

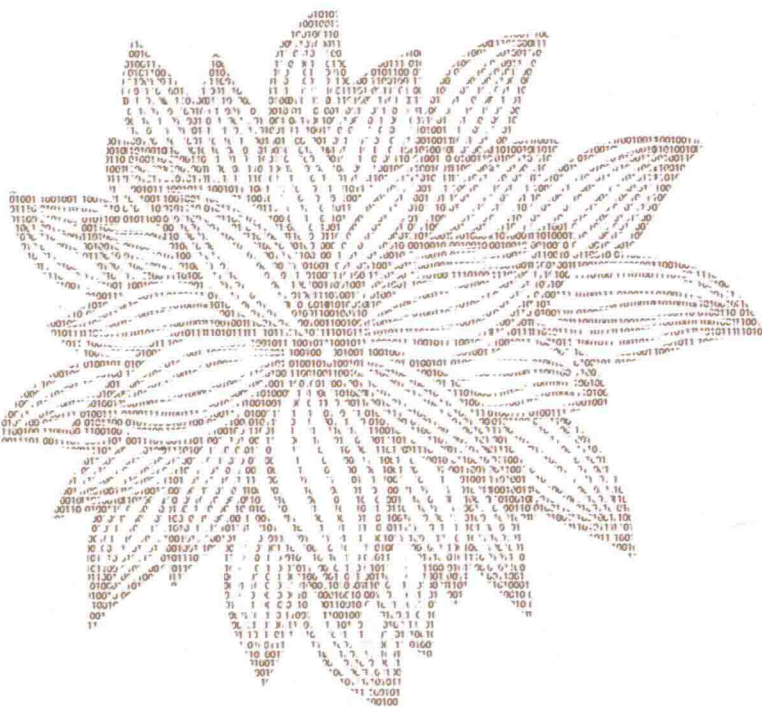
TURING

图灵程序  
设计丛书

# Unix内核源码剖析

超级计算机“京”【日】青柳隆宏 著  
的L1缓存设计者

殷中翔 译 黄炎 周金杰 审



进程 中断 块I/O系统 文件系统 字符I/O系统 启动系统

精读1万行代码，深入理解操作系统原理

 人民邮电出版社  
POSTS & TELECOM PRESS

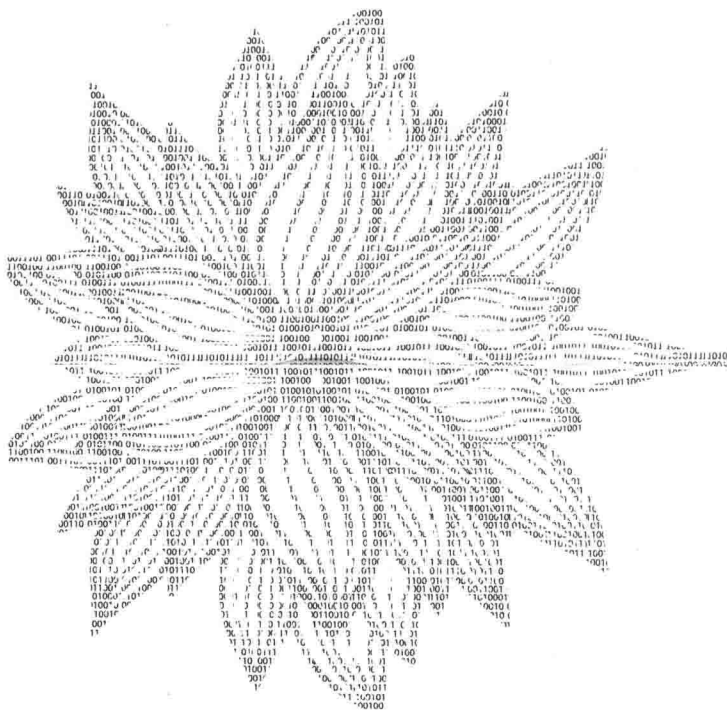
TURING

图灵程序  
设计丛书

# Unix内核源码剖析

【日】青柳隆宏 著

殷中翔 译 黄炎 周金杰 审



人民邮电出版社  
北京

## 图书在版编目(CIP)数据

Unix 内核源码剖析 / (日) 青柳隆宏著; 殷中翔译  
--北京: 人民邮电出版社, 2014.3  
(图灵程序设计丛书)  
ISBN 978-7-115-34521-9

I. ①U… II. ①青… ②殷… III. ①UNIX 操作系统—  
研究 IV. ①TP316.81

中国版本图书馆CIP数据核字(2014)第015530号

*HAJIMETE NO OS CODE READING* by Takahiro Aoyagi

Copyright © 2013 Takahiro Aoyagi

All rights reserved.

Original Japanese edition published by Gijyutsu-Hyoron Co., Ltd., Tokyo

This Simplified Chinese language edition published by arrangement with  
Gijyutsu-Hyoron Co., Ltd., Tokyo in care of Tuttle-Mori Agency, Inc., Tokyo

本书中文简体字版由 Gijyutsu-Hyoron Co., Ltd. 授权人民邮电出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

本书是一本 UNIX V6 内核源代码的阅读指南。作者结合 UNIX V6 已公开的相关文档, 对 UNIX V6 的内核源码进行详细剖析, 旨在让读者更深入地理解进程、中断、块 I/O 系统、文件系统、字符 I/O 系统、启动系统等操作系统的基本原理。

本书适合操作系统的初中学习者阅读, 特别适合通过大学课程和其他入门书对操作系统有所了解, 但是对具体细节缺乏深入理解的读者, 以及那些对操作系统的具体实现有兴趣的读者。

- 
- ◆ 著 [日] 青柳隆宏  
译 殷中翔  
审 黄炎 周金杰  
责任编辑 乐馨  
执行编辑 金松月  
责任印制 焦志炜
- ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号  
邮编 100164 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
北京天宇星印刷厂印刷
- ◆ 开本: 800×1000 1/16  
印张: 23  
字数: 433 千字 2014 年 3 月第 1 版  
印数: 1-4000 册 2014 年 3 月北京第 1 次印刷
- 著作权合同登记号 图字: 01-2013-4378 号

---

定价: 79.00 元

读者服务热线: (010)51095186 转 600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

# 译者序

本书作为学习 UNIX V6 内核源码的参考读物，值得向读者推荐。

国内计算机方面的翻译书籍以欧美作者的著作为主，这与相关的技术多起源于欧美有很大关系。在实际的学习和工作中，我也接触了许多日文书籍，我认为此领域的日文书籍也很有特色。因为日文书籍很擅长从读者的角度出发，配合大量的实例与图表，深入浅出地将一个复杂的问题讲解清楚，非常适合初学者或者工程技术人员阅读。

正如作者在前言中写到的，这本书是基于作者在学习 UNIX V6 内核源码过程中的笔记和博客而写成的，因此对初次接触 UNIX V6 内核的读者而言，无疑是一本很好的参考读物。

但是也应该注意到，正是因为这一点，一些作者已掌握的知识点，书中并未详细阐述。因此，将本书定位于参考书，同时配合附录中的其他参考资料，再借助模拟器 simh 实际动手操作，这样三管齐下才能真正达到理解 UNIX V6 内核源码的目的。

我一直很想将优秀的计算机日文书籍介绍给国内的广大读者。因为工作繁忙，迟迟没有付诸实际行动，直到今年才在图灵公司的大力支持下实现。在这里特别向本书的编辑表示衷心的感谢，他们从译文的逻辑表达和语法等方面提出了宝贵的意见。

同时，也要向原著作者青柳隆宏先生致敬。他的语言表达非常严谨，条理十分鲜明。这使得原著在日本亚马逊网站中获得了 4 星的高分。

最后，我要向家人表示谢意。这本书基本都是利用平时下班后和周末的时间来完成的，没有父母、妻子及两个可爱的孩子的支持，我很难顺利、按时完成翻译工作。

本书的翻译肯定会有许多不足，欢迎广大读者提出宝贵意见或来信交流。我的邮箱地址是 [unixv6.yinzx@gmail.com](mailto:unixv6.yinzx@gmail.com)，我会尽可能给与答复。

如果本书能给大家提供一些帮助的话，我会感到十分欣慰。

殷中翔

2013 年 10 月于福冈

# 前言

本书针对 1975 年由贝尔实验室<sup>①</sup>发布的 UNIX 第 6 版 (Sixth Edition Unix, 此后简称为 UNIX V6) 的内核源代码进行解说。面向的读者主要是计算机专业的学生, 以及从事计算机相关行业的具有初中级水平的技术人员。

考虑到一部分读者会有诸如“我对内核源代码根本不感兴趣”或者“与这种老古董相比, 我喜欢更现代的操作系统”等看法, 笔者想先阐述一下阅读内核源代码的引人入胜之处, 然后再解释 UNIX V6 为何适合初次接触内核源代码的读者。

## 阅读内核源代码的意义

我们可以将操作系统 (OS, Operating System) 看做是一种软件 (集合), 它对包括硬件和软件在内的计算机系统的各个部分进行管理, 并为用户提供便于使用的操作界面。内核作为操作系统的核心部分, 提供计算机系统必备的功能, 因此也被称为狭义的操作系统。例如, shell 之类的程序通常不是内核的一部分, 而是利用内核提供的功能来实现的。内核以外的程序通常被称为用户空间 (userland) 程序, 或用户程序。

通过阅读并理解内核源代码, 我们会有如下收获。

### 对计算机系统的全貌有更深入的了解

掌握了作为计算机系统核心部分的内核, 不仅对操作系统, 对计算机的全貌也会有更为深入的认识。对通过大学课程或其他途径学习的各种领域、各个层面的知识之间的关联性也会有更清晰的认识, 让人有醍醐灌顶的感觉。

### 让操作计算机成为一种令人愉快的体验

理解了计算机系统的全貌, 操作计算机本身也会变得更加令人愉快。比如, 在计算机上执行某个程序的时候, 如果能够准确把握系统内部所进行的操作, 是不是一件很令人兴奋的事情呢? 这种体验将加深读者对计算机的兴趣, 使读者更有动力去提高自己的技术水平。

<sup>①</sup> 由美国的 AT&T 公司和 Western Electric 公司于 1925 年设立的研究开发机构。

## 加深对知识的理解

阅读代码与否，对知识的理解程度会有云泥之差。如果只学习了概要，既容易遗忘也难以应用。相反，理解代码能够使你对学到的算法和思路举一反三，使之成为可以受用一生的财富。

## 提升技术人员自身的水平

作为计算机行业的技术人员，阅读并理解了内核源代码有助于在专业领域里将自己提升到一个新的层次。尽管在全球范围内这个领域的从业者不断增加，但是在了解应用层面的同时，对操作系统等底层的知识也有所了解，并且能够对系统做出整体优化的技术人员，仍是凤毛麟角。

但是恰恰是具备这种素质的人，才能在第一线发挥不可替代的作用。如果想拉大与竞争对手的差距，是必须理解系统内核的。

## 为何选择 UNIX V6

接下来想说明一下为什么不选择最新的操作系统，而将历史比较悠久的 UNIX V6 作为本书的题材。

### 代码行数约为 1 万行

UNIX V6 的内核源代码包括设备驱动程序在内约有 1 万行，这个数量的源代码，初学者是能够充分理解的。有一种说法是一个人所能理解的代码量上限为 1 万行，UNIX V6 的内核源代码从数量上看正好在这个范围之内。看到这里，大家是不是也有“如果只有 1 万行的话没准儿我也能学会”的想法呢？

另一方面，最近的操作系统，例如 Linux 最新版的内核源代码据说超过了 1000 万行<sup>①</sup>。就算不是初学者，想完全理解全部代码基本上也是不可能的。

### 充实的资料

UNIX V6 的用户手册、相关资料和论文都可以在网上找到。运行 UNIX V6 所需的处理装置 PDP-11 以及周边设备的设计文档，很大一部分也可以检索到。

<sup>①</sup> <http://www.h-online.com/open/features/Kernel-Log-15-000-000-lines-of-code-3-0-promoted-to-long-term-kernel-1408062.html>。

另外，有一本关于UNIX V6的指南已经问世多年。此书名为《莱昂氏UNIX源代码分析》(*Lions' Commentary on UNIX*，机械工业出版社出版)，由澳大利亚新南威尔士大学的约翰·莱昂(John Lions)教授撰写，也被称为Lions书(Lions Book)<sup>[3]①</sup>。这本书由于UNIX的版权问题，据说当时只能在学生之间私下传看，后来才广泛地流传开来，现在可以从网上免费下载<sup>[4]</sup>。Lions书对本书中基本没有涉及的PDP-11/40的汇编器及汇编代码也做了部分说明，与本书配合使用会有更好的效果。<sup>②</sup>

## 网罗了操作系统的基本功能

UNIX V6虽然比较老，但是它实现了构成操作系统的大部分基本功能。目前最新的操作系统大部分都是以它为基础发展而来的，因此以UNIX V6为入门教材对我们了解最新的操作系统来说会很有帮助。

线程、网络、GUI、多核支持、虚拟机等这些UNIX V6不具备的功能在近些年操作系统中得以实现。这些功能当中有很多其实是以UNIX V6实现的功能为基础的。

## 简化的设计

UNIX V6作为一种早期的操作系统，功能实现比较简单。而最新的操作系统要顾及更多的问题，同时也要考虑到性能的优化，因此实现也更为复杂。如果是首次阅读内核源代码，用相对简单的UNIX V6更合适。

## 便于读者对系统有完整的了解

前面已经说过，从代码量上看，通读UNIX V6内核源代码对个人来说是可以做到的。如果更进一步，对系统内置的用户程序集(如shell)的代码或是周边设备的设计文档也有所涉猎的话，就会对包括内核在内的计算机系统整体有更深入细致的了解。上述系统内置的用户程序集的代码或设计文档与最新产品相比，在实现上更为简单，也更容易理解。

## 有模拟器可供参考

simh<sup>[7]</sup> 这款模拟器能够模拟包括PDP-11系列的许多处理器，可以用来运行UNIX V6。

① 本书中以[数字]形式标注的部分在书尾附录中有相对应的内容。

② 同时审校者推荐《UNIX操作系统教程》，西安电子科技大学出版社于1985年6月出版，尤晋元主编。这是一本详细剖析UNIX V6机制和源码的教材。——审校者注



因此在阅读源代码时可以随时通过模拟器确认不太明白的地方。在 simh 上运行 UNIX V6 的方法请参考附录 A.1 的 [8]、[21] 和 [22]。

## 几个难点

当然，UNIX V6 也存在它特有的问题。UNIX V6 的大部分代码使用了 C 语言编写，而当时的 C 语言还处于初期阶段，在语法规格上与现在的 C 语言有所不同（顺便提一下，C 语言就是因为编写 UNIX 才诞生的）。当时的 C 语言使用 *K&R*<sup>①</sup> 之前的语法，因此也被称为 pre K&R。请看下面的例子（代码清单 1）。

代码清单 1 当时的 C 语言的示例

```
1 struct {
2   char high;
3   char low;
4 };
5
6 hogehoge(arg) {
7   int hoge;
8   hoge = arg;
9   return hoge->low;
10 }
```

估计会有读者提出“为什么函数 hogehoge() 没有定义返回值和参数”、“为什么对 int 型的 hoge<sup>②</sup> 可以使用 hoge-> 这样的写法”等疑问。这是由 C 语言当时的语法决定的。

虽然存在这种语法上的差异，但 pre K&R C 的语法规格说明书<sup>[16]</sup>可以在网上获取，而且如果是了解当代 C 语言的读者，应该不会很难理解。而比较特殊的 pre K&R C 语法，请参考本书的附录。

另外，和其他很多程序一样，UNIX V6 的一部分代码采用了汇编语言来编写。这对首次接触汇编语言的读者来说可能会有影响。但是，UNIX V6 使用的 PDP-11（参考第 1 章）的汇编语言的规格说明书<sup>[17]</sup>也可以在网上找到，慢慢熟悉就会适应了。

① 这里指的是 *C Programming Language* 这部讲述 C 语言的名著（Brian W. Kernighan、Dennis M. Ritchie 合著）。根据两位作者姓氏的首字母，此书也被称为 *K&R*，中文版书名是《C 程序设计语言》，由机械工业出版社。——译者注

② hoge 以及后文中出现的 fuga 等是日语中经常使用的伪变量名，类似于英语中的 foo、bar。——译者注



## 面向的读者

本书主要面向计算机专业的学生，以及从事计算机相关行业的具有初中级水平的技术人员。特别适合那些通过大学课程和其他入门书对操作系统有所了解，但是对具体细节缺乏深入理解的读者，或是那些对操作系统的具体实现有兴趣的读者。

## 需要具备的知识基础

UNIX V6内核的大部分代码都是用C语言写成的，因此读者必须具备C语言的知识基础，也要掌握栈、队列等基本数据结构和算法的知识。另外，如果能了解计算机的运行原理，比如程序在执行时需要首先加载到内存，然后从程序计数器指定的内存地址读取指令等，就更为理想了。

对那些完全不了解操作系统的朋友来说，本书的难度可能有点大。建议先参考在本书附录部分介绍的入门书籍（附录A.1中的[5]和[6]），等掌握了操作系统的基本知识后再来阅读本书，这样可能会达到事半功倍的效果。

## 本书的结构

第1章介绍UNIX V6的整体概要。

第2章到第4章介绍用来管理程序运行的进程（process）。第2章概述进程。第3章说明进程的控制方法。第4章说明以有效利用有限内存空间为目的的交换（swap）处理。

第5章和第6章介绍因某种事项中断当前进程运行，并转而处理该事项的几种机制。第5章首先介绍如何处理周边设备和CPU内部发生的中断请求。然后介绍了系统调用（system call），系统调用是利用中断请求来连接用户程序和内核的机制。第6章主要介绍信号（signal），信号用于实现进程间通信，会引起处理的中断或使处理内容发生变化。

第7章和第8章介绍磁盘等设备的I/O处理。第7章说明块（block）设备子系统，第8章说明块设备驱动程序。

第9章和第10章对文件系统进行说明。文件系统隐藏了块设备的存储细节，向用户提供访问数据的统一方法。第9章介绍文件系统的概要，第10章对文件的操作加以说明。

第11章介绍用来实现进程间数据通信的管道（pipe）。

第12章介绍行式打印机的I/O处理。

第13章介绍终端处理。终端处理使用户能够以会话的方式操作系统。

第14章说明系统的启动处理。

内核的整体结构和各章之间的对应关系如图1所示。

上述章节是根据笔者认为在阅读UNIX V6内核源代码时应该遵循的顺序而编排的，将读者可能更感兴趣的有关进程和文件系统的内容尽量安排在前半部分，其他的内容则安排在后半部分。另外，在阅读各章时所需的知识，尽量在前面的章节中加以说明。

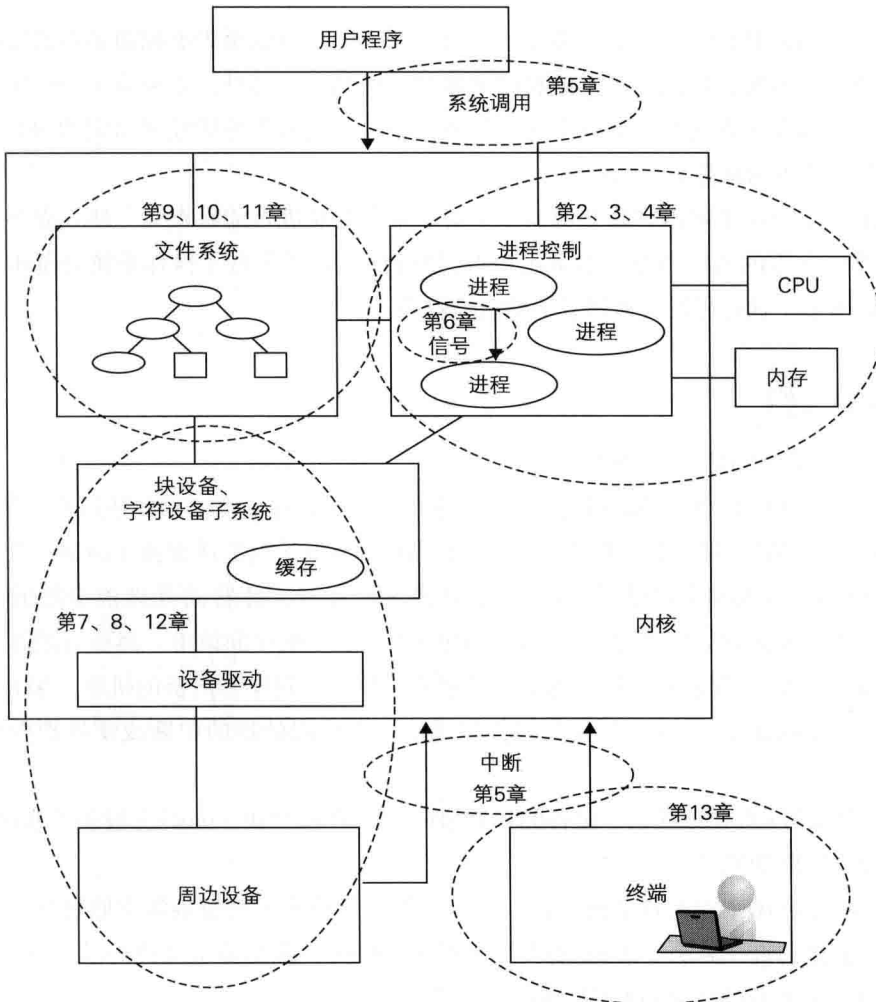


图1 内核整体结构

## 关于本书的说明

因为篇幅有限，所以本书以介绍内核的基本处理过程为主要目的，对比较少见的处理和异常处理没有进行特别细致的说明。此外，本书从所有代码中挑选出比较重要的部分（占有所有代码的70%~80%），省略不是很重要的部分或非常简单无需说明的代码。对用汇编语言编写的内容，本书只介绍了启动处理和上下文切换处理。本书中没有介绍的代码希望读者能够自己去分析理解。

另外，考虑到相关内容已经在本书中加以说明，因此在登载的代码中删除了UNIX V6开发者加入的注释。但是为了提高可读性，笔者还是保留了一部分原注释，同时也加入了一部分新的注释。

本书旨在成为UNIX V6内核源代码的阅读指南。如果想完全理解内核细节，仅阅读本书的说明部分是不够的，还需要深入理解书中涉及的甚至未涉及的代码。

## 关于代码的说明

本书按照下面的顺序，从抽象程度较高的内容向抽象程度较低的内容进行讲解。

- (1) 讲解进程或文件系统等功能
- (2) 讲解实现上述功能的函数或结构体
- (3) 给出上述函数或结构体的代码并讲解

为了便于读者阅读，本书按照下述格式讲解代码。根据需要，在有些地方调整了代码和表格的讲解顺序。

### 结构体的讲解

内核中定义了一些重要的结构体，用表格的形式介绍结构体的成员（代码清单2，表1）。在代码清单标题的括弧中，注明了定义该结构体的文件名。

讲解的过程统一采用“结构体名.成员”的形式表示某个构造体内的成员。

代码清单2 结构体示例（文件名）

```

1 struct hoge {
2     int aaa;
3     char bbb;
4     char ccc;
5     char *ddd;

```

包含结构体的文件名

```

6  int eee[3];
7  };

```

表 1 构造体示例

成员	含义
aaa	aaa 的讲解
bbb	bbb 的讲解
ccc	ccc 的讲解
*ddd	ddd 的讲解
eee[]	eee 的讲解

## 函数的讲解

首先以表格的形式讲解参数（表 2），然后给出代码和每行的讲解（代码清单 3）。在代码清单标题的括弧中，注明了定义该函数的文件名。虽然此处的例子对每行代码都做了详细（其实没有必要）的说明，但如果代码本身是无需解释的，或事先已做过介绍，那么在代码清单中就不会再做特别说明。

表 2 函数参数示例

参数	含义
fuga	在计算中使用

代码清单 3 函数示例（文件名）

```

1  hogehoge (fuga) {
2  int a, b;
3  a = 10;
4  b = fuga + a;
5  return b;
6  }

```

包含函数的文件名

### 2 局部变量的定义

3~5 返回参数 fuga 与 10 相加的结果。

对汇编语言代码也以相同的方式说明。因为本书省略了对汇编器规格的说明，所以如果有读者需要简单确认 PDP-11 指令集，请见附录 A.1 的 [9]，需要了解汇编器详细信息的读者详见附录 A.1 的 [20]。

## 栈的说明

在涉及进程拥有的栈状态时将用表格加以说明（表 3）。在第 2 章中会提到，各进程

的栈是从地址空间的最高位开始分配，向低位方向增长。但是在表格里位于上方的行表示较低位的地址，也就是说栈在表格里是由下向上增长的。

表 3 栈的示例

栈指针 (sp)	值
->	处于栈顶端的值
	处于栈顶端第 2 位的值

## 寄存器

在涉及 PDP-11 或周边设备的寄存器时将用表格加以说明（表 4）。对未使用的比特位不做介绍。

在讲解中如果需要调用寄存器的某一位，比如说第 0 位的时候，会以“寄存器名[0]”的形式标注。如果调用了多个比特位，比如说从第 3 位到第 1 位的时候，会以“寄存器名[3-1]”的形式标注。

表 4 寄存器示例

比特位	含义
15~5	数据
3~1	状态值
0	错误标志

## 数字的表示形式

UNIX V6 中有很多地方采用了八进制数。为了避免混淆基数，在介绍中采用了如下的表示形式。

- 十六进制数：0x10
- 十进制数：16
- 八进制数：020

对二进制数则直接采用 01 或 10 的形式表示。尽管十进制数和八进制数的表示形式比较相似，容易混淆，但相信读者通过上下文还是可以区分的。

另外请注意，PDP-11 的汇编器对不带前缀的数字一律按八进制数处理。

## 几点有助于增进理解的建议

UNIX V6 内核源代码尽管比较适合初次阅读代码的读者，但是仍然有一定的难度。

笔者在此举出几点便于读者增进理解的建议，在阅读代码遇到困难时请酌情参考。

## 首先通读全书

首先请从头到尾通读本书，掌握内核的全貌。在理解某个模块的代码时，往往需要对相关模块有所了解。前面已经说过，在编排章节时我尽量将相关内容靠前说明，但是这种做法毕竟还是有局限性的。在阅读中遇到难于理解的内容时，可以先跳过它，等到读完全书后再重读这一部分，这可能是一个比较好的办法。

## 阅读规格说明书和程序员手册

在阅读本书的同时，阅读规格说明书和程序员手册有助于加深理解。**不要通过阅读代码去理解程序规格，而应该在理解规格的基础上再去阅读代码。**特别是系统调用部分的规格非常重要。内核实现了最低限度的功能，而这些功能是通过系统调用提供给用户程序的，从这个角度上说，**将系统调用的规格看作是内核的规格也并不过分。**

在阅读设备驱动程序时，最好也读一下该设备的规格说明书。设备驱动程序是根据设备规格进行操作的，理解了设备规格后，对理解驱动程序进行的操作无疑会有帮助。

## 仔细阅读结构体

包括系统内核在内，程序的主要工作就是操作数据。因此，只要把握了作为处理对象的结构体，或是程序中出现的标志变量（flag），就能大致推测出该程序的处理内容。

## 区分内存空间地址

代码中涉及内存地址的部分，请注意区分该地址指向的是内核的地址空间，还是用户程序的地址空间。

## 寻找志同道合的朋友

在阅读本书、Lions书，或是UNIX V6内核源代码的时候，组织读书会是一个很好的方法。向他人提出自己的疑问，或是向他人介绍自己理解的内容，这都会加深对内容的理解。在博客上写出自己的想法和心得也是一个很好的方法。

## 本书的写作原委和谢辞

在写本书之前，笔者对操作系统的大概情况有一定认识，但是完全不了解具体的实

现方法，总希望有机会能阅读一下源代码。通过朋友的介绍接触到 *Lions' Commentary on UNIX* 这本书（也就是通称的 Lions 书），在了解到 UNIX V6 的源代码不到 1 万行等信息后，我马上就对此书产生了兴趣。

但是，Lions 书问世已久，写作的时代背景和现在相差迥异，因此对我而言很难理解，阅读速度非常缓慢。此时，又是前面提到的那位朋友向我介绍了一个关于 Lions 书的读书会，我就参加了他们的读书活动。在读书会里，通过请教对内核比较熟悉的成员，或是向他人介绍自己通过预习而掌握的知识，慢慢加深了对 UNIX V6 内核的理解。

理解了系统内核之后，我上学时学到的计算机相关知识之间的联系就在头脑中逐渐清晰起来，同时也切实体会到自己在这个领域的造诣明显地提高了一个层次。

为了和他人分享这种体验，同时也作为学习的备忘录，我开始在博客上记录学习过程。读完了 Lions 书后，博客的连载也告一段落后，我写了一篇总结性的文章，引起了很大反响。

当我确信读者对类似系统内核这种（与应用程序相比较）底层的知识有很大需求之后，便主动向出版社毛遂自荐，商谈以博客文章为基础出版图书的事宜，得到了技术评论社<sup>①</sup>的积极回应。

本书以笔者一人的微薄之力是难以完成的。如果没有参加读书会，是否能坚持读完 Lions 书都是一个问题。借此机会对读书会的各位成员表示感谢。

读书会成员大野徹先生、小泽广先生、坂顶佑树先生、高野了成先生、丰岛隆志先生、七志先生、浜田直树先生、细川生人先生、松泽裕先生和山本英雄先生参与了本书的审阅工作，为笔者提出了很多中肯的意见。一想到如果没有这些意见的话呈现在读者面前的将会是怎样一本书，不免有些后怕。正是因为大家的帮助，本书的品质得到了很大提高，我借此机会表示感谢。

## 小结

- 阅读内核源代码，可以提高自身的技术水平，加深对计算机的兴趣
- UNIX V6 的历史相对悠久，但是比较适合初次阅读内核源代码的读者
- 在理解规格说明书和手册的基础上，重复阅读本书和内核源代码将会加深理解
- 寻找志同道合的朋友十分重要

<sup>①</sup> 本书原著的出版社。——译者注



# 目录

## 第I部分

### 什么是UNIX V6 1

#### 第1章 UNIX V6的全貌 2

1.1	什么是UNIX V6 .....	2
1.2	UNIX的历史 .....	2
1.3	UNIX V6内核 .....	4
1.4	构成UNIX V6运行环境的硬件 .....	4
	PDP-11 .....	5
1.5	代码 .....	9
1.6	手册 .....	9
1.7	小结 .....	10

## 第II部分

### 进程 11

#### 第2章 进程 12

2.1	进程的概要 .....	12
	什么是进程 .....	12
	进程的并行执行 .....	12
	进程的执行状态 .....	14
	用户模式和内核模式 .....	14
	交换处理 .....	15
2.2	proc结构体和user结构体 .....	15
	proc结构体 .....	16
	user结构体 .....	17
2.3	为进程分配的内存 .....	21
	代码段 .....	21
	数据段 .....	21
	虚拟地址空间 .....	22
	变换地址 .....	24
2.4	小结 .....	26

**第3章 进程的管理 I****27**

3.1	进程的生命周期 .....	27
3.2	创建进程 .....	28
	进程的复制 .....	28
	父进程和子进程 .....	29
	系统调用 fork .....	29
	newproc() .....	32
	panic() .....	36
3.3	切换执行进程 .....	37
	中断执行进程 .....	37
	进程的执行状态 .....	37
	选择执行进程的算法 .....	38
	上下文切换 .....	39
	系统调用 wait .....	39
	sleep() .....	39
	swtch() .....	41
	swtch() 的返回位置 .....	44
	setpri() .....	51
	wakeup() .....	51
	setrun() .....	52
3.4	执行程序 .....	53
	程序执行文件的格式 .....	53
	系统调用 exec .....	54
	estabur() .....	62
	sureg() .....	65
	expand() .....	66
3.5	进程的终止 .....	68
	系统调用 exit .....	69
	系统调用 wait .....	71
3.6	数据区域的扩展 .....	73
	系统调用 break .....	73
3.7	管理内存和交换空间 .....	76
	map 结构体 .....	76
	获取未使用区域 .....	77
	释放区域 .....	79
3.8	小结 .....	81