

Linux汇编语言开发指南 PDF转换可能丢失图片或格式，建议
阅读原文

https://www.100test.com/kao_ti2020/224/2021_2022_Linux_E6_B1_87_E7_BC_c103_224222.htm 一、简介 作为最基本的编程语言之一，汇编语言虽然应用的范围不算很广，但重要性却毋庸置疑，因为它能够完成许多其它语言所无法完成的功能。就拿 Linux 内核来讲，虽然绝大部分代码是用 C 语言编写的，但仍然不可避免地某些关键地方使用了汇编代码，其中主要是在 Linux 的启动部分。由于这部分代码与硬件的关系非常密切，即使是 C 语言也会有些力不从心，而汇编语言则能够很好扬长避短，最大限度地发挥硬件的性能。大多数情况下 Linux 程序员不需要使用汇编语言，因为即便是硬件驱动这样的底层程序在 Linux 操作系统中也可以用完全用 C 语言来实现，再加上 GCC 这一优秀的编译器目前已经能够对最终生成的代码进行很好的优化，的确有足够的理由让我们可以暂时将汇编语言抛在一边了。但实现情况是 Linux 程序员有时还是需要使用汇编，或者不得不使用汇编，理由很简单：精简、高效和 libc 无关性。假设要移植 Linux 到某一特定的嵌入式硬件环境下，首先必然面临如何减少系统大小、提高执行效率等问题，此时或许只有汇编语言能帮上忙了。汇编语言直接同计算机的底层软件甚至硬件进行交互，它具有如下一些优点：能够直接访问与硬件相关的存储器或 I/O 端口；能够不受编译器的限制，对生成的二进制代码进行完全的控制；能够对关键代码进行更准确的控制，避免因线程共同访问或者硬件设备共享引起的死锁；能够根据特定的应用对代码做最佳的优化，提高运行速度；能够最大限度地发挥硬件

的功能。同时还应该认识到，汇编语言是一种层次非常低的语言，它仅仅高于直接手工编写二进制的机器指令码，因此不可避免地存在一些缺点：编写的代码非常难懂，不好维护；很容易产生 bug，难于调试；只能针对特定的体系结构和处理器进行优化；开发效率很低，时间长且单调。Linux 下用汇编语言编写的代码具有两种不同的形式。第一种是完全的汇编代码，指的是整个程序全部用汇编语言编写。尽管是完全的汇编代码，Linux 平台下的汇编工具也吸收了 C 语言的长处，使得程序员可以使用 `#include`、`#ifdef` 等预处理指令，并能够通过宏定义来简化代码。第二种是内嵌的汇编代码，指的是可以嵌入到 C 语言程序中的汇编代码片段。虽然 ANSI 的 C 语言标准中没有关于内嵌汇编代码的相应规定，但各种实际使用的 C 编译器都做了这方面的扩充，这其中当然就包括 Linux 平台下的 GCC。

二、Linux 汇编语法格式

绝大多数 Linux 程序员以前只接触过 DOS/Windows 下的汇编语言，这些汇编代码都是 Intel 风格的。但在 Unix 和 Linux 系统中，更多采用的还是 AT 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com