# 身份证信息识别

上传所需文件至 ~/env/目录下所有操作在此进行

**# cd ~/env/**

一般需要wget的包已下载好，若无请执行下载命令，有的话请忽略

## Python安装或升级

版本需 > 2.7

**若版本低或者未安装才执行**

**下载**

# wget https://www.python.org/ftp/python/2.7.13/Python-2.7.13.tgz --no-check-certificate

**解压**

# tar -xvf Python-2.7.13.tar.xz

# cd Python-2.7.13

**安装**

# ./configure && make && make install

**更新系统默认 Python 版本:**

**先把系统默认的旧版 Python 重命名：**

# mv /usr/bin/python /usr/bin/python.old

**再删除系统默认的 python-config 软链接**

# rm -f /usr/bin/python-config

最后创建新版本的 Python 软链接。

# ln -s /usr/local/bin/python /usr/bin/python

# ln -s /usr/local/bin/python-config /usr/bin/python-config

# ln -s /usr/local/include/python2.7/ /usr/include/python2.7

**安装完成后，验证**

使用

# python –V

查看Python版本

## Zlib 安装

**解压**

# tar -xvzf zlib-1.2.11.tar.gz

# cd zlib-1.2.11

#./configure    --prefix=/data/progam/zlib

**安装**

#make && make install

**配置一下系统的文件，加载刚才编译安装的zlib生成的库文件**

# vi /etc/ld.so.conf.d/zlib.conf

**加入如下内容后保存退出**

/data/program/zlib/lib

# ldconfig

**安装 zlib-devel**

#cd ~/env/

#rpm -ivh zlib-devel-1.2.3-29.el6.x86\_64.rpm

**重新编译 python**

#cd ~/env/Python-2.7.13

#make clean

#make && make install

## Setuptools 安装

**解压**

# tar -xzvf setuptools-0.6c11.tar.gz

# cd setuptools-0.6c11

**编译**

# python setup.py build

**安装**

# python setup.py install

## Pip安装

版本需 > 9.0

**解压**

# tar -xvzf pip-9.0.1.tar.gz

**安装**

# cd pip-9.0.1

# python setup.py install

**安装完成后，验证**

# pip –version

## GCC安装

版本需 > 4.7

**下载**

# wget <ftp://gcc.gnu.org/pub/gcc/infrastructure/mpfr-2.4.2.tar.bz2>

# wget <ftp://gcc.gnu.org/pub/gcc/infrastructure/gmp-4.3.2.tar.bz2>

# wget <ftp://gcc.gnu.org/pub/gcc/infrastructure/mpc-0.8.1.tar.gz>

**解压**

# tar –zxf mpfr-2.4.2.tar.bz2

# cd mpfr-2.4.2

**安装**

# ./configure && make && make install

**解压**

# tar –zxf gmp-4.3.2.tar.bz2

# cd gmp-4.3.2

**安装**

# ./configure && make && make install

**解压**

# tar –zxf mpc-0.8.1.tar.gz

# cd mpc-0.8.1

**安装**

# ./configure && make && make install

**下载GCC**

# wget <http://ftp.gnu.org/gnu/gcc/gcc-4.8.2/gcc-4.8.2.tar.bz2>

**解压**

# tar -jxvf gcc-4.8.2.tar.bz2

# cd gcc-4.8.2

**安装依赖（可选）**

# ./contrib/download\_prerequisites

**建立存放编译目录**

# mkdir gcc-build-4.8.2

# cd gcc-build-4.8.2

**生成Makefile文件**

# ../configure -enable-checking=release -enable-languages=c,c++ -disable-multilib

**编译**

# make -j4

**安装**

# sudo make install

**重启**

# reboot

**安装完成后，验证**

# GCC –version

## 安装cmake

版本需 >3.0

**下载**

# wget <http://www.cmake.org/files/v3.1/cmake-3.1.3.tar.gz> --no-check-certificate

**解压**

# tar -zxvf cmake-3.1.3.tar.gz

# cd cmake-3.1.3/

**安装**

# ./configure && make && make install

**安装完成后，验证**

# cmake –version

## 安装依赖的leptonica库

版本需 > 1.6

**下载**

# wget <http://www.leptonica.com/source/leptonica-1.72.tar.gz>

**解压**

# tar -xvf leptonica-1.72.tar.gz

# cd leptonica-1.72

**安装**

# ./configure && make && make install

## 安装tesseract

版本需 > 3.0

**所用版本 3.04，编译需要autoconf-2.65、automake、libtool**

**下载**

# wget <http://ftpmirror.gnu.org/autoconf/autoconf-2.65.tar.gz>

**解压**

# tar –zxf autoconf-2.65.tar.gz

# cd antoconf-2.65

**安装**

# ./configure && make && make install

**安装其余依赖**

# yum install automake

# yum install libtool

**安装tesseract**

# wget https://github.com/tesseract-ocr/tesseract/archive/3.04.zip --no-check-certificate

# unzip 3.04

# cd tesseract-3.04/

# ./configure && make && make install

# sudo ldconfig

**配置文件位置**

/usr/local/share/tessdata

**语言文件**

移动tessdata-lang.zip 到 /usr/local/share/tessdata/

# mv tessdata-lang.zip /usr/local/share/tessdata/

# unzip tessdata-lang.zip

# rm tessdata-lang.zip

**安装完成后，验证**

# tesseract --version

## 安装opencv

版本需 > 3.0

安装各种依赖包

# yum install libdc1394-devel

# yum install gtk2-devel

# yum install libv4l-devel

# yum install gstreamer-plugins-base-devel

# yum install -y gcc-c++ libjpeg-devel libtiff-devel jasper-devel libpng-devel zlib-devel cmake

# yum install git gtk2-devel pkgconfig python-devel gstreamer-plugins-base-devel libv4l ffmpeg-devel mplayer mencoder flvtool2

# yum install libdc1394

# yum install gtk\*

下载opencv 并安装

# wget https://github.com/Itseez/opencv/archive/3.2.0.tar.gz --no-check-certificate

# tar -zxf opencv-3.2.0.tar.gz

# mkdir downloads

# cd downloads

# mkdir linux-808b791a6eac9ed78d32a7666804320e

# cd ../../..

# mkdir build

# cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=./build -D WITH\_IPP=OFF ..

# make –j4

# make install

验证

# python

>>> import cv.2

没有报错即为成功

## 所需模块安装

numpy 安装

# cd /usr/local

# tar -xvzf numpy-1.9.1.tar.gz

# cd numpy-1.9.1

# python setup.py install

pillow 安装

# cd /usr/local

# tar -xvzf Pillow-4.2.1.tar.gz

# cd Pillow-4.2.1

# python setup.py install

pytesseract 安装

# cd /usr/local

# tar -xzvf pytesseract-0.1.7.tar.gz

# cd pytesseract-0.1.7

# python setup.py install

使用 #pip freeze检查所安装模块

## 出错处理

问题1： No module named cv2

将cv2.so 文件移动到python中的site-package文件夹中

查找文件

# find / -name 'cv2.so

查找后样例

# cp -r /root/ID/opencv-3.2.0/build/build/lib/python2.7/site-packages/cv2.so /root/bank\_env/lib/python2.7/site-packages/

问题2：ImportError: /usr/lib64/libstdc++.so.6: version `GLIBCXX\_3.4.15' not found (required by /root/ID/opencv-3.2.0/build/build/lib/libopencv\_stitching.so .3.2)

输入命令查询一下结果：

# strings /usr/lib64/libstdc++.so.6 | grep GLIBCXX

GLIBCXX\_3.4

GLIBCXX\_3.4.1

GLIBCXX\_3.4.2

GLIBCXX\_3.4.3

GLIBCXX\_3.4.4

GLIBCXX\_3.4.5

GLIBCXX\_3.4.6

GLIBCXX\_3.4.7

GLIBCXX\_3.4.8

GLIBCXX\_3.4.9

GLIBCXX\_3.4.10

GLIBCXX\_3.4.11

GLIBCXX\_3.4.12

GLIBCXX\_3.4.13

GLIBCXX\_FORCE\_NEW

GLIBCXX\_DEBUG\_MESSAGE\_LENGTH

我们看到当前GCC版本中的确没有GLIBCXX\_3.4.15。

libstdc++6\_4.7.2-5\_amd64.deb（去该文件目录下）

# ar -x libstdc++6\_4.7.2-5\_amd64.deb && tar xvf data.tar.gz

# cd usr/lib/x86\_64-linux-gnu

# cp libstdc++.so.6.0.17 /usr/lib64

# cd /usr/lib64

# rm libstdc++.so.6

# ln libstdc++.so.6.0.17 libstdc++.so.6

我们再输入命令查询一下结果：

# strings /usr/lib64/libstdc++.so.6 | grep GLIBCXX

GLIBCXX\_3.4

GLIBCXX\_3.4.1

GLIBCXX\_3.4.2

GLIBCXX\_3.4.3

GLIBCXX\_3.4.4

GLIBCXX\_3.4.5

GLIBCXX\_3.4.6

GLIBCXX\_3.4.7

GLIBCXX\_3.4.8

GLIBCXX\_3.4.9

GLIBCXX\_3.4.10

GLIBCXX\_3.4.11

GLIBCXX\_3.4.12

GLIBCXX\_3.4.13

GLIBCXX\_3.4.14

GLIBCXX\_3.4.15

GLIBCXX\_3.4.16

GLIBCXX\_3.4.17

GLIBCXX\_DEBUG\_MESSAGE\_LENGTH

已经安装好GLIBCXX\_3.4.15啦。