

**Linux**第**1**章

如何才能学好 Shell 编程

1.1 为什么要学习 Shell 编程

Shell 脚本语言是实现 Linux/UNIX 系统管理及自动化运维所必备的重要工具，Linux/UNIX 系统的底层及基础应用软件的核心大都涉及 Shell 脚本的内容。每一个合格的 Linux 系统管理员或运维工程师，都需要能够熟练地编写 Shell 脚本语言，并能够阅读系统及各类软件附带的 Shell 脚本内容。只有这样才能提升运维人员的工作效率，适应日益复杂的工作环境，减少不必要的重复工作，从而为个人的职场发展奠定较好的基础。那么，Shell 脚本编程的学习是否容易呢？学习 Shell 编程到底需要什么样的 Linux 基础呢？

1.2 学好 Shell 编程所需的基础知识

本节首先来探讨一下在学习 Shell 编程之前需要掌握的基础知识，需要说明的是，并不是必须具备这些基础知识才可以学习 Shell 编程，而是，如果具备了这些基础知识，那么就可以把 Shell 编程学得更好，领悟得更深。如果只是想了解 Shell 脚本语言，那么就无须掌握太多的系统基础知识，只需要会一些简单的命令行操作即可。

学好 Shell 编程并通过 Shell 脚本轻松地实现自动化管理企业生产系统的必备基础如下：

1) 能够熟练使用 vim 编辑器，熟悉 SSH 终端及“.vimrc”等的配置。

在 Linux 下开发 Shell 脚本最常使用的编辑器是 vim, 因此如果能够熟练使用并配置好 vim 的各种高级功能设置, 就可以让开发 Shell 脚本达到事半功倍的效果。这部分内容在本书的第 16 章有相应的讲解, 读者在开始编写脚本之前可以考虑先看看第 16 章并搭建出高效的 Shell 开发环境。

 说明: 在本书的第 16 章讲解 Shell 脚本开发环境的配置调整和优化时, 提到了高效搭建 Shell 开发环境的方法, 之所以把这部分内容安排在第 16 章, 是希望读者能体验一下比较原始的 Shell 开发过程, 然后再来掌握搭建高效的开发环境的方法, 老男孩从教学的角度认为这是一个比较好的过程, 读者可以根据自身的情况来决定要不要提前学习第 16 章, 搭建好高效的 Shell 开发环境。

2) 要有一定的 Linux 命令基础, 至少需要掌握 80 个以上 Linux 常用命令, 并能够熟练使用它们 (Linux 系统的常用命令请参见本书的附录)。

和其他的开发语言 (例如 Python) 不同, Shell 脚本语言很少有可以直接使用的外部函数库, 老男孩就将 Linux 系统的命令看作 Shell 的函数库, 因此, 对 Linux 系统常用命令的掌握程度就直接决定了运维人员对 Shell 脚本编程的掌握高度。一些 Shell 类图书在开篇花费大量章节来讲解 Linux 基础命令也许就是因为这点, 本书主要侧重于 Shell 编程企业案例实战讲解, 因此不会进行大且全的介绍, 也不会过多地讲解 Linux 的常用命令, 而是采用小而美的实战策略, 本书结尾会以附录的形式给出常用的 Linux 基础命令的相关知识。此外, 如果读者想学习 Linux 基础命令, 可以关注老男孩即将出版的新书——《跟老男孩学习 Linux 运维: 常用命令实战》^①, 或者其他相关图书。

3) 要熟练掌握 Linux 正则表达式及三剑客命令 (grep、sed、awk)。

Linux 正则表达式及三剑客命令 (grep、sed、awk) 是 Linux 系统里所有命令中最核心的 3 个命令, 每个命令加上正则表达式的知识后, 功能都会变得异常强大。如果能够掌握它们, 就可以在编写 Shell 脚本时轻松很多。如读者想学习这部分知识, 可以关注老男孩即将出版的新书——《跟老男孩学习 Linux 运维: 三剑客命令实战》^②, 或者其他相关图书。

4) 熟悉常见的 Linux 网络服务部署、优化、日志分析及排错。

学习 Shell 编程最直接的目的是在工作中对系统及服务等进行自动化管理, 因此, 如果不熟悉工作中的网络服务, 就会很难使用 Shell 编程处理这些服务; 如果不掌

^① 《跟老男孩学习 Linux 运维: 常用命令实战》(预计书名) 也将由机械工业出版社出版, 时间预计为 2017 年。

^② 《跟老男孩学习 Linux 运维: 三剑客命令实战》(预计书名) 也将由机械工业出版社出版, 时间预计为 2017 年。

握网络服务等知识，就会让 Shell 开发者的能力大打折扣，甚至学习到的仅仅是 Shell 的语法及简单的基础，那么想要学好 Shell 编程的想法也就落空了。需要掌握的基础网络服务包括但不限于：Cron、Rsync、Inotify、Nginx、PHP、MySQL、Keepalived、Memcached、Redis、NFS、Iptables、SVN、Git，老男孩 IT 教育的老师在教学的过程中也是先讲解 Linux 常用命令和系统网络服务，然后再讲解 Shell 编程，目的就是不要让学员仅仅掌握 Shell 的语法皮毛，而是让他们能在学完 Shell 编程之后，自动搭建中型集群架构等，有关基础网络服务的知识可以参考机械工业出版社的《跟老男孩学习 Linux 运维：Web 集群实战》一书，或者其他相关图书。

1.3 如何才能学好 Shell 编程之“老鸟”经验谈

学好 Shell 编程的核心：多练→多思考→再练→再思考，坚持如此循环即可！

从老男孩 IT 教育毕业的一名学生^①曾在工作多年后返校分享了一篇“如何学好 Shell 编程”的讲稿，经过老男孩的整理后和读者分享如下。

（1）掌握 Shell 脚本基本语法的方法

最简单有效的方法就是将语法敲 $n+1$ 遍。为什么不是 n 遍呢？因为这里的 n 指的是你刚开始为掌握语法而练习的那些天（21 天法则），而 1 则是指在确定掌握语法后每天都要写一写、想一想，至少是要看一看，保持一个与 Shell 脚本接触的热度。

（2）掌握 Shell 脚本的各种常见语法

要掌握各类条件表达式、if 多种判断、for 循环的不同用法、While 多种读文件的循环等，这样做不是为了什么都学会，而是为了能够看懂别人写的代码。掌握常见的各种语法，也就是要经常写，而且要持续写一段时间（让动作定型，在大脑和肌肉里都打上深刻烙印），各种语法都要用。

（3）形成自己的脚本开发风格

当掌握了各种常见的语法之后，就要选定一种适合自己的语法，形成自己的开发风格，例如：if 语句的语法就只用一种，条件表达式的语法只用一种，函数的写法也只用一种，有些语法需要根据场景去选择，除非你是像师傅（老男孩）一样要教学育人。否则，没有必要什么语法都掌握。在解决问题的前提下，掌握一种语法，然后将其用精、用透就是最好的，切记横向贪多，要多纵深学习。

（4）从简单做起，简单判断，简单循环

初学者一定要从简单做起，最小化代码学习，简单判断，简单循环，简单案例练习，所有的大程序都是由多个小程序组成的，因此，一开始没必要写多大的程序，免得给自己带来过多的挫败感，形成编程恐惧症。可先通过小的程序培养兴趣及成就感，到

① 说明：该分享人是老男孩的早期学员，毕业后曾任职于一家近千人公司的运维经理岗位。目前就职于小米科技公司，担任资深工程师。

碰到大的程序时,即使遇到困难也能坚持下去了。

(5) 多模仿,多放下参考资料练习,多思考

多找一些脚本例子来仔细分析一下,或者是系统自带的,或者是别人写的(本书就包含大量例子),不要只看,看着会并不是真的会。当你闭上眼睛的时候,还能完整地回忆起来,甚至还能完整口述或手写出来才是真的会。

(6) 学会分析问题,逐渐形成编程思维

在编写程序或脚本时,先将需求理解透,对大的需求进行分解,逐步形成小的程序或模块,然后再开发,或者先分析最终需求的基础实现,最后逐步扩展批量实现。例如师傅(老男孩)在编写批量关闭不需要自启动服务的脚本时,就采用了这种分析方法,思路如下:

1) 掌握关闭一个服务的命令,即“chkconfig 服务名 off”。

2) 批量处理时,会有多个服务名,那么就要用到多条以上的命令。

3) 仔细分析以上命令,会发现需要处理的所有命令中,只有“服务名”不同,其他地方都一样,那么自然就会想到用循环语句来处理。

如果是你,能想到这些吗?若是想到了,则表示你已经形成了初级的编程思维了,恭喜你。

如果你能够通过分析将一个大的需求细分为各个小的单元,然后利用函数、判断、循环、命令等实现每一个小的单元,那么最后把所有程序组合起来就是一个大的脚本程序了。

如果达到了上述的水平,你就算会编程了,对于领导提出的需求,就能够进行合理的分解,只要在机器上多进行调试,相信一定能写出来。

(7) 编程变量名字要规范,采用驼峰语法表示

oldboyAgeName 用的就是驼峰表示法。记住,在学习的初期,不要去看不大的脚本,要从小问题和小的方面着手,当你觉得小的判断、循环等在你的脑子里瞬间就能出来时,再开始去看和写大的脚本,进行深入练习。

师傅(老男孩)常说,新手初期最好的学习方法就是多敲代码,并针对问题进行分解练习,多敲代码就是让自己养成一个编程习惯,使肌肉、视觉和思维形成记忆,分解问题实际上就是掌握软件的设计和实现思想。

对于最高的编程境界,我个人的理解是:能把大问题进行完整的分析、分解且高效解决。

完整性:就是指预先考虑到各种可能性,将问题分解后,合理模块化并实现。

高效率:例如,在求“1+2+3...+100”的和时,考虑使用算法“(1+100)×100/2”,而不是逐个去加。

(8) 不要拿来主义,特别是新手

好多网友看书或学习视频时,喜欢要文档、要代码,其实,这是学习的最大误区。

有了文档和代码，你会变得非常懒惰，心里面会觉得已经学会了，而实际上并没有学会。因此无论是看书还是学习视频，都要自己完成学习笔记及代码的书写，这本身就是最重要的学习过程，在学习上要肯于花时间和精力，而不是投机取巧。如果你至今都没有学好 Linux 运维，那么可以想一想是不是也犯了这个错误？

1.4 学完本书后可以达到何种 Shell 编程高度

如果读者具备了前文提到 Linux 基础知识，认真地阅读并按照书中的内容去勤加练习，相信很快便可熟练掌握 Shell 编程，搞定企业场景中的绝大多数 Shell 编程问题，本书介绍了大量的核心互联网运维场景企业案例，相信对大家的工作会很有帮助。

如果再配合老男孩的 Shell 脚本教学视频，定能使你如虎添翼，相关视频一共有 14 部（数百课时），观看地址为：<http://edu.51cto.com/pack/view/id-546.html>，读者也可以扫描下面的二维码，注册付费后开始学习。

