

文章编号: 1005—0523(2006)05—0075—04

智能仪器中微型 Linux 的构建

杨丰萍, 邢 剑, 卢伟海

(华东交通大学 电气与电子工程学院, 江西 南昌 330013)

摘要: 在新型智能仪器仪表中, 功能强大但资源消耗较小的微型 Linux 有很好的应用价值. 本文以主机/目标板为开发模式, 介绍了裁减系统内核、构建根文件系统和设置引导装入器等过程. 并根据仪器仪表的具体情况, 在构建微型 Linux 过程中对需要注意的问题进行了详细的说明.

关 键 词: 微型 Linux; 系统裁减; 根文件系统; 引导装入器

中图分类号: TP316

文献标识码: A

1 概述

随着以计算机技术为代表的高新技术的迅速发展, 电子测量仪器的发展取得了重大突破. 电子测量仪器的演化与发展从总体上看沿着两条主线展开.

一是从所采用的技术上看经历了模拟仪器、数字仪器、智能仪器的发展过程. PC 仪器作为智能仪器发展的一种新阶段, 实际上就是建立在 PC 机系统上的测量仪器, 它由硬件和软件两大部分组成. 仪器的硬件部分主要包括 CPU、存储器、总线、各种 I/O 接口、输入输出通道, 仪器面板(键盘、开关、按钮、显示器)等.

二是从仪器的结构形式上看, 经历了独立仪器、模块化仪器、虚拟仪器和网络化仪器的发展过程. 网络化仪器的应用有效地降低了测试系统的成本, 实现了远程测控和资源共享, 实现了测试设备的远程诊断与维护. 网络化正成为测试技术发展的新动向.

Linux 是开放源代码的操作系统, 与其他操作系统相比, 具有安全性和可靠性高、成本低、对硬件的广泛支持、网络功能强大等特点, 微型 Linux 是从标准 Linux 裁减而来, 继承了 Linux 的所有优点, 又大

大的降低了硬件资源的消耗, 在仪器 PC 化和网络化的进程中, 将成为首选操作系统之一.

2 实现微型 Linux 的方案选择

包括仪器仪表在内的嵌入式系统, 通常是资源受限的系统, 无论处理器速度计算能力还是 RAM 或其他存储器容量都比较“小”. 因此如何创建一个微型的 Linux 作为系统的软件平台成为首先考虑的问题. 一般宿主机和目标机硬件平台的异构, 需要首先建立交叉平台开发环境, 交叉平台开发环境包括交叉编译器、交叉调试器和系统仿真器, 比如嵌入式 Linux 开发经常用的 GUN 工具链. 开发者需要根据目标平台选择合适的 GUN 交叉编译器, 然后在宿主机上重新编译内核和其他软件, 这样得到的目标代码才能拿到目标机上面运行. 这个过程相当烦琐而且容易出错.

近年来, 越来越多的目标系统选择了性价比不断提高的 X86 处理器和成熟的 PC 架构作为硬件平台, LinuxDevices.com 网站 2003 年 5 月进行的调查显示, 嵌入式开发者在过去 2 年和未来 2 年选择 X86 处理器作为目标平台的比例分别为 31% 和 35%, 高居首位.

对于宿主机和目标机都是 PC 兼容平台的开发

收稿日期: 2006—06—25

作者简介: 杨丰萍(1967—), 女, 江西萍乡人, 副教授.

者来说,除了沿用一般交叉编译开发的模式外,有更简单的创建微型 Linux 的办法:一个普通 Linux 发行版本为原型,通过编译裁减内核,移植必须的文件,创建小型的文件系统,重新引导操作系统,就实现了一个微型的 Linux。

开发嵌入式 Linux 系统有三种不同的主机/目标板架构:连接式设置、可抽换存储装置设置、独立式设置。

连接式设置:如图 1 所示,这种设置的目标板和主机会一直被缆线连接在一起.此连接通常就是一条串行线或一条 Ethernet 连接.这种设置的主要特色是,目标板与主机之间并未用到实际的硬件存储装置来传送数据.所有数据都是通过连接传送的.主机包含了跨平台开发环境,而目标板则包含了适当的引导加载程序、可用的内核以及最起码的根文件系统。

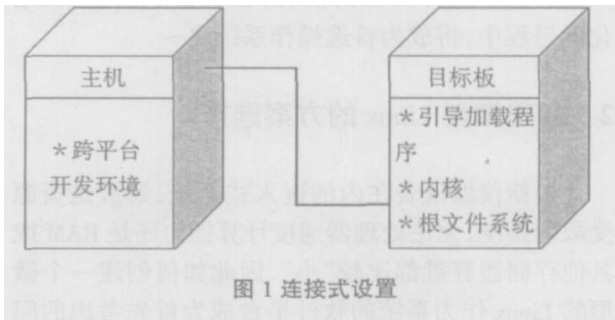


图 1 连接式设置

可抽换存储装置设置:如图 2 所示,主机和目标板之间没有实际的连接,取而代之的是,先由主机将数据写入存储装置,然后将存储装置连接到目标板,并用该存储装置引导.主机包含了跨平台开发环境,而目标板则只包含了最起码的引导加载程序,其余的组件被存放在抽换式存储媒体上.先用主机将这些主件写入抽换式存储媒体,然后由目标板上最起码的引导加载程序在启动时加载。

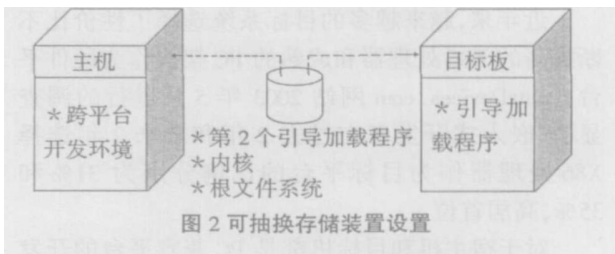


图 2 可抽换存储装置设置

独立式设置:目标板是个独立的开发系统,它包含了引导、操作以及开发额外软件所必需的任何软件.本质上,这种设置就象在使用一台工作站,只不过其底下的硬件并非传统工作站,而是嵌入式系统本身.这种设置不需要任何跨平台开发环境,因为所有的开发环境都会在固有的环境中执行.而

且,目标板与主机之间不需要任何数据传送工作,因为所有必要的存储装置都放在目标板本地。

鉴于测试系统硬件资源限制,无法采用独立式设置架构,又因为测试系统采用和主机平台一样的 X86 架构的硬件平台,而在每次内核或根文件系统有变动的时候,可以先在主机平台上模拟实现,不必通过插拔存储装置来实现.所以本系统开发采用最受欢迎的可抽换存储装置设置架构来实现。

3 构建微型 Linux

以红帽子 Linux 发行版为例,构建一个微型嵌入式 Linux 操作系统大致分为三个步骤:

根据功能需要裁减编译内核、设计根文件系统和设置引导装入器。

3.1 裁减编译内核

内核,是一个操作系统的核心.它负责管理系统的进程、内存、设备驱动程序、文件和网络系统,决定着系统的性能和稳定性.Linux 内核是一个高度模块化、可配置的,可以通过裁减掉不需要的功能来减小内核的大小,也可以根据实际情况选择所需的功能。

- 1) 下载并解压 linux-2.4.XX 内核到 /usr/src;
- 2) # cd /usr/src/linux-2.4.XX //进入内核源码目录;
- 3) # make mrproper //该命令确保源代码目录下没有不正确的 .o 文件以及文件的互相依赖;
- 4) # make xconfig //进行内核配置,去除不需要的选项,根据实际需要选择所需功能,例如在系统中使用安装稳定可靠、抗震性能比较好 DOC2000 作为存储设备,需要内核支持以下选项:

Memory Technology Device(MTD) support 选项

MTD partitioning support 选项

Direct char device access to MTD devices 选项

NFTL(NAND Flash Translation Layer)support 选项

Write support for NTFL(BETA)选项

Self-contained MTD device drivers 次级菜单里的 M-Systems Disk-On-Chip 2000 and Millennium 选项

- 5) # make dep //读取配置过程生成的配置文件,来创建对应于配置的依赖关系树;

- 6) # make clean //完成删除前面步骤留下的文件,避免出现一些错误;

- 7) # make bzImage //实现完全编译内核;

```
8) # make modules //生成相应的模块;
9) # make modules-install //把模块拷贝到需要的目录中;
10) # cp /usr/src/linux/arch/i386/bzImage /boot/vmlinuz-2.4.XX //拷贝新内核到目标目录.
```

3.2 设计根文件系统

Linux 根文件系统采用的是树型结构.最上层是根目录,其他的所有目录都是从根目录出发而生成.根文件系统用“/”表示,是存放 Linux 系统运行所必须的库文件、各种工具软件、脚本、配置文件和其他特殊文件(例如设备节点)的地方.

创建根文件系统就是建立如下目录:/bin、/boot、/dev、/etc、/lib、/sbin、/mnt、/usr、/tmp、/var、/home、/initrd、/proc 等等.并移植微型 Linux 系统运行所必需的文件、文件夹到相应目录下.

鉴于系统硬件资源有限,也不需要过多的功能,所以需要重新编写/etc/inittab、/etc/rcs、reboot、halt 等脚本;如果使用 DOC2000,必须在/dev 目录下新创建 mtd0, nftla, nftla1 等设备文件.

如果存储设备容量十分有限,可以使用 BusyBox 和 uClibc 链接库来构建系统. BusyBox 是一个小程序,可以根据配置,执行 shell,以及几十个各种小应用程序,包括 vi 编辑器、/sbin/init 初始化程序以及 ifconfig、halt、reboot、mkdir、mount、ln、ls、echo、cat 等等系统上必不可少的程序,并对它们进行了缩减,压缩到 100K 左右,根据用户需要,还可以进一步缩小.相对于越来越大而且越来越全的 glibc 链接库,相对小巧的 uClibc 链接库是 glibc 链接在嵌入式领域的重要替代品,它提供 glibc 链接库类似的功能.

基本上所有的仪器仪表都不用键盘作为输入设备,所以登陆系统时需要取消身份验证,自动以 root 身份登陆,为此需要重新修改并编译 mingetty 和 login 的源代码.

首先,安装 mingetty 的源代码包,打开 mingetty.c 文件,并找到如下代码:

```
... ..
while ((logname = get __logname ()) == 0);
execl (__PATH __LOGIN, __PATH __LOGIN, "
", logname, NULL);
error ("%s: can't exec \"%PATH __LOGIN\": %s",
tty, sys __errlist[errno]);
... ..
```

将其修改为:

```
//while ((logname = get __logname ()) ==
0);
logname = "checker";
execl (__PATH __LOGIN, __PATH __LOGIN,
"- ", logname, NULL);
error ("%s: can't exec \"%PATH __LOGIN\": %s",
tty, sys __errlist[errno]);
... ..
重新编译,生成 mingetty.
然后安装 util-linux-2.11r-10.src.rpm 软件包,修改 login.c 文件,
... ..
fflag = hflag = pflag = 0;
//passwd-req = 1 //缺省时,要求进行密码验证,注释掉本行
passwd-req = 0 //添加本行
... ..
```

编译生成 login.将生成的 mingetty 和 login 拷贝到目标系统相应位置,即可实现自动登录.

3.3 设置引导装入器

引导装入器的任务是从引导设备装载内核,引导系统运行. Linux 系统常见的引导装入器包括 LILO 和 GRUB 等. GRUB 功能更强、更加灵活.所以本文以 GRUB 为例来说明引导装入器的设置.

一般磁盘设备的引导配置和红帽子安装版默认安装的配置是类似的,只要稍微修改磁盘设备名、分区和路径即可.相比较而言,DOC 设备的引导比较困难.

DOC 设备包含了一个称为 IPL 的 ROM 程序,系统启动时它会被视为 BIOS 的扩展,由 BIOS 负责执行.当它执行时, IPL 会安装另一个程序 SPL.要在 DOC 设备上使用 GRUB 引导, SPL 必须被替换成“为了从 DOC 引导而定制的”GRUB 版本.

下载 grub-0.92-doc.patch 和 GRUB-0.92,移到 grub-0.92 目录并运行

```
# patch -p0 < grub-0.92-doc.patch
# aclocal && automake && autoconf
# ./configure --enable-diskonchip2000 --enable-diskonchip-ctrlbypass
--enable-ext2fs --disable-ffs --disable-xfss --disable-jfs
--disable-vstafs --disable-reiserfs --disable-minix --disable-fat
# make
```

编译好后, stage/grub _firmware 就是将被写入 DOC 的 GRUB 映象, 将映象安装到 DOC 设备, 最后编写 GRUB 的配置文件 menu.lst 文件将 GRUB 设定成从 DOC 引导.

4 总结

将大小为几兆或者十几兆的微型 Linux 移植到 X86 等嵌入式处理器上, 加上外围设备如液晶触摸和微型打印机, 并转接一些网口、USB 接口以供数据通讯, 可以较容易实现仪器仪表的 PC 化和网络化. 相对一些极耗硬件资源操作系统和一些价格昂贵的嵌入式操作系统, 将微型 Linux 应用在仪器仪表

中成本较低, 因此具有较高的实用价值. 在仪器仪表的 PC 化和网络化进程中, 微型 Linux 将发挥重要作用.

参考文献:

[1] 林 涛, 等. Linux 在嵌入式系统中的实现. 微计算机信息[J]. 2005(7).
[2] Karim Yagbmour. 构建嵌入式 Linux 系统[M]. 北京: 中国电力出版社, 2004.
[3] 罗 奕. Linux 操作系统的定制和精简[J]. 计算机时代, 2005(5)
[4] 怀石工作室编著. Linux 上的 C 编程[M]. 北京: 中国电力出版社, 2001

The Construction of Minisize Linux in Intelligent Instruments

YANG Feng-ping, XING Jian, LU Wei-hai

(School of Electrical and Electronic Eng., East China Jiaotong University, Nanchang 330013, China)

Abstract: Minisize Linux with powerful function but lesser resource consumed is paying a more and more important role in new intelligent instruments. Under the development mode of main machine / target machine, this paper introduces some important processes such as reducing system kernel, constructing root file system and setting boot loader. And as constructing minisize Linux some needing attention problems are explained in detail according to concrete situations of instruments.

Key words: Minisize Linux; system reduction; root file system; bootloader