

《嵌入式系统》

（第三讲）

厦门大学信息学院软件工程系 曾文华

2023年9月26日

第3章 嵌入式Linux操作系统

- 3.1 嵌入式Linux简介
- 3.2 内存管理
- 3.3 进程管理
- 3.4 文件系统

- 嵌入式系统对操作系统（OS）的要求：

- ① 高度简练

- ② 界面友善

- ③ 质量可靠

- ④ 易开发

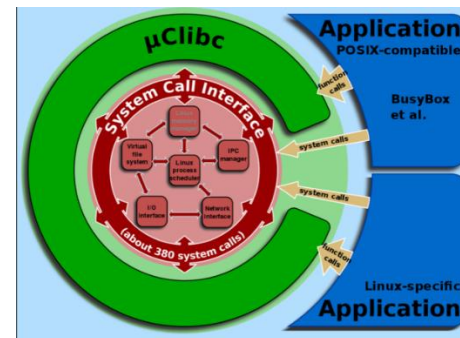
- ⑤ 多任务

- ⑥ 价格低

3.1 嵌入式Linux简介

• 3.1.1 μ Clinux

- μ : Micro, 微
- C: Control, 控制
- μ Clinux: 用于微控制器（单片机）的Linux操作系统



– μ Clinux的特点:

MMU: Memory Management Unit, 内存管理单元

- ① 针对微控制领域设计的Linux系统。
- ② 编译后的目标文件可控制在几百KB大小。
- ③ 专门针对**没有MMU**（存储管理单元）的处理器设计的。
- ④ 无法使用**虚拟内存管理技术**，采用实存储器管理策略，所有程序访问的地址都是实际的物理地址。

虚拟内存管理技术是一种利用虚拟存储器来逻辑扩充物理内存的管理方式，其基本思想是用软硬件技术把内存与外存这两级存储器当成一级存储器来用，从而给用户提供了一个比内存也比任何应用程序大得多的虚拟存储器，使得用户编程时再也不用考虑内存大小的限制了，给用户编程带来极大的方便。

• 3.1.2 RT-Linux（硬实时操作系统）

- RT-Linux（Real-Time Linux）：具有**硬实时**特性的多任务操作系统，是美国新墨西哥州立大学计算机科学系Victor Yodaiken和Micae Brannanov开发的。
- 通过在Linux内核与硬件中断之间增加一个精巧的**可抢先的实时内核**，把标准的Linux内核作为实时内核的一个进程与用户进程一起调度，标准的Linux内核的优先级最低，可以被实时进程抢断。
- 正常的Linux进程仍可以在Linux内核上运行，这样既可以使用标准分时操作系统（即Linux的各种服务），又能提供低延时的实时环境。

Linux内核

可抢先的实时内核

硬件中断



• 3.1.3 红旗嵌入式Linux

- 北京红旗软件技术有限公司推出的，国内做得较好的嵌入式Linux操作系统。
- 红旗嵌入式Linux的特点：
 - ① 精简内核，适用于多种常见的嵌入式处理器。
 - ② 提供完善的嵌入式GUI和嵌入式X-Windows。
 - ③ 提供嵌入式浏览器、邮件程序和多媒体播放程序。
 - ④ 提供完善的开发工具和平台。

自2000年6月成立以来，红旗软件迅速实现了主营业务的全面盈利，形成了以北京为总部，在广州、上海等地设有分公司的架构，并拥有立体的销售渠道和遍布全国的服务网络。红旗软件提供的产品涵盖了高端Linux服务器操作系统、集群系统、桌面版操作系统、嵌入式系统以及技术支持服务和培训等领域，用户广泛分布在政府、邮政、教育、电信、金融、保险、交通、运输、能源、物流、媒体和制造等各个行业。2014年2月10日，中科红旗北京总部的大门上粘贴了一张最新的公告，通知全体员工，公司正式解散，员工劳动合同全部终止，公司进入清算程序。伴随着这张公告，这家成立14年之久的国产操作系统厂商的发展历史可能就此终止。2014年6月27日，中科红旗发布公告，宣布拟以公开竞价方式转让公司全部注册商标、全部软件著作权等资产，正式启动资产出售进程。

X Windows 系统其实就是 Linux 上的图形界面系统。众所周知，Linux 是一种基于命令行的操作系统，本身是没有图形界面的，但是后来为了让更多的人喜欢 Linux，人们就专门开发了一种图形界面系统，因为在英文字母表中 X 排在 Windows 的首字母 W 之后，所以人们玩笑式的将它命名为“X Windows”。与微软的 Windows 图形界面不同，X Windows 系统并没有集成在 Linux 操作系统的内核中，而仅仅是 Linux 系统上的一套应用程序罢了，所以你是可以自由安装和卸载 X Windows 的，而微软的图形界面由于集成在内核中，所以是不能自由卸载的。

3.2 内存管理

• 3.2.1 内存管理和MMU

– 内存管理（也称存储管理）包括：

- ① 地址映射
- ② 内存空间的分配
- ③ 地址访问的限制（即保护机制）
- ④ I/O地址的映射
- ⑤ 代码段、数据段、堆栈段空间的分配

– MMU（Memory Management Unit）的主要作用：

- ① 地址映射
- ② 对地址访问的保护和限制（访问保护、保护机制）

– MMU就是提供一组寄存器，依靠这组寄存器来实现地址映射和访问保护；MMU可以在CPU芯片中，也可以作为协处理器（以协处理器的形式实现）。

• 3.2.2 标准Linux的内存管理

- 标准Linux使用**虚拟存储器**技术。
- 为了支持虚拟存储管理器的管理，标准Linux系统采用**分页**的方式来载入进程。
- 虚拟存储器由存储器管理机制（**MMU**）及一个大容量的快速硬盘存储器支持。
- **虚拟地址**被送到MMU，由MMU将虚拟地址映射为**物理地址**。

– 标准Linux的内存管理具有以下功能：

- ① 运行比内存还要大的程序。
- ② 先加载部分程序运行，缩短了程序启动的时间。
- ③ 可以使多个程序同时驻留在内存中，以提高CPU的利用率。
- ④ 可以运行重定位程序，即程序可以放在内存中的任何一处，且可以在执行过程中移动。
- ⑤ 写机器无关的代码，程序不必事先约定机器的配置情况。
- ⑥ 减轻程序员分配和管理内存资源的负担。
- ⑦ 可以进行内存共享，例如，如果两个进程运行同一个程序，它们应该可以共享程序代码的同一个副本。
- ⑧ 提供内存保护，进程不能以非授权方式访问或修改页面，内核保护单个进程的数据和代码以防止其它进程修改它们，否则，用户程序可能会偶然（或恶意）的破坏内核或其它用户程序。

• 3.2.3 μ CLinux的内存管理

- μ CLinux: 用于微控制器（单片机）的Linux操作系统。
- μ CLinux没有MMU，而是采用**实存储器管理**（Real Memory Management）。
- μ CLinux不能使用虚拟内存管理技术（虚拟存储器技术），但仍然采用存储器的**分页管理**。
- μ CLinux将整个物理内存划分为若干个**4KB**大小的**页面**。

3.3 进程管理

• 3.3.1 进程和进程管理

- **进程**是一个运行程序并为其提供执行环境的实体，它包括一个地址空间和至少一个控制点，进程在这个地址空间上执行单一指令序列。进程地址空间包括可以访问或引用的内存单元的集合，进程控制点通过程序计数器（PC）控制和跟踪进程指令序列。
- **进程管理**是操作系统的功能之一，特别是多任务处理的状况下，这是必要的功能。操作系统将资源分配给各个进程，让进程间可以分享与交换信息，保护每个进程拥有的资源，不会被其他进程抢走，以及使进程间能够同步。为了达到这些要求，操作系统为每个进程分配了一个数据结构，用来描述进程的状态，以及进程拥有的资源。操作系统可以通过这个数据结构，来控制每个进程的运作。
- **任务调度**主要是协调任务对计算机系统内资源（如内存、I/O设备）的争夺使用。
- **进程调度**又称为CPU调度，其根本任务是按照某种原则为处于就绪状态的进程分配CPU。
- 很多场合下，**任务调度**和**进程调度**的概念是一致的。

– 与进程管理相关的几个概念：

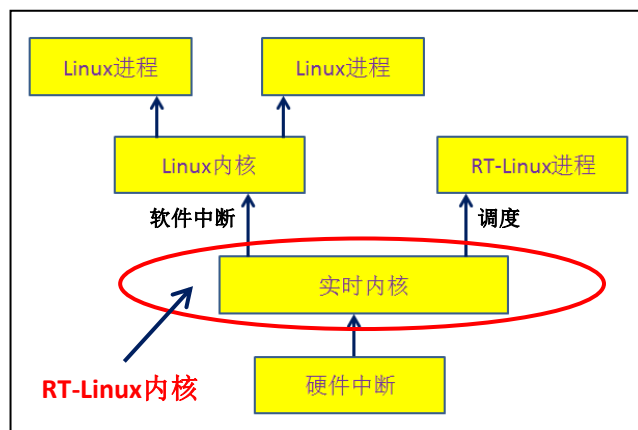
- ① **上下文**：进程运行的环境。
- ② **上下文切换**：**CPU**的控制权由运行进程转移到另外一个就绪进程时所发生的事件，当前运行进程转为就绪（或者挂起、删除）状态，另外一个被选定的就绪进程成为当前进程。
- ③ **抢占**：正在执行的进程可以被打断，让另外一个进程运行。
- ④ **优先抢占**：每一个进程都有一个优先级，如果有进程优先级高于当前的进程优先级，系统立即保存当前进程的上下文，切换到优先级高的进程上下文。
- ⑤ **轮转调度**：使所有相同优先级、状态为就绪的进程，公平分享**CPU**（分配一定的时间间隔，使各进程轮流享有**CPU**）。

— 进程调度策略（两种）：

- ① **抢占式调度**：一旦就绪状态中出现优先权更高的进程，或者运行的进程已用满了规定的时间片时，便立即抢占当前进程的运行（将其放回就绪状态），把CPU分配给其他进程。
- ② **非抢占式调度**：一旦某个进程被调度执行，该进程一直执行下去，直至该进程结束，或者由于某种原因自行放弃CPU进入等待状态，才将CPU重新分配给其他进程。

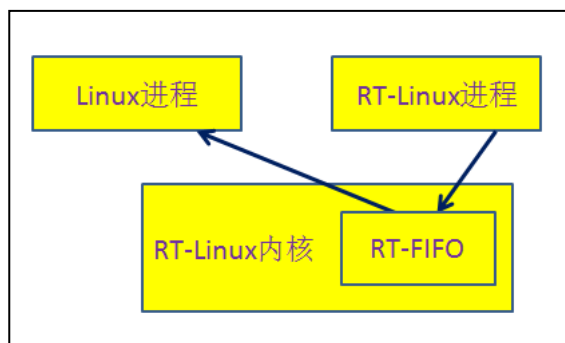
• 3.3.2 RT-Linux的进程管理

- RT-Linux：硬实时Linux操作系统。
- RT-Linux有两种中断：
 - ① 硬中断：是实现实时Linux内核的前提。
 - ② 软中断：是常规Linux内核中断。
- RT-Linux的系统结构：在Linux内核和硬件中断之间增加了一个**实时内核**（RT-Linux内核），该实时内核是一个高效的、可抢占的实时调度核心。

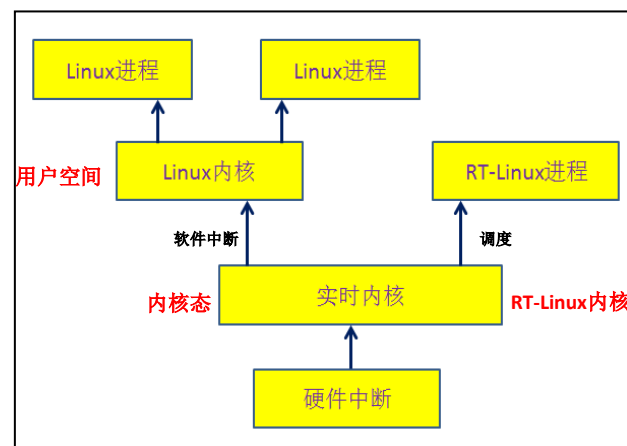


RT-Linux的系统结构

- RT-Linux的**调度策略**：默认的情况下采用优先级的调度策略，也支持其他的调度策略（如RM、EDF）。
 - RM（Rate Monotonic，速率单调调度）：任务周期确定优先级。
 - EDF（Earliest Deadline First，最早期限调度）：截止期限确定优先级。
- **RT-FIFO队列**：为保证RT-Linux实时进程与非实时Linux进程进行数据交换，RT-Linux引入了RT-FIFO队列。
 - FIFO（First In First Out）：先进先出。
- RT-Linux程序运行于**用户空间**和**内核态**两个空间：
 - ① 实时处理部分编写成内核模块，并装载到RT-Linux内核中，运行于RT-Linux的**内核态**。
 - ② 非实时部分的应用程序则在Linux下的**用户空间**中执行。



带有RT-FIFO的RT-Linux系统运行图



用户空间和内核态

• 3.3.3 标准Linux的进程管理

– 1、Linux进程

- 进程的**属性**:
 - ① 进程标识符（**PID**）。
 - ② 父进程和父进程的ID（**PPID**）。
 - ③ 启动进程的用户ID（**UID**）和所归属的组（**GID**）。
 - ④ 进程状态：运行**R**、休眠**S**、僵尸**Z**。
 - ⑤ 进程执行的优先级。
 - ⑥ 进程所连接的终端名。
 - ⑦ 进程资源占用：如占用资源大小（内存、**CPU**占用量等）。
- **父进程**：已创建一个或多个子进程的进程。
- **子进程**：子进程是父进程的副本，它将获得父进程数据空间、堆、栈等资源的副本。
- **fork函数**（分叉函数）：一个现有进程可以调用**fork**函数创建一个新进程。
- **vfork函数**：**vfork**函数和**fork**函数一样都是在已有的进程中创建一个新的进程，但它们创建的子进程是有区别的。
- **inode**：索引节点（**index node**）。

– 2、Linux进程的调度

- Linux操作系统**进程调度策略**（三种）：
 - ① 分时调度策略： **SCHED_OTHER**
 - ② 先到先服务的实时调度策略： **SCHED_FIFO**
 - ③ 时间片轮转的实时调度策略： **SCHED_RR**
- **实时进程**：将得到优先调用，并根据实时优先级决定调度权限。
- **分时进程**：通过**nice**和**counter**值决定权值，**nice**越小、**counter**越大，被调用的概率越大，即曾经使用**CPU**最少的进程将会得到优先调度。

• 3.3.4 μ CLinux的进程管理

– μ CLinux: 用于微控制器（单片机）的Linux操作系统。

– μ CLinux进程管理的特点:

- ① 由于没有MMU，在实现多个进程时需要实现**数据保护**。
- ② 由于没有MMU，系统虽然支持fork系统调用，但实际上是**vfork**系统调用。
- ③ 启动新的应用程序时，系统必须为应用程序**分配存储空间**，并立即将应用程序加载到内存。
- ④ 必须在可执行文件加载阶段对可执行文件**Relocation**（重定位）处理，使得程序执行时能够**直接使用物理内存**。
- ⑤ 操作系统对内存空间没有保护，各个进程实际上共享一个运行空间，会导致用户程序使用的空间可能占用到系统内核空间，容易导致**系统崩溃**。

3.4 文件系统

• 3.4.1 文件系统定义

- 包含在磁盘驱动器或者磁盘分区的目录结构，整个磁盘空间可以给一个或者多个文件系统使用。
- 在对某个文件系统做挂载（**mount**）操作后，就可以使用该文件系统了。

```
mount -t nfs 192.168.33.129:/imx6 /mnt
```

将虚拟机（Ubuntu，IP地址为192.168.33.129）的/imx6目录，挂载到实验箱的/mnt目录，这样访问实验箱的/mnt目录相当于访问虚拟机的/imx6目录

• 3.4.2 Linux文件系统

– 1、常见的Linux文件系统

- ① **ext** (Extended File System) : 扩展文件系统。
- ② **ext2** (Extended File System 2) : 扩展文件系统 (版本2) 。
- ③ **ext3** (Extended File System 3) : 扩展文件系统 (版本3) 。
- ④ **JFS2** (Journal File System 2) : 日志文件系统 (版本2) 。
- ⑤ **XFS**: 一种高性能的日志文件系统, 是 **Silicon Graphics**公司于 90 年代初开发的文件系统。
- ⑥ **ReiserFS**: 一种新型的文件系统, 它支持海量磁盘和磁盘阵列, 并能在上面继续保持很快的搜索速度和很高的效率。
- ⑦ **CFS**: 加密文件系统。
- ⑧ **VFS**: 虚拟文件系统。

– 2、**ext2**文件系统（扩展文件系统版本2）

- 是Linux事实上的**标准文件系统**。
- **ext2**的优点：
 - ① 支持达4TB的内存
 - ② 文件名称最长可达1012个字符
 - ③ 可以选择逻辑块的大小（1024、2048、4096字节）
 - ④ 实现快速符号链接
 - ⑤ 具有稳定性、可靠性和健壮性
 - ⑥ 是为像IDE设备那样的块设备设计的
 - ⑦ 没有提供对基于扇区的擦除/写操作的良好管理
 - ⑧ 在出现电源故障时，不是防崩溃的
 - ⑨ 不支持损耗平衡
 - ⑩ 没有特别完美的扇区管理

– 3、**ext3**文件系统（扩展文件系统版本3）

- 是一种**日志式文件系统**，它会将整个磁盘的写入动作完整记录在磁盘的某个区域上，以便需要时可以回溯追踪。
- **ext3**的优点：
 - ① 具有高可靠性
 - ② 具有很好的数据完整性
 - ③ 访问速度快
 - ④ 兼容性好
 - ⑤ 多种日志模式

• 3.4.3 嵌入式Linux文件系统

– 嵌入式系统很少使用IDE硬盘，而是选用Flash ROM（闪存）代替硬盘。

– 1、Flash ROM（闪存）简介

- 属于Non-Volatile内存（非挥发性内存，即断电信息不丢失）。

– 2、Flash ROM上的两种技术NOR和NAND

- ① **NOR Flash**: Intel于1988年首先开发出NOR Flash技术（采用或非门），彻底改变了原先由EPROM和EEPROM一统天下的局面。
- ② **NAND Flash**: 1989年，东芝公司发表了NAND Flash结构（采用与非门），强调降低每比特的成本，更高的性能，并且象磁盘一样可以通过接口轻松升级。

NAND Flash和NOR Flash的比较

1989年东芝公司

1988年Intel公司

	NAND Flash	NOR Flash
芯片容量	<32GBit	<1GBit
访问方式	顺序读写	随机读写
接口方式	任意I/O口	特定完整存储器接口
读写性能	读取快（顺序读） 写入快 擦除快（可按块擦除）	读取快（RAM方式） 写入慢 写入慢
使用寿命	百万次	十万次
价格	低廉	高昂

容量大
价格低
顺序读写

容量小
价格高
随机读写

– 3、嵌入式文件系统分类

- 嵌入式系统主要的存储设备为：

- ① **Flash ROM**：闪存（IMX6实验箱采用8GB NAND的Flash ROM）

- ② **SDRAM**：Synchronous Dynamic Random Access Memory，同步动态RAM（IMX6实验箱采用4片256MB的DDR 3 SDRAM，共1GB）

- 嵌入式系统主要的文件系统类型：

- ① **JFFS2**

- ② **YAFFS2**

- ③ **Cramfs**

- ④ **Romfs**

- ⑤ **RamDisk**

- ⑥ **Ramfs/Tmpfs**

– 4、基于Flash ROM的文件系统（NAND Flash）

① JFFS2文件系统

- **Jouralling Flash File System 2**（闪存日志型文件系统第2版）
- 主要用于NOR Flash

② YAFFS2文件系统

- **Yet Another Flash File System 2**，是一种和JFFS类似的闪存文件系统
- 主要用于NAND Flash

③ Cramfs文件系统

- **Compressed ROM File System**，一种只读的压缩文件系统

④ Romfs文件系统

- **ROM File System**，一种简单的只读文件系统

⑤ 其他文件系统

- **FAT/FTA32**：主要是为了与Windows系统兼容
- **ext2**：是GNU/Linux系统中的标准文件系统，特点是存取文件的性能好，具有存取文件的性能极好，但是用于Flash上会有很多弊端

– 5、基于RAM的文件系统（SDRAM）

① RamDisk

- **RamDisk**（虚拟内存盘），是通过软件将一部分内存(**RAM**)模拟为硬盘来使用的一种技术。相对于直接的硬盘文件访问来说，这种技术可以极大的提高在其上进行的文件访问的速度。但是**RAM**的易失性也意味着当关闭电源后这部分数据将会丢失。但是在一般情况下，传递到**RAM**盘上的数据都是在硬盘或别处永久贮存的文件的一个拷贝。经由适当的配置，可以实现当系统重启后重新建立虚拟盘。

② Ramfs/Tmpfs

- **Ramfs**（基于**RAM**的文件系统）是**Linus Torvalds**开发的一种基于内存的文件系统，工作于虚拟文件系统(**VFS**)层，不能格式化，可以创建多个，在创建时可以指定其最大能使用的内存大小。
- **Tmpfs**（临时文件系统），是一种基于内存的文件系统，它和虚拟磁盘**RamDisk**比较类似像，但不完全相同，和**RamDisk**一样，**Tmpfs**可以使用**RAM**，但它也可以使用**swap**分区来存储，而且传统的**RamDisk**是个块设备，要用**mkfs**来格式化它，才能真正地使用它；而**Tmpfs**是一个文件系统，并不是块设备，只是安装它，就可以使用了。**Tmpfs**是最好的基于**RAM**的文件系统。

– 6、网络文件系统（NFS）

- **NFS: Network File System**（网络文件系统）
 - 是FreeBSD支持的文件系统中的一种，它允许网络中的计算机之间通过TCP/IP网络共享资源。
 - 在NFS的应用中，本地NFS的客户端应用可以透明地读写位于远端NFS服务器上的文件，就像访问本地文件一样。
- **NFS至少包括两个主要部分：**
 - ① 1台服务器
 - ② 1台（或多台）客户机

```
mount -t nfs 192.168.33.129:/imx6 /mnt
```

小结

- 介绍了常见的**嵌入式Linux操作系统**。
- 针对是否支持**MMU**，介绍了嵌入式Linux的不同处理方法（**内存管理**）。
- 针对不同的进程调度模式，分别介绍了实时操作系统和分时操作系统的进程管理（**进程管理**）。
- 在标准Linux文件系统的基础上，介绍了嵌入式Linux文件系统的种类和特点（**文件系统**）。

进一步探索

- 进一步了解各种嵌入式Linux操作系统的特点，以及它们的应用领域以及典型设备。

Thanks