

<<理解Unix进程>>

图书基本信息

书名：<<理解Unix进程>>

13位ISBN编号：9787115316899

10位ISBN编号：7115316899

出版时间：2013-6

出版时间：人民邮电出版社

作者：Jesse Storimer

译者：门佳

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>

<<理解Unix进程>>

内容概要

《理解Unix进程》从Unix编程的基础概念着手，采用循序渐进的方法，详细介绍了Unix进程的内部工作原理。

《理解Unix进程》提供的许多简单而强大的技术，能够帮助Web开发人员深入了解Unix系统的并发性、守护进程、生成进程(spawning process)与信号等。

同时，读者也可以使用这些技术和方法编写并调试自己的服务器。

此外，《理解Unix进程》附录部分也涉及了一些流行的Ruby项目，让读者进一步了解如何巧妙运用Unix进程。

《理解Unix进程》适合Unix程序员、Web开发人员阅读。

<<理解Unix进程>>

作者简介

Jesse Storimer 高级软件工程师与自出版作家，服务于美国电子商务方案提供商Shopify。

痴迷于编程，编程之余喜欢阅读、园艺以及徒步旅行等户外运动。

除了经常写技术博客文章外，他还著有Working with TCP Sockets, Working with Ruby Threads两本颇有影响力的电子书。

<<理解Unix进程>>

书籍目录

第1章 引言 1 第2章 基础知识 3 2.1 干嘛要在意？
 3 2.2 驾驭神力！
 4 2.3 概述 4 2.4 系统调用 5 2.5 命名法，wtf（2） 6 2.6 进程：Unix之本 7 第3章 进程皆有标识 9 3.1 交叉参考 9 3.2 实践领域 10 3.3 系统调用 10 第4章 进程皆有父 12 4.1 交叉参考 12 4.2 实践领域 13 4.3 系统调用 13 第5章 进程皆有文件描述符 14 5.1 万物皆为文件 14 5.2 描述符代表资源 14 5.3 标准流 17 5.4 实践领域 18 5.5 系统调用 18 第6章 进程皆有资源限制 19 6.1 找出限制 19 6.2 软限制与硬限制 20 6.3 提高软限制 20 6.4 超出限制 21 6.5 其他资源 22 6.6 实践领域 22 6.7 系统调用 23 第7章 进程皆有环境 24 7.1 这是个散列吗？
 25 7.2 实践领域 25 7.3 系统调用 26 第8章 进程皆有参数 27 8.1 这是个数组！
 27 8.2 实践领域 28 第9章 进程皆有名 29 9.1 进程命名 29 9.2 实践领域 30 第10章 进程皆有退出码 31 第11章 进程皆可衍生 34 11.1 Luke，使用fork（2） 34 11.2 多核编程？
 37 11.3 使用block 38 11.4 实践领域 38 11.5 系统调用 38 第12章 孤儿进程 39 12.1 失控 39 12.2 弃子 40 12.3 管理孤儿 40 第13章 友好的进程 41 13.1 对CoW好点 41 13.2 MRI/RBX用户 43 第14章 进程可待 44 14.1 看顾（Babysitting） 45 14.2 Process.wait一家子 46 14.3 使用Process.wait2进行通信 46 14.4 等待特定的子进程 48 14.5 竞争条件 49 14.6 实践领域 50 14.7 系统调用 51 第15章 僵尸进程 52 15.1 等待终有果 52 15.2 僵尸长什么样子？
 53 15.3 实践领域 54 15.4 系统调用 54 第16章 进程皆可获得信号 55 16.1 捕获SIGCHLD 55 16.2 SIGCHLD与并发 56 16.3 信号入门 59 16.4 信号来自何方？
 59 16.5 信号一览 61 16.6 重定义信号 62 16.7 忽略信号 63 16.8 信号处理程序是全局性的 64 16.9 恰当地重定义信号处理程序 64 16.10 何时接收不到信号？
 66 16.11 实践领域 66 16.12 系统调用 67 第17章 进程皆可互通 68 17.1 我们的第一个管道 68 17.2 管道是单向的 70 17.3 共享管道 70 17.4 流与消息 72 17.5 远程IPC？
 74 17.6 实践领域 74 17.7 系统调用 74 第18章 守护进程 75 18.1 首个进程 75 18.2 创建第一个守护进程 76 18.3 深入Rack 76 18.4 逐步将进程变成守护进程 77 18.5 进程组和会话组 78 18.6 实践领域 82 18.7 系统调用 83 第19章 生成终端进程 84 19.1 fork+exec 84 19.2 exec的参数 86 19.3 实践领域 90 19.4 系统调用 91 第20章 尾声 92 20.1 抽象 92 20.2 通信 93 20.3 再会，而非永别 93 附录A Resque如何管理进程 95 附录B Unicorn如何收割工作进程 100 附录C preforking服务器 106 附录D Spyglass 113 索引 116

<<理解Unix进程>>

章节摘录

版权页： 11.3使用block 在上面的例子中，我们使用if/else结构演示了fork的用法。

其实也可以通过block来使用fork，这种方法在Ruby中更常见。

如果你将一个block传递给fork方法，那么这个block将在新的子进程中执行，而父进程则会跳过block中的内容。

子进程执行完block之后就会退出，它并不会像父进程那样执行随后的代码。

fork do #此处的代码仅在子进程中执行 end #此处的代码仅在父进程中执行 11.4实践领域 请参看附录或Spyglass项目来学习fork (2) 的实践用法。

11.5系统调用 Ruby的Kernel#fork对应于fork (2) 。

第12章 孤儿进程 12.1 失控 在运行上一章中的例程时你可能就已经注意到了，如果涉及子进程，就不能再像我们已经习惯的那样从终端来控制一切。

当通过终端启动单个进程时，通常只有这个进程向STDOUT写入，从键盘获取输入或是侦听Ctrl—C以待退出。

一旦进程衍生出了子进程，这一切就变得不那么简单了。

如果你按下Ctrl—C，哪一个进程应该退出？

是全部退出还是只有父进程退出？

了解这些问题还是有好处的，因为实际上很容易创建孤儿进程。

```
fork do 5.times do sleep 1 puts " I ' m an orphan !
```

```
" end end abort " Parent process died..." 如果从终端运行这个程序，你会注意到父进程结束后，立刻返回到终端命令提示符下，此时终端被子进程输出到STDOUT的内容所重写！
```

在进行进程衍生的时候，就会发生这些莫名其妙的事。

12.2弃子 当父进程结束之后，子进程会怎样？

简短的回答就是“安然无恙”。

也就是说，操作系统并不会对子进程区别对待。

因此父进程结束后，子进程照常继续运行。

父进程可不会带着子进程同归于尽。

12.3管理孤儿 那么还能够对孤儿进程进行管理吗？

对于这个问题我们接触得有点早了，不过这涉及两个有意思的概念。

第一个是守护进程。

守护进程是一种长期运行的进程，为了能够一直保持运行，它们有意作为孤儿进程存在。

在后面我们会对其进行详述。

另一个是与脱离终端会话的进程进行通信。

你可以使用Unix信号来做到这一点。

同样我们会在随后的章节中详述。

接下来我们很快就会讲述如何正确管理和控制子进程。

<<理解Unix进程>>

编辑推荐

《理解Unix进程》是唯一一本针对Web开发人员的Unix编程书籍，让你无需借助C语言即可玩转Unix进程，教给你如何自主编写并调试高效服务器。
教你游刃有余的利用系统编程技术，许多WEB开发人员或者对UNIX系统的基本原理知之甚少，理解unix进程可弥补这方面的缺憾，帮你从底层了解unix系统的进程机制。

<<理解Unix进程>>

名人推荐

“《理解Unix进程》让我意识到Unix编程是多么简单而强大，而我对Unix编程的了解又是何其之少！在下一个版本的Thin的设计中，这本书无疑会对我有很大的启发。

” ——Marc-André Cournoyer，创建了Thin Web服务器 “《理解Unix进程》这本书填补了当今许多开发人员的空白。

掌握Unix的基本原理无需编写C代码。

本书就说明了这一点。

” ——David Bryant Copeland，著有Build Awesome Command-Line Applications in Ruby

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>