

## 第4章 嵌入式Linux开发环境搭建

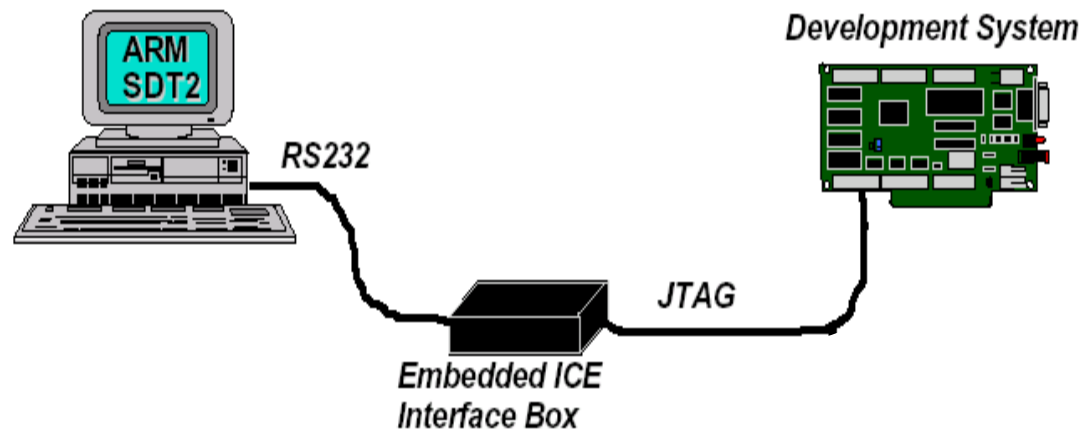
4.1 嵌入式交叉编译环境

4.2 相关开发工具

4.3 嵌入式开发环境搭建

## 4.1 嵌入式交叉编译环境

- 在嵌入式开发过程中有宿主机和目标机的角色之分：宿主机是执行编译、链接嵌入式软件的计算机；目标机是运行嵌入式软件的硬件平台



**通常将这种在主机上开发编译，在目标平台上调试运行的开发模式称为交叉开发。**

# 交叉开发环境

- 目标机(Target): 嵌入式程序运行的硬件平台
- 宿主机(Host):
  - 安装开发工具
  - 编译、编译运行在目标机中的OS
  - 编辑、编译运行在目标板中的应用程序
- 宿主机环境下开发, 目标机上运行。
- 宿主机开发环境
  - 安装Linux
  - Windows下安装虚拟机(VMWare), 在虚拟机上安装Linux

# 目标板与主机之间的连接

- 串口
  - 向目标板发送命令、打印目标板信息
  - 通信速率慢、不适合大数据量传输
  - windows附件 “超级终端”
  - Linux下, Minicom
- 以太网接口
- USB接口
- JTAG接口
  - 一种嵌入式调试技术

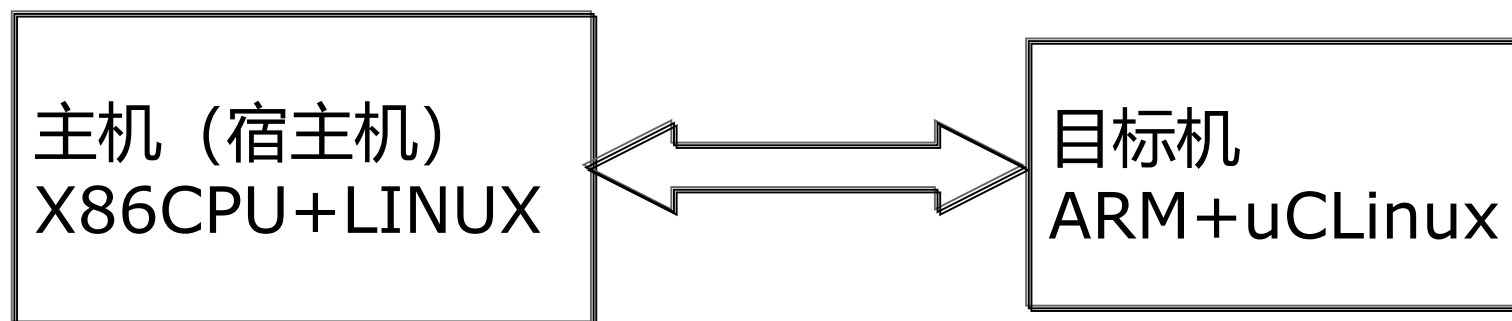
# 交叉编译

- 嵌入式软件所采用的编译方式称为交叉编译。普通软件所采用的编译方式为一般编译
- 交叉编译：在一个平台上生成在另一个平台上可以执行的代码。
- 编译的主要工作：把源程序，通过编译器转换成运行该程序的CPU所能认识的机器码，不同的体系结构具有不同的指令系统，同一个源程序，通过不同的编译器的编译，就可以生成可以运行在不同体系结构CPU上的可执行程序。

# 交叉编译模型

主机：进行交叉编译的主机，普通计算机

目标机：程序实际运行的环境，嵌入式系统环境



- hello.c----一般编译 (GCC) ----hello----在主机上运行
- hello.c----交叉编译 (arm-linux-gcc) ----hello----在目标机上运行，需要在主机上安装 arm-linux-gcc交叉编译器

# 交叉编译工具链安装

- 进行嵌入式开发前，首先要安装交叉工具链，步骤如下：

- 下载arm-linux-gcc-4.4.3.tar.gz
- 解压 `tar xvzf arm-linux-gcc-4.4.3.tar.gz -C /home/wxq/`
- 把编译器的路径加入系统的环境变量

`sudo gedit /etc/environment`

在`PATH=$PATH:/user/……:/user/local/games`

后增加`:/home/wxq/opt/FridendlyARM/4.4.3/bin`

- 重启系统（或注销）
- 运行`arm-linux-gcc -v` 正确表示安装配置成功

# 交叉编译工具链使用

- 编译器: arm-linux-gcc  
arm-linux-gcc hello.c -o hello
- 反汇编工具: arm-linux-objdump  
arm-linux-objdump -D -S hello
- ELF文件查看工具: arm-linux-readelf  
arm-linux-readelf -a hello  
arm-linux-readelf -d hello 查看hello使用的动态库



## 4.2相关开发工具

- minicom安装配置
- TFTP服务器配置
- NFS服务配置
- Samba服务配置

# minicom

- 串口是我们常用的嵌入式调试端口，minicom是linux下的串口设置、通讯软件。类似于windows下的“超级终端”。
- 安装： `sudo apt-get install minicom`
- 配置： `sudo minicom -s`  
设置端口ttyS0，usb转串口用ttyUSB0  
设置串行通信的波特率、字符位数、校验、停止位等信息

# TFTP服务器

- Tftp服务器作为工作于宿主机上的软件， 主要提供对目标机的映像文件的下载。
- Tftp安装：
  - 安装： tftp-hpa, tftpd-hpa, xinetd
  - `sudo apt-get install tftp-hpa tftpd-hpa`
  - `sudo apt-get install xinetd`
  - 建立tftp的工作目录
    - `mkdir tftpboot`
    - `sudo chmod -R 777 /home/wxq/tftpboot`

# TFTP配置

## •修改配置文件

- 配置/etc/default/tftpd-hpa, 内容如下:

```
# /etc/default/tftpd-hpa
```

```
TFTP_USERNAME="tftp"
```

```
TFTP_DIRECTORY="/var/tftp"
```

```
TFTP_ADDRESS="0.0.0.0:69"
```

```
TFTP_OPTIONS="--secure -c"
```

- 然后重启服务: `sudo /etc/init.d/tftpd-hpa restart`
- 通过tftp的get和put进行测试:

```
$ tftp localhost
```

```
tftp> get xx
```

```
tftp> put yy
```

# NFS介绍

- 网络文件系统(NFS, Network File System)

是一种将远程主机上的分区（目录）经网络挂载到本地的一种机制，通过对网络文件系统的支持，用户可以在本地系统上像操作本地分区一样来对远程主机的共享分区（目录）进行操作。（类似于windows共享目录）

# NFS安装

- `sudo apt-get install nfs-kernel-server`(安装nfs-kernel-server时, apt会自动安装nfs-common和portmap)
- NFS配置
  - `vi /etc/exports`
  - 加入允许其他计算机访问的目录和访问权限
  - 例: `/home/wxq/linuxc 192.168.0.* (rw,sync,no_root_squash)`
  - 重启NFS服务:  
`sudo /etc/init.d/nfs-kernel-server restart`

# NFS配置

- /home/wxq/linuxc: 允许其他计算机访问的目录
- 192.168.0.\*: 被允许访问该目录的客户端IP地址
- rw: 可读可写
- sync: 同步写磁盘 (async的资料会先暂存于内存中, 而非直接写入硬盘)
- no\_root\_squash: 表示客户端root用户对目录具有备写权限

# NFS使用

- 使用mount命令来挂载NFS服务器上的共享目录：

`mount -t nfs servername:/shared_dir /localdir`      (主机ip:共享目录 挂载点)

例如：

`mount -t nfs 192.168.0.47:/home/wxq/linuxc /mnt/nfs -o nolock`

说明： nfs mount 默认选项包括文件锁，依赖于portmap提供的动态端口分配功能， 需要添加参数`mount -o nolock`；



# Samba

- Linux与Linux之间通过NFS实现共享， windows与windows之间通过共享目录实现共享

Linux与Windows之间怎样实现共享  
通过**Samba**

- samba安装：  
    sudo apt-get install samba  
    sudo apt-get install smbfs

# Samba配置

- 假设共享的目录为/home/wxq/share
- 登录samba的用户为wxq,可以通过useradd添加一个用户, 只要是ubuntu中的用户即可
- 设置登录密码:
  - sudo smbpasswd -a wxq,输入两次密码
  - 注意: 该密码不是用户登录ubuntu的密码, 是专用于登录samba的密码
- 新建文件 "/etc/samba/smbusers"
- sudo gedit /etc/samba/smbusers 添加:  
wxq = "network username"

# Samba配置

- 修改配置文件 `sudo gedit /etc/samba/smb.conf`

- 修改"security = user"

`security = user`

`username map = /etc/samba/smbusers`

- 修改[global], 添加如下代码

`workgroup = WORKGROUP`

`display charset = UTF-8`

`unix charset = UTF-8`

`dos charset = cp936`

# Samba配置

- 将下面代码放到文件的尾部

```
[Share]
```

```
comment = Share Folder with username and password
```

```
path = /home/glb/share
```

```
public = yes
```

```
writable = yes
```

```
valid users = glb      #确定的samba用户
```

```
create mask = 0700
```

```
directory mask = 0700
```

```
available = yes
```

```
browseable = yes
```

# Samba配置

- 启动samba服务：
  - /etc/init.d/samba start
- Window下访问linux服务器
  - \\192.168.0.47

