

普通高校本科计算机专业

特色

教材精选

实用 UNIX 教程

蒋砚军 高占春 编著

<http://www.tup.com.cn>

31-43

清华大学出版社



普通高校本科计算机专业 **特色** 教材 精选

实用 UNIX 教程

蒋砚军 高占春 编著

清华大学出版社
北 京

内 容 简 介

本书面向 UNIX 的普通用户, 介绍 UNIX 操作系统提供的用户接口。书中讲述了 UNIX 中的基本概念和原理, 在此基础上讲述 UNIX 的主要命令和函数调用。书中重点突出, 理论学习和实际操作相结合, 强调在理解基本原理的基础上理解和掌握 UNIX 提供给操作员的命令和提供给程序员的函数调用, 为深入研究 UNIX 系统打下基础。主要内容包括 UNIX 的命令, shell 编程语言, 进程控制和进程间通信, 以及网络 socket 编程。本书适合高等院校本科生和研究生以及工程技术人员学习 UNIX 使用, 读者不需要事先具备 UNIX 的操作经验, 但需要有 C 语言的基本知识。

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

本书防伪标签采用特殊防伪技术, 用户可通过在图案表面涂抹清水, 图案消失, 水干后图案复现; 或将表面膜揭下, 放在白纸上用彩笔涂抹, 图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

实用 UNIX 教程/蒋砚军, 高占春编著. —北京: 清华大学出版社, 2005. 1

(普通高校本科计算机专业特色教材精选)

ISBN 7-302-09825-5

I. 实… II. ①蒋… ②高… III. UNIX 操作系统—高等学校—教材 IV. TP316.81

中国版本图书馆 CIP 数据核字 (2004) 第 111834 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社 总 机: 010-62770175 客户服务: 010-62776969

责任编辑: 汪汉友

印 刷 者: 北京国马印刷厂

装 订 者: 三河市金元印装有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185 × 260 印张: 21 字数: 485 千字

版 次: 2005 年 1 月第 1 版 2006 年 1 月第 2 次印刷

书 号: ISBN 7-302-09825-5/TP · 6778

印 数: 5001 ~ 8000

定 价: 28.00 元

编审委员会

主任：蒋宗礼

副主任：李仲麟 何炎祥

委员：（排名不分先后）

王向东 宁 洪 朱庆生 吴功宜 吴 跃

张 虹 张 钢 张为群 余雪丽 陈志国

武 波 孟祥旭 孟小峰 胡金初 姚放吾

原福永 黄刘生 廖明宏 薛永生

秘书长：王听讲

出版说明

INTRODUCTION

在我国高等教育逐步实现大众化后，越来越多的高等学校将会面向国民经济发展的第一线，为行业、企业培养各级各类高级应用型专门人才。为此，教育部已经启动了“高等学校教学质量和教学改革工程”，强调要以信息技术为手段，深化教学改革和人才培养模式改革。如何根据社会的实际需要，根据各行各业的具体人才需求，培养具有特色显著的人才，是我们共同面临的重大问题。具体地，培养具有一定专业特色的和特定能力强的计算机专业应用型人才则是计算机教育要解决的问题。

为了适应21世纪人才培养的需要，培养具有特色的计算机人才，急需一批适合各种人才培养特点的计算机专业教材。目前，一些高校在计算机专业教学和教材改革方面已经做了大量工作，许多教师在计算机专业教学和科研方面已经积累了许多宝贵经验。将他们的教研成果转化为教材的形式，向全国其他学校推广，对于深化我国高等学校的教学改革是一件十分有意义的事。

清华大学出版社在经过大量调查研究的基础上，决定组织出版一套“普通高校本科计算机专业特色教材精选”。本套教材是针对当前高等教育改革的新形势，以社会对人才的需求为导向，主要以培养应用型计算机人才为目标，立足课程改革和教材创新，广泛吸纳全国各地的高等院校计算机优秀教师参与编写，从中精选出版确实反映计算机专业教学方向的特色教材，供普通高等院校计算机专业学生使用。

本套教材具有以下特点：

1. 编写目的明确

本套教材是在深入研究各地各学校办学特色的基础上，面向普通高校的计算机专业学生编写的。学生通过本套教材，主要学习计算机科学与技术专业的基本理论和基本知识，接受利用计算机解决实际问题的基本训练，培养研究和开发计算机系统，特别是应用系统的基本能力。

2. 理论知识与实践训练相结合

根据计算学科的三个学科形态及其关系,本套教材力求突出学科的理论与实践紧密结合的特征,结合实例讲解理论,使理论来源于实践,又进一步指导实践。学生通过实践深化对理论的理解,更重要的是使学生学会理论方法的实际运用。在编写教材时突出实用性,并做到通俗易懂,易教易学,使学生不仅知其然,知其所以然,还要会其如何然。

3. 注意培养学生的动手能力

每种教材都增加了能力训练部分的内容,学生通过学习和练习,能比较熟练地应用计算机知识解决实际问题。既注重培养学生分析问题的能力,也注重培养学生解决问题的能力,以适应新经济时代对人才的需要,满足就业要求。

4. 注重教材的立体化配套

大多数教材都将陆续配套教师用课件、习题及其解答提示,学生上机实验指导等辅助教学资源,有些教材还提供能用于网上下载的文件,以方便教学。

由于各地区各学校的培养目标、教学要求和办学特色均有所不同,所以对特色教学的理解也不尽一致,我们恳切希望大家在使用教材的过程中,及时地给我们提出批评和改进意见,以便我们做好教材的修订改版工作,使其日趋完善。

我们相信经过大家的共同努力,这套教材一定能成为特色鲜明、质量上乘的优秀教材,同时,我们也希望通过本套教材的编写出版,为“高等学校教学质量和教学改革工程”作出贡献。

清华大学出版社

前言

PREFACE

在信息技术发展日新月异的今天，UNIX操作系统可以算作是一种很“古老”的操作系统。UNIX自20世纪70年代问世以来，得到了广泛的推广和应用。目前，从硬件配置简单、功能单一的小型嵌入式系统到广为流行的PC、功能强大的工作站以及小型和中大型计算机系统中，都有UNIX操作系统在运行。UNIX成为一种事实上的操作系统标准。UNIX在维持最初的核心框架结构未发生大的变化的基础上，经历了三十多年的发展和完善，衍生出了众多的UNIX版本，并融入了许多新的技术，依然活跃在当今信息技术的各个领域，并赢得了健壮和高效的好名声。近几年，随着可以免费使用的Linux的出现和迅速普及，UNIX系统更具有生命力。UNIX也一直是各种新技术诞生的温床，C语言和Internet都源自UNIX的研究和开发。UNIX以它的简洁、高效、灵活以及开放性，得到了高校师生、研究机构和专业技术人员的推崇，并得到了广泛的应用。可以说UNIX在学术和商业上都取得了巨大的成功。完全可以预测这个“古老”的操作系统，还会在今后相当长的时期内发挥重要作用。

UNIX是一个功能很强而且十分灵活的操作系统和计算环境。系统为操作员提供了丰富的命令，每个命令都有很多选项；系统也为程序员提供了灵活的编程接口。为了全面介绍UNIX系统，本书以UNIX系统的重要概念和基本原理为重点和主要线索，介绍UNIX提供给操作员和程序员的系统功能。可以使得读者从理论上掌握UNIX系统的主体框架，在实际操作理解 and 掌握UNIX常用的命令和函数调用，为进一步深入地研究和拓展UNIX的使用打下坚实的基础。

本书主要包括四部分内容。

第1~4章，介绍UNIX常用的命令。包括常用的处理文本文件的实用程序，vi编辑程序，文件和目录的管理命令。

第5~6章，介绍UNIX的shell。主要介绍C-shell的交互功能和B-shell脚本程序设计。没有兴趣设计shell脚本程序的读者，也应当了解6.5节介绍的shell元字符，以了解交互式使用shell时需要注意的问题。

第7章, 介绍进程的概念和进程间通信。包括新进程的创建, 信号、管道、多进程之间的通信机制(消息队列、信号量、共享内存), 以及文件和记录的锁定。

第8章, 主要介绍基于TCP/IP协议的网络socket编程。

书末附有关键词的索引, 便于读者查阅。书中也含有部分与Windows相似功能的对照, 便于读者学习。

为了便于读者阅读, 书中对印刷字体作了一些约定。

加黑的等宽体并带下划线的内容表示实际上机操作时需要使用键盘输入的内容, 输入后应当再键入Enter键, 不再作特别说明。

等宽体不带下划线的内容, 是实际上机操作时系统显示的内容。没有特别的声明, 命令输入时使用的shell是Bourne shell, 提示符为美元符(\$)。上机操作实例中的一些注释, 用括号内的楷体文字标出。例如:

```
$ echo "Terminal type is $TERM"
```

(带下划线的部分是键盘输入的命令, 其余部分为系统显示的内容)

```
Terminal type is vt100
```

一些组合键, 如: Ctrl+C键, 需要同时按下键盘的Ctrl键和C键, 不再作特别说明。

在本书的写作过程中, 得到了马跃、喻炜、王炜、孙礼、罗强和刘辰等同志的帮助, 在此表示感谢。

本书适于高校本科生和研究生以及工程技术人员使用。

正如程序员不得不面对自己软件中的bug一样, 由于作者水平有限, 书中的错误和不足之处在所难免, 欢迎您把批评和建议发送到作者的电子信箱: jiangy@public3.bta.net.cn。

作 者

2004年7月

目 录

CONTENTS

第1章	UNIX 简介	1
1.1	UNIX 的发展过程和标准化	1
1.1.1	UNIX 的发展过程	1
1.1.2	什么是 UNIX	3
1.1.3	UNIX 的标准化	3
1.2	系统的登录与退出	3
1.2.1	UNIX 的主机和终端	3
1.2.2	登录	5
1.2.3	退出	6
1.2.4	关机	6
1.3	使用系统命令	6
1.3.1	man:查阅联机手册	6
1.3.2	获取系统情况	9
1.3.3	passwd:更换用户口令	10
1.3.4	与其他用户通信	11
1.3.5	与其他主机通信	12
1.3.6	几个实用工具	17
第2章	基本 UNIX 实用程序	21
2.1	more,less,pg:逐屏显示文件内容	22
2.2	cat:列出文本文件内容	24
2.3	od:列出文件每个字节的内容	24
2.4	head 与 tail:打印文件头或尾	25
2.5	wc:字计数	26
2.6	sort:对文件内容排序	28
2.7	tee:三通	29
2.8	正则表达式的概念	29

2.9	grep, egrep 与 fgrep: 在文件中查找字符串	31
2.10	awk: 文本处理语言	34
2.11	sed: 流编辑	36
2.12	tr: 翻译字符	37
2.13	cmp 和 diff: 比较两个文件	38
第 3 章	全屏编辑程序 vi	41
3.1	vi 的启动方法	41
3.2	vi 的选项	41
3.3	vi 的工作方式	42
3.4	vi 的编辑命令	42
3.4.1	正文插入命令	43
3.4.2	光标移动命令	43
3.4.3	设置书签	45
3.4.4	删除	45
3.4.5	字符替换	45
3.4.6	取消和重复	46
3.4.7	文件命令	46
3.4.8	段落的删除、复制和移动	46
3.4.9	剪贴板	47
3.4.10	其他命令	48
3.4.11	模式查找	48
3.4.12	模式替换	48
3.4.13	编辑命令小结	52
第 4 章	UNIX 的文件和目录	55
4.1	文件和目录的层次结构	55
4.2	文件和目录的命名	56
4.3	shell 的文件名通配符	56
4.3.1	规则	56
4.3.2	与 DOS 文件名通配符的区别	56
4.3.3	文件名通配符的处理过程	57
4.3.4	验证文件名匹配的结果	58
4.4	文件管理	60
4.4.1	ls: 文件名列表	60
4.4.2	cp: 复制文件	64
4.4.3	mv: 移动文件	65
4.4.4	rm: 删除文件	65
4.4.5	find: 查找文件	67

4.5	目录管理	69
4.5.1	路径名	69
4.5.2	pwd:打印当前工作目录	70
4.5.3	cd:改变当前工作目录	70
4.5.4	mkdir:创建目录	70
4.5.5	rmdir:删除目录	70
4.5.6	cp:复制目录	71
4.6	文件的归档与压缩处理	71
4.6.1	tar:文件归档	71
4.6.2	compress:文件压缩	72
4.6.3	应用	73
4.7	文件系统的存储结构	73
4.7.1	基本文件系统与子文件系统	73
4.7.2	文件系统的结构	75
4.7.3	目录结构	76
4.7.4	命令 df 与 du	77
4.8	硬连接与符号连接	79
4.8.1	硬连接	79
4.8.2	符号连接	82
4.8.3	硬连接与符号连接的比较和应用	85
4.9	系统调用	86
4.10	文件和目录的访问	88
4.10.1	文件存取	88
4.10.2	目录访问	93
4.10.3	获取文件系统的信息	96
4.11	获取文件的状态信息	97
4.12	设备文件	99
4.13	文件和目录的权限	101
4.13.1	权限控制的方法	101
4.13.2	查看文件和目录的权限	103
4.13.3	chmod:修改权限	103
4.13.4	umask:改变文件创建状态掩码	106
4.13.5	SUID 权限和 SGID 权限	107
第 5 章	C-shell 的交互功能	113
5.1	UNIX 的 shell	113
5.2	csh 启动与终止	114
5.3	使用 csh 的历史机制	115

5.3.1	历史表大小	115
5.3.2	查看历史表	115
5.3.3	引用历史机制	115
5.4	别名	116
5.4.1	在别名表中增加一个别名	116
5.4.2	查看别名表	117
5.4.3	给别名传递参数	117
5.4.4	取消别名	118
5.5	csh 提示符	118
5.6	csh 的管道和重定向	118
5.6.1	标准输入, 标准输出, 标准错误输出	119
5.6.2	标准输出和标准错误输出重定向	120
5.6.3	管道	121
第 6 章	B-shell 及编程	123
6.1	启动 B-shell	123
6.1.1	启动一个交互式 B-shell	123
6.1.2	#!/bin/sh:脚本文件的执行	123
6.2	重定向与管道	125
6.2.1	输入重定向	125
6.2.2	输出重定向	127
6.2.3	管道	129
6.3	变量	129
6.3.1	变量赋值和引用	129
6.3.2	read:读用户的输入	130
6.3.3	环境变量和局部变量	131
6.3.4	内置变量	132
6.3.5	shell 的标准变量	134
6.4	替换	135
6.4.1	文件名生成	135
6.4.2	变量替换	135
6.4.3	命令替换	136
6.5	元字符	136
6.5.1	空格、制表符和转义符	137
6.5.2	回车和分号	138
6.5.3	文件名通配符	138
6.5.4	美元符和反撇号	138
6.5.5	重定向和管道	138

6.5.6	启动程序后台执行	140
6.5.7	括号	140
6.5.8	转义符	141
6.5.9	双引号和单引号	142
6.5.10	转义符与引号及反撇号	142
6.6	条件判断	144
6.6.1	条件	144
6.6.2	最简单的条件判断	146
6.6.3	命令 true 与命令 false	147
6.6.4	命令 test 与命令[	147
6.6.5	{ }与()	149
6.6.6	条件结构 if	152
6.6.7	case 结构	153
6.7	循环结构	154
6.7.1	while 结构	154
6.7.2	expr:计算表达式的值	155
6.7.3	for 结构	157
6.7.4	break 与 continue	159
6.8	函数	161
6.9	shell 开关和位置变量	163
6.9.1	set:设置 B-shell 内部开关	163
6.9.2	set:设置 shell 位置变量	165
6.9.3	shift:位置变量的移位	166
第 7 章	进程控制与进程间通信	169
7.1	进程控制	169
7.1.1	进程的基本概念	169
7.1.2	fork:创建新进程	180
7.1.3	exec:重新初始化进程	183
7.1.4	wait:等待子进程运行结束	186
7.1.5	xsh0.c:最简单的 shell	188
7.1.6	vfork 和 _exit	191
7.1.7	system:在程序中运行一个命令	193
7.1.8	ps:列出进程的状态	195
7.2	信号	197
7.2.1	信号的产生及信号类型	197
7.2.2	kill:发送信号	200
7.2.3	信号的捕捉与处理	204

7.2.4	longjmp:全局跳转	210
7.2.5	信号对进程执行的影响	213
7.2.6	sleep, pause 与 alarm	214
7.2.7	trap 命令:shell 对信号的处理	216
7.3	进程与文件描述符	218
7.3.1	内核中的文件打开结构	218
7.3.2	文件描述符的继承和复制	221
7.3.3	管道操作	226
7.4	消息队列	232
7.4.1	有关的系统调用	232
7.4.2	程序举例	235
7.4.3	消息队列的特点	239
7.4.4	死锁	240
7.4.5	命令 ipcs 和 ipcrm	241
7.4.6	有关的主要系统参数	242
7.5	信号量	242
7.5.1	创建或获取信号量组	243
7.5.2	信号量组的控制	243
7.5.3	信号量操作	244
7.5.4	相关命令和系统参数	245
7.6	共享内存	245
7.6.1	创建或获取共享内存段	246
7.6.2	获取指向共享内存段的指针	246
7.6.3	共享内存段的控制	246
7.6.4	相关的命令和系统参数	246
7.7	信号量和共享内存使用举例	247
7.8	内存映射文件 I/O	253
7.9	文件和记录的锁定	255
7.9.1	设置文件锁定的必要性	255
7.9.2	锁定操作	256
7.9.3	使用举例	258
7.9.4	咨询式锁定和强制性锁定	261
7.9.5	死锁	262
7.9.6	文件锁的隐式释放	262
7.9.7	获取记录的加锁状态	264

第 8 章 网络程序设计 267

8.1	概述	267
-----	----	-----

8.1.1	socket	267
8.1.2	TCP 与 UDP	268
8.1.3	网络字节次序	269
8.2	TCP 客户—服务器程序	270
8.2.1	TCP 客户端程序	270
8.2.2	最简单的 TCP 服务器端程序	273
8.2.3	多进程并发处理的 TCP 服务器端程序	275
8.3	与 socket 有关的系统调用	279
8.3.1	socket:创建一个 socket	279
8.3.2	bind:指定本地端点名	279
8.3.3	listen:开始监听到达的连接请求	280
8.3.4	accept:接受一个连接请求	280
8.3.5	connect:建立连接	280
8.3.6	read 和 write:接收和发送	281
8.3.7	send/sendto 和 recv/recvfrom	284
8.3.8	fcntl:无阻塞 I/O	285
8.3.9	getpeername:获取对方的端点名	286
8.3.10	getsockname:获取本地的端点名	286
8.3.11	shutdown:禁止发送或接收	286
8.3.12	close:关闭一个连接	287
8.3.13	getsockopt, setsockopt 和 ioctl	287
8.3.14	库函数	289
8.4	单进程并发处理	291
8.4.1	select:I/O 多路转接	291
8.4.2	单进程并发处理的 TCP 服务器端程序	295
8.5	UDP 通信	298
8.5.1	UDP 客户端程序	298
8.5.2	UDP 服务器端程序	300
8.6	与 TCP/IP 通信相关的命令	302
8.6.1	查看 IP 接口状态	302
8.6.2	打印 IP 路由表	303
8.6.3	协议统计信息	303
8.6.4	查看连接和端点名	304
附录 A	ASCII 码表	307
参考文献		311
索引		313

第 1 章

CHAPTER

UNIX 简介

1.1 UNIX 的发展过程和标准化

1.1.1 UNIX 的发展过程

20 世纪 60 年代末期, 由美国通用电气、麻省理工学院和 AT&T 的贝尔实验室联合创立了一个庞大的项目, 就是开发一种功能强大的操作系统 MULTICS (MULTiplexed Information and Computing System, 多路信息与计算系统)。该系统设计目标宏伟, 功能强大, 是一个分时和友好的作业环境。MULTICS 的原始版本于 1969 年在 GE645 计算机上运行了, 但由于效率低, 功能庞杂, 没能取得预期的成功。但是, MULTICS 的研究对操作系统的发展起到了很重要的作用。

1969 年, 从事 MULTICS 研究的贝尔实验室的 Ken Thompson 和他的同事 Dennis Ritchie, 在 DEC 的 PDP-7 型计算机上编写了一个简易的新型操作系统, 是一个两用户的多任务操作系统。整个系统都用汇编语言编写。从技术角度上, 这个新的操作系统并没有引用更新的技术, 主要对 MULTICS 的技术做了合理的裁减, 不追求大而全, 追求小而精。Brian Kernighan 在 1971 年给这套系统取名时, 针对于它的前身 MULTICS, 半开玩笑式的命名为 UNICS (UNiplexed Information and Computing System)。UNICS 的尾音-ICS 与-IX 类似, 于是被改写成 UNIX, 名字就这样被确定下来。众所周知, UNIX 在后来取得了巨大成功。

UNIX 最初用汇编语言编制, 在 PDP-7 上实现。在把 UNIX 移植到更新的计算机系统 PDP-11 上时遇到了一些麻烦。PDP-11 不完全兼容 PDP-7 的指令, 不像今天的 Pentium 完全兼容 80386。这样, 由于汇编语言不兼容, 在汇编语言级上的移植变得比较麻烦。而且以汇编语言编写的程序由于可读性差, 维护起来又很困难。因此, 必须寻找一种高级语言, 提高可读性和可移植性, 同时又要求不丧失汇编语言高效的优点。C 语言在这种需求下应运而生。Dennis Ritchie 发明了 C 语言, 并与 Thompson 用 C 语言

改写了 UNIX 的源程序，为以后的开发和移植奠定了基础。DEC 公司的 PDP-11 是 20 世纪 70 年代小型机的主流机型，被广泛用在大学的实验室中。贝尔实验室把 UNIX 的 C 语言源程序代码和说明书赠送给美国的大学，使得 UNIX 成为许多大学操作系统课程的范例，为 UNIX 的普及和以后的成功奠定了基础。

C 语言最初是为了能够在不同的硬件平台上移植 UNIX 而产生的。UNIX 和 C 语言都取得了巨大成功，UNIX 自身就是 C 程序设计语言在系统软件领域的成功范例。反过来，UNIX 推动了 C 语言的应用和普及。C 语言后续的衍生版本 C++ 和 Java 都是目前非常活跃的编程语言。

1983 年美国计算机协会的最高年度奖“图灵奖”，颁发给了 UNIX 和 C 语言的主要奠基人 Ken Thompson 和 Dennis Ritchie，以表彰他们在 UNIX 和 C 语言上的贡献。

随后，UNIX 被移植到各种各样的计算机系统上，UNIX 也经历了二三十年的发展，引进了许多新的技术，成为更加成熟的操作系统。

UNIX 发展过程中，曾经有两大流派，一个是以 UNIX 的鼻祖 AT&T 最早于 1983 年发表的 UNIX System V，这里的 V 是罗马数字 5，最新的版本是 Release 4，简称为 SVR4。另一个是学术派的加州大学伯克利分校计算机系统研究小组 (CSRG) 发表的 BSD UNIX (Berkeley Software Distributions UNIX)，比较有影响的版本是 4.3，记作 4.3BSD。它在 UNIX 中引进了许多新技术，风靡全球的 Internet 上的 TCP/IP 协议，最早就是在 BSD 上开发的，它的后继版本就是现在的 FreeBSD。早期的 Internet 上的大部分计算机都运行 UNIX。两种流派的 UNIX 随着不断的发展，互相吸收对方的新技术，趋于统一到逐渐完备的 POSIX 标准。

UNIX 后来成为了 AT&T 贝尔实验室操作系统产品的注册商标，所以，许多操作系统由于商业上的原因，在命名时不能够使用 UNIX 这个字眼，如 IBM RISC/6000 小型机上的 AIX，Sun 工作站上的 SunOS 以及后续版本 Solaris，CDC 小型机上的 EP/IX，惠普 Hewlett-Packard 的 HP-UX，等等。几乎所有的工作站平台上都运行 UNIX 操作系统。

UNIX 成为一种软件商品后，其源代码也就不再随意公开。早期的 UNIX 用作许多大学操作系统课程的范例，澳大利亚的 John Lions 教授还为 UNIX 的源程序逐行写了注释，这就是著名的《莱昂氏 UNIX 源代码分析》一书。后来由于其源代码不公开，为了教学上的需要，Andrew S. Tanenbaum 教授在 UNIX 的基础上开发了教学版的 UNIX，取名 MINIX，意为 Mini-UNIX。它的功能简单，模块清晰，结构易懂，可以运行在流行的 Intel x86 系列的 PC 上，非常适于操作系统课程的教学使用。1991 年，芬兰学生 Linus Benedict Torvalds，对于 MINIX 系统十分熟悉，在 PC 上独立编写了基于 Intel 80386 体系结构的 UNIX 系统，相对简单易懂的纯教学工具 MINIX 有更好的性能，命名为 freax。作者 Linus 承认这套系统有点像 MINIX，但是完全没有用到 MINIX 的源代码。Linus 将这套系统的所有源程序免费公布在芬兰最大的 FTP 站点，由于这套系统是“Linus 的 MINIX”，因此，就建立了一个名为 Linux 的目录存放这些代码，并通过 Internet 请世界各地的软件爱好者参加 Linux 的开发工作。最终 Linux 成了这套系统的名字。世界各地的软件爱好者都可以通过 Internet 免费获得 Linux 的全部源代码和二进制发行版本。现在，Linux 更是发展迅速。从比 PC 还要简单得多的小型嵌入式环境，一直到大型的服务器，