

Linux内核体系简介Linux认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022_Linux_E5_86_85_E6_A0_c103_644990.htm 1.Linux 内核简介 现在让我们从一个比较高的高度来审视一下 GNU/Linux 操作系统的体系结构。您可以从两个层次上来考虑操作系统。最上面是用户（或应用程序）空间。这是用户应用程序执行的地方。用户空间之下是内核空间，Linux 内核正是位于这里。GNU C Library（glibc）也在这里。它提供了连接内核的系统调用接口，还提供了在用户空间应用程序和内核之间进行转换的机制。这点非常重要，因为内核和用户空间的应用程序使用的是不同的保护地址空间。每个用户空间的进程都使用自己的虚拟地址空间，而内核则占用单独的地址空间。Linux 内核可以进一步划分成 3 层。最上面是系统调用接口，它实现了一些基本的功能，例如 read 和 write。系统调用接口之下是内核代码，可以更精确地定义为独立于体系结构的内核代码。这些代码是 Linux 所支持的所有处理器体系结构所通用的。在这些代码之下是依赖于体系结构的代码，构成了通常称为 BSP（Board Support Package）的部分。这些代码用作给定体系结构的处理器和特定于平台的代码。在Linux 内核中，包括了进程管理（process management）定时器（timer）中断管理（interrupt management）内存管理（memory management）模块管理（module management）虚拟文件系统接口（VFS layer）文件系统（file system）设备驱动程序（device driver）进程间通信（inter-process communication）网络管理（network management）系统启动（system init）等操作系统

功能的实现。 2.Linux 内核的主要子系统 现在使用分类说明 Linux 内核的主要组件。 内核是什么？ 内核实际上仅仅是一个资源管理器。 不管被管理的资源是进程、内存还是硬件设备，内核负责管理并裁定多个竞争用户对资源的访问（既包括内核空间也包括用户空间）系统调用接口 SCI 层提供了某些机制执行从用户空间到内核的函数调用。正如前面讨论的一样，这个接口依赖于体系结构，甚至在相同的处理器家族内也是如此。SCI 实际上是一个非常有用的函数调用多路复用和多路分解服务。在 `./linux/kernel` 中您可以找到 SCI 的实现，并在 `./linux/arch` 中找到依赖于体系结构的部分。有关这个组件的更详细信息可以在 参考资料 一节中找到。 进程管理 进程管理的重点是进程的执行。在内核中，这些进程称为线程，代表了单独的处理器虚拟化（线程代码、数据、堆栈和 CPU 寄存器）。在用户空间，通常使用进程这个术语，不过 Linux 实现并没有区分这两个概念（进程和线程）。内核通过 SCI 提供了一个应用程序编程接口（API）来创建一个新进程（`fork`、`exec` 或 Portable Operating System Interface [POSIX] 函数），停止进程（`kill`、`exit`），并在它们之间进行通信和同步（`signal` 或者 POSIX 机制）。进程管理还包括处理活动进程之间共享 CPU 的需求。内核实现了一种新型的调度算法，不管有多少个线程在竞争 CPU，这种算法都可以在固定时间内进行操作。这种算法就称为 $O(1)$ 调度程序，这个名字就表示它调度多个线程所使用的时间和调度一个线程所使用的时间是相同的。 $O(1)$ 调度程序也可以支持多处理器（称为对称多处理器或 SMP）。您可以在 `./linux/kernel` 中找到进程管理的源代码，在 `./linux/arch` 中可以找到依赖于体系结构的源代码

。在参考资料一节中可以了解有关这个算法的更多内容。
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