

# 《UNIX系统编程》

## 图书基本信息

书名：《UNIX系统编程》

13位ISBN编号：9787111161905

10位ISBN编号：7111161904

出版时间：2005-5

出版社：机械工业出版社

作者：罗宾斯

页数：656

译者：陈涓

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu000.com](http://www.tushu000.com)

## 内容概要

本书深入阐述了能使UNIX操作系统发挥出量大功效的软件设计方法，对UNIX编程的精髓进行了清晰易懂的介绍。两位作者具有多年教学与科研经验，匠心独运，精耕细做，以一些用来说明如何使用系统调用的短小代码段开始，巧妙地过渡到实际项目中，逐渐在拓展了读者的技术水平。本书对通信、并发和多线程问题进行了透彻的研究；对复杂的概念，比如信号和并发，进行了全面、明晰的解释，并以此闻名。本书的特色还在于它提供了大量实例、练习、可重用的代码以及有和于网络通信程序的简化了的库。

本书同时也是一本基于最新的UNIX标准的完备的参考书，它完全覆盖了与文件、信号、信号量、POSIX线程和客户机-服务器通信相关的内容。此外，本版本中新增拉在于Web、UDP和服务器性能的章节也为本书增色不少。书中包含大量有趣的项目设计及其参考答案，例如命令解释程序、WWW重定向、因特网音频广播、服务器性能等，这些项目设计稍加完善就可以成为可用的程序，可供软件开发人员参考。

本书的Web站点<http://usp.cs.utsa.edu/usp>为读者提供了大量的资料，内容包括所有可下载的程序，这些程序可以免费使用。此外，这个站点中还包含到模拟器、测试工具、作者准备的课程资料以及勘误表的链接。

本书是UNIX系统编程的经典教材，并基于最新的UNIX标准进行了更新，其内容完全覆盖文件、信号、POSIX线程、UDP、Web和客户机 - 服务器等相关内容，并对通信、并发和多线程问题进行了透彻的研究，对复杂的概念进行了全面、完善以及明晰的解释，并以此而闻名。本书不仅提供了大量实例和练习，还专门设计了有针对性的项目，并给出了参考答案，帮助读者理解相关的概念，增强应用能力。本书适合作为高等院校计算机专业教材，也可供软件开发人员参考。

# 《UNIX系统编程》

## 作者简介

Kay A. Robbins和Steven Robbins从麻省理工学院获得博士学位。他们现任教于得克萨斯大学圣安东尼奥分校的计算机科学系。

## 书籍目录

### 第一部分 基础知识

第一章 技术对程序的冲击

第二章 程序、进程和线程

第三章 UNIX系统中的进程

第四章 UNIX I/O

第五章 文件和目录

第六章 UNIX特殊文件

第七章 项目：令牌环

### 第二部分 异步事件

第八章 信号

第九章 时间和定时器

第十章 项目：虚拟定时器

第十一章 项目：破解命令解释程序

### 第三部分 并发

第十二章 POSIX线程

第十三章 线程同步

第十四章 临界区和信号量

第十五章 POSIX IPC

第十六章 项目：生产者消费者同步

第十七章 项目：非完全并行虚拟机

### 第四部分 通信

第十八章 面向连接的通信

第十九章 项目：WWW重定向

第二十章 无连接通信和多播

第二十一章 项目：因特网音频广播

第二十二章 项目：服务器性能

前言：本书是1995年Prentice Hall出版的《Practical UNIX Programming：A Guide to Communication，Concurrency and Multithreading》的第2版。为了更好地反映本书的内容，我们修改了书名（英文版）。同上一版相比，除了书名，我们还对许多地方进行了修改。因特网已经成为计算和社会领域中的一个主导因素。我们的个人信息联网了，我们的软件经常会受到攻击。因此，编写正确的代码显得更加重要。在新的版本中，我们尽量去生成能够正确地处理错误和特殊情况的代码。我们知道，只是声称“处理了所有的错误”，却在给出的代码示例中省略掉错误处理，是没有什么说服力的。但是，错误处理使代码变得更加复杂，我们想尽力使代码清晰一些。本书对前一版的另一个重大改进就是采用了单一UNIX规范，在本书中我们称之为POSIX。我们再也不用为选用哪个厂商的库函数而发愁了——现在已有正式的版本了。在编写本书的过程中，我们尽了最大的努力来遵循这个标准。本书与众不同之处是含有练习和项目章。实际上，本书就是从一个项目手册发展而来的，这个项目手册是美国国家科学基金课题（National Science Foundation Grant）的一部分。在完成这个项目的初期开发之后，我们逐渐认识到：完成这个项目所需的资料分散在很多地方——通常可以在那些提供了大量细节但很少进行概念描述的参考书中找到这些资料。这样，这本书就逐步演变成一本基于最新UNIX标准的完备的参考书了。本书分为四个部分，每个部分中都包括一些主题章和项目章。主题章以循序渐进的方式涵盖了特定的资料，其中有很多例子和小练习。本书还针对进程管理、并发和通信过程中的很多基本概念提供了编程练习（即项目章）。这些编程练习和传统科学课程中的实验作用相当。只有通过实践，才能真正理解书中的概念。这些练习的难度由浅入深，很多练习都只要不到100行的代码就可以实现。下表对本书的结构进行了总结。有14个主题章，这些主题章与8个项目章相互独立。第一次通读本书时可以略过项目章。项目章通过开发一个扩展的应用将几个主题章中的资料整合起来。这些项目包含两个层面：除了说明编程思想之外，还引导读者去理解与应用相关的更高级的主题。这些项目都是分阶段设计的，大多数完整的实现都只需要几百行代码。由于不需要编写大量的代码，因此读者可以将注意力集中在对概念的理解而不是代码的调试上。为了简化编程，我们提供了网络通信和日志输出的库。对专业的程序员来说，主题章结尾处的练习提供了基本的练习内容。通常，使用本书的教师可以在一个学期的课程中挑选几个练习和一个项目章让学生们实现。每个项目都有很多变化，因此这些项目可以在多个学期中反复使用。读者可以选择不同的方式来读这本书。第一部分中的主题章是学习本书其他部分的基础。读完第一部分的主题章之后，读者就可以按任意的顺序来学习第二部分到第四部分了。但后继各章结尾处有关交互（例如，线程是怎样与信号交互的）的讨论除外。我们假定读者是一名很好的C程序员，尽管不一定是UNIX C的程序员。读者应该熟悉C语言编程和基本的数据结构。如果你刚刚接触UNIX，附录A中给出了程序开发的一些必备知识。本书中包括标准函数的概要，在“概要”的右下角列出了用来说明函数的相关标准。知识的更新如此之快，然而书的内容却是有限的。我们欢迎读者给我们提出意见和建议。读者可以给我们发电子邮件authors@usp.cs.utsa.edu。我们尽了最大的努力来编写一本无差错的书。但是，如果你是第一个向我们指出某个错误的人，我们会在本书的网站上向你表示诚挚的谢意。我们在Web网站<http://usp.cs.utsa.edu/usp>上提供了与本书相关的信息。本书包含的所有代码都可以从这个Web站点上下载。译者序：本书是一本介绍UNIX系统的工作原理以及学习UNIX系统程序编写方法的经典教材。对于程序员来说，本书同样是一本不可多得的案头必备参考书。读者在掌握了操作系统、C语言以及计算机网络的基本知识后学习本书，会发现本书将使读者对UNIX系统的理解登上一个新的高度，并将读者编写多线程和网络通信程序的能力提高到一个新的境界。本书有以下几个突出特点：其一，在讲述基本概念时，本书围绕主题设计了大量实例程序和练习来加深读者对概念的理解；其二，在各部分的最后，作者精心设计了生动有趣的项目来总结各部分的主要内容。这些项目，例如破解命令解释程序、WWW重定向、因特网音频广播、服务器性能等的取材都有很强的实际应用背景，稍加完善就是一个可用的程序；其三，本书采用了最新的UNIX标准：Single UNIX Specification，并在书中加入了大量的最新信息，如火星探路者中的优先级反转缺陷等。另外，本书各章末尾处的扩展阅读也给读者指出了相关领域经典的或最新的参考资料并进行了细致的评价，对读者进一步研究提供了很好的指导。如果您希望了解并掌握操作系统接口及其工作原理，我们相信本书将是一本合适的教材及参考书；如果您已经具备了初步的UNIX程序编写能力，本书将会引领您进入系统程序编写者的行列；如果您已经是一个熟练的系统程序员了，本书丰富的内容实例、充实的知识架构也一定会让您对计算机系统的理解有新的升华。本书的第20、21、22章由赵振平翻译

## 《UNIX系统编程》

，其他部分由陈涓翻译。郭传雄和赵振平校阅了全书。致谢 我们非常感谢Mike Speciner和Bob Lynch，他们通读了全部手稿，并提出了很多有益的建议。我们尤其要对Mary Lou Nohr细心睿智的编辑工作表示感谢。我们还要对Neal Wagner和Radia Perlman给予的鼓励和建议表示感谢。从1988年至今（2003年），我们为大学本科生和研究生开设了操作系统课程，本书中很多资料都曾作为这些教学课程的一部分使用。学习此课程的学生们都经历了书稿发展的不同阶段，并对不断出现的项目进行了实际的测试。他们提供的程序故障、注释、抱怨和建议都使本书更加出色，并且使我们对这些主题之间的相互联系有了更深刻的认识。在早期书稿中发现错误的学生有Joseph Bell、Carlos Cardenas、Igor Grinshpan、Jason Jendrusch和James Manion。我们要感谢美国国家科学基金会通过NSF-ILI授权的USE-0950497对我们组建实验室提供的支持，这样我们才有机会去开展最初的课程，而这些课程正是本书的基础。NSF（DUE-975093、DUE-9752165和DUE-0088769）还为开发那些用来探索和分析操作系统概念的工具提供了支持。我们要感谢Prentice Hall的编辑Greg Doench在整个过程中为我们提供的指导，还要感谢我们的出版编辑William Mara使此书得以出版。我们用对本书进行了排版，的制作者提供了可以免费使用的软件，我们还要向他们表达我们的谢意。我们特别要感谢我们的家人，他们给了我们无限的爱和支持，尤其要感谢我们的孩子Nicole和Thomas，他们对这项艰巨的任务表示了热情和理解。

## 精彩短评

- 1、很好，配合APUE一起看
- 2、大学读了3年的书
- 3、这是一本学习Linux知识不可多得的好书!
- 4、确实有点难啊
- 5、目前我还没有认真看这本书因为手头的事情还没有做完大概暑假为会系统看书！不过听很多用过的朋友介绍说这是很好大一本经典书籍，学习linux编程是很好大书
- 6、还不错
- 7、 This book stands from the view of Practical Programming. There are lots of program examples to illustrate the system call. You can reuse the program directly whenever you need it.
- 8、这本书先从基础讲起，然后慢慢地由浅入深，特别讲解原理。这本书代码实例特别多，而且都是一些很实用的例子，非常好，对代码的讲解也是非常的精彩。严谨也是本书的一大特色，例题中的代码有极高的健壮性，非常正规。这点非常值得我们学习。问答式也是本书的一大特色，书中提的问题有很多正是读者的疑惑点，更甚有些问题读者并没有想到，但却很重要。问题的后面都为读者一一解答，更令读者欢喜的是，有时针对一个知识点所提出的问题层层深入，解答得非常精彩。里面还会提供专题项目，包括项目的介绍，分析，解答。项目代码可以在网上下载，代码非常正规，有时可以直接运用于实际项目中。本书还一大特色就是附有图表解释，一目了然！每章节的最后还为读者推荐一些额外的资料。当然还有很多亮点了，我就不一一说了，总之，买这本书觉对让你有“物超所值”的感觉！
- 9、七八年初学unix编程时，买的第一本书就是它。当然那时还是第一版，名字也不叫unix system programming, 而是Practical UNIX programming, 中文版叫实用unix编程。对比这两版，可以清晰地看到这些年来技术的飞速发展，同时也可以明白无论技术如何发展，其实核心的东西总是一脉相承。总之，个人以为单纯就unix编程而言，先读R.Steven的unix高级编程，再读Robins的系统编程，基本上就没问题了。
- 10、一本不错的介绍UNIX下使用C语言编程的书籍，很全面。
- 11、.....我就是想抄抄代码什么的
- 12、 都是关于系统编程的各个方面的具体的讲解。先介绍函数，然后给出的代码实例。每隔一段就给出用前面所学可以做到的一个实用小项目（这是有点难度的，觉得没难度的大概有过一些编程经验了）。  
我觉得最好读者有一些关于操作系统的基础知识，比如看过操作系统原理之类的书籍，还有相应的C语言基础  
本人看过之后觉得获益良多
- 13、比较详细的讲解POSIX接口，及多线程，并行等东西，另一本高级环境编程也不错
- 14、n年了，终于读的差不多了
- 15、学习unix的宝典
- 16、绝对的UNIX系统编程经典，很好的介绍了UNIX下的各种概念，书中的代码可以几乎不经任何修改的嵌入到自己的系统中。
- 17、有的代码纯属扯淡
- 18、可以与APUE媲美的书，最好两本参阅着看，边做点相关的小项目是最好的方式。作为手边的参阅手册来说也是不错的选择。
- 19、标记以后在读！
- 20、重量级的书，很好。
- 21、你牛逼你牛逼你牛逼你牛逼你牛逼你牛逼你牛逼你牛逼你牛逼你牛逼你牛逼你牛逼你牛逼
- 22、我以前在学校时在图书馆里看过这本书，但由于时间的原因未能细看。我觉得挺好的就买了一本仔细看看。
- 23、大而全，可以当reference用。要学习还是看UNIX环境高级编程
- 24、很好，如果你想将你的C和系统知识系统学习的话
- 25、工作之余以及在枯燥的旅途中，抱着这本大厚书看。很酷的样子，点点滴滴就把UNIX系统下的C

语言编程给消化了，呵呵



## 章节试读

## 1、《UNIX系统编程》的笔记-第132页

```

char bufin[10]="empty";
char bufout[] = "hello";
int bytesin;
pid_t childpid;
int fd[2];

if(pipe(fd)==-1)
/*error processing..*/

bytesin = strlen(bufin);
childpid = fork();

if(childpid == -1)
/*error processing..*/

if(childpid) /*父进程*/
    write(fd[1],bufout,strlen(bufout)+1);
else
    bytesin = read(fd[0],bufin,10);

fprintf(stderr,"%ld]: my bufin is {%. *s},my bufout is {%s}\n",
        (long) getpid(),bytesin,bufin,bufout);
return 0;

```

父进程在创建子进程前创建了一个管道，然后往管道里写个字符串，并向标准错误里输出一条消息。子进程从管道中读入，然后也打印到标准错误。

## 2、《UNIX系统编程》的笔记-第68页

```

int copyfile(int fromfd,int tofd){
    char *bp;
    char buf[BLKSIZE];
    int bytesread,byteswritten;
    int totalbytes = 0;
    for(;;){
        while(((bytesread = read(fromfd,buf,BLKSIZE))==-1 &&&
            (errno == EINTR)); //处理由于信号产生的中断
        if(bytesread &lt;= 0) break;
        bp = buf;
        while(bytesread>0){
            while((byteswritten = write(tofd,bp,bytesread))==-1) &&& (errno == EINTR));
            if(byteswritten &lt;= 0) break;
            totalbytes += byteswritten;
            bytesread -= byteswritten;
        }
    }
}

```

```
bp += byteswritten;  
}  
if(byteswritten == -1)break;  
  
}  
return totalbytes;  
}
```

## 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:[www.tushu000.com](http://www.tushu000.com)