

环境搭建教程

目录

环境搭建教程.....	1
1、虚拟机 ubuntu 安装.....	2
2、建立 Windows 和虚拟机 Ubuntu 共享文件夹	2
3、ubuntu 配置 tftp 服务器.....	4
第 1 步：安装 tftp.....	4
第 2 步：配置/etc/xinetd.conf.....	4
第 3 步：配置/etc/default/tftpd-hpa	4
第 4 步：配置/etc/default/xinetd.d/tftp	5
第 5 步：修改权限.....	5
第 6 步：重启服务.....	5
总结：	6
4、ubuntu 配置 nfs 服务器	6
一 安装 NFS	6
二 配置 NFS	7
三 配置 hosts.deny	7
四 配置 hosts.allow	7
五 配置/etc/exports.....	8
六 启用.....	8
挂载测试.....	8
5、使用 SecureCRT 远程登录 Ubuntu	9
6、ubuntu 中配置静态 IP.....	9
7、ubuntu 中安装 arm-linux-gcc-4.1.2.....	10

1、虚拟机 ubuntu 安装

---by aston---

嵌入式开发学习中目前最常用的方式是在 Windows 下安装 VMWare 虚拟机，然后在虚拟机中安装 Ubuntu 作为嵌入式 linux 开发宿主机使用。关于 VMWare 和 Ubuntu 的安装网上资料很多，这里就不赘述了，谨强调以下：

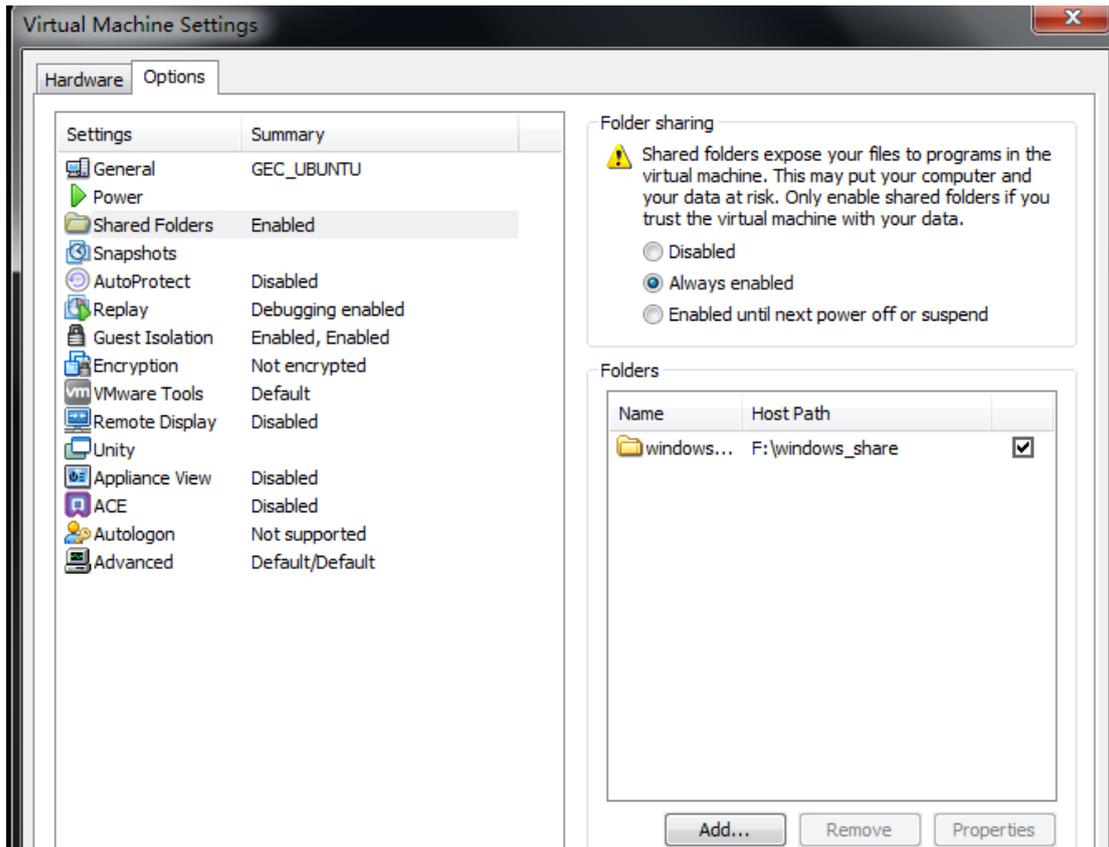
- VMWare 一般选用 7.1 及以上版本。因为已经内置了 VMWareTools，不需要再另外安装配置。(VMWareTools 最有用的功能莫过于在 Windows 和 Ubuntu 之间建立共享文件夹)
- 虚拟机网络连接选择桥接方式 (Bridged network)，不要选择 NAT。

2、建立 Windows 和虚拟机 Ubuntu 共享文件夹

---by aston---

在 Windows 和虚拟机 Ubuntu 之间共享文件有好几种方法，包括 tftp, samba, VMWareTools 等。每种方法在网上都能找到不少对应的介绍文档，大家可先了解一下再根据自己的喜好选择一种使用。

在这里介绍通过 VMWareTools 建立共享文件夹的方法，这也是我最常使用的方法。首先选择 VMWare 软件的菜单栏 VM->Settings->Options->SharedFolders，然后在右侧选择 Always Enable，点击下方 Add 按钮，浏览添加一个事先建立好的 Windows 下的文件夹作为共享文件夹。如下图是添加好共享文件夹后的样子，我定义的共享文件夹为 F:/windows_share。



启动 Ubuntu，进入/mnt 目录，可见该目录下有一个名为 hgfs 的文件夹，其内有一个和我们在 Windows 中建立的共享文件夹同名的文件夹，此即为我们在 Ubuntu 中访问该共享文件夹的目录。

为了访问方便，我一般会为该共享文件夹建立一个符号链接。首先切换到 root 用户（我们在做开发时一般都是在 root 下，以避免因为权限问题造成的一些不必要的麻烦。），然后进入 root 用户的根目录/root，并执行 `ln -s /mnt/hgfs/windows_share /root/windows_share`，即可在/root 下成功创建共享文件夹的符号链接，便于以后实验时与 Windows 下交互。

```
root@ubuntu:/# cd root
root@ubuntu:~# ls
root@ubuntu:~#
root@ubuntu:~#
root@ubuntu:~# ln -s /mnt/hgfs/windows_share/ windows_share
root@ubuntu:~# ls
windows_share
root@ubuntu:~# ls -l
total 0
lrwxrwxrwx 1 root root 24 2012-09-10 23:57 windows_share -> /mnt/hgfs/windows_share/
```

使用 `su` 或 `su root` 命令切换到 root 时需要输入 root 用户的密码，很多人觉得奇怪我在安装 ubuntu 时并没有要求输入 root 用户的密码啊。其实 root 账户密码是我们安装好 ubuntu 以后再设置的，我们在终端中自己的用户下输入 `sudo su root`，然后输入自己账户的密码，系统即会允许我们为 root 账户设置密码，重复输入两次即可完成设置。建议 root 用户和本用户使用同样的密码，并且不要太长，否则经常输入会很麻烦。密码也不宜太生僻否则长时间不用会忘记。

3、ubuntu 配置 tftp 服务器

---来源于网络---

第 1 步：安装 tftp

安装 tftp 所需的软件。首先需要安装 tftp-hpa, tftpd-hpa, 前者是客户端, 后者是服务程序,在终端下输入 **sudo apt-get install tftp-hpa tftpd-hpa**, 安装 tftp-hpa 和 tftpd-hpa。然后还需要安装 xinetd, 在终端下输入 **sudo apt-get install xinetd**, 安装好 xinetd。

第 2 步：配置/etc/xinetd.conf

配置相关服务文件。进入根目录下的 etc 文件夹 (**cd /etc/**), 首先看目录中有没有一个 xinetd.conf 文件, 如果没有则新建一个, 有的话查看内容, 看是否与下面的一致, 若不一致则修改, 内容如下:

```
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/
defaults
{
# Please note that you need a log_type line to be able to use log_on_success
# and log_on_failure. The default is the following :
# log_type = SYSLOG daemon info
}
includedir /etc/xinetd.d
```

第 3 步：配置/etc/default/tftpd-hpa

配置 tftp 服务器

命令:

```
sudo vim /etc/default/tftpd-hpa
```

将内容修改成

```
# /etc/default/tftpd-hpa
TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/tftpboot"
```

#这是你 tftp 服务器的工作目录, 自行修改, 注意, 在新建工作目录时, 最好修改其权限为 777, 命令 **sudo chmod 777 /tftpboot**

```
TFTP_ADDRESS="0.0.0.0:69"  
TFTP_OPTIONS="-l -c -s"
```

第 4 步：配置/etc/xinetd.d/tftp

然后进入 xinetd.d 文件夹（**cd xinetd.d**），查看是否有一个 tftp 文件，如果没有就新建一个，如果有的话就查看内容是否与下面的一致，不一致则修改，内容如下：

```
service tftp  
{  
  socket_type = dgram  
  wait = yes  
  disable = no  
  user = root  
  protocol = udp  
  server = /usr/sbin/in.tftpd  
  server_args = -s /tftpboot  
  #log_on_success += PID HOST DURATION  
  #log_on_failure += HOST  
  per_source = 11  
  cps = 100 2  
  flags = IPv4  
}
```

其中 server_args 一行是配置服务器的文件存放的位置，就是进行 tftp 传输的时候，都是从该文件夹中搜索文件的

第 5 步：修改权限

修改所需文件夹的权限。需要修改的文件夹也就是上一步提到的那个服务器文件存放的文件夹，以我的配置文件为例，创建一个文件夹（**sudo mkdir /tftpboot**），然后把它可以设置成访问权限最宽松的（**sudo chmod 777 /tftpboot**），也可以设置成合适的权限。

第 6 步：重启服务

重新启动服务。**sudo service tftpd-hpa restart**，这也是我经常疏忽的一步，当配置好 tftp 的配置文件后，需要重新启动一下 xinetd，在终端中输入 **sudo /etc/init.d/xinetd reload**，重新加载一下进程，再输入 **sudo /etc/init.d/xinetd restart**，重启服务。记住，每次修改完配置文件后，都需要重新启动一下服务。

执行次序：

```
sudo service tftpd-hpa restart
```

```
sudo /etc/init.d/xinetd reload
```

```
sudo /etc/init.d/xinetd restart
```

总结:

通过以上四步就可以建立好 tftp 服务器了,可以在本地先进行一下测试。还是以我的配置文件为例,首先在/tftpboot 中新建一个文件 file,然后在其中随便输入内容;然后进入一个不是/tftpboot 的目录下(原因是避免混淆,因为在获取文件是,默认是将想要获取的文件存放在当前目录下的);再在终端中输入 sudo tftp localhost,进入 tftp 命令符下(可以在其中输入 help 查看命令和命令的作用),输入 get file,如果没有任何提示,就说明传输成功了,然后输入 q 退出 tftp 命令符,在当前目录下就可以看到一个 file 文件,内容是与开始新建的那个 file 的内容是一致的。同样也可以在 tftp 命令符中输入 put xx,把 xx 文件上传到服务器文件夹中。如果一切 ok,那么就么一个可用的 tftp 服务器就顺利搭建成功了。

对于上面的测试是针对本地测试的,如果想要对其他电脑或者开发板进行 tftp 传输,则在终端输入 sudo tftp xxx(即目标电脑或目标开发板的 ip 地址),而且还要关闭 ubuntu 自带的防火墙(书上说的),我是直接把防火墙给卸载了(sudo apt-get remove iptables,或 sudo ufw disable 可以关掉防火墙),因为我觉得对于 Linux 来说,防火墙可能没什么需要把,至少目前为止我还没有受到过攻击。。。当然,想要与其他局域网内的电脑或开发板互联进行传输,还要将这些机器的 ip 改到同一个段内。好了,通过以上的步骤,就可以通过 tftp 协议,在电脑和开发板之间进行文件传输了,目前的目的就是下载 Linux 内核和文件系统到开发板。如果有朋友发现了我的不足,都可以给我提出来,我们也可以进行讨论。不过还要强调一点,我的这些配置都是针对于 ubuntu 的,对于其他的系统可能会有所出入的。

4、ubuntu 配置 nfs 服务器

---来自网络---

环境 ubuntu 10.4 vm 7.1

终端

ifconfig 得到 ubuntu 资料

INET ADDR 192.168.1.116

BCAST 192.168.1.255

MASK 255.255.255.0

一 安装 NFS

```
$ sudo apt-get install nfs-kernel-server
```

\$ sudo apt-get install nfs-common

二 配置 NFS

1 **\$ sudo dpkg-reconfigure portmap** , 对 Should portmap be bound to the loopback address? 选 N

2 **\$ sudo vi /etc/default/portmap** 末行清除 "-i 127.0.0.1"

三 配置 hosts.deny

\$ sudo vi /etc/hosts.deny

文本未添加

```
### NFS DAEMONS
portmap: ALL
lockd:    ALL
mountd:   ALL
rquotad:  ALL
statd:    ALL
```

四 配置 hosts.allow

\$ sudo vi /etc/hosts.allow

文本未加入

```
### NFS DAEMONS
portmap: 192.168.1.
lockd:    192.168.1.
rquotad: 192.168.1.
mountd:   192.168.1.
statd:    192.168.1.
```

\$ sudo /etc/init.d/portmap restart 重启 portmap daemon.

显示如下

Since the script you are attempting to invoke has been converted to an Upstart job, you may also use the restart(8) utility, e.g. restart portmap
portmap start/running, process XXXX

五 配置/etc/exports

```
$ sudo vi /etc/exports
```

文本未添加

```
/root/rootfs *(rw,sync,no_root_squash)
```

然后在终端执行 **\$chmod 777 -R /root/rootfs**

```
$ sudo showmount -e
```

显示 clnt-create : RPC : Program not registered

```
$ sudo exportfs -r 更新
```

```
$ sudo showmount 192.168.1.116 -e
```

显示

Export list for 192.168.1.116

```
/root/rootfs *
```

六 启用

```
$ sudo /etc/init.d/nfs-kernel-server restart 重启 nfs 服务
```

显示如下

```
* Stopping NFS kernel daemon
```

```
[ OK ]
```

```
* Unexporting directories for NFS kernel daemon...
```

```
[ OK ]
```

```
* Exporting directories for NFS kernel daemon...
```

```
exportfs: /etc/exports [1]: Neither 'subtree_check' or 'no_subtree_check' specified for  
export "*/home/work".
```

```
Assuming default behaviour ('no_subtree_check').
```

```
NOTE: this default has changed since nfs-utils version 1.0.x
```

```
* Starting NFS kernel daemon
```

```
[ OK ]
```

挂载测试

```
$ mount -t nfs localhost:/root/rootfs /mnt
```

```
$ mount -t nfs -o nolock localhost:/root/rootfs /mnt
```

执行后，进入/mnt 目录中，如果可以看到/root/rootfs 中的内容，则说明 nfs 搭建成功！

5、使用 SecureCRT 远程登录 Ubuntu

---by aston---

1. 首先要明白什么是 ssh?

可以把 ssh 看做是 telnet 的加强版，telnet 的密码和信息都是不加密的，而 ssh 则加密。

2. 开启 ubuntu 上的 ssh 功能

先安装 ssh，安装后就自动开启了。

\$ sudo apt-get install openssh-server openssh-client

3. 安装 SecureCRT

安装过程省略

4. 查看 ubuntu 的 ip

\$ ifconfig(注意是 ifconfig,不是 windows 上的 ipconfig)。

5. 连接

SecureCRT => Quick Connect, 在打开的对话框中：协议一栏选择 SSH2。主机名输入 ubuntu 的 IP 地址。端口号默认为 22 不用更改。用户名输入自己要登录的 ubuntu 用户，这里我们选择使用 root 账户登录，因此直接输入 root。最后选择“在一个标签中打开”，勾选这个选项可以让我们在同一个 SecureCRT 窗口内打开多个链接，这样用起来很方便。

最后点击连接。在弹出的对话框中输入正确的密码就可以看到登录信息了。推荐选上保存密码，这样就不用每次都输入密码这么麻烦了。

如果还连接不上，可能是 Ubuntu 上的 ssh 服务没有开启。可以使用 **\$ /etc/init.d/ssh**

restart 以启动 ssh-server。使用 **\$ netstat -tlp** 来确认 ssh-server 已经正常工作。

```
tcp6 0 0 *:ssh:*:* LISTEN -
```

看到上面这一行输出说明 ssh-server 已经在运行了。

6、ubuntu 中配置静态 IP

---by aston---

首先要明白为什么要在 Ubuntu 中配置静态 IP。在整个 uboot 移植、linux 移植和驱动开发过程中，经常需要开发板与宿主机（ubuntu）通往网络连接。而 IP 地址正是开发板找到宿主机的索引地址，因此将宿主机更改成一个固定好记的静态 IP，可以在实验中省不少事。

在 ubuntu 中设置静态 IP 比较简单，设置前首先确定虚拟机网络设置为桥接方式（NAT 等模式不要）。然后在 ubuntu 的终端中输入 **\$ sudo vi /etc/network/interfaces**,

在打开的网络配置文件中输入以下内容：

```
auto lo
iface lo inet loopback
```

```
auto eth0
iface eth0 inet static
address 192.168.1.100
netmask 255.255.255.0
gateway 192.168.1.1
```

这里我将静态地址设置为 192.168.1.100。要注意的是在桥接方式下，虚拟机相当于是局域网中一台独立的主机，因此这个静态地址不能和局域网中别的主机冲突。

如果对网络设置做了更改后无法连接网络，可以尝试使用

\$ sudo /etc/init.d/networking restart 命令来重启网络。

7、ubuntu 中安装 arm-linux-gcc-4.4.3

编译 uboot 和 linux kernel 都需要 gnu 交叉工具链支持，这里我们选择的是 arm-linux-gcc-4.4.3 版本。使用 eabi 版本有诸多优势，具体大家可以到网上搜索，此文档以在 ubuntu 中安装 arm-linux-gcc-4.4.3 为主要内容。

首先得到 arm-linux-gcc-4.4.3 工具的压缩包，解压后重命名为 4.4.3。然后按照以下步骤：

1、首先切换到 root 用户。 **\$ su root**

2、在 /usr/local/ 文件夹下建立名为 arm 的文件夹。

```
# cd /usr/local/      $ mkdir arm
```

3、通过共享文件夹或者 samba 的方式拷贝 arm-linux-gcc-4.4.3.tar.gz 到本目录下

```
# cp /root/winshare/s3c2440/arm-linux-gcc-4.4.3.tar.gz /usr/local/arm
```

4、解压到当前目录

```
# tar -xvf arm-linux-gcc-4.4.3.tar.gz
```

5、重命名

```
# mv opt/FriendlyARM/toolschain/4.4.3/ 4.4.3
```

```
# rm -rf opt
```

6、导出路径到环境变量。

```
# Vi /root/.bashrc
```

然后在打开的文件中最后空白处添加以下内容：

```
# aston added for export arm-linux-gcc-4.4.3
export PATH=$PATH:/usr/local/arm/4.4.3/bin
```

#end of aston added

保存后退出。然后在终端中执行 **# source .bashrc** 以使刚才添加的内容生效。

5、测试。在终端内输入 **#arm-linux-gcc -v** 如果可以看到 arm-linux-gcc 的版本信息则说明成功。若提示找不到或者不识别等错误信息，请详细检查以上各步骤。