ubuntu配置arm虚拟机

# 1、下载安装QEMU仿真器：

git:/repo.or.cz/w/qemu

下载最新的qemu源码，安装依赖包，配置（设定安装目录），编译，添加环境变量

安装依赖包:

sudo apt-get install build-essential pkg-config zlib1g-dev

sudo apt-get install libglib2.0-0 libglib2.0-dev

sudo apt-get install libsdl1.2-dev

sudo apt-get install libpixman-1-dev libfdt-dev

sudo apt-get install autoconf automake libtool

sudo apt-get install librbd-dev

sudo apt-get install libaio-dev

sudo apt-get install flex bison

配置（设定安装目录）: sudo ./configure --prefix=/usr/local/qemu --target-list=arm-softmmu --audio-drv-list=

编译:sudo make && sudo make install

添加环境变量:export PATH=$PATH:/usr/local/qemu/bin

# 2、安装交叉编译链

交叉编译工具链有多种，不同的工具链有不同应用针对性，其区别查看链接： <https://blog.csdn.net/qq_16149777/article/details/82349868>

本初主要是从一下网站下载最新的交叉编译工具，这里使用由 Linaro 公司基于GCC推出的的ARM交叉编译工：

arm-linux-gnueabihf-gcc：<https://www.linaro.org/downloads/>

解压后，添加环境变量即可。

# 3、下载并交叉编译linux内核

从清华镜像网站下载所需版本的内核：

<https://mirrors.tuna.tsinghua.edu.cn/kernel/>

1、修改顶级目录的Makefile文件,使：

ARCH=arm

CROSS\_COMPILE=arm-linux-gnueabihf-

2、选择使用的开发板的配置文件，生成对应板的默认配置文件

make vexpress\_defconfig

3、在交互界面定制自己需要的内核

make menuconfig

4、编译内核和dts文件

make -j2 zImage #编译内核

make dtbs #编译dts文件

# 4、下载并交叉编译u-boot

1、从此处网站下载所需版本的u-boot:

<ftp://ftp.denx.de/pub/u-boot/>

2、配置顶层目录的Makefile，修改或添加:

ARCH=arm

CROSS\_COMPILE=arm-linux-gnueabihf-

3、配置u-boot，修改include/configs/vexpress\_common.h

设置网络等内容，详情请查询

4、生成目标开发板的默认配置文件：

make vexpress\_ca9x4\_defconfig

5、编译u-boot

make –j2

# 5、安装编译busybox

1、从此处下载最新且稳定的busybox版本：

<https://busybox.net/downloads/>

2、配置顶层目录的Makefile，修改或添加:

ARCH=arm

CROSS\_COMPILE=arm-linux-gnueabihf-

3、生成默认配置文件：

make defconfig

4、在在交互界面设置编译选项

make menuconfig （#配置busybox,然后在弹出窗中：setting->build options->build static binary;然后Y;退出并保存；）

5、编译并安装

make CROSS\_COMPILE=arm-linux-gnueabihf- ARCH=arm #编译

make install CROSS\_COMPILE=arm-linux-gnueabihf- ARCH=arm # 安装

备注：如果安装出现问题可能是缺乏依赖库，那么先安装依赖库：

sudo apt-get update #更新apt

sudo apt install aptitude #安装一个应用安装管理软件

sudo aptitude install libncurses5-dev #安装busybox需要的依赖库

然后：make clean

再按照上述过程编译安装busybox

# 6、编译配置根文件系统

备注：这里使用的目录为busybox的父目录

-p ：连同档案的属性一起复制过去，而非使用预设属性；

-r ：递归持续复制，用于目录的复制行为；

1、使用sudo在自己想要保存的位置，创建文件夹rootfs

sudo mkdir -p rootfs/{dev,etc/init.d,lib}

2、将busybox文件内的文件复制到rootfs

①将busybox编译安装生成的\_install目录下的文件复制到rootfs

sudo cp –r ./busybox-1.30.1/\_isntall/\* ./rootfs/

②把busybox源码目录下的example /bootfloppy/etc的内容拷贝到rootfs目录下的etc下

cp –r ./busybox-1.30.1/examples/bootfloppy/etc/\* ./rootfs/etc

3、将交叉编译工具链中的lib文件下的文件复制到rootfs下的lib文件夹

sudo cp –p /usr/arm-linux-gcc/lib/\* ./rootfs/lib

4、创建4个tty终端设备

sudo mknod rootfs/dev/tty1 c 4 1

sudo mknod rootfs/dev/tty2 c 4 2

sudo mknod rootfs/dev/tty3 c 4 3

sudo mknod rootfs/dev/tty4 c 4 4

5、创建终端和回收站

sudo mknod -m 666 console c 5 1

sudo mknod -m 666 null 1 3

6、使用dd命令制作文件系统镜像，然后格式化生成ext4文件系统

dd if=/dev/zero of=a9rootfs.ext4 bs=1M count=32

mkfs.ext4 a9rootfs.ext4

7、创建文件系统挂载目录

sudo mkdir tmpfs

8、挂载文件系统

sudo mount -t ext4 a9rootfs.ext4 tmpfs/ -o loop