

Linux 编程典型实例解析

电脑编程技巧与维护杂志社 编著



清华大学出版社
北京

内容简介

本书通过编程实例清晰地剖析了 Linux 内核的内存管理中进程线性区的管理、内核的物理内存的分配与回收、内核的物理内存的页面周转与缓冲区、内核的虚拟内存的管理、内核的进程创建和执行、内核的进程调度、管道与信号以及嵌入式 Linux 内核模块的配置与编译等；深入分析了 Linux 系统应用基础编程；通过大量编程实例导航对 Linux 在数据库、系统安全与多媒体、网络与通信等方面的应用编程作了示范性的说明。

本书强调理论与实践相结合，突出实用性，通过一个个典型编程实例导航，帮助读者掌握 Linux 环境下应用编程方法和技巧，对深入研究 Linux 环境下的编程有很大的帮助。

本书适合 Linux 的初学者及希望利用 Linux 进行开发的程序设计人员阅读，也适合作为高等院校计算机相关专业师生的参考教材。

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。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目 (CIP) 数据

Linux 编程典型实例解析/电脑编程技巧与维护杂志社编著. —北京：清华大学出版社，2009.1
ISBN 978-7-302-19041-7

I. L… II. 电… III. Linux 操作系统—程序设计 IV. TP316.89

中国版本图书馆 CIP 数据核字 (2008) 第 190845 号

责任编辑：应勤 孙兴芳

封面设计：杨玉兰

责任校对：李玉萍

责任印制：何芊

出版发行：清华大学出版社

地址：北京清华大学学研大厦 A 座

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

邮编：100084

社总机：010-62770175

邮购：010-62786544

投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

印刷者：北京密云胶印厂

装订者：三河市新茂装订有限公司

经销：全国新华书店

开本：185×260 印张：31.5 字数：760 千字

版次：2009 年 1 月第 1 版 印次：2009 年 1 月第 1 次印刷

印数：1~4000

定价：49.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010)62770177 转 3103 产品编号：027358 01

前 言

时至今日，Linux 系统已经从一个个人作品发展为一个可以用于各种关键任务的成熟、高效和稳定的操作系统。因为具备跨平台、开源、支持众多应用软件和网络协议等优点，它得到了各大主流软硬件厂商的支持，也成为广大程序设计人员理想的开发平台。

本书从《电脑编程技巧与维护》杂志的最新稿件中精选了有关 Linux 系统应用编程的 74 个典型实例，共分 5 章。

第 1 章为 Linux 内核剖析及其编程，本章通过编程实例的介绍清晰地剖析了 Linux 内核的内存管理中进程线性区的管理、物理内存的分配与回收、物理内存的页面周转与缓冲区、虚拟内存的管理、进程创建和执行、进程调度、管道与信号以及嵌入式 Linux 内核模块的配置与编译等。

第 2 章为 Linux 系统应用基础编程，本章通过 Linux 编程常用函数、Linux 平台下创建和使用用户函数库、多进程应用程序调试技巧、Linux 中守护进程的原理及其程序设计方法、Linux 下 Java 程序的编译与调试、Linux 文件与目录编程、Linux 字符终端界面面向对象的编程、Linux 下自动隐藏窗口的程序实现、基于嵌入式 Linux 的应用程序开发以及 Linux 下 newt 的高级编程等编程实例，深入地分析了 Linux 系统应用基础编程。

第 3 章为 Linux 数据库编程，本章通过在 Linux 平台下用 PHP 访问 MS SQL Server 数据库、安装配置与 SQL 2000 数据库相连的 JSP 服务器、实现 Oracle 9I 数据库的自动异地备份、基于 Linux 的 VPN 服务器中连接用户数据的获取等编程实例，指导读者掌握在 Linux 环境中进行数据库编程的方法与技巧。

第 4 章为 Linux 安全与多媒体应用编程，本章通过 Linux 系统中超级用户密码的安全问题、Linux 文件系统安全检查实现、Winux 病毒感染 Linux 下 ELF 文件的分析、基于 Linux 的路由器和防火墙技术、实现 Linux 架构下的防火墙扩展技术及入侵检测、基于 Linux 的视频采集系统的研究、嵌入式 Linux 下的视频和图像采集以及 Linux 音频应用程序设计等应用实例导航，对 Linux 在系统安全与多媒体等方面的应用编程作了示范性的解析与说明。

第 5 章为 Linux 网络与通信应用编程。本章通过 Linux 网关设置、基于以太网络的应用程序开发、Linux 下构建企业级 FTP 服务器的方案、基于 Linux 的 Client/Server 程序设计、基于 Web 的应用开发、Linux 平台 POP3 信箱客户端实现技术、开发 Linux 环境下的服务器程序、Linux 系统网络设备驱动程序的设计和实现、基于嵌入式 Linux 的设备驱动程序开发、在 Linux 平台上实现不同厂商 ORB 间通信的程序以及 Linux 进程间的管道通信等典型编程实例，解析了在 Linux 环境下进行网络与通信应用编程的方法与技巧。

本书通过编程实例的介绍，清晰地剖析了 Linux 内核的内存管理中进程线性区的管理、物理内存的分配与回收、物理内存的页面周转与缓冲区、虚拟内存的管理、进程创建和执行、进程调度、管道与信号以及嵌入式 Linux 内核模块的配置与编译等；深入地分析了 Linux 系统应用基础编程；示范性地说明了 Linux 在数据库、系统安全与多媒体、网络

与通信等方面的应用编程。

本书强调理论与实践相结合，突出实用性，通过一个个的典型编程实例导航，来帮助读者掌握 Linux 环境下的应用编程方法和技巧，对读者深入研究 Linux 环境下的编程有很大的帮助。本书的特色如下：

一、每一章都是通过一个个的实例来介绍 Linux 应用编程方法和技巧的，避免了枯燥、空洞的理论，并且每个实例都具有很强的实用性和代表性。在实例的讲解上，一般都是先给出设计目标，然后介绍实现该目标的基本思想和方法，最后详细给出其核心部分的源代码，并对程序的关键部分作出讲解和给出程序的运行结果。

二、本书所选的每个实例都是从事 Linux 应用编程的人员的经验总结，具有很强的实用性，其中很多编程技巧可供借鉴。

三、每个实例的程序源代码都是上机调试通过的，这给程序开发人员移植源代码带来了方便，加快了编程应用的步伐。

本书适合 Linux 的初学者及希望利用 Linux 进行程序开发的设计人员阅读，也适合作为高等院校计算机相关专业师生的参考教材。

本书由王路敬任主编，田真、孙茹萍、候穆蕾任副主编，编委由姬振伟、苏加友、杨月会、刘艳彬、韦玉发、夏义宇、李相兰、张振华、李晓鸿、姚建华、张森、袁伟、苏萌、高燕云、刘巍、杨杰、周源、王万中、韩大为、渠素真组成。

书中若有不足和错误之处，敬请读者批评指正。

目 录

第 1 章 Linux 内核剖析及其编程

1.1 Linux 内存管理中进程线性区的管理	1
1.1.1 概述	1
1.1.2 线性空间的管理	1
1.1.3 结束语	9
1.2 Linux 物理内存的分配与回收	9
1.2.1 概述	9
1.2.2 物理内存的分配和回收	10
1.2.3 结束语	19
1.3 Linux 物理内存的页面周转与缓冲区	20
1.3.1 概述	20
1.3.2 物理页面的周转	20
1.3.3 缓冲区	28
1.3.4 结束语	31
1.4 Linux 虚拟内存的管理	31
1.4.1 概述	31
1.4.2 虚拟内存的管理	31
1.4.3 结束语	41
1.5 Linux 内核的进程创建和执行	42
1.5.1 概述	42
1.5.2 创建进程	43
1.5.3 系统调用	50
1.5.4 结束语	56
1.6 Linux 的进程调度	56
1.6.1 概述	56
1.6.2 数据结构	57
1.6.3 进程调度	59
1.7 Linux 的管道与信号	65
1.7.1 概述	65
1.7.2 管道	66
1.7.3 信号	76
1.7.4 结束语	82
1.8 嵌入式 Linux 内核模块的配置与编译	82
1.8.1 简介	82
1.8.2 内核模块	83

1.8.3 将内核模块编入操作系统	84
1.8.4 内核配置模块格式	85
1.9 实现 Linux 中的魔幻键编程	87
1.9.1 魔幻键概述	87
1.9.2 利用内核接口实现魔幻编程	88
1.9.3 利用魔幻键原理实现魔幻键编程	90
1.9.4 魔幻键实现的核心机制分析	92
1.9.5 结束语	94
第 2 章 Linux 系统应用基础编程	
2.1 Linux 编程常用函数	95
2.1.1 C 语言的标准函数	95
2.1.2 结束语	99
2.2 Linux 下创建和使用用户函数库	99
2.2.1 Linux 下有关函数库的概念	99
2.2.2 用 Gcc 创建函数库	100
2.2.3 创建一个用于数值计算的特殊函数库	101
2.2.4 结束语	102
2.3 Linux 下的多进程应用程序调试技巧	102
2.3.1 程序的实现	103
2.3.2 tty1 的 shell 下的演示	104
2.3.3 在 tty8 上的情景	105
2.3.4 tty1 中调试父进程, tty8 中调试子进程	105
2.4 Linux 下 Qt GUI 应用程序框架的建立	106
2.4.1 Qt 简介及其主要类与 MFC 的比较	106
2.4.2 Linux 下 Qt 开发环境的建立	107
2.4.3 GUI 应用程序框架的建立	107
2.5 UNIX/Linux 下 newt 的高级编程	109
2.5.1 编程基本步骤	109
2.5.2 基本组件	110
2.5.3 高级对话框	110
2.5.4 窗格布局的高级组件 Grid	113
2.5.5 结束语	117
2.6 Linux 中守护进程的原理及其程序设计方法	117
2.6.1 关闭文件描述符	118
2.6.2 甩开控制终端	118
2.6.3 改变工作目录	118
2.6.4 重新设置文件创建掩码	119
2.6.5 处理 SIGCHLD 信号	119
2.7 Linux 进程调度程序的改写	122

2.7.1 问题的提出	122
2.7.2 程序的实现	123
2.8 Linux 环境下的二进制代码重用	130
2.8.1 问题的提出	130
2.8.2 程序的实现	130
2.8.3 结束语	140
2.9 Linux 下 Java 程序的编译与调试	140
2.9.1 引言	140
2.9.2 开发环境的安装与配置	141
2.9.3 利用 GCJ 编译 Java 代码	142
2.9.4 利用 gdb 调试 Java 程序	144
2.10 Linux 多线程环境下 System V 信号灯的使用	146
2.10.1 引言	146
2.10.2 System V 信号灯的实现	147
2.10.3 结束语	152
2.11 基于 Linux 环境下的 Sniffer 设计与实现	152
2.11.1 Sniffer 原理分析	152
2.11.2 Sniffer 具体实现	154
2.11.3 运行结果及结论	157
2.12 Linux 下使用 C++ 进行 Socket 编程	161
2.12.1 引言	161
2.12.2 Socket 通信原理	162
2.12.3 一个简单的客户/服务器通信的 C 语言实现	163
2.12.4 使用 C++ 封装 Socket 操作	165
2.12.5 Socket 通信的 C++ 实现	167
2.12.6 结束语	168
2.13 Linux 守护进程的编写	169
2.13.1 守护进程及其特性	169
2.13.2 守护进程的编程要点	169
2.13.3 守护进程实例	171
2.14 在 Linux 下编制 RPC 程序	172
2.15 基于嵌入式 Linux 的应用程序开发	175
2.15.1 开发方式	175
2.15.2 NFS 的建立与配置	177
2.15.3 应用程序	179
2.16 Linux 下文件系统的编程	182
2.16.1 Linux 文件系统简介	182
2.16.2 文件的基本操作	183
2.16.3 实例	188
2.17 Linux 下 Makefile 的 automake 生成全攻略	190

2.17.1	Makefile 介绍	190
2.17.2	使用的环境	190
2.17.3	从 helloworld 入手	191
2.17.4	深入浅出	194
2.17.5	结束语	196
2.18	Linux 文件与目录编程	197
2.18.1	引言	197
2.18.2	操作函数说明	198
2.18.3	文件与目录函数实用例子	201
2.18.4	结束语	204
2.19	Linux 字符终端界面面向对象的编程	205
2.19.1	概述	205
2.19.2	设计方法	205
2.19.3	程序的运行过程	206
2.19.4	源程序	206
2.19.5	结束语	212
2.20	Linux 下自动隐藏窗口的程序实现	212
2.20.1	自动隐藏功能目的	212
2.20.2	自动隐藏功能介绍	212
2.20.3	Linux 下自动隐藏程序实现	213
2.21	Linux 下用 C++ 进行 OOP 窗口编程	216
2.21.1	X Window 系统	216
2.21.2	Qt 库	217
2.21.3	第一个“Hello World!”程序	217
2.21.4	Signal - Slot 机制	218
2.21.5	使用控件(Widget)	220
2.21.6	响应鼠标事件和在窗口中画图	223
2.21.7	一个完整的应用程序框架	224
2.21.8	进一步讨论	227
2.22	Linux 应用经验与技巧	228
2.22.1	root 口令遗忘后的解决方法	228
2.22.2	查看系统版本信息	229
2.22.3	查看特定主机的 MAC 地址	229
2.22.4	在 Linux 中直接使用 iso 镜像文件	230
2.22.5	以 root 身份 ftp 到服务器	230
2.22.6	以 root 身份 telnet 到系统	230
2.22.7	启动或关闭 ftp 或 telnet 服务	230
2.22.8	制作 Linux 应急启动盘	231
2.22.9	阻止 Linux 系统响应 Ping 询问	232

第3章 Linux 数据库编程

3.1 如何在 Linux 平台下用 PHP 访问 MS SQL Server 数据库	233
3.2 Linux 下 Oracle 宿主语言编程	235
3.2.1 引言	235
3.2.2 环境设置	236
3.2.3 Pro * C 编程	237
3.2.4 JDBC 编程	239
3.2.5 SQLJ 编程	241
3.2.6 结束语	242
3.3 在 Linux 下安装配置与 SQL 2000 数据库相连的 JSP 服务器	243
3.3.1 安装前准备	243
3.3.2 软件包安装与配置	244
3.3.3 配置自己的工作目录	245
3.3.4 SQL 2000 JDBC 驱动程序的安装与配置	246
3.3.5 数据库连接测试	247
3.4 Linux 环境中使用 Flex、Bison 进行 SQL 语法分析	249
3.4.1 引言	249
3.4.2 Flex、Bison 工具	250
3.4.3 SQL 语法分析的特点与目标	250
3.4.4 SQL 词法分析	251
3.4.5 SQL 语法分析	253
3.4.6 分析过程中的错误处理	259
3.4.7 SQL 分析结果应用	260
3.4.8 结束语	260
3.5 在 Linux 系统下实现 Oracle 9i 数据库的自动异地备份	261
3.5.1 系统环境	261
3.5.2 技术实现	261
3.5.3 应用效果	264
3.6 基于 Linux 的 VPN 服务器中连接用户数据的获取	264
3.6.1 引言	264
3.6.2 在 RedHat 9 中安装 PPTP VPN 服务器	265
3.6.3 安装 MySQL	266
3.6.4 修改 PPP 代码	267
3.6.5 测试与结果	271

第4章 Linux 安全与多媒体应用编程

4.1 Linux 文件系统安全检查实现	272
4.1.1 漏洞分析	272
4.1.2 设计思想	273
4.1.3 具体实现	274

4.1.4	结束语	276
4.2	Linux 系统中超级用户口令的安全问题	276
4.2.1	问题的提出	276
4.2.2	问题的分析	276
4.2.3	解决的思路与方法	277
4.2.4	结束语	280
4.3	Winux 病毒感染 Linux 下 ELF 文件的分析	280
4.3.1	前言	280
4.3.2	ELF 文件的基本格式	280
4.3.3	Winux 病毒感染 ELF 文件分析	281
4.3.4	结束语	283
4.4	基于 Linux 的全功能防火墙	284
4.4.1	安装基本系统	285
4.4.2	Iptables 和入侵检测	285
4.4.3	辅助功能	291
4.5	基于 Linux 的路由器和防火墙技术	292
4.5.1	引言	292
4.5.2	包过滤防火墙	292
4.5.3	代理防火墙	293
4.6	基于 Linux 环境下的包过滤防火墙的设计与实现	296
4.6.1	防火墙工作原理分析	296
4.6.2	防火墙的具体实现及功能评述	299
4.6.3	运行结果及结论	302
4.6.4	附源代码	302
4.7	实现 Linux 架构下的防火墙扩展技术及入侵检测	312
4.7.1	Linux 防火墙的扩展 Netfilter/Iptables 的技术	312
4.7.2	Linux 防火墙入侵检测扩展匹配设计	312
4.7.3	结束语	316
4.8	基于 Video4Linux 的 USB 设备视频图像采集	316
4.8.1	Video4Linux 简介	316
4.8.2	加载驱动模块	316
4.8.3	Video4Linux 程序设计	317
4.8.4	运行结果	320
4.8.5	结束语	320
4.9	嵌入式 Linux 下的视频和图像采集	321
4.9.1	引言	321
4.9.2	设计思路	321
4.9.3	程序框架	322
4.9.4	程序模块	322
4.9.5	程序流程	326

4.9.6	问题分析	326
4.10	基于 Linux 视频采集系统的深入研究	328
4.10.1	Video 4 Linux 视频采集编程接口分析	329
4.10.2	基于 Video 4 Linux 的视频采集处理流程	332
4.10.3	结束语	334
4.11	Linux 音频应用程序设计	334
4.11.1	引言	334
4.11.2	OSS 编程接口	335
4.11.3	OSS 编程机制	336
4.11.4	OSS 的一般框架	338
4.11.5	OSS 程序设计示例	338
4.11.6	结束语	347
第 5 章	Linux 网络与通信应用编程	
5.1	Linux 网络编程	348
5.1.1	前言	348
5.1.2	结构体	348
5.1.3	同步 Socket 服务程序介绍	349
5.2	Linux 下的网关设置	354
5.2.1	安装网卡	354
5.2.2	启动 IP 转发	355
5.2.3	添加路由信息	355
5.3	基于以太网络的应用程序开发	356
5.3.1	基本知识	356
5.3.2	编程示例	357
5.4	在 Linux 环境中向 PHP 加入自编函数的方法	361
5.4.1	简介	361
5.4.2	加入自编函数的方法	361
5.4.3	结束语	364
5.5	基于 Web 的应用开发	364
5.5.1	嵌入式 Web 服务器的移植	364
5.5.2	基于 Web 的应用开发	366
5.6	Linux 下预创建线程并发网络服务器的两种实现方法	376
5.6.1	引言	376
5.6.2	使用条件变量	376
5.6.3	并发调用 accept	379
5.6.4	结束语	380
5.7	Linux 下构建企业级 FTP 服务器的方案	381
5.7.1	安装 vsftpd	381
5.7.2	启动/重新启动/停止 vsftpd 服务	381

5.7.3	配置 vsftpd	382
5.7.4	优化工作	383
5.7.5	最终的 vsftpd.conf 文件	384
5.8	基于 Linux 的 Client/Server 程序设计	386
5.8.1	客户/服务器编程模型	386
5.8.2	客户/服务器模型的 Socket 实现	387
5.8.3	测试	391
5.8.4	结束语	391
5.9	Linux 平台 POP3 信箱客户端实现技术	392
5.9.1	POP3(邮局协议 - 版本 3)简介	392
5.9.2	编程要点	392
5.9.3	实现过程	394
5.9.4	实现结果	399
5.10	在 Linux (UNIX) 中配置 Sendmail 阻止垃圾邮件的发送和接收	401
5.10.1	邮件用户代理和邮件传输代理	401
5.10.2	Sendmail 的用法及选项	401
5.10.3	实际配置举例	404
5.11	开发 Linux 环境下的服务器程序	404
5.11.1	循环服务器模型	405
5.11.2	即时创建型并发服务器模型	405
5.11.3	预创建型并发服务器模型	406
5.11.4	另一种预创建型并发服务器模型	407
5.12	Linux 网络设备驱动程序的设计和实现	408
5.12.1	引言	408
5.12.2	运行机理	409
5.12.3	实现模式	410
5.12.4	应用实例	411
5.12.5	结束语	413
5.13	在 Linux 平台上实现不同厂商 ORB 间通信的程序	413
5.14	Linux 进程间的管道通信	417
5.14.1	管道的概念	417
5.14.2	管道的建立和使用	418
5.14.3	管道操作	420
5.14.4	示例程序	421
5.14.5	几个问题	422
5.15	Linux 管道通信应用示例	422
5.16	嵌入式 Linux 下 PCI 驱动程序的设计与实现	425
5.16.1	编写 PCI 驱动程序	425
5.16.2	操作函数的实现	429
5.16.3	结束语	430

5.17 基于嵌入式 Linux 的设备驱动程序开发	430
5.17.1 基于字符设备的驱动程序	430
5.17.2 将设备驱动程序添加到 Linux 内核	433
5.17.3 编写应用程序访问设备	433
5.17.4 中断处理、硬件操作及内核定时器的实现	435
5.17.5 结束语	439
5.18 嵌入式 Linux 的 SD 卡驱动程序	439
5.18.1 引言	439
5.18.2 Linux 的设备驱动程序	439
5.18.3 SD 存储卡的操作	440
5.18.4 Linux 的 SD 卡驱动程序	442
5.18.5 结束语	444
5.19 基于串行通信的应用程序开发	445
5.19.1 基本知识	445
5.19.2 编程示例	447
5.19.3 编程方法	449
5.19.4 结束语	453
5.20 Linux X Window 环境下用 Kylix 3(C++)实现串口通信	454
5.20.1 RS-232C(DB-9 型号)的引脚定义	455
5.20.2 串口的具体操作	455
5.20.3 Kylix3 中 *.inf 文件的使用	459
5.21 基于 Linux 的串行通信类的封装及其应用	462
5.21.1 终端控制函数介绍	462
5.21.2 Qt 串口类的封装	463
5.21.3 具体应用	466
5.21.4 结束语	467
5.22 基于 Linux 的并口驱动程序	467
5.22.1 模型	468
5.22.2 并口硬件	468
5.22.3 原理	469
5.22.4 并口驱动	470
5.22.5 结束语	474
5.23 Linux 套接字编程及其在进程间通信中的应用	474
5.23.1 套接字应用中的基本概念	474
5.23.2 基本进程间通信实例的实现	476
5.23.3 程序代码及注释	478
5.23.4 结束语	481
5.24 基于 Linux 的远程过程调用	481
5.24.1 引言	481
5.24.2 RPC 的标识	481

5.24.3	RPC 机制	481
5.24.4	构造 RPC 应用	482
5.24.5	结束语	487

第 1 章

Linux 内核剖析及其编程

通过编程实例清晰地剖析了 Linux 操作系统内存管理、物理内存的分配与回收、页面周转与缓冲区、内核的进程创建和执行、内核的进程调度、管道与信号等内核模块编程及对内核系统结构作了详细的解析；展示了 Linux 系统内核及其应用编程方法和技巧。

1.1 Linux 内存管理中进程线性区的管理

本节将剖析 Linux 内存管理中进程线性区管理的算法和实现。

1.1.1 概述

根据实际的使用情况，Linux 进程的线性地址空间可以分为多个线性区，每个线性区结构记载着该块区域相应的使用情况。本节就 Linux 线性区管理的算法和源代码的分析，探究其高效的奥秘。

1.1.2 线性空间的管理

1. 线性空间的概念

线性空间就是虚拟地址空间。Linux 内核从一开始就是支持 i386 的 32 位系统，因此进程可以寻址 0 ~ 4GB 的虚拟内存，这是进程的线性地址空间。操作系统对地址空间作了划分，Windows 2000 的 0 ~ 2GB 被划分为用户空间，3 ~ 4GB 是核心空间；而 Linux 的 0 ~ 3GB 是用户空间，最高的 1GB 作为核心空间。用户进程无法直接访问核心空间，否则会发生页保护异常；此外，访问不存在的页或写只读页也会发生页异常。

2. 线性区的数据结构

根据不同的使用需要，进程的 0 ~ 3GB 用户地址空间被划分为不同的线性区。线性区是一块地址连续的区域，具有相同的保护模式和标志，并且映射同一个文件。例如一个线

性区是用户程序段空间，另一个线性区是内存堆空间。

进程的线性区由 `vm_area_struct` 结构表示，并且按照地址从小到大的顺序组成了链表。线性区结构定义如下：

```
struct vm_area_struct {
    struct mm_struct * vm_mm;
    // 指向进程的内存管理结构 mm_struct
    unsigned long vm_start;
    // 线性区的起始地址
    unsigned long vm_end;
    // 线性区的结束地址
    struct vm_area_struct * vm_next;
    // 指向链表中的下一个线性区
    pgprot_t vm_page_prot;
    // 线性区中的页保护位
    unsigned long vm_flags;
    // 线性区的标志 (线性区特性, 读写权限等)
    short vm_avl_height;
    // 在 AVL 中, 以该线性区为根节点的子树高度
    struct vm_area_struct * vm_avl_left;
    // 指向在 AVL 树中的左子...
    struct vm_area_struct * vm_avl_right;
    // 指向在 AVL 树中的右子...
    struct vm_area_struct * vm_next_share;
    // 指向在共享链表中的下一个线性区
    struct vm_area_struct ** vm_pprev_share;
    struct vm_operations_struct * vm_ops;
    // 操作函数指针集的结构
    unsigned long vm_pgoff;
    struct file * vm_file;
    // 线性区映射文件时, 指向文件结构
    unsigned long vm_raend;
    void * vm_private_data;
};
```

其中，`vm_mm` 是指向进程的内存管理结构的指针，`vm_start` 和 `vm_end` 分别是该区域的起始和终止地址，`vm_next` 指向链表中的下一个结构，`vm_page_prot` 是该区域的页保护标志，`vm_ops` 指向一个 `vm_operations_struct` 结构，该结构又包含了一些页处理函数(如缺页的处理)。为了提高链表的搜索效率，`vm_area_struct` 除了被组织成线性链表外，当链表中的结构个数大于 1 个阈值时，还被组织成 AVL 树(高度平衡二叉树)的形式，从而搜索的平均效率可以达到 $\log n$ 。`vm_avl_height`、`vm_avl_left`、`vm_avl_right` 这 3 个域就是用于 AVL 算法的结构。

3. 线性区的搜索和插入操作

1) 操作介绍

线性区的主要操作包括搜索、插入、删除，其中删除操作是插入操作的逆过程。链表结构和 AVL 树是线性区的两种组织形式，当线性区数量 n 不大($n < 32$)时，采用线性链