.61\_375.26\_linux-run

说明：在提示是否安装driver的时候，选择n（如果第2步做完的话）

安装结束后，按照提示修改PATH，LD\_LIBRARY\_PATH

检测CUDA是否安装成功的办法：

nvcc --version

4. 安装CUDNN

tar -xvf cudnn-8.0-linux-x64-v5.0-ga.tgz

cp libcudnn\* /usr/local/cuda-8.0/lib64/

cp cudnn.h /usr/local/cuda-8.0/include

**5. 安装ffmpeg**

dependencies

访问<https://trac.ffmpeg.org/wiki/CompilationGuide/Centos> 安装ffmpeg的依赖：Yasm、libx264

安装依赖Yasm

curl -O http://www.tortall.net/projects/yasm/releases/yasm-1.3.0.tar.gz

tar xzvf yasm-1.3.0.tar.gz

cd yasm-1.3.0

./configure --prefix="/usr/local" --bindir="/usr/local/bin"

make

make install

安装依赖libx264

git clone --depth 1 http://git.videolan.org/git/x264

cd x264

PKG\_CONFIG\_PATH="/usr/local/lib/pkgconfig" ./configure --prefix="/usr/local" --bindir="/usr/local/bin" **--enable-shared**

说明：一定要用--enable-shared开关，用于生成动态库。因为ffmpeg依赖libx264的动态库

make

make install

echo

安装ffmpeg

curl -O http://ffmpeg.org/releases/ffmpeg-snapshot.tar.bz2

tar xjvf ffmpeg-snapshot.tar.bz2

cd ffmpeg

./configure --enable-shared --disable-static --prefix=/usr/local --enable-gpl --enable-pthreads --enable-nvenc --enable-cuda --enable-cuvid --extra-cflags="-Invidia\_sdk -I/usr/local/cuda/include" --extra-ldflags="-Lnvidia\_sdk -L/usr/local/cuda/lib64 -L/usr/local/ffmpeg/lib" --enable-nonfree --enable-libnpp --enable-libx264

make

make install

ffmpeg -hwaccel cuvid -c:v h264\_cuvid -i /home/lipeng/我的前半生第六集V2.mp4 -c:v h264\_nvenc out.mp4

**6. 安装opencv2.4.13**

说明：一定要看清版本。误装了2.4.11的版本，出现了很多编译错误

1) install jasper-devel-1.900.1-29.el7 (if jasper version conflicts, you should remove the newer version first)

<https://www.rpmfind.net/linux/rpm2html/search.php?query=jasper-devel>

2) yum install gcc gcc-c++ gtk2-devel gimp-devel gimp-devel-tools gimp-help-browser zlib-devel libtiff-devel libjpeg-devel libpng-devel gstreamer-devel libavc1394-devel libraw1394-devel libdc1394-devel jasper-utils swig python libtool nasm

3) 下载并解压opencv-2.4.13.3.zip

cd opencv-2.4.13.3

mkdir release

cd release

cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=/usr/local -D CUDA\_GENERATION=Auto -D WITH\_CUDA=ON -D WITH\_CUBLAS=ON CUDA\_FAST\_MATH=ON -D WITH\_NVCUVID=ON ..

make -j10

make install

ldconfig

**7. 安装CMAKE 3.2.2**

1) 下载并解压cmake-3.2.2.tar.gz

cd cmake-3.2.2

./bootstrap

说明：如果遇到错误CMake Error: The current CMakeCache.txt directory /home/sensetime/installation\_package/cmake-3.2.2/CMakeCache.txt is different than the directory /home/sensetime/Documents/cmake-3.2.2 where CMakeCache.txt was created. This may result in binaries being created in the wrong place. If you are not sure, reedit the CMakeCache.txt， 删掉 cmake-3.2.2/CMakeCache.txt，重新执行./bootstrap

gmake

gmake install

**8. 安装grpc**

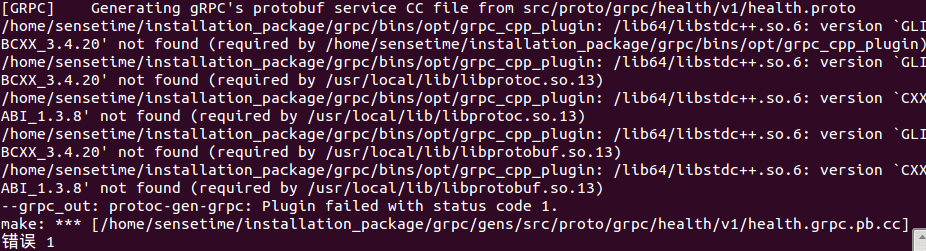
参考：http://dreamlikes.cn/archives/555

说明：

先安装protobuf 3.3.0

再安装grpc

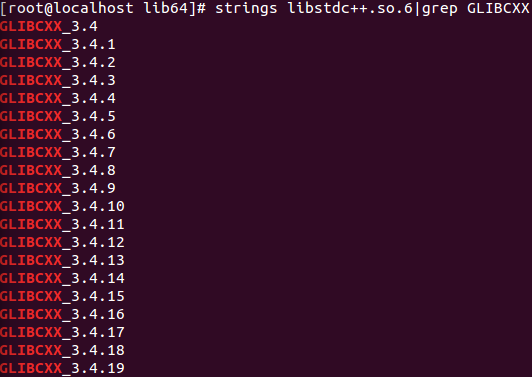
出现下面的错误：



解决方法：

1. 检查libstdc++.so.6中的GLIBCXX的版本号

strings libstdc++.so.6|grep GLIBCXX



确实没有3.4.20的版本。

发现gcc-4.9.4下的libstdc++版本是3.4.20的，所以，

cp /usr/local/gcc-4.9.4/lib64/libstdc++.so.6.0.20 /usr/lib64/

ln -sf libstdc++.so.6.0.20 libstdc++.so.6

**9. 安装libjpeg**

zai <http://www.ijg.org/files/> 中下载jpegsrc.v8d.tar.gz

tar xzvf jpegsrc.v8d.tar.gz

cd jpeg-8d

./configure --enable-shared

make

make install

ldconfig

**10. 安装libpng**

yum install libpng12.x86\_64