



Linux
CAOZUO XITONG JIAOCHENG

Linux 操作系统教程

刘百峰 宋 翠 主编

Linux 操作系统教程

主 编：刘百峰 宋 翠

副主编：闫冬梅 耿仲华 王爱华

内容提要

本书以 RHEL5 为基础,结合高等院校教学特点,全面、系统、深入地介绍了 Linux 操作系统的基本使用和在网络服务方面的应用,为读者介绍了 RHEL5 的安装、Linux 图形桌面系统、字符界面与 Shell、vi 文本编辑器、用户与组群管理、文件系统管理、进程管理、GCC 编译器、GDB 调试器以及如何配置 Squid 代理服务器、DHCP 服务器、Samba 服务器、FTP 服务器、DNS 服务器、WWW 服务器等内容。本书既可以作为高等院校相关专业的 Linux 操作系统课程的教材,也可以作为 Linux 网络服务的培训教材,还可以作为构建网络服务技术人员的参考书。

版权专有 侵权必究

图书在版编目 (CIP) 数据

Linux 操作系统教程/刘百峰,宋翠主编. —北京:北京理工大学出版社,2016.1
ISBN 978-7-5640-7201-8

I. ①L… II. ①刘… ②宋… III. ①Linux 操作系统-高等学校-教材 IV. ①TP316.89
中国版本图书馆 CIP 数据核字 (2013) 第 002862 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)

(010) 82562903 (教材售后服务热线)

(010) 68948351 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 三河市华骏印务包装有限公司

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 17

字 数 / 400 千字

版 次 / 2016 年 1 月第 1 版 2016 年 1 月第 1 次印刷

定 价 / 48.00 元

责任编辑 / 高 芳

文案编辑 / 高 芳

责任校对 / 周瑞红

责任印制 / 李志强

图书出现印装质量问题,请拨打售后服务热线,本社负责调换



前言

Linux 操作系统是一种真正的多任务、多用户的操作系统，它诞生于 1991 年，是目前最为优秀的操作系统之一。

Linux 是采用 GPL 协议发布的自由软件，用户可以通过网络或其他途径免费获得它的源代码。正是由于 Linux 坚持开放源代码的策略，才使全球优秀的开发人员能够在 Linux 内核基础上加以改良，从而让 Linux 不断成熟。与 Unix 一样，Linux 是多任务、多用户的操作系统，各个用户对自己的资源有特定权限，用户可以同时运行多个应用程序；Linux 采用了多种安全策略，如对读写进行权限控制、带保护的子系统、审计跟踪、核心授权等，这些安全策略技术为多用户环境中的用户提供了安全保障；Linux 是以 UNIX 为设计原型的，是一个类 UNIX 的操作系统，它继承了 UNIX 的高稳定性特点，可以随时终止出故障的进程，而不会导致整个系统崩溃，这使得平稳运行数月或数年的 Linux 操作系统随处可见。Linux 在开发过程中参考了 POSIX 标准，完全兼容 POSIX 标准，还增加了部分 System V 和 BSD 系统的接口，使其能直接运行一些 UNIX 上的应用程序。另外，Linux 在对工业标准的支持上也做得非常好，得到众多 Linux 发行厂商的支持。Linux 具有良好的可移植性，支持在多种平台上运行，从大型计算机到 PC，都能运行。同时，Linux 还是一种优秀的嵌入式操作系统，可以运行于掌上电脑、手机、机顶盒等。

Linux 操作系统是一个成长于网络、成熟于网络的操作系统，它最重要的特点是其网络功能非常强大，许多网络功能都内置于内核模块之中，提高了各种网络服务的运行速度，所以 Linux 操作系统在网络中的应用非常广泛，几乎成为搭建各种网络服务的首选操作系统。使用 Linux 可以搭建的网络服务有 WWW 服务、DHCP 服务、DNS 服务、FTP 服务、网络代理服务等。此外，利用 Linux 操作系统的防火墙技术可以实现路由、NAT 转换等网络功能。

本书以 RHEL5 为背景，紧密围绕 Linux 的网络服务功能进行编写。全书共 15 章，第 1 章介绍操作系统的基本知识，Linux 的发展历史、特点；第 2 章介绍 RHEL5 的安装；第 3 章介绍 Linux 的图形桌面系统；第 4 章介绍字符界面和 vi 文本编辑器的使用；第 5 章介绍 Linux 如何对用户与组群进行管理；第 6 章介绍 Linux 的文件系统、磁盘管理、包管理等功能；第 7 章介绍 Linux 的进程管理、调度与系统性能监视；第 8 章阐述了网卡的配置、虚拟网卡的配置、网卡设备别名的配置；第 9 章介绍如何配置 Squid 代理服务器；第 10 章介绍如何搭建 DHCP 服务器；第 11 章介绍如何配置 Samba 服务；第 12 章介绍如何配置 FTP 服务；第 13 章介绍如何构建 DNS 服务器；第 14 章介绍如何构建各种类型的 WWW 服务器；第 15 章介绍 Linux 操作系统的 GCC 编译器、GDB 调试器、make 项目管理工具。

本书由刘百峰、宋翠担任主编，由闫冬梅、耿仲华、王爱华担任副主编。具体编写分工如下：第1章由耿仲华、王爱华老师编写，第5章由闫冬梅老师编写，第15章由宋翠老师编写，其余章节由刘百峰老师编写，全书由刘百峰、宋翠老师统稿。

由于编者水平有限，书中错误和不妥之处在所难免，恳请广大读者批评指正。

编 者

目 录

第 1 章 Linux 操作系统概述	001
1.1 操作系统概论	001
1.1.1 操作系统的发展	001
1.1.2 操作系统的基本特性	002
1.1.3 操作系统的主要功能	003
1.2 Linux 操作系统	006
1.2.1 Linux 操作系统的发展历史	006
1.2.2 自由软件	008
1.2.3 Linux 操作系统的版本	009
1.2.4 Linux 操作系统的特点	011
1.2.5 Linux 操作系统的组成	012
第 2 章 Linux 操作系统的安装	014
2.1 Red Hat 产品与安装方式	014
2.1.1 Red Hat 产品简介	014
2.1.2 Linux 操作系统的安装方式	015
2.1.3 Red Hat Enterprise Linux 5 server 安装的硬件要求	016
2.2 Linux 操作系统的磁盘分区	016
2.3 RHEL5 的安装	018
第 3 章 Linux 的桌面系统	040
3.1 Linux 桌面环境概述	040
3.2 GNOME 的桌面与面板	040
3.3 文件浏览器的使用	048

3.4	使用移动存储介质	053
3.5	设置面板	054
3.6	设置桌面	056
3.7	设置打印机	062
3.8	使用用户配置文件	065
3.9	KDE 桌面系统	067

第4章 Linux 的字符界面 071

4.1	字符界面简介	071
4.2	Shell 概述	071
4.3	Linux 的运行级别	073
4.3.1	字符界面的登录	073
4.3.2	图形系统到字符界面的切换	073
4.3.3	启动系统时进入字符界面	074
4.3.4	在字符界面下启动图形系统	076
4.3.5	字符界面下的注销、关机、重启	076
4.4	Shell 常用命令	076
4.5	vi 文本编辑器	083

第5章 Linux 用户与组群管理 088

5.1	用户与组群概述	088
5.2	图形方式的用户与组群管理	089
5.3	字符界面的用户与组群管理	093
5.4	用户与组群的配置文件	095
5.5	批量建立用户	096
5.6	用户与组群管理实例	099

第6章 Linux 文件系统与管理 102

6.1	文件系统与挂载	102
6.1.1	Linux 常用的文件系统	102
6.1.2	Linux 支持的文件系统	104
6.1.3	Linux 文件系统挂载	105
6.1.4	挂载 U 盘	106

6.2	磁盘管理	111
6.2.1	磁盘分区命令 fdisk	111
6.2.2	分区格式化 mkfs 命令	112
6.2.3	检查修复文件系统 fsck 命令	113
6.3	Linux 文件系统	113
6.4	文件权限的设置	116
6.5	文件的打包与压缩	119
6.6	RPM 软件包的管理	122

第 7 章 进程与系统管理

7.1	进程基础知识	125
7.2	字符界面下进程的管理	126
7.3	图形方式下进程的管理	132
7.3.1	进程管理	132
7.3.2	系统性能监视	134
7.4	进程调度	136
7.4.1	at 调度	136
7.4.2	cron 调度	138

第 8 章 网络配置基础

8.1	网络基础知识	141
8.1.1	网络基本概念	141
8.1.2	Linux 的网络接口与端口	142
8.2	物理网卡的配置	143
8.2.1	图形方式下配置网卡	143
8.2.2	配置文件与相关命令	145
8.2.3	字符界面下配置网卡	149
8.3	虚拟网卡的配置	150
8.3.1	图形方式下设置虚拟网卡	150
8.3.2	字符界面下设置虚拟网卡	152
8.4	配置网卡的设备别名	153
8.4.1	图形方式下设置网卡的设备别名	153
8.4.2	字符界面下设置网卡的设备别名	158

第 9 章 配置 Squid 代理服务	160
• 9.1 Squid 代理服务器概述	160
9.2 Squid 代理服务器配置	160
9.3 配置客户端的 Internet 浏览器	164
9.3.1 配置 Windows 系统的客户端	164
9.3.2 配置 Linux 系统的客户端	165
第 10 章 配置 DHCP 服务	168
• 10.1 DHCP 概述	168
10.2 DHCP 工作过程	169
10.3 DHCP 配置文件	169
10.4 DHCP 服务器配置实例	172
10.4.1 DHCP 服务器的配置	172
10.4.2 DHCP 客户端的配置	173
第 11 章 配置 Samba 服务	177
• 11.1 Samba 服务器的安装	177
11.2 Samba 服务器的配置基础	178
11.3 构建 Samba 服务器实例	182
11.3.1 构建 Samba 服务器实例一	182
11.3.2 构建 Samba 服务器实例二	183
11.3.3 构建 Samba 服务器实例三	184
11.4 建立 Samba 用户	185
11.5 Linux 字符界面访问 Samba 服务器	187
11.6 图形方式下配置 Samba 服务器	188
11.7 设置防火墙与 SELinux	193
11.8 Windows 客户端访问 Samba 服务器	196
第 12 章 配置 FTP 服务	198
• 12.1 FTP 服务器概述	198
12.2 vsftpd 服务器	199
12.2.1 vsftpd 服务器的用户	199

12.2.2	vsftpd 配置文件	199
12.3	vsftpd 服务器的配置	201
12.3.1	vsftpd 服务器的安装	201
12.3.2	配置 vsftpd 服务器实例一	202
12.3.3	配置 vsftpd 服务器实例二	205
12.3.4	配置 vsftpd 服务器实例三	208
12.3.5	配置 vsftpd 服务器其他功能	209

第 13 章 配置 DNS 服务

13.1	DNS 服务器基础	210
13.1.1	域名服务器简介	210
13.1.2	DNS 相关的配置文件	211
13.2	DNS 软件包的安装	214
13.3	配置主域名服务器	215
13.4	启动域名服务器	217
13.5	检测主域名服务器的配置	218
13.5.1	Linux 环境下检测域名服务器	218
13.5.2	Windows 环境下检测域名服务器	219

第 14 章 配置 WWW 服务

14.1	WWW 服务配置基础	221
14.1.1	安装 Apache 软件包	221
14.1.2	Apache 服务器的配置文件	222
14.1.3	Apache 服务器的启动	223
14.2	利用 Apache 默认配置构建 Web 站点	223
14.3	构建基于本地用户的 Web 站点	225
14.4	配置虚拟主机	227
14.4.1	配置基于 IP 地址的虚拟主机	228
14.4.2	配置基于域名的虚拟主机	230
14.5	Apache 的授权与认证	232
14.5.1	Apache 基于主机的授权与认证	233
14.5.2	Apache 基于用户的授权与认证	234

第 15 章 Linux C 编程基础	236
• 15.1 Linux 编程基础	236
15.2 GCC 编译器	236
15.2.1 GCC 简介	236
15.2.2 GCC 命令格式	237
15.2.3 GCC 的库函数处理	238
15.2.4 GCC 编译过程	238
15.3 GNU make	244
15.3.1 GNU make 简介	244
15.3.2 makefile 文件	245
15.3.3 make 运用实例	246
15.4 调试器 GDB	249
15.4.1 GDB 基础	249
15.4.2 GDB 调试程序过程	250
15.4.3 GDB 应用实例	250
15.5 GDB 命令	256

Linux 操作系统概述



1.1 操作系统概论

1.1.1 操作系统的发展

1. 人工操作方式

从第一台计算机诞生到 20 世纪 50 年代中期,属于第一代计算机,这时还未出现 OS (操作系统)。这时的计算机操作是由用户采用人工操作方式直接使用计算机硬件系统,即由程序员将事先已穿孔的纸带装入纸带输入机,再启动它们,将程序和数据输入计算机,然后启动计算机运行。当程序运行完毕并取走计算结果后,才让下一个用户上机。这种人工操作方式有以下两方面的缺点:用户独占全机、CPU 等待人工操作。

2. 脱机输入/输出方式

脱机 I/O 方式的主要优点:减少了 CPU 的空闲时间;提高 I/O 速度。

3. 单道批处理系统

单道批处理系统是最早出现的一种 OS,严格地说,它只能算是 OS 的前身而并非是现在人们所理解的 OS。尽管如此,该系统比起人工操作方式的系统已有很大进步。该系统的主要特征:自动性、顺序性、单道性。

4. 多道批处理系统

在单道批处理系统中,内存中仅有一道作业,它无法充分利用系统中的所有资源,致使系统性能较差。为了进一步提高资源的利用率和系统吞吐量,在 20 世纪 60 年代中期又引入了多道程序设计技术,由此而形成了多道批处理系统。在该系统中,用户所提交的作业都先存放在外存上并排成一个队列,称为“后备队列”;然后,由作业调度程序按一定的算法从后备队列中选择若干个作业调入内存,使它们共享 CPU 和系统中的各种资源,多道程序的特征是多道性、无序性、调度性。多道程序设计有如下的优点。

1) 提高 CPU 的利用率

当内存中仅有一道程序时,如果该程序在运行中发出 I/O 请求后,CPU 空闲,必须在其 I/O 完成后才继续运行;尤其因 I/O 设备的低速性,更使 CPU 的利用率显著降低。引入多道程序设计技术后,由于同时在内存中装有若干道程序,并使它们交替地运行,这样,当正在运行的程序因 I/O 而暂停执行时,系统可调度另一道程序运行,从而保持了 CPU 处于忙碌状态。

2) 提高内存和 I/O 设备的利用率

为了能运行较大的作业，通常内存都具有较大容量，但由于 80% 以上的作业都属于中小型，因此在单道程序环境下，也必定造成内存的浪费。类似地，对于系统中所配置的多种类型的 I/O 设备，在单道程序环境下也不能充分利用。如果允许在内存中装入多道程序，并允许它们并发执行，则无疑会大大提高内存和 I/O 设备的利用率。

3) 增加系统的吞吐量

在保持 CPU、I/O 设备不断忙碌的同时，也必然会大幅度地提高系统的吞吐量。

5. 分时系统

多道批处理系统的形成和发展提高了资源利用率和系统吞吐量，但交互性能比较差。分时系统使用户能与自己的作业进行交互，即当用户在自己的终端上输入命令时，系统应能及时接收并及时处理该命令，再将结果返回给用户。此后，用户可继续输入下一条命令，此即人一机交互。应强调指出，即使有多个用户同时通过自己的键盘输入命令，系统也应能全部及时地接收并处理。分时系统的特征是多路性、独立性、及时性、交互性。

6. 实时系统

实时系统 (Real-Time System) 是指系统能及时、即时响应外部事件的请求，在规定的时间内完成对该事件的处理，并控制所有实时任务协调一致地运行。它的主要特征是多路性、独立性、及时性、交互性、可靠性。

1.1.2 操作系统的基本特性

1. 并发

并行性和并发性是既相似又有区别的两个概念。并行性是指两个或多个事件在同一时刻发生；而并发性是指两个或多个事件在同一时间间隔内发生。在多道程序环境下，并发性是指在一段时间内，宏观上有多个程序在同时运行，但在单处理机系统中，每一时刻却仅能有一道程序执行，故微观上这些程序只能是分时地交替执行。倘若在计算机系统中有多个处理机，则这些可以并发执行的程序便可被分配到多个处理机上实现并行执行，即利用每个处理机来处理一个可并发执行的程序，这样，多个程序便可同时执行。

2. 共享

在操作系统环境下，所谓共享是指系统中的资源可供内存中多个并发执行的进程（线程）共同使用。由于资源属性的不同，进程对资源共享的方式也不同，目前主要有以下两种资源共享方式：互斥共享方式、同时访问方式。

3. 虚拟

操作系统中的所谓“虚拟”，是指通过某种技术把一个物理实体变为若干个逻辑上的对应物。物理实体是实的，即实际存在的；而后者是虚的，是用户感觉上的东西。相应地，用于实现虚拟的技术，称为虚拟技术。在 OS 中利用了多种虚拟技术，分别用来实现虚拟处理机、虚拟内存、虚拟外部设备和虚拟信道等。

(1) 虚拟处理机技术：通过多道程序设计技术让多道程序并发执行的方法，来分时使用一台处理机。此时，虽然只有一台处理机，但它能同时为多个用户服务，使每个终

端用户都认为是有一个 CPU 在专门为它服务。亦即，利用多道程序设计技术，把一台物理上的 CPU 虚拟为多台逻辑上的 CPU，也称为虚拟处理机。用户所感觉到的 CPU 称为虚拟处理器。

(2) 虚拟存储器技术：将一台机器的物理存储器变为虚拟存储器，以便从逻辑上来扩充存储器的容量。此时，虽然物理内存的容量可能不大，但它可以运行比它大得多的用户程序。这使用户所感觉到的内存容量比实际内存容量大得多，当然这时用户所感觉到的内存容量是虚的。用户所感觉到的存储器称为虚拟存储器。

(3) 虚拟设备技术：将一台物理 I/O 设备虚拟为多台逻辑上的 I/O 设备，并允许每个用户占用一台逻辑上的 I/O 设备，这样便可使原来仅允许在一段时间内由一个用户访问的设备，变为在一段时间内允许多个用户同时访问的共享设备。例如，原来的打印机属于临界资源，而通过虚拟设备技术，可以把它变为多台逻辑上的打印机，供多个用户“同时”打印。此外，也可以把一条物理信道虚拟为多条逻辑信道。

4. 异步性

在多道程序环境下，允许多个进程并发执行，但只有进程在获得所需的资源后方能执行。在单处理机环境下，由于系统中只有一个处理机，因而每次只允许一个进程执行，其余进程只能等待。当正在执行的进程提出某种资源要求时，如打印请求，而此时打印机正在为其他某进程打印，由于打印机属于临界资源，因此正在执行的进程必须等待，且放弃处理机，直到打印机空闲，并再次把处理机分配给该进程时，该进程方能继续执行。可见，由于资源等因素的限制，使进程的执行通常都不是一气呵成，而是以停停走走的方式运行。内存中的每个进程在何时能获得处理机运行，何时又因提出某种资源请求而暂停，以及进程以怎样的速度向前推进，每道程序总共需多少时间才能完成等，都是不可预知的。由于各用户程序性能的不同，比如，有的侧重于计算而较少需要 I/O，而有的程序其计算少而 I/O 多，这样，很可能是先进入内存的作业后完成，而后进入内存的作业先完成。或者说，进程是以人们不可预知的速度向前推进的，此即进程的异步性。尽管如此，但只要运行环境相同，作业经多次运行，都会获得完全相同的结果。因此，异步运行方式是允许的，是操作系统的一个重要特征。

1.1.3 操作系统的主要功能

1. 处理机管理功能

1) 进程控制

在传统的多道程序环境下，要使作业运行，必须先为它创建一个或几个进程，并为之分配必要的资源。当进程运行结束时，立即撤销该进程，以便能及时回收该进程所占用的各类资源。进程控制的主要功能是为作业创建进程、撤销已结束的进程以及控制进程在运行过程中的状态转换。在现代 OS 中，进程控制还应具有为一个进程创建若干个线程的功能和撤销（终止）已完成任务的线程的功能。

2) 进程同步

为使多个进程能有条不紊地运行，系统中必须设置进程同步机制。进程同步的主要任务是对多个进程（含线程）的运行进行协调。有以下两种协调方式。

(1) 进程互斥方式：指诸进程在对临界资源进行访问时，应采用互斥方式；

(2) 进程同步方式：指在相互合作去完成共同任务的诸进程间，由同步机构对它们的执行次序加以协调。为了实现进程同步，系统中必须设置进程同步机制。

3) 进程通信

在多道程序环境下，为了加速应用程序的运行，应在系统中建立多个进程，并且再为一个进程建立若干个线程，由这些进程相互合作去完成一个共同的任务。而在这些进程之间，又往往需要交换信息。例如，有三个相互合作的进程，它们是输入进程、计算进程和打印进程。输入进程负责将所输入的数据传送给计算进程；计算进程利用输入数据进行计算，并把计算结果传送给打印进程；最后，由打印进程把计算结果打印出来。进程通信的任务就是用来实现在相互合作的进程之间的信息交换。

4) 调度

在后备队列上等待的每个作业，通常都要经过调度才能执行。在传统的操作系统中，包括作业调度和进程调度两步。作业调度的基本任务是：从后备队列中按照一定的算法，选择出若干个作业，为它们分配其必需的资源，分别为它们建立进程，使它们都成为可能获得处理机的就绪进程，并按照一定的算法将它们插入到就绪队列。而进程调度的任务则是：从进程的就绪队列中选出一新进程，把处理机分配给它，并为它设置运行现场，使进程投入执行。

2. 存储器管理功能

1) 内存分配

OS 在实现内存分配时，可采取静态和动态两种方式。在静态分配方式中，每个作业的内存空间是在作业装入时确定的；在作业装入后的整个运行期间，不允许该作业再申请新的内存空间，也不允许作业在内存中移动；在动态分配方式中，每个作业所要求的基本内存空间也是在装入时确定的，但允许作业在运行过程中，继续申请新的附加内存空间，以适应程序和数据的动态增长，也允许作业在内存中移动。

2) 内存保护

内存保护的主要任务是确保每道用户程序都只在自己的内存空间内运行，彼此互不干扰。为了确保每道程序都只在自己的内存区中运行，必须设置内存保护机制。一种比较简单的内存保护机制是：设置两个界限寄存器，分别用于存放正在执行程序的上界和下界。系统须对每条指令所要访问的地址进行检查，如果发生越界，便发出越界中断请求，以停止该程序的执行。如果这种检查完全用软件实现，则每执行一条指令，便须增加若干条指令去进行越界检查，这将明显降低程序的运行速度。因此，越界检查都由硬件实现。

3) 地址映射

一个应用程序经编译后，通常会形成若干个目标程序；这些目标程序再经过链接便形成了可装入程序。这些程序的地址都是从 0 开始的，程序中的其他地址都是相对于起始地址计算的；其中的地址称为逻辑地址。此外，由内存中的一系列单元所限定的地址范围称为物理地址。在多道程序环境下，每道程序不可能都从 0 地址开始装入，这就致使逻辑地址和内存空间中的物理地址不一致。为了使程序能正确运行，存储器管理必须提供地址映射功能，将逻辑地址转换成物理地址。该功能应在硬件的支持下完成。

4) 内存扩充

存储器管理中的内存扩充任务，并非是去扩大物理内存的容量，而是借助于虚拟存储技术，从逻辑上去扩充内存容量，使用户所感觉到的内存容量比实际内存容量大得多；或者是让更多的用户程序能并发运行。这样，既满足了用户的需要，改善了系统的性能，又基本上不增加硬件投资。为了能在逻辑上扩充内存，系统必须具有内存扩充机制，用于实现请求调入功能、置换功能等。

3. 设备管理功能

设备管理用于管理计算机系统中所有的外围设备，而设备管理的主要任务是：完成用户进程提出的 I/O 请求；为用户进程分配其所需的 I/O 设备；提高 CPU 和 I/O 设备的利用率；提高 I/O 速度；方便用户使用 I/O 设备。为实现上述任务，设备管理应具有缓冲管理、设备分配和设备处理以及虚拟设备等功能。

1) 缓冲管理

CPU 运行的高速性和 I/O 低速性间的矛盾自计算机诞生时起便已存在。而随着 CPU 速度迅速、大幅度地提高，此矛盾更为突出，严重降低了 CPU 的利用率。如果在 I/O 设备和 CPU 之间引入缓冲，则可有效地缓和 CPU 和 I/O 设备速度不匹配的矛盾，提高 CPU 的利用率，进而提高系统的吞吐量。

2) 设备分配

设备分配的基本任务是：根据用户进程的 I/O 请求、系统的现有资源情况以及按照某种设备分配策略，为之分配其所需的设备。如果在 I/O 设备和 CPU 之间，还存在着设备控制器和 I/O 通道时，还须为分配出去的设备分配相应的控制器和通道。

3) 设备处理

设备处理程序又称为设备驱动程序，其基本任务是用于实现 CPU 和设备控制器之间的通信，即由 CPU 向设备控制器发出 I/O 命令，要求它完成指定的 I/O 操作；反之由 CPU 接收从控制器发来的中断请求，并给予迅速的响应和相应的处理。

4. 文件管理功能

1) 文件存储空间的管理

文件存储空间的管理是指由文件系统对诸多文件及文件的存储空间实施统一的管理。其主要任务是为每个文件分配必要的外存空间，提高外存的利用率，并能有助于提高文件系统的运行速度。

2) 目录管理

为了使用户能方便地在外存上找到自己所需的文件，通常由系统为每个文件建立一个目录项。目录项包括文件名、文件属性、文件在磁盘上的物理位置等。若干个目录项又可构成一个目录文件。目录管理的主要任务是为每个文件建立其目录项，并对众多的目录项加以有效的组织，以实现方便的按名存取。用户只需提供文件名，即可对该文件进行存取。其次，目录管理还应能实现文件共享，这样，只需在外存上保留一份该共享文件的副本。此外，还应能提供快速的目录查询手段，以提高对文件的检索速度。

3) 文件的读/写管理和保护

根据用户的请求，从外存中读取数据，或将数据写入外存。在进行文件读/写时，系统先根据用户给出的文件名，去检索文件目录，从中获得文件在外存中的位置，然后，利用文

件读/写指针，对文件进行读/写操作。

5. 用户接口

1) 命令接口

它由一组键盘操作命令及命令解释程序所组成。当用户在终端或控制台上每输入一条命令后，系统便立即转入命令解释程序，对该命令加以解释并执行该命令。在完成指定功能后，控制又返回到终端或控制台上，等待用户输入下一条命令。这样，用户可通过先后输入不同命令的方式，来实现对作业的控制，直至作业完成。

2) 程序接口

该接口是为用户程序在执行中访问系统资源而设置的，是用户程序取得操作系统服务的唯一途径。它由一组系统调用组成，每一个系统调用都是一个能完成特定功能的子程序，每当应用程序要求 OS 提供某种服务时，便调用具有相应功能的系统调用。早期的系统调用都是用汇编语言提供的，只有在用汇编语言书写的程序中，才能直接使用系统调用；但在高级语言以及 C 语言中，往往提供了与各系统调用一一对应的库函数，这样，应用程序便可通过调用对应的库函数来使用系统调用。但在近几年所推出的操作系统中，如 UNIX、OS/2 版本中，其系统调用本身已经采用 C 语言编写，并以函数形式提供，故在用 C 语言编制的程序中，可直接使用系统调用。

3) 图形接口

图形用户接口采用了图形化的操作界面，用非常容易识别的各种图标（icon）来将系统的各项功能、各种应用程序和文件，直观、逼真地表示出来。用户可用鼠标或通过菜单和对话框，来完成对应用程序和文件的操作。



1.2 Linux 操作系统

1.2.1 Linux 操作系统的发展历史

Linux 操作系统是 UNIX 操作系统的一种克隆系统。它诞生于 1991 年的 10 月 5 日（这是第一次正式向外公布的时间），之后借助于 Internet 网络，并通过全世界各地计算机爱好者的共同努力，现已成为今天世界上使用最多的一种 UNIX 类操作系统，并且使用人数还在迅猛增长。Linux 操作系统的诞生、发展和成长过程始终依赖着以下五个重要支柱：UNIX 操作系统、MINIX 操作系统、GNU 计划、POSIX 标准和 Internet 网络。

1. UNIX 操作系统的诞生

UNIX 操作系统是美国贝尔实验室的 Ken. Thompson 和 Dennis Ritchie 于 1969 年夏在 DEC PDP-7 小型计算机上开发的一个分时操作系统。当时 Ken Thompson 为了能在闲置不用的 PDP-7 计算机上运行他非常喜欢的星际旅行游戏，在 1969 年夏天的一个月内开发出了 UNIX 操作系统的原型。当时使用的是 BCPL 语言，后经 Dennis Ritchie 于 1972 年用移植性很强的 C 语言进行了改写，使得 UNIX 系统在大专学校得到了推广。

2. MINIX 操作系统

MINIX 系统是由 Andrew S. Tanenbaum（AST）在 1987 年开发的。AST 在荷兰 Amsterdam