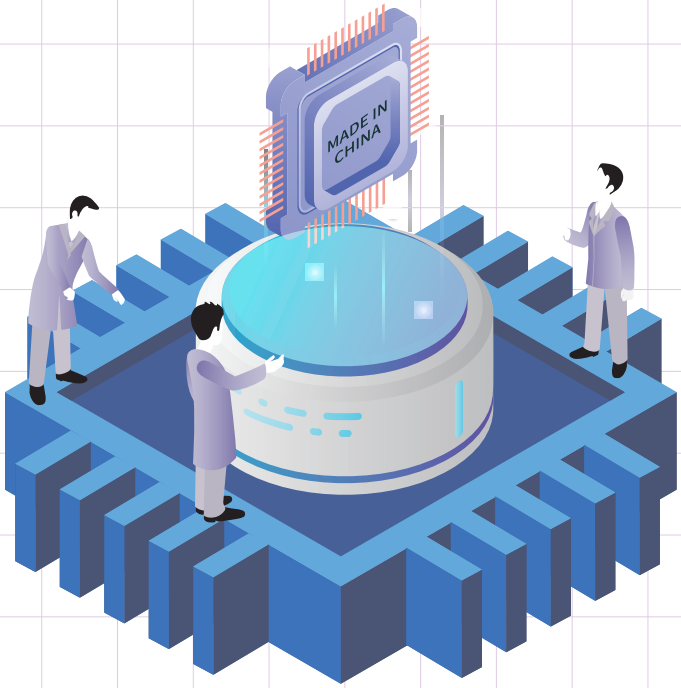
嵌入式开发人才培养系列教材
“互联网+”新形态一体化教材

嵌入式Linux应用开发编程基础

主编◎田　晶　张永华　刘孝国

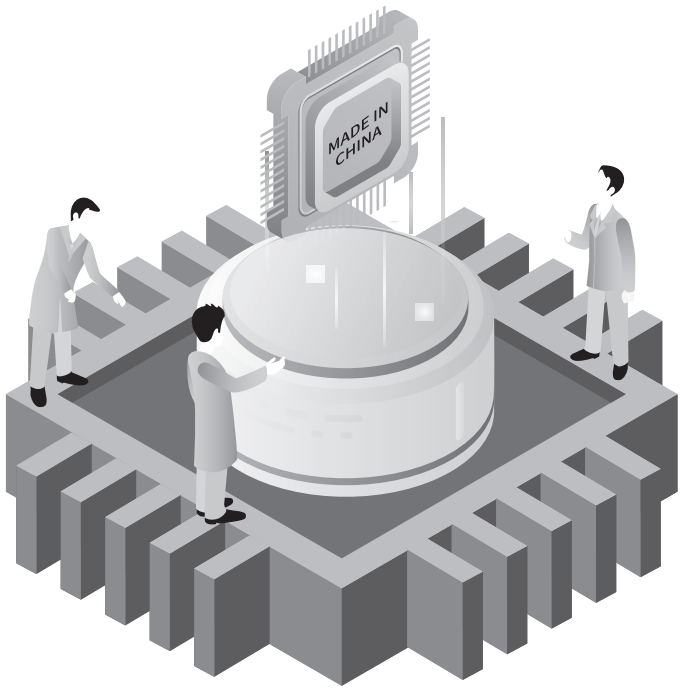


上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

嵌入式开发人才培养系列教材
“互联网+”新形态一体化教材

嵌入式Linux应用开发编程基础

主编◎田　晶　张永华　刘孝国



扫一扫
学习资源库

上海交通大學出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

内容提要

Linux 应用开发是嵌入式开发过程必不可少的环节。本书以任务驱动为导向，根据企业岗位需求抽取技能点组织成实训任务，内容涵盖搭建嵌入式 Linux 开发环境、嵌入式 Linux 文件 I/O 编程、嵌入式 Linux 多任务编程、嵌入式 Linux 进程间通信、嵌入式 Linux 多线程编程、嵌入式 Linux 网络编程、嵌入式 Linux 驱动编程等多个方面。本书详细介绍了 Linux 应用开发过程中的重点步骤，可操作性强，可作为物联网、嵌入式等相关专业的教学用书，也可作为广大嵌入式开发爱好者的自学用书。

图书在版编目（CIP）数据

嵌入式 Linux 应用开发编程基础 / 田晶，张永华，刘孝国主编. —上海：上海交通大学出版社，2024.3

ISBN 978-7-313-30217-5

Ⅰ. ①嵌… Ⅱ. ① 田… ② 张… ③ 刘… Ⅲ. ① Linux 操作系统—程序设计 Ⅳ. ① TP316.85

中国国家版本馆 CIP 数据核字（2024）第 006247 号

嵌入式 Linux 应用开发编程基础

QIANRUSHI Linux YINGYONG KAIFA BIANCHENG JICHU

主　　编：田晶　张永华　刘孝国
出版发行：上海交通大学出版社
邮政编码：200030
印　　制：北京荣玉印刷有限公司
开　　本：889mm × 1194mm　1/16
字　　数：406 千字
版　　次：2024 年 3 月第 1 版
书　　号：ISBN 978-7-313-30217-5
定　　价：49.80 元

地　　址：上海市番禺路 951 号
电　　话：021-6407 1208
经　　销：全国新华书店
印　　张：14.5
印　　次：2024 年 3 月第 1 次印刷
电子书号：ISBN 978-7-89424-547-2

编写委员会

主　编　田　晶　张永华　刘孝国

副主编　许春艳　金　鑫　佟　彤

编　委　宋海兰　王　莹　王　爽

于　薇　王占亮

前言

习近平总书记在党的二十大报告中指出：“教育是国之大计、党之大计。培养什么人、怎样培养人、为谁培养人是教育的根本问题。”其中，重点强调了要“深化教育领域综合改革，加强教材建设和管理”。本书是为适应新时期国家对职业教育的新要求，加快推进人才强国战略，健全现代化职业教育体系而开发的新形态教材。

Linux 是一种流行的操作系统，其内核和源代码都可以自由获得，因此它在全球范围内广泛应用于各种设备和平台。作为一种开放源代码的系统，Linux 不仅具有高度的安全性和稳定性，还提供了更多的自由度和灵活性。因此，越来越多的软件开发者选择使用 Linux 进行应用程序开发。

本教材是校企合作开发的新形态教材，以“企业岗位（群）任职要求、职业标准、工作过程或产品”为教材主体内容，以嵌入式 Linux 开发环境为依据，以嵌入式开发人员的职业岗位能力要求为出发点，确定学生应该掌握的专业知识和技术能力。本教材采用“理论讲解 + 实例解析 + 实训”的一体化教学方式，将系统开发所必需的理论知识构建于任务中，学生在完成具体项目的过程中完成相应工作任务，从而使学生掌握相应的理论知识及实际工作所需的职业能力。

本书以嵌入式 Linux 应用开发工作内容为基础，共设置了 7 个教学项目，项目 1 是搭建嵌入式 Linux 开发环境，项目 2 是嵌入式 Linux 文件 I/O 编程，项目 3 是嵌入式 Linux 多任务编程，项目 4 是嵌入式 Linux 进程间通信，项目 5 是嵌入式 Linux 多线程编程，项目 6 是嵌入式 Linux 网络编程，项目 7 是嵌入式 Linux 驱动编程。本教材的特点如下。

（1）切实贴合初学者和入门者的需求，讲解内容浅显易懂，注重实践操作，让读者通过实际操作来深入了解 Linux 应用程序开发。

（2）内容全面而丰富，涵盖了嵌入式开发环境的搭建、Linux 文件 I/O 编程、Linux 多任务编程、Linux 进程间通信、Linux 多线程编程、Linux 网络编程、Linux 内核驱动程序的编写等方面，可以帮助读者全面了解 Linux 应用程序开发所需的知识和技能。

（3）采用了大量的案例和实例，可以帮助读者更好地理解和掌握教材中的知识点，同时也可以提高读者的实际应用能力。

（4）突出了实践操作，每个项目都提供了一些实践任务，通过这些实践任务，读者可以深入了解 Linux 应用程序开发所需的技能，为将来的实际应用奠定基础。

（5）落实立德树人根本任务，贯彻《高等学校课程思政建设指导纲要》和党的二十大精神，将专

业知识与思政教育有机结合，实现价值引领、知识传授和能力培养紧密结合。

本教材的目标读者是想要学习 Linux 应用开发的人，包括那些想要开发跨平台应用程序的开发者。本教材假设读者已经有一定的编程经验，并且熟悉 Linux 操作系统的基本操作。如果您是一个新手，我们建议您先学习一些 Linux 操作系统的基本操作和基本的编程语言。

本书充分考虑初学者的学习特点，全书内容安排循序渐进、由易到难，同时尽可能地采取一步一步详解的教学方法，并为所有代码编写了详尽的注释，帮助读者更好地理解书中的内容。此外，编者还为广大一线教师提供了服务于本书的教学资源库，有需要者可致电 13810412048 或发邮件至 2393867076@qq.com 获取。

最后，我们希望这本教材可以帮助读者更好地了解 Linux 应用开发，并且能够使用 Linux 操作系统进行应用程序开发。如果您有任何问题或建议，请随时联系我们。

编者

2023 年 8 月

目录

项目 4　嵌入式 Linux 进程间通信 / 88

项目 5　嵌入式 Linux 多线程编程 / 130

项目 6　嵌入式 Linux 网络编程 / 153

项目 1

搭建嵌入式 Linux 开发环境

学习目标

知识目标

1. 了解嵌入式系统的概念。
2. 熟悉嵌入式 Linux 开发的流程。
3. 掌握嵌入式 Linux 开发环境搭建的步骤及内容。

能力目标

1. 会搭建嵌入式 Linux 上位机开发平台。
2. 会搭建嵌入式 Linux 目标机开发平台。

素质目标

1. 关注 Linux 发展现状，培养创新精神。
2. 了解我国在操作系统领域取得的成就。

项目导入

在以信息家电为代表的互联网时代，嵌入式产品不仅为嵌入式市场展现了美好前景，注入了新的生命，同时也对嵌入式操作系统（operating system，OS）技术提出了新的挑战。而 Linux 以其源代码开放、文档齐全、内核可裁剪等诸多优势，受到全球的众多嵌入式系统开发爱好者的喜爱。

嵌入式 Linux 是嵌入式系统开发中广泛使用的操作系统，它是 Linux 操作系统的一个精简版本，可以运行在低功耗、资源受限的嵌入式设备上，广泛应用在智能家居、工业自动化控制、智能医疗、智能安防、智能交通、智能农业、智能城市等物联网系统当中，使社会生活更加智能化。

目前，国产 Linux 操作系统的发展非常活跃。中国政府一直在推动自主研发操作系统，以减少对外国技术的依赖。中国已经有多个国产 Linux 操作系统项目在不断发展。其中，中标麒麟操作系统是最为知名的国产 Linux 操作系统之一。中标麒麟操作系统基于 Linux 内核，支持多种硬件架构，并提供了丰富的应用软件和工具，它在政府部门、教育机构和企业中得到了广泛应用。

总体来说，嵌入式 Linux 操作系统在中国的发展态势良好。随着技术的进步和国家政策的支持，可以预见，作为物联网底层的嵌入式 Linux 技术将继续发展壮大，并在未来发挥更加重要的作用。

任务 1.1　Windows 和 Linux 文件系统共享

1.1.1　嵌入式系统


Windows 和 Linux 文件系统共享 (1)


Windows 和 Linux 文件系统共享 (2)

嵌入式系统是指被嵌入到特定设备或系统中，用于完成控制、监测、处理和通信等特定任务的计算机系统。它通常集成在各种电子设备和系统中，包括智能手机、家电、汽车、医疗设备、工业控制系统等。

嵌入式系统是以应用为中心，以现代计算机技术为基础，能够根据用户需求、功能、可靠性、成本、体积、功耗、环境等因素的要求，灵活裁剪软硬件模块的专用计算机系统。

嵌入式系统被设计用于执行特定的任务，因此具有高度定制化和专用化的特点。与通用计算机系统相比，嵌入式系统的硬件和软件都经过精心设计，可以满足特定的需求和约束条件。这使得嵌入式系统在性能、功耗、大小和成本方面具有优势。

嵌入式系统通常由以下几个核心组件构成。

（1）处理器。嵌入式系统使用各种不同类型的处理器，包括微控制器（microcontroller unit，MCU）、数字信号处理器（digital signal processor，DSP）和嵌入式处理器（embedded processor，EP）。处理器负责执行系统的指令和算法。

（2）存储器。嵌入式系统需要存储程序代码、数据和配置信息。存储器可以分为内部存储器和外部存储器。内部存储器通常用于存储程序代码和数据，外部存储器用于扩展系统的存储容量。

（3）输入 / 输出接口。嵌入式系统需要与外部设备通信，如传感器、显示屏、键盘、网络等。输入 / 输出接口提供了与外部设备通信的方式，使得嵌入式系统能够接收输入的数据并输出处理结果。

（4）实时操作系统（real−time operating system，RTOS）。嵌入式系统通常需要满足实时性要求，即在特定时间范围内对输入做出响应并产生输出。实时操作系统提供了任务调度、中断处理和资源管理等功能，以确保系统能够按时完成任务。

嵌入式系统的应用非常广泛。在消费电子领域，智能手机、智能电视等都是嵌入式系统的典型应用。在汽车领域，嵌入式系统用于控制发动机、导航系统等。在工业控制领域，嵌入式系统被广泛应

用于自动化生产线、机器人控制、仪器仪表等。医疗设备、航空航天、军事装备等领域也离不开嵌入式系统的支持。

随着物联网（internet of things，IoT）的兴起，嵌入式系统的重要性进一步凸显。嵌入式系统支持连接到互联网，实现设备之间的通信和数据交换，推动了智能家居、智慧城市、智能交通等领域的发展。

随着技术的不断进步，嵌入式系统将继续发挥重要作用，并推动各个行业的创新与发展。

1.1.2　交叉编译

1. 交叉编译的概念

交叉编译的概念主要和嵌入式开发有关。所谓交叉编译（cross-compile），就是在一个平台上生成另一个平台上的可执行代码。这里的平台有两层含义：一是指处理器的体系结构；二是指运行的操作系统，如可以在 32 位的 Windows 操作系统开发环境上生成可以在 64 位 Linux 操作系统上运行的二进制程序。

使用交叉编译的主要原因是嵌入式系统中的资源太少，交叉编译出来的程序所要运行的目标环境拥有的各种资源都相对有限。编译开发会占用比较多的 CPU（central processing unit，中央处理器）、内存、硬盘等资源，嵌入式开发板的那点资源只够嵌入式（Linux）系统运行，没太多剩余的资源来进行本地编译。例如，在进行嵌入式开发时，目标平台（嵌入式开发板，采用最大主频为 200 MHz 的 ARM 处理器，32 MB 的随机存储器等）的硬件资源比较紧张，在运行嵌入式 Linux 的前提下，无法很方便地直接在嵌入式 Linux 系统中进行本地编译（在 ARM 的 CPU 下编译出来供 ARM 的 CPU 运行的程序）。

简单来说，交叉编译是指在宿主机上开发，在目标机上运行。

2. 宿主机和目标机

一般把进行交叉编译的主机称为宿主机（host machine），也就是普通的通用计算机，而把程序实际的运行环境称为目标机（target machine），也就是嵌入式系统环境。由于一般通用计算机拥有非常丰富的系统资源、使用方便的集成开发环境和调试工具，而嵌入式系统的系统资源非常紧缺，没有相关的编译工具，因此，嵌入式系统的开发需要借助宿主机（通用计算机）来编译出目标机的可执行代码。

宿主机：编辑和编译程序的平台，一般是基于 x86 的计算机，通常也称为主机。

目标机：用户开发的系统，通常都是非 x86 平台。宿主机编译得到的可执行代码在目标机上运行。

3. 交叉编译的模式

交叉编译器一般有两种模式，一种是 Java 模式，另一种是 GCC（GNU compiler collection，GNU 编译器套件；GNU 为一个自由软件社区）模式。本书只讲述 GCC 模式，即在宿主机上交叉编译得到可执行文件，通过调试器下载到目标系统中调试运行。GCC 交叉开发模式模型如图 1-1 所示。

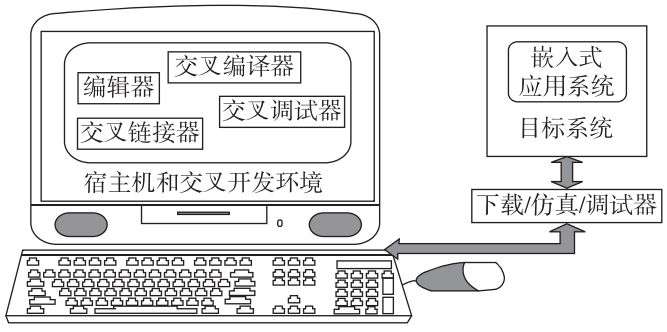


图 1-1　GCC 交叉开发模式模型

搭建交叉开发环境是嵌入式开发的第一步。搭建交叉开发环境要安装交叉编译链以实现文件共享，这是必须要做的第一个任务。

实验 ——Windows 和 Linux 文件系统共享


Windows 和 Linux 文件系统共享 (3)

常用的实现 Windows 和 Linux 文件系统共享的方式有三种，分别为使用虚拟工具——VMware Tools；使用 Samba Server；使用 FTP（file transfer protocol，文件传输协议）软件 FileZilla。下面对这三种方式分别进行介绍。

1. 安装虚拟工具——VMware Tools

VMware Tools 包含一系列服务和组件，可在各种 VMware 产品中实现多种功能，从而使用户能够更好地管理客户操作系统，以及与客户操作系统进行无缝交互。

在 Ubuntu 环境下安装 VMware Tools 的具体步骤如下。

（1）开启虚拟机，运行想要安装 VMware Tools 的系统。进入系统后，单击虚拟机上方菜单栏中的“虚拟机 (M)”→“安装 VMware Tools”，图 1-2 所示是系统已经安装过该软件的界面。

（2）第（1）步完成后，系统桌面会有一个 VMware Tools 文件，进入文件目录，可以看到如图 1-3 所示的界面。

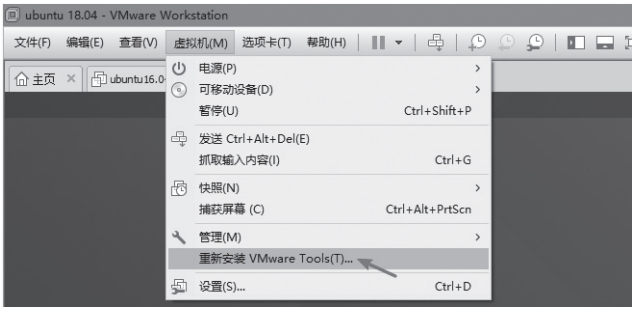


图 1-2 选择“重新安装 VMware Tools(T)…”

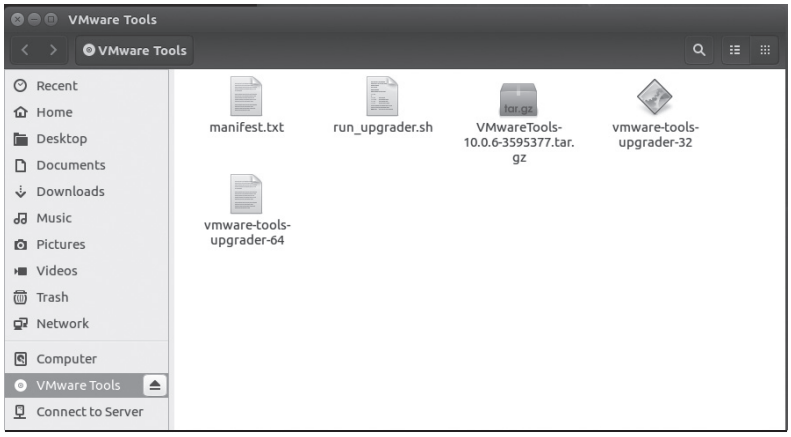


图 1-3 VMware Tools 目录下文件

（3）使用终端解压 VMwareTools-xxxx.tar.gz 文件，解压完成后进入该目录，使用 ls 命令查看当前目录是否有 VMware Tools 的安装程序 vmware-install.pl。如果存在，则输入命令安装。

```
#sudo ./vmware-install.pl
```

（4）在安装的过程中，按照提示继续完成安装，出现如图 1-4 所示的界面表示安装成功。

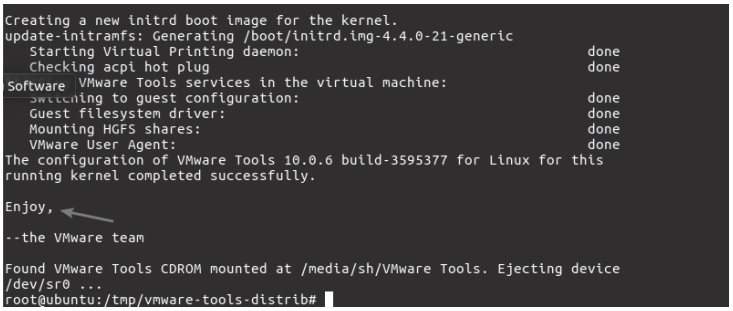


图 1-4　安装成功界面

（5）重启系统，就可使用 VMware Tools 了。

接下来，在系统中设置共享文件夹，具体步骤如下。

（1）打开 VMware 的界面（注意系统不要启动），单击“虚拟机 (M)”→“设置”→“选项”，出现如图 1-5 所示的界面。

图 1-5　虚拟机设置界面

（2）在添加共享文件夹向导中，选择要共享的文件夹，如图 1-6 所示。

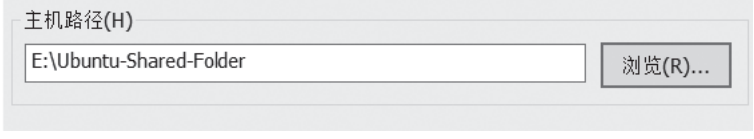


图 1-6　共享文件夹主机路径设置

（3）选择“启用此共享”（见图 1-7），单击完成。

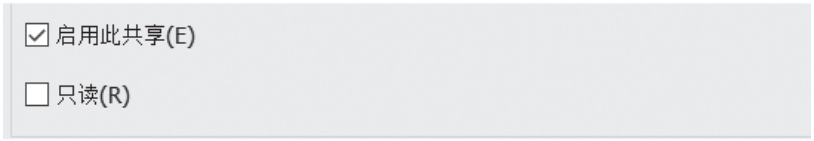


图 1-7　选择“启用此共享 (E)”

（4）打开虚拟机进入 Ubuntu 系统，在根目录下，进入“ mnt”→“ hgfs”就可以看到在主机共享的文件夹，如图 1-8 所示。

图 1-8　共享目录“hgfs”下文件显示界面

2. 使用 Samba Server 共享文件

Samba 是在 Linux 系统上实现 SMB（server message block，信息服务块）协议的一款免费软件。它可以实现在局域网内共享文件和打印机，是一个客户机 / 服务器型协议。客户机通过 SMB 协议访问服务器上的共享文件。

Samba 服务器的工作原理：客户端向 Samba 服务器发起请求访问共享目录，Samba 服务器接受请求后查询 smb.conf（/etc/samba/smb.conf）文件，查看共享目录是否存在以及访问者的权限。如果访问者具有相应的权限，则允许客户端访问，并将访问过程中系统的信息以及采集的用户行为放在日志文件（/var/log/samba）中。

使用 Samba Server 共享文件的步骤如下。

（1）确认是否安装 Samba 服务。

```
# dpkg -l |grep samba
```

（2）建立一个共享文件夹，并修改权限。

```
# mkdir /home/share
# chmod 777 share
```

（3）设置 Linux 和 Windows 操作系统的 IP 为同一网段。

```
Linux:  10.130.120.105
Windows: 10.130.120.5
```

（4）配置 Samba , 修改 smb.conf 文件。习惯上先创建备份文件。

```
# cp  /etc/samba/smb.conf  /etc/samba/smb.conf.bak
```

（5）使用 vi 命令打开 smb.conf 文件，在该文件中添加如图 1-9 所示的内容。

```
# vi  /etc/samba/smb.conf
```

```
[share]
comment=Shared folder with username and password
path=/home/share
writable=yes
valid users=root
create mask=0770
directory mask=0770
force user=root
force group=root
available=yes
browseable=yes
```

图 1-9　smb.conf

（6）测试配置文件。

```
# sudo testparm
```

（7）启动 Samba 服务。

```
# sudo service smbd restart
# sudo service nmbd restart
```

（8）测试 SMB。

```
# smbclient – L  10.130.120.105
```

（9）登录 SMB。

```
# smbclient  //10.1.103.200/share
```

（10）测试文件共享是否成功。

在 Windows 下，按“Win+R”组合键调出运行窗口，在命令行内输入“\\”+ Linux 的 IP 地址。

3. Windows 系统下小工具——FTP 软件 FileZilla

FileZilla 是一个免费开源的 FTP 软件，分为客户端版本和服务器版本，具备 FTP 软件所有的功能。可控、有条理的界面和管理多站点的简化方式使得 FileZilla 客户端成为一个方便高效的 FTP 客户端工具，而 FileZilla Server 则是一个小巧、可靠且支持 FTP&SFTP（secure file transfer protocol，安全文件传输协议）的 FTP 服务器软件。该软件的图标如图 1-10 所示。

图 1-10　FileZilla 软件图标

注意事项

在使用 Samba 服务器实现文件共享时，需要注意以下两点：

（1）在 Windows 系统下设置共享文件夹一定要设置密码；

（2）确定 Windows 和 Linux 系统哪一端是服务器，哪一端是客户端。

任务 1.2　上位机 Linux 和开发板 Linux 文件共享

嵌入式开发的特点是在宿主机（上位机）上开发，在目标机（下位机）上运行。在宿主机上开发的程序，如何烧写到目标机上？这是一个跨平台的操作，该如何处理呢？

这一节的任务就是解决上位机 Linux 系统下的文件和目标机 Linux 系统文件的共享，采用的方法是利用 NFS 服务。

1.2.1　NFS 网络文件系统

NFS 是 network file system 的缩写，即网络文件系统。NFS 的主要功能是通过局域网络让不同的主机系统之间可以共享文件或目录。 NFS 由 SUN 公司开发，已经成为文件服务的一种标准（RFC 1904、RFC 1813）。其最大功能是可以通过网络让不同操作系统的计算机共享数据，所以也可以将其看作一台文件服务器，如图 1-11 所示。NFS 提供了除 Samba 之外，实现 Windows 与 Linux、UNIX 与 Linux 之间通信的方法。

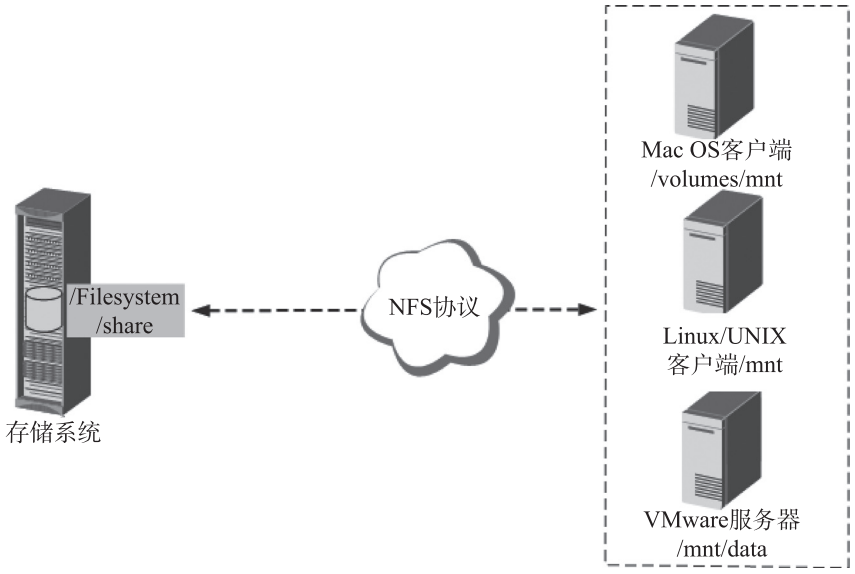


图 1-11　网络文件系统

NFS 系统和 Windows 系统的网络共享、网络驱动器类似，只不过 Windows 用于局域网，NFS 用于企业集群架构。如果是大型网站，还会用到更复杂的分布式文件系统 GlusterFS（大文件 ISO 镜像等）、Ceph 或者 OSS（阿里云对象存储系统）。

客户端计算机可以挂载 NFS 服务器所提供的目录，并且挂载之后这个目录看起来如同本地的磁盘分区，可以使用 cp、cd、mv、rm 及 df 等与磁盘相关的命令。

NFS 有属于自己的协议与使用的端口号，但是在传输资料或者其他相关信息时，NFS 服务器使用一个称为远程过程调用（remote procedure call，RPC）的协议来协助 NFS 服务器本身的运行。

1.2.2　NFS 工作原理

NFS 服务器可以让 PC（personal computer，个人计算机）将网络中的 NFS 服务器共享的目录挂载到本地端的文件系统中，而在本地端的系统来看，那个远程主机的目录就好像是自己的一个磁盘分区一样，在使用时相当便利。NFS 挂载原理如图 1-12 所示。

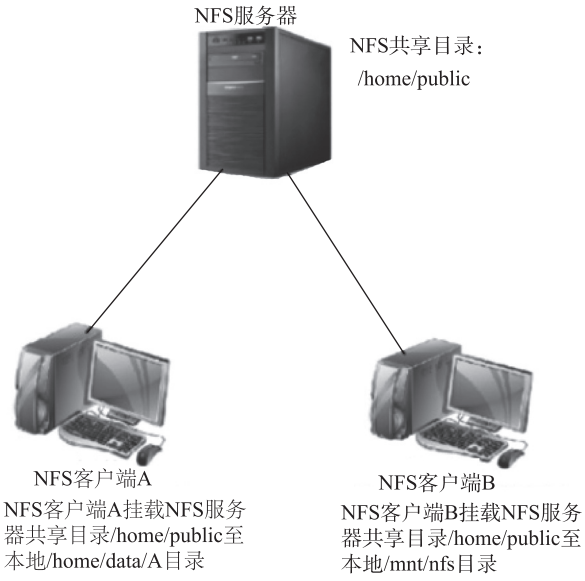


图 1-12　NFS 挂载原理

1.2.3　NFS 常用命令

NFS 常用的命令如表 1-1 所示。

表 1-1　NFS 常用命令

序号	常用选项	描述
1	rw	允许读写
2	ro	只读
3	sync	同步写入
4	async	先写入缓冲区，必要时才写入磁盘，速度快，但会丢数据
5	subtree_check	若输出一个子目录，则 NFS 服务将检查其父目录权限
6	no_subtree_check	输出子目录时，NFS 服务不检查其父目录权限，提高效率
7	no_root_squash	客户端以 root 登录时，赋予其本地 root 权限
8	root_squash	客户端以 root 登录时，将其映射为匿名用户
9	all_squash	将所有用户映射为匿名用户

实验　——利用 NFS 服务实现文件共享

常用的实现上位机 Linux 和开发板 Linux 文件系统共享的方式就是利用 NFS 服务。先在本机测试 NFS 服务的功能，再连接开发板，实现宿主机与目标机 Linux 文件共享。

实现上位机 Linux 本机挂载，测试 NFS 服务的功能，具体步骤如下。

（1）查看是否安装 NFS 服务，并确认其版本。

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

（2）设置 NFS 共享目录。

上位机 Linux 和开发板 Linux 文件共享

```
/home/nfs
```

（3）在 /etc/exports 中添加如下语句，保存后退出。

```
/home/nfs  *(rw,sync,no_root_squash,no_subtree_check)
```

（4）启动 NFS 服务。

```
# sudo /etc/init.d/portmap restart
# sudo /etc/init.d/nfs-kernel-server restart
```

（5）测试 NFS 服务，以本机 mnt 目录为例。

```
# mount –t nfs –o nolock  192.168.100.192:/home/nfs /mnt
```

（6）切换到 mnt 目录下查看。

```
# cd /home/nfs/mnt
# ls
```

（7）查看共享挂载信息。

```
# df -h
```

（8）解除挂载。

```
# umount mnt
```

（9）停止 NFS 服务。

```
# sudo /etc/init.d/nfs-kernel-server stop
```

在本机测试通过以后连接开发板，利用 NFS 实现上位机 Linux 和开发板 Linux 文件共享，这是本任务的重点内容。实现上位机 Linux 和开发板 Linux 文件共享的步骤如下。

（1）连接主机与开发板设备。实验选用的开发板设备如图 1-13 所示。

（2）打开 SecureCRT 工具，设置串口连接，如图 1-14 所示。

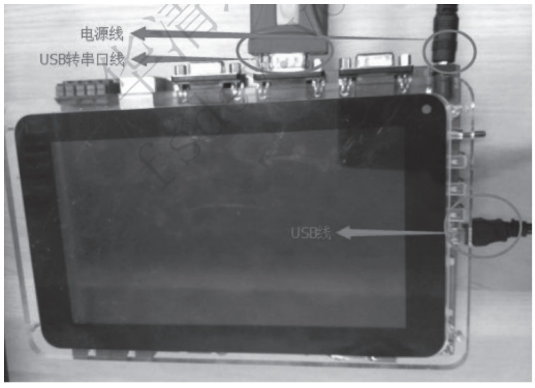


图 1-13　开发板设备

图 1-14　设置串口连接

（3）重启开发板，进入主界面。

（4）在终端输入以下命令。

```
# mount –t nfs –o nolock  192.168.100.192:/home/nfs /mnt
```

注意事项

在 NFS 服务使用过程中注意以下 3 点：

（1）NFS 服务多用于局域网内；

（2）搭建服务时一定要先启动 RPC 服务，后启动 NFS 服务；

（3）配置文件中的信息格式一定要正确，否则会报错。

任务 1.3　构建嵌入式 Linux 目标平台

最终是在目标机（下位机）上运行程序，那么在目标机上搭建环境时需要做哪些准备工作呢？

嵌入式操作系统与通用操作系统的最显著的区别之一就是它的可移植性。一个嵌入式操作系统通常可以运行在不同体系结构的处理器和开发板上。为了使嵌入式操作系统可以在某款具体的目标设备上运行，嵌入式操作系统的编写者通常无法一次性完成整个操作系统的代码，而必须把一部分与具体硬件设备相关的代码作为抽象的接口保留出来，让提供硬件的 OEM（original equipment manufacture，原厂委托制造）厂商来完成，这样才可以保证整个操作系统的可移植性。

嵌入式系统从软件角度来看有引导加载程序（固化在 firmware 固件中的程序 + Bootloader）、Linux 内核、根文件系统三个层次，如图 1–15 所示。

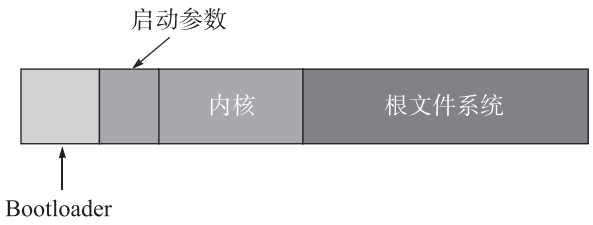


图 1–15　嵌入式系统软件的三个层次

1.3.1　Bootloader

1. Bootloader 的概念

Bootloader 是在操作系统运行之前运行的一段程序。这段程序可以初始化硬件设备、建立内存空间的映像表，从而建立适当的系统软硬件环境，为最终调用操作系统内核做好准备。

引导加载程序包括固化在固件（firmware）中的 boot 代码（可选）和 Bootloader 两部分。

有一些 CPU 在运行 Bootloader 之前先运行一段固化的程序，比如 x86 结构的 CPU 就是先运行 BIOS（basic input output system，基本输入输出系统）固件，然后才运行磁盘上第一个分区（MBR）中的 Bootloader。大多嵌入式系统中并没有 BIOS 固件，Bootloader 是上电（接通电源）后执行的第一个程序。

对于嵌入式系统来说，Bootloader 是基于特定硬件平台来实现的，因此不能通用，其不但依赖于

CPU 体系结构还依赖于嵌入式系统板级设备的配置。

系统加电或复位后，所用的 CPU 通常都从 CPU 制造商预先安排的地址开始执行。比如 S3C2410 在复位后从地址 0x00000000 起开始执行。而嵌入式系统则将固态存储设备（比如 FLASH）安排在这个地址上，而 Bootloader 程序又安排在固态存储器的最前端，这样就能保证在系统加电后，CPU 首先执行 Bootloader 程序，如图 1-16 所示。

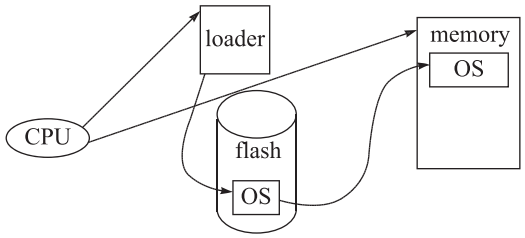


图 1-16　CPU 加载系统

2. Bootloader 的工作流程

Bootloader 启动大多数分成 Bootloader Stage1 和 Bootloader Stage2 两个阶段。

（1）第一阶段（Stage1）完成以下工作。

①硬件设备初始化，包括芯片内的寄存器，如 ARM 的 CPSR（current program status register，程序状态寄存器）。

②为加载 Bootloader 的第二个阶段代码准备 RAM 空间。

③将 Bootloader 第二个阶段代码拷贝到 RAM 空间中。

④设置好堆栈。

⑤跳转到第二阶段代码的 C 程序入口点。

（2）第二阶段（Stage2）完成以下工作。

①初始化本阶段要使用的硬件设备（开发板上的网卡等）。

②将内核映像和根文件系统映像从 flash 上读取到 RAM 中。

③启动内核。

第一阶段使用汇编来实现，它完成一些依赖于 CPU 体系结构的初始化，并调用第二阶段的代码；第二阶段则通常使用 C 语言来实现，这样可以实现更加复杂的功能，而且代码会有更好的可读性和可移植性。

3. 常见的 Bootloader 的种类

嵌入式系统中常见的 Bootloader 有以下几种。

（1）GRUB（GRand Unified Bootloader）：这是一个广泛使用的开源引导加载程序，用于在多个操作系统之间进行选择和引导。

（2）LILO（Linux Loader）：这是一个较早的引导加载程序，主要用于引导 Linux 操作系统。

（3）Windows Boot Manager：这是 Windows 操作系统中的引导加载程序。

（4）U-Boot：这是一个开源的引导加载程序，通常用于嵌入式系统和嵌入式 Linux 设备。

（5）Clover Bootloader：这是一个专为“黑苹果”（Hackintosh，指非苹果公司产品但使用苹果操作系统的设备）设计的引导加载程序，允许在非苹果硬件上运行苹果的 macOS 操作系统。

（6）Systemd-Boot：这是 Systemd 初始化系统的一部分，用于引导 Linux 操作系统。

这里仅列举了一些常见的引导加载程序，还有其他一些用于特定硬件或操作系统的引导加载程序，此处不再赘述。

1.3.2　Linux 内核

1. Linux 内核概述

Linux 是一个开源的操作系统。它由 Linus Torvalds 构思设计而成，当时还在读大学的 Linus 想要基于 UNIX 的原则和设计来创建一个免费的开源系统，从而代替 MINIX 操作系统。如今，Linux 不仅是公共互联网服务器上常用的操作系统，还是速度排名前 500 的超级计算机上使用的唯一一款操作系统。

Linux 最大的优势是它的开源属性。Linux 是一款基于 GNU 通用公共许可证（general public license，GPL）发布的操作系统，这意味着所有人都能运行、研究、分享和修改这个软件。经过修改后的代码还能重新发布，甚至出售，但必须基于同一个许可证。这一点与传统操作系统（如 UNIX 和 Windows）截然不同，传统操作系统都是锁定供应商、以原样交付且无法修改的专有系统。

Linux 内核源代码官方网站为 http://www.kernel.org。Linux 主要包括桌面版本和 Linux 内核。

（1）桌面版本。桌面版本是面向计算机用户的桌面发行的版本，常见的有 RedHat、Fedora、Debian、Ubuntu、SUSE、红旗等。

（2）Linux 内核。内核是所有 Linux 系统的中心软件组件。嵌入式领域所说的 Linux 一般是指 Linux 内核。移植也是指移植 Linux 内核到目标平台。

2.Linux 内核的主要功能

（1）进程管理。进程是计算机中资源分配的最小单元。进程管理控制系统中的多个进程对 CPU 的访问，使得多个进程能在 CPU 中“微观串行，宏观并行”地执行。进程调度处于系统的中心位置，内核中其他的子系统都依赖它，因为每个子系统都需要挂起或恢复进程。

（2）内存管理。内存是计算机系统中重要的资源。内存管理的主要作用是控制多个进程安全地共享主内存区域。当 CPU 提供存储管理部件（memory management unit，MMU）时，Linux 内存管理为每个进程完成虚拟内存到物理内存的转换。Linux 2.6 引入了对无 MMU CPU 的支持。

（3）文件管理。Linux 系统中的任何一个概念几乎都可以看作一个文件，而虚拟文件系统（virtual file system，VFS）隐藏了各种硬件的具体细节，为所有的设备提供了统一的接口。而且，它独立于各个具体的文件系统，是对各文件系统的抽象。它使用超级块 super block 存放文件系统的相关信息，使用索引节点 inode 存放文件的物理信息，使用目录项 dentry 存放文件的逻辑信息。

（4）进程间通信管理。该功能用于支持多种进程间的信息交换。

（5）网络管理。Linux 内核支持各种网络标准协议和网络设备，提供了对各种网络标准的存取和对各种网络硬件的支持。

Linux 内核的功能模块之间的关系如图 1-17 所示。

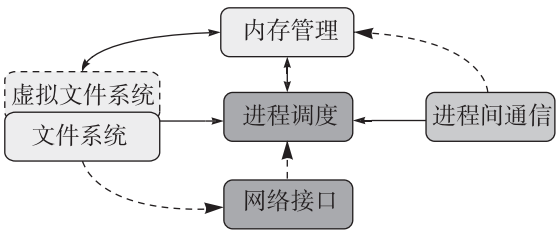


图 1-17　Linux 内核的功能模块之间的关系

由图 1-17 可以看出，所有的模块都与进程调度模块存在依赖关系，因为它们都需要依靠进程调度程序来挂起或重新运行它们的进程。通常，一个模块会在等待硬件期间被挂起，在操作完成后才可继续运行。当一个进程试图将一个数据块写到软盘上去时，软盘驱动程序就可以在启动软盘旋转期间将该进程设置为挂起等待状态，在软盘进入正常转速后再让该进程继续运行。另外 3 个模块也是由于类似的原因而与进程调度模块存在依赖关系。

3. Linux 内核的结构

Linux 内核的目录结构源代码非常庞大，随着版本的发展不断增加。它使用目录树结构，并且使用 Makefile 组织配置编译。初次接触 Linux 内核，最好仔细阅读顶层目录的 README 文件，它包括 Linux 内核的概述和编译命令说明。README 的说明更加针对 x86 等通用的平台，对于某些特殊的体系结构，可能有些特殊的地方。内核源码很复杂，包含多级目录，形成一个庞大的树状结构，通常称为 Linux 源码目录树。Linux 的文件系统目录如表 1-2 所示。

表 1-2　Linux 的文件系统目录

目录	描述
/bin	存放二进制可执行文件，这些命令在单用户模式下也能够使用。可以被 root 和一般的账户使用
/boot	存放内核和启动文件，如 vmlinuz-xxx。grub 引导装载程序
/dev	设备驱动文件
/etc	存放一些系统配置文件，如用户账户和密码文件，以及各种服务的起始地址
/home	系统默认的用户主文件夹。一般创建用户账户的时候，默认的用户主文件夹都会放到此目录下
/lib	存放库文件
/media	此目录下放置可插拔设备，如 SD 卡或者 U 盘就是挂载到这个目录中
/mnt	用户可使用的挂载点，如果要挂载一些额外的设备，那么可以挂载到此处
/opt	可选的文件和程序的存放目录，放置第三方软件的目录
/root	root 用户目录，也就是系统管理员目录
/sbin	和 /bin 类似，也是存放一些二进制可执行文件。sbin 目录中存放的一般是系统开机过程中所需要的命令
/srv	服务相关信息的目录，比如网络服务
/sys	记录内核信息，虚拟文件系统
/tmp	临时目录
/var	存放一些变化的文件，如日志文件
/usr	usr 不是 user 的缩写，而是 UNIX software resource 的缩写，存放与系统用户有关的文件，会占用很大的存储空间

1.3.3　Linux 的文件系统与根文件系统

1. Linux 的文件系统

Linux 支持多种文件系统，包括 ext2、ext3、vfat、jffs、romfs 和 nfs 等。为了对各类文件系统进行统一管理，Linux 引入了虚拟文件系统（virtual file system，VFS），为各类文件系统提供一个统一的应用编程接口。

根据存储设备硬件特性和系统需求不同来选择以下几种。

jffs：专用于 NOR flash（或非型闪存），可读写，支持数据压缩的日志型文件系统。

yaffs：专用于 NAND flash，可读写，不压缩。

cramfs：既可用于 NOR flash 又可用于 NAND flash，只读。

nfs：在计算机系统之间通过网络共享文件的文件系统。

ramdisk：基于 RAM 的文件系统。

2. Linux 的根文件系统

Linux 系统由 Linux 内核与根文件系统两部分构成，两者缺一不可。

Linux 根文件系统首先是一种文件系统，但是相对于普通的文件系统，它的特殊之处在于，它是内核启动时所挂载的第一个文件系统。若没有根文件系统，Linux 将无法正常启动。

Linux 的根文件系统以树型结构组织，包含内核和管理系统所需要的各种文件和程序，一般来说根目录 / 下的顶层目录都有一些比较固定的命名和用途。

内核启动的最后步骤是挂载根文件系统，包含以下内容。

（1）开启 init 进程。

（2）开启 shell（命令解释程序）。

（3）读取文件系统、网络系统等工具集。

（4）读取系统配置文件。

（5）创建链接库。

实验——构建开发平台


构建嵌入式 Linux 目标平台

目标平台是嵌入式 ARM Cortex-A9（FS4412）开发平台。移植共分为三个部分，分别为引导程序、内核和根文件系统。使用 fastboot 工具来烧写，需要上位机与目标机用 USB（universal serial bus，通用串行总线）连接。

利用 fastboot 工具烧写的具体步骤如下。

（1）打开目录，找到镜像文件（见图 1-18），然后根据设备型号选择对应名称的文件夹，将该目录下的文件拷贝到“E:\ShareVMware”。

（2）连线接好后，启动串口调试助手 putty 并对开发板上电，启动开发板，putty 在倒数计时的过程中，可按任意键停止。

（3）首先输入“fdisk -c 0 600 3000 600”，会出现如图 1-19 所示的界面。

图 1-18　镜像文件

```
FS4412 # fdisk -c 0 600 3000 600
.fdisk is completed

partion #    size(MB)     block start #    block count    partition_Id
   1         10684          8634368         21880832         0x0C
   2           600            32768          1228800         0x83
   3          3000          1261568          6144000         0x83
   4           600          7405568          1228800         0x83
FS4412 #
```

图 1-19　输入 fdist 后

（4）然后输入“fastboot”，会出现如图 1-20 所示的界面。

（5）在 Windows 系统中进行烧写，按“Win+R”组合键调出运行窗口，输入“cmd”并单击“确定”按钮进入命令行界面，如图 1-21 所示。

（6）进入烧写镜像的文件夹里面进行烧写，这里放在 E:\ ShareVMware 文件夹下。

（7）输入“E：”后按回车键，再次输入“cd ShareVMware”并按回车键切换到 E:\ ShareVMware 目录下。

（8）搭建好 fastboot 环境后输入“fastboot -w”。

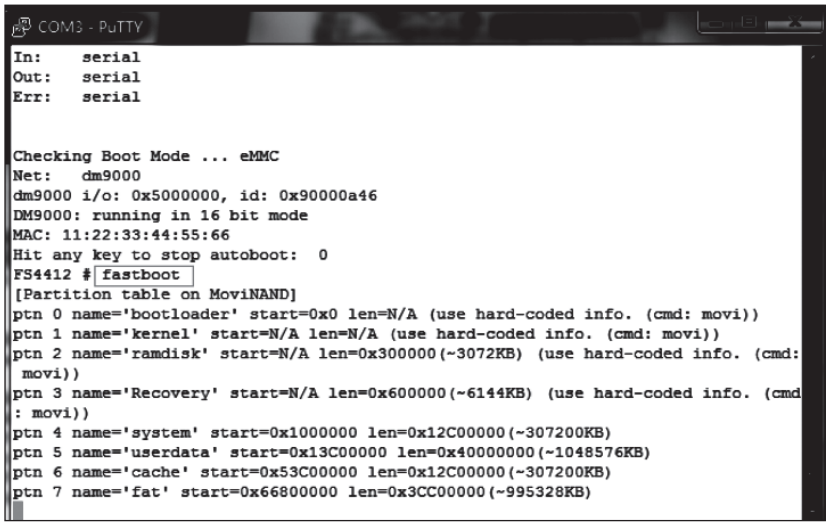


图 1-20　输入 fastboot 后

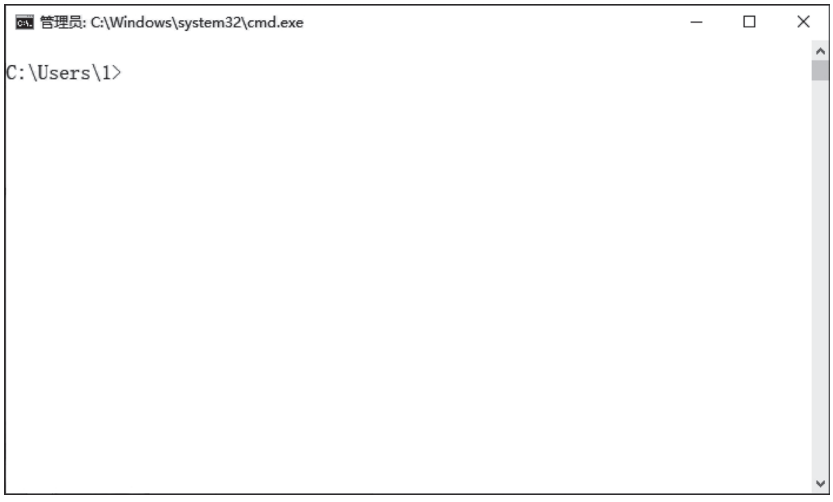


图 1-21　命令行界面

（9）烧写 uboot。

```
fastboot flash bootloader u-boot-fs4412.bin
```

（10）烧写 zImage。

```
fastboot flash kernel zImage
```

（11）烧写根文件系统。

```
fastboot flash system system.img -S 300M
```

（12）测试。

到这里为止，Linux 系统所需要的镜像全部烧写完毕，重启开发板并在倒数计时的时候按任意键，在 putty 里面输入“print”命令查看当前的环境变量，如图 1-22 所示。

```
Environment size: 319/16380 bytes
FS4412 # print
bootdelay=5
baudrate=115200
ethaddr=11:22:33:44:55:66
ethact=dm9000
ipaddr=192.168.11.191
gatewayip=192.168.11.1
serverip=192.168.11.10
bootcmd=movi read kernel 40008000;bootm 40008000
bootargs=root=/dev/mmcblk0p2 rootfstype=ext4 init=/linuxrc console=ttySAC2,115200 lcd=WA101S
stdin=serial
stdout=serial
stderr=serial

Environment size: 319/16380 bytes
```

图 1-22　查看当前环境变量

（13）设置 bootcmd 和 bootargs 参数，命令如下。

```
setenv bootargs root=/dev/mmcblk0p2 rootfstype=ext4 init=/linuxrc console=ttySAC2,115200
lcd=WA101S
setenv bootcmd movi read kernel 40008000\;bootm 40008000
save
```

（14）重新启动开发板完成烧写，如图 1-23 所示。

```
[    3.465341] input: gt818 as /devices/virtual/input/input3
[    3.472898] <<-GTP-INFO->>[1368]IC VERSION:18c3_0083
[    3.477017] <<-GTP-INFO->>[1706]Chip type:GT818X.
[    3.481818] <<-GTP-INFO->>[1725]GTP works in interrupt mode.
[    3.487350] <<-GTP-INFO->>[199]Applied memory size:2562.
[    3.492656] <<-GTP-INFO->>[220]Create proc entry success!
[    3.498019] gt818--------probe success
[    3.502238] drivers/rtc/hctosys.c: unable to open rtc device (rtc1)
[    3.508476] FIMC0 registered successfully
[    3.512516] FIMC1 registered successfully
[    3.516344] FIMC2 registered successfully
[    3.520425] FIMC3 registered successfully
[    3.524204] S5P TVOUT Driver v3.0 (c) 2010 Samsung Electronics
[    3.565471] EXT4-fs (mmcblk0p2): mounted filesystem with ordered data mode.
pts: (null)
[    3.572177] VFS: Mounted root (ext4 filesystem) on device 179:2.
[    3.578165] Freeing init memory: 224K
[root@farsight ]#
```

图 1-23　完成烧写

注意事项

关于目标平台的搭建，需要注意以下 3 点：

（1）移植 Bootloader 时，注意 Bootloader 的选型；

（2）烧写过程中，注意烧写镜像文件的大小是否与原文件大小一致；

（3）设置 bootcmd 和 bootargs 的参数时格式要正确。

任务 1.4　安装交叉编译器

前面做好了准备工作，现在正式进入安装交叉编译器的环节，选择适当的版本和型号，下载并安装交叉编译器。

1.4.1 交叉编译器

在一种计算机环境中运行的编译程序能编译出在另外一种环境下运行的代码，就称这种编译器支持交叉编译，这个编译过程就叫交叉编译。简单地说，就是在一个平台上生成另一个平台上的可执行代码。这里需要注意的是，所谓平台，实际上包含两个概念：体系结构、操作系统。同一个体系结构可以运行不同的操作系统；同样，同一个操作系统也可以在不同的体系结构上运行。举例来说，人们常说的 x86 Linux 平台实际上是 Intel x86 体系结构和 Linux for x86 操作系统的统称；x86 WinNT 平台实际上是 Intel x86 体系结构和 Windows NT for x86 操作系统的统称。

交叉编译器可以在上位机编写源程序代码，编译生成在目标机上可执行的机器代码。例如，在通用计算机上生成在 Android 智能手机上运行的代码的编译器就是交叉编译器。

交叉编译器的基本用途是将构建环境与目标环境分开，这在以下几种情况下很有用。

（1）设备资源极其有限的嵌入式计算机。例如，微波炉有一个非常小的计算机来读取它的键盘和门传感器，向数字显示器和扬声器提供输出，并控制烹饪食物。这台计算机通常不够强大，无法运行编译器、文件系统或开发环境。

（2）多台机器编译。例如，公司可能希望支持多个不同版本的操作系统，通过使用交叉编译器，可以设置单个构建环境来为这些目标中的每一个进行编译。

（3）在服务器上编译。与为多台机器编译类似，一个涉及许多编译操作的复杂构建可以在任何免费机器上执行，无论其底层硬件或运行的操作系统版本如何。

（4）引导到新平台。在为新平台或未来平台的模拟器开发软件时，人们使用交叉编译器来编译必要的工具，如操作系统和本地编译器。

（5）使用在当前平台上运行的交叉编译器（如在 Windows 下运行的 AztecC 的 MS-DOS 6502 交叉编译器）为现在已经过时的旧平台（如 Commodore 64 或 Apple II）编译模拟器的本机代码。

虚拟机（如 Java 虚拟机 JVM）解决了开发交叉编译器的一些问题。虚拟机范例允许跨多个目标系统使用相同的编译器输出，尽管这并不总是理想的，因为虚拟机通常较慢，并且编译的程序只能在具有该虚拟机的计算机上运行。

通常硬件架构不同时使用交叉编译，例如，在 x86 计算机上编译面向 MIPS（microprocessor without interlocked pipeline stages，无互锁流水线微处理器）架构的程序。但只有操作系统环境不同时，交叉编译也适用，例如在 Linux 下编译 FreeBSD 程序，甚至只是系统库，就像在 glibc 主机上用 uClibc 编译程序一样。

1.4.2 常用的交叉编译工具

1. 交叉编译器 arm-linux-gcc

该编译器和 x86 平台下的 gcc 的基本用法是完全一样的，不同之处在于标准的 gcc 所引用的头文件路径为 /usr/include/stdio.h，而 arm-linux-gcc 所引用的头文件路径为它的安装路径，如 /usr/local/armtools/4.5.1/bin/...。

2. 交叉链接器 arm-linux-ld

arm-linux-ld 是 ARM 平台下的交叉链接器，把程序链接成可以在 ARM 平台下运行的状态。

用法：

```
arm-linux-ld -T led.lds led.o -o led.elf
```

说明：把 led.o 链接成 led.elf 文件；led.lds 是链接器脚本。

3. 交叉 elf 文件工具 arm-linux-readelf

用法：

```
arm-linux-readelf -a led.elf
```

说明：-a 参数用来查看 .elf 文件的所有内容。

4. 交叉反汇编器 arm-linux-objdump

把 hello.c 文件编译成 hello 可执行文件：

```
arm-linux-gcc hello.c -o hello
```

把 hello 可执行文件反汇编后输入保存到 dump 文件中：

```
arm-linux -objdump -D -S hello >dump
```

说明：-D -S 是反汇编参数，>dump 把 hello 反汇编后的内容保存到 dump 文件中。

【注意】编译的程序运行不了有两个原因：一是要看运行平台对不对，二是要看处理器的大小端跟编译的程序的大小端是否对应，可以使用 arm-linux-readelf -a xxx.elf 命令查看编译出来的程序的大小端情况和程序运行平台。

5. 文件格式转换器 arm-linux-objcopy

使用 arm-linux-objcopy 命令可以把 elf 格式的文件转换成二进制文件。

文件格式转换的原因：elf 格式的文件不能直接在 ARM 上运行（ARM 只能运行二进制格式的文件）。

用法：

```
arm-linux-objcopy -O binary led.elf led.bin
```

说明：把 led.elf 格式的文件转换成 led.bin 二进制文件。

6. 库管理器 arm-elf-ar

arm-elf-ar 将多个可重定位的目标模块归档为一个函数库文件。采用函数库文件，应用程序能够从该文件中自动装载要参考的函数模块，同时将应用程序中频繁调用的函数放入函数库文件中，易于应用程序的开发管理。arm-elf-ar 支持 elf 格式的函数库文件。

实验　——安装交叉编译器

交叉编译常用的工具有 gcc、readelf、size、nm、strip、strings、objdump 和 addr2line。本实验使用的交叉编译器是 arm-linux-gcc，版本是 4.6.4，读者可以根据 Linux 操作系统及目标机系统选择适用的交叉编译器型号及版本。

安装交叉编译器 – 子任务 2

安装交叉编译器的步骤如下。

（1）创建交叉编译器存储路径。

```
# mkdir /usr/local/toolchain
```

（2）拷贝 gcc-4.6.4.tar.xz 到 toolchain 目录。

```
# cp  gcc-4.6.4.tar.xz  /usr/local/toolchain
```

（3）解压。

```
# tar xvf gcc-4.6.4.tar.xz
```

（4）配置环境变量，修改文件 /etc/bash.bashrc，在文件中添加如下内容。

```
# export PATH=$PATH:/usr/local/toolchain/4.6.4/bin
```

（5）重启配置文件。

```
# source /etc/bash.bashrc
```

（6）测试。

```
# arm-none-linux-gnueabi-gcc  -v
```

【思考】在安装后，通过查看交叉编译器的版本信息可以确认交叉编译器已经安装成功。那么如何测试它是否好用呢？

可以利用编辑器编写一个 Helloworld 程序 hello.c，然后使用如下命令来测试能否生成二进制文件 hello。

```
$arm-none-linux-gnueabi-gcc hello.c -o hello
```

注意事项

关于安装交叉编译器，需要注意的 3 点：

（1）交叉编译器的版本选择；

（2）交叉编译器类型的选取要满足使用的硬件设备；

（3）安装后需要测试编译器。

学习评价

任务 1.1：Windows 和 Linux 文件系统共享			
能否独立完成操作任务 不能掌握□	仅能理解□	仅能操作□	能理解会操作 □
任务 1.2：上位机 Linux 和开发板 Linux 文件共享			
能否独立完成操作任务 不能掌握□	仅能理解□	仅能操作□	能理解会操作 □
任务 1.3：构建嵌入式 Linux 目标平台			
能否独立完成操作任务 不能掌握□	仅能理解□	仅能操作□	能理解会操作 □

续表

任务 1.4：安装交叉编译器
能否独立完成操作任务 不能掌握□　　仅能理解□　　仅能操作□　　能理解会操作 □

项目总结

搭建嵌入式 Linux 开发环境是在进行嵌入式开发时非常重要的一个环节，正确的开发环境可以提高开发人员的工作效率，并能够保证开发过程的质量。开发人员的操作从如下几方面入手。

（1）硬件准备。嵌入式开发需要一些硬件设备，如开发板、串口线、USB 转串口设备。先确定需要什么样的硬件，然后再购买和准备。

（2）系统安装。嵌入式开发主要在 Linux 环境下进行，因此需要先将 Linux 系统安装到开发机上。可以使用虚拟机进行安装，也可以直接在物理机上安装。

（3）软件安装。Linux 系统安装好后，还需要安装一些必要的开发软件，如 gcc、make、git 等。此外，还需要安装交叉编译器，以保证编译出来的程序可以运行在目标设备上。

（4）配置开发环境。在安装软件后，还需要对开发环境进行一些配置，如配置环境变量、添加 PATH 等，方便编译和运行程序。

（5）下载内核源码。在进行嵌入式开发时需要使用内核源码，因此需要下载内核源码并进行编译。可以选择自己编译内核，也可以使用已经编译好的内核。

（6）编译和烧录程序。编译程序时需要使用交叉编译器，将程序编译成目标设备可以运行的二进制文件。然后再使用烧录软件将程序烧录到目标设备的存储器中。

搭建嵌入式 Linux 开发环境需要耐心和一定的经验，考虑到系统涉及硬件和软件集成的特性，一次操作不一定能成功，如果遇到困难，可以根据本书中的操作步骤，重复进行操作。

拓展训练

一、填空题

1. 在嵌入式系统开发中，所谓交叉编译环境，就是在（　　）上开发，在（　　）下运行。
2. 解除挂载的命令是（　　）。
3. 利用 NFS 服务实现上位机 Linux 和目标机 Linux 文件共享时，（　　）是服务器，（　　）是客户端。
4. 系统移植是指（　　）。
5. 构建嵌入式 Linux 目标平台，需要在目标机上移植（　　）、（　　）和（　　）。

二、选择题

1. 在创建 Linux 分区时，一定要创建（　　）两个分区。

   A. FAT/NTFS　　B. FAT/SWAP　　C. NTFS/SWAP　　D. SWAP/ 根分区

2. 当使用 mount 进行设备或者文件系统挂载的时候，需要用到的设备名称位于 (　　) 目录。

   A. /home　　B. /bin　　C. /etc　　D. /dev

3. 如果要列出一个目录下的所有文件，需要使用命令 (　　)。

A. ls –l　　B. ls　　C. ls –a　　D. ls –d

4. Samba 服务器的配置文件是 (　　)。

A. httpd.conf　　B. inetd.conf　　C. rc.samba　　D. smb.conf

5. 以下哪一个是 Linux 内核的稳定版本（　　）

A. 2.5.24　　B. 2.6.17　　C.1.7.18　　D. 2.3.20

6. NFS 是（　　）系统。

A．文件　　B. 磁盘　　C. 网络文件　　D. 操作

7. 在 vi 编辑器里，命令 dd 用来删除当前的 (　　)。

A. 行　　B. 变量　　C. 字　　D. 字符

8. 在搭建嵌入式 Linux 目标平台时，需要移植下列哪些部分 (　　)?（多选）

A. 引导系统 Bootloader　　B. 内核

C. 根文件系统　　D. 应用程序

三、简答题

1. 请举例说明目前使用的主流 Linux 的发行版本。
2. 请举例说明你所了解的嵌入式产品。
3. 如何理解交叉编译。