

# Chap1 Linux系统概述

- Linux系统特点
- Linux系统组成
- 什么是POSIX标准？为什么现代操作系统的设计必须遵循POSIX标准
- 什么是GNU？Linux与GNU有什么关系
- Linux内核由哪几个子系统组成？各子系统的主要功能是什么？能够写出各子系统之间的关联
- 能够认识Linux内核版本

## 1. Linux的起源与发展

- 操作系统原型/内核：**进程切换+文件系统+设备驱动程序**
- Linux之父：Linus Torvalds

## 2. 自由软件与GPL

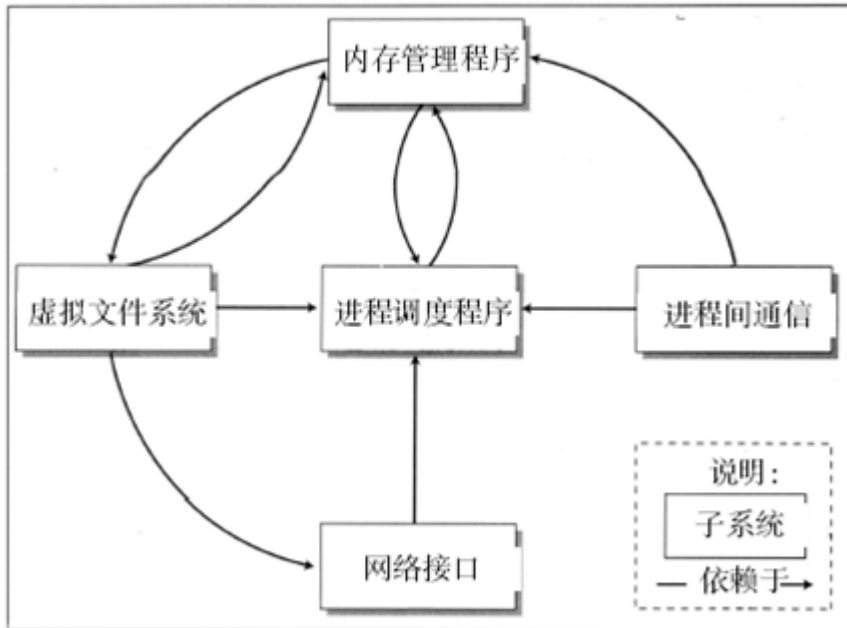
- 自由软件：允许自由地**使用、复制、修改、分发**的软件，软件的**源代码必须是可得到的**。
- **POSIX**：Portable Operating System Interface Standard
  - 基于Unix的可移植操作系统
  - 一套标准的操作系统接口和工具
- GNU：GNU's Not Unix
  - 由自由软件基金FSF赞助的项目
  - 开发一个完全自由的，UNIX类型的操作系统，成为GNU
- GPL：General Public License
  - 保证自由软件对所有用户都是自由的
  - 所有GNU软件和派生工作都遵循GPL
  - GPL类软件都**遵循一定规则**：
    - 传播者不能限制购买软件的用户自由权。如果用户买了一套GPL软件，就可免费复制、传播或出售。
    - 传播者必须清楚告诉用户该软件属于GPL软件。
    - 传播者必须免费提供软件的完整源代码，且任何用户能以源代码形式将软件复制或发布给别的用户。
    - 如果用户的软件使用了受GPL保护的任何软件的一部分，则该软件也成为GPL软件，它必须随应用程序一起发布源代码。
- GNU与Linux：
  - **Linux内核的开发工作都是基于GNU推出的自由软件完成的**。Linux与GNU相辅相成
    - Linux：内核
    - GNU/Linux:操作系统名称
      - Linux中的窗口管理工具，编译器，shell等都是GNU软件。

### 3. Linux系统的主要特点

- Linux是一套**免费使用**，遵循**GPL、POSIX标准的、与UNIX兼容**的操作系统
- 具有**良好的可移植性**
- 可运行于**多种平台**(是目前**支持硬件最多**的操作系统)
- 具有**现代操作系统**的所有内容：
  - 抢占式多任务处理，多用户
  - 内存保护
  - SMP, TCP/IP
  - 支持绝大多数的32位/64位CPU
- NFS, VFS, EXT文件系统
- 成熟的企业应用、服务器级操作系统、嵌入式领域
- 设备独立性、丰富的网络功能、性能高和安全性强、便于定制和再开发
- 杰出的学习系统

### 4. Linux系统及内核的组成

- **Linux系统组成**：
  - 用户进程、系统调用接口(OS服务层)、Linux内核、硬件层
- **Linux内核组成**：
  - 进程调度程序
    - 负责控制进程访问CPU。确保所有的进程都能公平访问CPU。
  - 内存管理程序
    - 使多个进程可以安全地共享机器的主存系统，并支持虚拟内存。
  - 虚拟文件系统
    - 提供一个所有设备的公共文件接口，抽象了不同硬件设备的细节
  - 网络接口
    - 提供对多种网络协议标准和网络设备的访问。
  - 进程间通信
    - 进程间通信支持进程之间的通信，Linux支持进程间的多种通信机制。这些机制可协助多个进程、多资源的互斥访问、进程间的同步和消息传递。
- **内核间关系**：
  - **进程调度与内存管理**：**互相依赖**。在多程序环境下，程序要运行，则必须为之创建进程，而创建进程的第一件事情，就是将程序和数据装入内存。
  - **进程间通信与内存管理**：**进程间通信要依赖内存管理**支持共享内存通信机制。这种机制允许两个进程除了拥有自己的私有空间之外，还可以存取共同的内存区域。
  - **虚拟文件系统与网络接口**：**虚拟文件系统利用网络接口**支持网络文件系统（NFS），也利用内存管理支持RAMDISK设备。
  - **内存管理与虚拟文件系统**：**内存管理利用虚拟文件系统**支持交换，交换进程定期由调度程序调度，这也是内存管理依赖于进程调度的原因。当一个进程存取的内存映射被换出时，内存管理向虚拟文件系统发出请求，同时，挂起当前正在运行的进程。
  - 除了这些依赖关系外，内核中的所有子系统还要依赖于一些共同的资源。这些资源包括所有子系统都用到的API，如分配和释放内存空间的函数、输出警告或错误消息的函数及系统提供的调试接口等。



- Linux**内核特征**:

1. 使用单一内核结构
  - 可以动态装入和卸载内核中的部分代码，称为模块
2. 进程调度方式简单有效
3. 动态加载内核模块
4. 支持对称多处理机制
5. Linux内核可以抢占
6. Linux对线程的支持

## 5. Linux版本

- 内核版：稳定版&测试版
- 发行版：内核+tools
- Linux内核采用双树系统：
  - **稳定树**：主要用于发行。修正Bug，加入新设备驱动
  - **开发树**：用于产品开发，变化快。
- 内核版本号：\$r.x.y\$
  - 主版本号.从版本号.修订版本号.(稳定本号)
  - “**从版本号**”：偶数为稳定版；奇数为测试版。

## 6. Linux内核源代码组织

- **源代码位置**：/usr/src/linux
- 组织方法：树形结构组织
  - arch:所有体系结构的相关的核心代码
    - 每个体系结构子目录下包含几个主要子目录：
    - kernel与体系结构相关的**核心代码**。
    - mm内存管理
    - lib库代码
  - include：用来编译的头文件
  - init：内核初始化代码
  - mm：**独立于CPU体系结构**的所有内存管理代码
  - modules：仅包含已建好的模块
  - fs：文件系统
  - kernel：核心模块。包括进程管理和调度的主要核心代码。中断，时钟，同步机制
  - net：网络部分
  - lib：核心库代码
  - script：用于编译时的脚本文件
  - drivers：所有设备的驱动
  - ipc：内核的进程间通讯
- 每个目录下的'.depend'和'makefile'文件在编译时使用

## 7. Linux与其他系统的区别

- MS-DOS与Linux
  - 单任务操作系统/多任务操作系统
- Windows与Linux
  - 封闭有偿/开放免费
  - Linux：软件丰富，稳定性好，硬件适应强，网络功能丰富。

## 8. Linux中的C语言与汇编语言

- 汇编语言文件'.s'
- 已预编译的汇编语言文件'.S'
- 编译后的目标文件'.o'
- Linux源代码中的C语言代码：
  - 使用GNU的C语言编写
  - 从C++中吸收了'inline'和'const'
  - 支持“属性描述符”(attribute)
  - 增加新数据类型'long long int'，用于支持64位CPU
- Linux源码中的**汇编语言**：'.s'文件与嵌入C中的
  - 汇编采用AT&T格式，非Intel格式

区别项	AT&T格式	Intel格式
指令大小写	多用小写,如: <code>movw %eax,%ebx</code>	多用大写, 如PUSH AX
寄存器名	寄存器名前加%号,如: <code>%eax,%edi</code>	寄存器名前不加任何符号, EAX, EDI
源/目的操作数顺序	操作指令 源数 目的数 <code>mov %eax,%ebx;eax→ebx</code>	操作指令 目的数 源数 MOV EAX,EBX;EBX→EAX
操作数长度数位表示	操作码最后一字母表示长度位数, 其中b(16位), w(32位) ,l(64位) <code>movb FOO,%al</code>	操作数前加 BYTE PTR, WORD PTR, DWORD PTR 限定位数 MOV AL,BYTE PTR FOO
立即数表示	立即数前加\$号, 如: <code>push \$2</code>	不加任何符号, 如 PUSH 2
远程转移/调用指令	指令前加字符l,如lcall,ljmp	指令后加FAR 如CALL FAR, JMP FAR

- 汇编嵌入C语言中
  - AT&T行内汇编: 关键字'`asm()`','`_asm()`'
  - GCC处理汇编时, 将asm括号中的汇编代码拷贝到汇编文件中。
- 扩展行内汇编
  - `:output_regs` //输出寄存器说明
  - `:input_reg` //输入寄存器说明
  - `:clobbered_regs` //被改变的寄存器