

## <<UNIX/Linux程序设计教程>>

### 图书基本信息

书名：<<UNIX/Linux程序设计教程>>

13位ISBN编号：9787111403890

10位ISBN编号：7111403894

出版时间：2012-12

出版时间：机械工业出版社

作者：赵克佳 等编著

页数：483

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<UNIX/Linux程序设计教程>>

### 内容概要

《unix/linux程序设计教程》遵循最新的“统一unix规范版本4”，以linux为平台，系统地讲述了unix api各种函数的编程方法。

本书内容包括unix的发展历程与标准、标准i/o和低级i/o、文件与目录操作、进程环境与进程控制、信号处理、时间与定时、终端i/o、高级i/o、进程之间的通信、套接字与网络通信、多线程编程。

《unix/linux程序设计教程》在介绍unix api各种函数的功能和用途的同时，清晰地阐述了它们所隐含的操作系统基本原理。书中给出了大量程序设计示例程序，有助于读者更好地掌握这些函数的功能、使用方法及编程技巧。

《unix/linux程序设计教程》特别适合于高等院校计算机专业的教师、高年级本科生、研究生作为教材和参考书，也特别适合从事计算机系统软件和应用软件开发的工程技术人员作为实用编程手册查阅。

# <<UNIX/Linux程序设计教程>>

## 书籍目录

前言

第1章 unix导论

1.1 unix简史

1.1.1 unix的诞生

1.1.2 unix的早期发展

1.1.3 bsd unix

1.1.4 系统v unix

1.1.5 unix的商业化

1.1.6 linux

1.2 标准

1.2.1 svid

1.2.2 posix

1.2.3 统一unix规范

1.2.4 c标准

1.3 unix基本概念

1.3.1 程序和进程

1.3.2 内核

1.3.3 shell

1.3.4 用户名与用户id、用户组与组id

1.3.5 特权用户

1.3.6 系统调用与库函数

1.4 系统库

1.4.1 头文件

1.4.2 保留字

1.4.3 特征测试宏

1.5 示例程序和编译环境

1.6 错误处理

1.7 系统信息

1.7.1 机器标识

1.7.2 硬件/软件类型识别

1.8 系统能力限制

1.8.1 一般能力限制值

1.8.2 系统和文件特征选项

1.8.3 sysconf()、pathconf()和fpathconf()函数

1.9 思考与练习

第2章 标准输入输出

2.1 unix 输入输出基本概念

2.2 流和file对象

2.3 打开和关闭流

2.4 读和写流

2.4.1 字符i/o

2.4.2 行i/o

2.4.3 读回退

2.4.4 块i/o

2.5 文件定位

## &lt;&lt;UNIX/Linux程序设计教程&gt;&gt;

- 2.6 文件结束和错误指示器
- 2.7 流缓冲
- 2.8 格式i/o
  - 2.8.1 格式输出
  - 2.8.2 格式输入
- 2.9 临时文件
- 2.10 思考与练习
- 第3章 低级输入输出
  - 3.1 文件描述字的打开、创建和关闭
  - 3.2 read()和write()函数
  - 3.3 设置描述字的文件位置
  - 3.4 dup()和dup2()函数
  - 3.5 fdopen()和fileno()函数
  - 3.6 文件控制函数fcntl()
    - 3.6.1 重复文件描述字
    - 3.6.2 文件描述字标签
    - 3.6.3 文件状态标签
  - 3.7 非阻塞i/o
  - 3.8 readv()和writev()函数
  - 3.9 fsync()和fdatasync()函数
  - 3.10 思考与练习
- 第4章 文件与目录
  - 4.1 文件
    - 4.1.1 stat()、fstat()和lstat()函数
  - 4.2 文件类型
    - 4.2.1 普通文件
    - 4.2.2 目录
    - 4.2.3 链接与link()函数
    - 4.2.4 符号链接与symlink()和readlink()函数
    - 4.2.5 特别文件
    - 4.2.6 测试文件的类型
  - 4.3 文件的属主和用户组
    - 4.3.1 chown()、fchown()和lchown()函数
  - 4.4 文件方式
    - 4.4.1 文件访问权限
    - 4.4.2 调整用户id和调整组id
    - 4.4.3 sticky位
    - 4.4.4 文件方式位小结
  - 4.5 确定和改变文件方式
    - 4.5.1 umask()函数
    - 4.5.2 chmod()和fchmod()函数
    - 4.5.3 access()函数
  - 4.6 文件大小
    - 4.6.1 截断文件
  - 4.7 文件时间
    - 4.7.1 utime()和utimes()函数
  - 4.8 文件的删除与换名

## <<UNIX/Linux程序设计教程>>

- 4.8.1 删除文件和目录
- 4.8.2 文件换名
- 4.9 目录操作
  - 4.9.1 工作目录
  - 4.9.2 创建目录
  - 4.9.3 读目录流
  - 4.9.4 对目录流的随机访问
- 4.10 思考与练习
- 第5章 进程环境
  - 5.1 main()函数
  - 5.2 命令行参数
    - 5.2.1 命令行参数的语法约定
    - 5.2.2 扫描命令行中的选项
  - 5.3 环境变量
    - 5.3.1 环境表
    - 5.3.2 访问环境
  - 5.4 终止进程
    - 5.4.1 出口状态
    - 5.4.2 终止前的清理
    - 5.4.3 流产程序
  - 5.5 进程的存储空间
    - 5.5.1 进程的地址空间
    - 5.5.2 动态存储分配与释放
    - 5.5.3 释放分配的存储单元
  - 5.6 setjmp()和longjmp()函数
  - 5.7 进程资源
    - 5.7.1 查看与设置资源限制
    - 5.7.2 资源使用统计
  - 5.8 用户信息
    - 5.8.1 用户名
    - 5.8.2 用户数据库
    - 5.8.3 组数据库
  - 5.9 进程的身份凭证
  - 5.10 调整进程的身份
  - 5.11 思考与练习
- 第6章 进程控制
  - 6.1 进程标识
  - 6.2 进程创建
  - 6.3 执行一个新程序
  - 6.4 等待进程完成
  - 6.5 进程终止与僵死进程
  - 6.6 system()函数
  - 6.7 进程组
  - 6.8 会晤期
  - 6.9 控制终端
  - 6.10 作业控制
  - 6.11 思考与练习

## &lt;&lt;UNIX/Linux程序设计教程&gt;&gt;

## 第7章 信号处理

## 7.1 信号概念

## 7.2 unix 信号

## 7.2.1 程序错误类信号

## 7.2.2 程序中止类信号

## 7.2.3 闹钟类信号

## 7.2.4 i/o类信号

## 7.2.5 作业控制类信号

## 7.2.6 操作错误类信号

## 7.2.7 其他信号

## 7.3 生成信号

## 7.3.1 raise()函数

## 7.3.2 kill()函数

## 7.4 设置信号的动作

## 7.4.1 signal()函数

## 7.4.2 进程初启时的信号动作

## 7.4.3 不可靠信号

## 7.4.4 sigaction()函数

## 7.5 信号句柄

## 7.5.1 正常返回的信号句柄

## 7.5.2 终止进程的句柄

## 7.6 阻塞信号

## 7.6.1 sigset\_t类型和信号集操作

## 7.6.2 设置信号屏蔽

## 7.6.3 检查悬挂信号

## 7.7 等待信号

## 7.7.1 pause()函数

## 7.7.2 sigsuspend()函数

## 7.8 使用分开的信号栈

## 7.9 信号句柄编程技巧

## 7.9.1 句柄内非局部控制转移

## 7.9.2 可重入函数与异步信号安全函数

## 7.9.3 被信号中断的系统调用

## 7.9.4 原子数据

## 7.10 实时信号

## 7.10.1 sa\_siginfo标志

## 7.10.2 发送实时信号

## 7.10.3 等待实时信号

## 7.11 思考与练习

## 第8章 时间与定时

## 8.1 系统时钟

## 8.1.1 time()函数

## 8.1.2 gettimeofday()函数

## 8.2 时间格式转换

## 8.2.1 分解的日历时间

## 8.2.2 格式化日期与时间

## 8.3 cpu时间与墙钟时间

## <<UNIX/Linux程序设计教程>>

- 8.3.1 clock()函数
- 8.3.2 times()函数
- 8.4 睡眠与定时
  - 8.4.1 sleep()函数
  - 8.4.2 设置定时器
- 8.5 实时时钟与定时
  - 8.5.1 实时时钟
  - 8.5.2 实时睡眠
  - 8.5.3 实时定时器
  - 8.5.4 创建和删除实时定时器
  - 8.5.5 设置实时定时器
  - 8.5.6 定时器超期计数
- 8.6 思考与练习
- 第9章 终端i/o
  - 9.1 需要改变终端设置的例子
  - 9.2 终端i/o概述
    - 9.2.1 终端
    - 9.2.2 串行端口
    - 9.2.3 终端设备文件
    - 9.2.4 输入输出队列
    - 9.2.5 加工和非加工输入方式
  - 9.3 gti控制接口
    - 9.3.1 termios数据结构
    - 9.3.2 gti控制函数
  - 9.4 终端属性
    - 9.4.1 输入方式
    - 9.4.2 输出方式
    - 9.4.3 控制方式
    - 9.4.4 局部方式
    - 9.4.5 特殊字符
    - 9.4.6 stty命令
  - 9.5 终端标识
  - 9.6 改变终端属性
  - 9.7 加工方式与非加工方式输入
    - 9.7.1 加工方式输入
    - 9.7.2 非加工方式输入
  - 9.8 设置波特率
  - 9.9 行控制函数
  - 9.10 串行端口程序设计
  - 9.11 思考与练习
- 第10章 高级i/o
  - 10.1 文件锁
    - 10.1.1 fcntl()文件锁操作
    - 10.1.2 锁的测试、请求和释放
    - 10.1.3 文件锁与进程和文件的关系
    - 10.1.4 死锁
    - 10.1.5 建议锁与强制锁

## &lt;&lt;UNIX/Linux程序设计教程&gt;&gt;

- 10.2 信号驱动的i/o
- 10.3 多路转接i/o
  - 10.3.1 select()函数
  - 10.3.2 poll()函数
- 10.4 异步i/o
  - 10.4.1 异步i/o控制块
  - 10.4.2 i/o完成时的信号交付
  - 10.4.3 异步i/o的优先级
  - 10.4.4 异步i/o函数
  - 10.4.5 异步i/o之例
  - 10.4.6 异步i/o注意事项
- 10.5 存储映射i/o
- 10.6 思考与练习
- 第11章 进程间通信
  - 11.1 管道
    - 11.1.1 创建管道
    - 11.1.2 父子进程间的管道通信
    - 11.1.3 连接标准输入和标准输出的管道
    - 11.1.4 popen()和pclose()函数
    - 11.1.5 管道i/o的原子性
  - 11.2 fifo特别文件
    - 11.2.1 创建fifo
    - 11.2.2 fifo操作
    - 11.2.3 fifo用于客户/服务通信
  - 11.3 系统v ipc
    - 11.3.1 关键字和标识
    - 11.3.2 ipc资源描述结构与成员ipc\_perm
    - 11.3.3 ipcs和ipcrm命令
  - 11.4 消息队列
    - 11.4.1 创建和获得消息队列
    - 11.4.2 消息队列的查询、设置和删除
    - 11.4.3 发送和接收消息
  - 11.5 共享存储段
    - 11.5.1 创建和获得共享存储段
    - 11.5.2 共享存储段的查询、设置和删除
    - 11.5.3 共享存储段的连接和分离
  - 11.6 信号量
    - 11.6.1 创建和获得信号量标识
    - 11.6.2 信号量的查询、设置和删除
    - 11.6.3 信号量操作
  - 11.7 思考与练习
- 第12章 套接字与网络通信
  - 12.1 tcp/ip协议
  - 12.2 套接字
  - 12.3 套接字地址结构
    - 12.3.1 ip地址
    - 12.3.2 域名地址

## &lt;&lt;UNIX/Linux程序设计教程&gt;&gt;

- 12.3.3 服务与端口号
  - 12.3.4 套接字地址数据结构
  - 12.3.5 字节顺序
  - 12.4 命名套接字
  - 12.5 套接字通信模式
  - 12.6 流套接字操作
    - 12.6.1 请求连接
    - 12.6.2 接收连接
    - 12.6.3 getsockname()和getpeername()函数
    - 12.6.4 多客户服务
    - 12.6.5 send()和recv()函数
  - 12.7 套接字选项
  - 12.8 带外数据
    - 12.8.1 tcp带外数据
    - 12.8.2 带外数据的发送和接收
    - 12.8.3 带外数据标志
  - 12.9 数据报套接字操作
    - 12.9.1 sendto()和recvfrom()函数
    - 12.9.2 数据报套接字客户/服务之例
    - 12.9.3 使用connect()函数
  - 12.10 超时处理
  - 12.11 思考与练习
- 第13章 线程
- 13.1 线程概念
    - 13.1.1 什么是线程
    - 13.1.2 线程的好处
    - 13.1.3 pthreads线程
    - 13.1.4 线程标识
  - 13.2 创建线程
  - 13.3 终止线程
    - 13.3.1 等待线程终止
    - 13.3.2 可汇合与分离的线程
  - 13.4 创建特殊属性的线程
  - 13.5 互斥变量
    - 13.5.1 互斥变量的初始化和销毁
    - 13.5.2 互斥变量属性
    - 13.5.3 互斥变量的加锁与解锁
    - 13.5.4 互斥变量与spin锁
  - 13.6 读写锁
    - 13.6.1 读写锁的初始化和销毁
    - 13.6.2 读写锁的上锁与解锁
  - 13.7 条件变量
    - 13.7.1 创建和销毁条件变量
    - 13.7.2 条件变量属性
    - 13.7.3 等待条件变量
    - 13.7.4 唤醒条件变量等待
  - 13.8 思考与练习

## <<UNIX/Linux程序设计教程>>

### 第14章 线程高级特征

#### 14.1 线程专有数据

##### 14.1.1 线程专有数据键的创建和删除

##### 14.1.2 使用线程专有数据

#### 14.2 取消线程

##### 14.2.1 线程的可取消属性

##### 14.2.2 取消线程与取消点

##### 14.2.3 异步取消的安全性

##### 14.2.4 现场清理

#### 14.3 线程调度

##### 14.3.1 线程调度竞争范围

##### 14.3.2 调度策略与优先级

##### 14.3.3 线程调度属性

##### 14.3.4 动态改变线程的调度策略和优先级

#### 14.4 线程与信号

##### 14.4.1 信号动作

##### 14.4.2 信号屏蔽

##### 14.4.3 向线程发送信号

##### 14.4.4 等待信号

##### 14.4.5 一种新的事件通知方法：sigev\_thread

#### 14.5 思考与练习

#### 参考文献

## &lt;&lt;UNIX/Linux程序设计教程&gt;&gt;

## 章节摘录

1.1.3 BSD UNIX 在UNIX V7期间，UNIX已发展成为一个可运行于许多不同处理机上的相当稳定的操作系统。

贝尔实验室广泛使用UNIX，但没有任何技术服务。

由于美国法律的原因，其父公司AT&T也没有提供服务（受1956年美国司法部对AT&T以及Western Electric公司反托拉斯诉讼协议的约束，在该协议的有效期内，禁止AT&T生产任何与电话或电报无关的设备或从事其他非公共载体通信服务的商务）。

加州大学伯克利分校于1974年12月得到了第一个UNIX许可。

在随后的几年中，贝尔实验室研究组的一些成员，包括Ken Thompson，利用休假在那里讲授UNIX，并参加一些研究工作。

加州大学伯克利分校的很多研究生和教授对UNIX系统产生了极大兴趣，其中包括Bill Joy和Chuck Haley。

他们为这个UNIX开发了一些实用程序，包括Pascal编译器和ed编辑器，这个编辑器后来成为著名的vi编辑器。

Bill Joy将这些新增的内容与一些广泛发布的软件集中在一起，打成一个软件包，形成了“Berkeley Software Distribution”（BSD），并在1978年春季将它以每个许可协议50美元售出。

同年早些时候又推出了2BSD。

据说前两个BSD版本只包含应用和实用程序，并没有修改也没有包含UNIX操作系统，第一个包含操作系统的版本是1979年发布的3BSD。

3BSD运行于32位的VAX-11/780，这个版本在内核加入了页式请求和虚拟内存等新功能。

3BSD虚拟内存功能引起了国防高级研究项目局（the Defense Advanced Research Project Agency，DARPA）的注意，他们决定为伯克利UNIX系统的开发提供基金。

DARPA项目的主要目的之一是集成TCP/IP网络协议包。

在DARPA的资助下，UNIX开始蓬勃发展。

1980年产生了后来统称为4BSD的几个版本：4.

0BSD（1980）、4.1BSD（1981）、4.2BSD（1983）、4.3BSD（1986）以及4.

4BSD（1993）。

伯克利UNIX的研究工作是由计算机科学研究组（Computer Science Research Group，CSRG）进行的。

4.4BSD之后，由于UNIX系统变得越来越大，以致很难由一个小组来维护和发展，同时也由于经费问题，CSRG决定不再继续进行UNIX的开发。

伯克利小组对UNIX作出了许多重要性的贡献。

除了虚拟内存和纳入了TCP/IP协议外，BSD UNIX还引入了快速文件系统（Fast File System，FFS）、可靠信号以及套接字功能。

1.1.4 系统V UNOX ..... P3

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>