
Linux 操作系统概述

Linux 是一套可以免费使用和自由传播的，类似于 UNIX 风格的操作系统。

Linux 最早是由芬兰人托瓦兹（Linus Torvalds）设计的。

1. 体系结构概述

按照 Garlan 和 Shaw 提出的 Linux 操作系统分层方法：Linux 操作系统分为 4 层次，即四个子系统，分别是用户进程、系统调用接口、Linux 内核、硬件控制器。下面简单介绍一下这四个组成部分。

用户进程：用户应用程序是运行在 Linux 操作系统最高层的一个庞大的软件及核。当一个用户程序在操作系统之上运行时，它就是操作系统的一个进程。计算机不同，程序的集合大小会有所变化。

系统调用接口：为了在应用程序中实现特定的任务，可以通过系统调用来调用操作系统内核中特定的过程，以实现特定的服务。一般认为，这些调用和服务是操作系统的一部分，内核和编程接口也属于这一部分。系统调用本身也是由若干条指令组成的，但与一般过程不同的是：系统调用运行在内核模式，而一般的进程运行在用户模式。

Linux 内核：内核式操作系统的灵魂，包括内核抽象核对硬件资源（如 cpu）的间接访问，它负责管理磁盘上的文件、内存，负责启动系统并运行程序，负责从网络上接收和发送数据包等等

硬件：这个子系统包括了 Linux 安装时需要的所有可能的物理设备。

2.Linux 内核

从程序员的角度来讲，操作系统的内核提供了一个虚拟的机器接口。它抽象了许多硬件细节，程序可以以某种统一的方式来进行数据管理，而内核将所有的硬件抽象成统一的虚拟借口。

Linux 以统一的方式支持多任务，而这种方式对用户进程是透明的，每一个进程运行起来就好像只有它一个进程在计算机上运行一样，独占内存和其他的硬件资源。实际上内核在并发的运行几个进程。并且能够让几个进程公平合理地使用硬件资源，也能使各个进程之间互不干扰安全的运行。读到这里你就明白了为什么 linux 不会象 windows 那样常死机、蓝屏。linux 如果使用图形界面可能会发现计算机好像死机了，其实没有死，而是某个进程死了，可能就是你的 kde,gnome.杀死这个进程就可以了不必 reset.

Linux 内核也要完成一般操作系统必须完成的任务：

对文件系统的读写进行管理，把对文件系统的操作映射成对磁盘或者其他块设备的操作，Linux 系统把所有的设备对定义为文件了，哈哈，这可与 windows 不同。

管理程序的运行，为程序分配资源，并且处理程序之间的通讯。

管理存储器，为程序分配内存，并且管理虚拟内存

管理输入输出，将设备映射成文件。

管理网络：有“网络之子”之称的 Linux,对网络的管理可是她的强项。使用 Linux 你就会感受到它的网络功能的强大，可以使用 Linux 模拟强大的 Csico 高级路由器，那种感觉真实太棒了，只要几百块钱买一台 486 的计算机哦，就可

以在上面自己构造强大功能的路由器，是不是动心了，哈哈，不过软件实现的路由器性能肯定比不上硬件路由器。

内核必须包含虚拟文件系统 (VFS) 管理程序以及各种具体文件系统映射成 VFS 的程序。这可是 Linux 较有特设的一部分，这就是为什么 Linux 支持的文件系统 (minix 文件系统、ext2/ext3 文件系统、msdos/vfat/ntfs 文件系统、iso9600CD-ROM 的标准文件系统、hpfs OS/2 用的文件系统、ufs/sysv 文件系统) 很多的原因。对于内存的管理，Linux 使用虚拟存储管理方式，利用现代处理器的页面映射能力，在 x86 处理器上，Linux 使用 4GB 的地址空间，操作系统处理利用物理存储器外还支持将硬盘空间映射成虚拟内存。所有的存储器 (物理内存和虚拟内存) 被分成大小相等的页面，系统通过给出页号和页面内偏移量对某个内存地址进行访问。物理内存紧张的时候，操作系统把某些没有使用的页面从内存移动到硬盘上以便腾出空闲的页面供程序使用，这个过程叫做交换 (SWAP)。Linux 使用交换分区来处理交换需要的虚拟存储空间，在硬盘上开设一个独立的分区专门用于映射虚拟内存，交换分区可以有多个，之所以这样是因为早期的 Linux 核心要求每一个交换分区不能超过 128MB。对于较重负荷的服务器，交换内存用到 256MB 甚至更多都是很正常的事情，因此那时的系统经常有多个交换分区。目前这个限制已经去除。

内核的另外一个任务是执行用户程序，为此核心必须支持可执行格式。Linux 使用多种可执行文件格式，诸如 elf、aout 等等，这可与 windows 不同，没有办法从名字上区分一个文件到底是什么格式，核心只关心二进制文件的具体形式。

linux 内核由五个主要的子系统组成：进程调度(SCHED)、内存管理(MM)、虚拟文件系统(VFS)、网络接口(NET)、进程间通信 (IPC)。进程调度处于核心位

置，所有的子系统都依赖于它，因为每一个子系统都需要挂起或者恢复进程。一般情况下，当一个进程等待硬件操作完成时，它会被挂起；当操作真正完成时，进程恢复执行。各个子系统之间的依赖关系如下：

进程调度与内存管理之间的关系：这两个子系统互相依赖。在多道程序环境下，程序运行必须为之创建进程，而创建进程的第一件事就是要将程序和数据装入内存。

进程间通信与内存管理的关系：进程间通信子系统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内存外，还可存取共同的内存区域。

虚拟文件系统与网络接口之间的关系：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之间的关系：内存管理利用虚拟文件系统支持交换，交换进程定期地由调度程序调度。