**香蕉派开发板安装流媒体服务器**

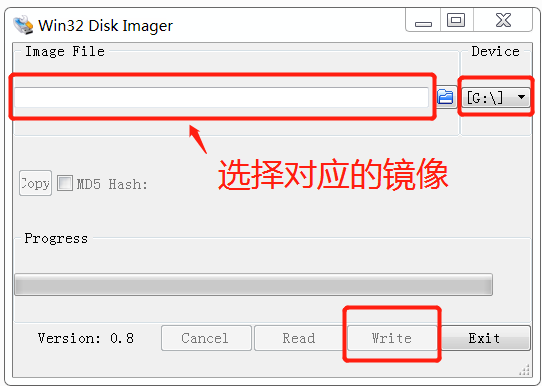
1. **给香蕉派开发板烧写centos-7-lite系统镜像**
2. **将SD卡插入至读卡器中，按如图方式，由电脑进行读取**



1. **将该SD卡利用SDFormatter.exe软件进行格式化（即使是新的SD卡也要进行格式化，不然后面烧写程序会出错）**

****

1. **利用Win32DiskImager.exe软件将系统烧写至SD卡中**

****

1. **初始化系统**
2. **安装系统下载软件所必须的工具 yum -y install wget**
3. **任意选定一个文件夹，下载初始化的压缩包（oneinstack-full.tar.gz）**

**当前选定的文件夹为：/usr/local/src**

**wget <http://mirrors.linuxeye.com/oneinstack-full.tar.gz>**

1. **解压oneinstack-full.tar.gz**

**tar -xzvf oneinstack-full.tar.gz**

1. **进入解压之后的文件夹下进行初始化安装**

**cd oneinstack-full**

**./install.sh**

1. **安装nginx-1.14.0服务器**

**1.选定一个文件夹，以便后面存储压缩包**

**cd /usr/local/src 后面所有安装包都在这个文件夹下，以便于编译的时候添加模块**

**2.安装PCRE库**

**wget ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.40.tar.gz**

**tar -zxvf pcre-8.40.tar.gz  
cd pcre-8.40  
./configure  
make && make install**

**3.安装zlib库**

**wget http://zlib.net/zlib-1.2.11.tar.gz  
tar -zxvf zlib-1.2.11.tar.gz  
cd zlib-1.2.11  
./configure  
make && make install**

**4.安装openssl库**

**wget https://www.openssl.org/source/openssl-1.1.0e.tar.gz**

**tar -zxvf openssl-1.1.0e.tar.gz**

**5.下载nginx-rtmp-module模块**

**git clone [https://github.com/arut/nginx-rtmp-module.git](https://github.com/arut/nginx-rtmp-module.git" \t "https://blog.csdn.net/tangyaliang11/article/details/_blank)**

**下载下来会自动生成文件夹 nginx-rtmp-module**

**6.源码编译nginx-1.14.0（这个版本是目前较为稳定的）**

**wget http://nginx.org/download/nginx-1.14.0.tar.gz  
tar -zxvf nginx-1.14.0.tar.gz  
cd nginx-1.14.0**

**./configure --prefix=/usr/local/nginx --with-http\_ssl\_module --with-pcre=/usr/local/src/pcre-8.40 --with-zlib=/usr/local/src/zlib-1.2.11 --with-openssl=/usr/local/src/openssl-1.1.0e --add-module=/usr/local/src/nginx-rtmp-module（注：这是一个命令）**

**make && make install**

1. **安装ffmpeg-3.3.2和ffmpeg-4.0**
2. **首先安装x264**

**git clone git://git.videolan.org/x264.git**

**cd x264**

**touch config\_x264\_bpi.sh**

**chmod +x config\_x264\_bpi.sh （+号和x之间不能有空格）**

**vim config\_x264\_bpi.sh 点击i （在终端的左下角会出现insert的字样），然后将一下脚本复制进这个文件当中**

#!/bin/sh

./configure \

--disable-shared --enable-static \

--enable-strip \

--disable-cli

**./config\_x264\_bpi.sh**

1. **安装ffmpeg-4.0（安装时间大概2-3个小时，请耐心等待）**

**wget http://ffmpeg.org/releases/ffmpeg-4.0.tar.bz2**

**tar jxvf ffmpeg-4.0.tar.bz2**

**cd ffmpeg-4.0**

**./configure --arch=armel --target-os=linux --enable-gpl --enable-libx264 --enable-nonfree （这是一个命令）**

**make**

**make install （或者执行：make && make install）**

1. **安装ffmpeg-3.3.2**

**wget http://ffmpeg.org/releases/ffmpeg-3.3.2.tar.bz2**

**tar jxvf ffmpeg-3.3.2.tar.bz2**

**cd ffmpeg-3.3.2**

**touch config\_ffmpeg\_bpi.sh**

**chmod +x config\_ffmpeg\_bpi.sh**

**vim config\_ffmpeg\_bpi.sh 点击i 添加下列脚本（节选）**

#!/bin/sh

PREFIX=/usr/local

./configure \

--enable-gpl --enable-version3 --enable-nonfree \

--enable-static --disable-shared \

\

--prefix=/usr

--bindir=/usr/bin

--datadir=/usr/share/ffmpeg

--incdir=/usr/include/ffmpeg

--libdir=/usr/lib64

--mandir=/usr/share/man

--optflags='-O2 -g -pipe -Wall -Wp,-D\_FORTIFY\_SOURCE=2 -fexceptions -fstack-protector-strong

--param=ssp-buffer-size=4 -grecord-gcc-switches -m64 -mtune=generic'

--extra-ldflags='-Wl,-z,relro ' --enable-libopencore-amrnb

--enable-libopencore-amrwb --enable-libvo-amrwbenc

--enable-version3 \

--enable-bzlib \

--disable-crystalhd \

--enable-gnutls \

--enable-ladspa \

--enable-libass \

--enable-libcdio

--enable-libdc1394 \

--disable-indev=jack \

--enable-libfreetype \

--enable-libgsm \

--enable-libmp3lame \

--enable-openal \

--enable-libopenjpeg \

--enable-libopus \

--enable-libpulse \

--enable-libschroedinger \

--enable-libsoxr \

--enable-libspeex \

--enable-libtheora \

--enable-libvorbis \

--enable-libv4l2 \

--enable-libx264 \

--enable-libxvid \

--enable-x11grab \

--enable-avfilter \

--enable-avresample \

--enable-postproc \

--enable-pthreads \

--disable-static \

--enable-shared \

--enable-gpl \

--disable-debug \

--disable-stripping \

--shlibdir=/usr/lib64 \

--enable-runtime-cpudetect\

\

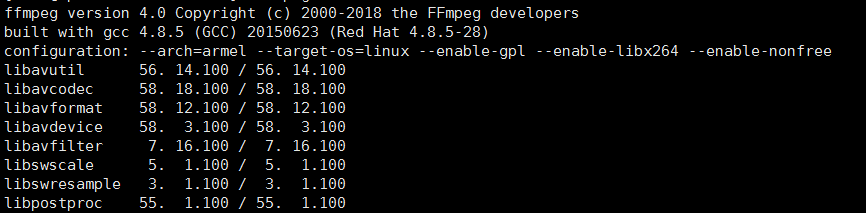
**./config\_ffmpeg\_bpi.sh**

**Make**

**Make install （或者执行：make && make install）**

1. **验证是否安装上ffmpeg**

**①执行命令：ffmpeg -version 可看到以下结果**

****

**②执行命令：whereis ffmpeg 可看到以下结果**

**1542243479(1)**

1. **安装jdk-8u191（由于香蕉派是arm32系统，java也要选择相对应的：jdk-8u191-linux-arm32-vfp-hflt.tar.gz）**
2. **下载jdk-8u191-linux-arm32-vfp-hflt.tar.gz压缩包**

**终端执行rz，然后选择jdk-8u191-linux-arm32-vfp-hflt.tar.gz**

1. **解压jdk-8u191-linux-arm32-vfp-hflt.tar.gz**

**tar -xzvf jdk-8u191-linux-arm32-vfp-hflt.tar.gz**

1. **为java添加环境变量**

**1.终端执行：vim /etc/profile 进入这个文件夹下按i进行添加**

**##java**

**export JAVA\_HOME=/usr/local/src/jdk1.8.0\_191  
export CLASSPATH=.:${JAVA\_HOME}/jre/lib/rt.jar:${JAVA\_HOME}/lib/dt.jar:${JAVA\_HOME}/lib/tools.jar  
export PATH=$PATH:${JAVA\_HOME}/bin**

**2.输入java -version 和 javac测试安装是否成功**

1. **安装apollo-1.7**
2. **终端执行rz，然后选择apache-apollo-1.7.1-unix-distro.tar.gz**
3. **解压apache-apollo-1.7.1-unix-distro.tar.gz压缩包**
4. **进入apache-apollo-1.7.1/bin 文件夹下，执行如下命令**

**进入文件夹： cd apache-apollo-1.7.1/bin**

**创建mybroker文件夹： ./apollo create mybroker**

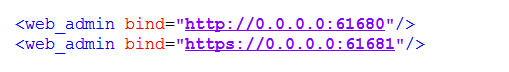
**进入文件夹查看是否安装成功：mybroker文件夹下有bin、data、etc、log、tmp五个文件夹**

**4.修改配置文件apollo.xml，以免网页能够访问**

**找到以下位置，**

****

**将这个位置修改为：**

****

1. **安装tomcat9**

**1下载apache-tomcat-9.0.13.tar.gz压缩包**

**终端执行rz，然后选择apache-tomcat-9.0.13.tar.gz**

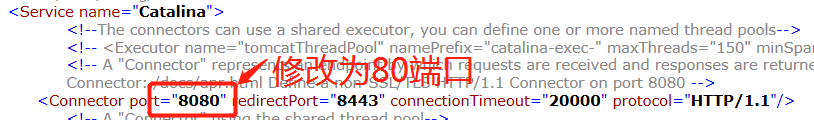
**2.解压apache-tomcat-9.0.13.tar.gz**

**tar -xzvf apache-tomcat-9.0.13.tar.gz**

1. **进入apache-tomcat-9.0.13/conf/文件夹下，修改server.xml文件的端口**

**cd apache-tomcat-9.0.13/conf/**

**vim server.xml**

****

1. **为tomcat添加环境变量**

**终端执行：vim /etc/profile 进入这个文件夹下按i进行添加**

**##tomcat**

**CATALINA\_HOME=/usr/local/src/apache-tomcat-9.0.13**

**CATALINA\_BASE=/usr/local/src/apache-tomcat-9.0.13  
PATH=$PATH:$CATALINA\_BASE/bin  
export PATH CATALINA\_BASE**

1. **测试安装是否成功**

**cd apache-tomcat-9.0.13/bin**

**./startup.sh**

****