一：内核追踪

Linux内核是一个很大的模块，如果只是看源码有时会难以理解Linux内核的一些代码设计情况，如果可以结合Linux内核运行同时阅读源码再好不过，本文大致介绍Linux内核追踪方式，采用工具为qemu模拟器和gdb工具完成。

1、先决条件

（1）工具：需要使用qemu模拟器和gdb，在默认的Linux发行版中基本都有gdb工具；而对于qemu模拟器则需要自己下载安装，可以使用qemu源码安装，比较麻烦，不建议使用，可以使用Linux发行版中的包管理器进行下载安装，对于ubuntu和debian使用apt-get install命令，而对于centos和federa使用yum包管理器安装；在ubuntu下安装qemu模拟器，对应的包的名字可以在http://packages.ubuntu.com/搜索

（2）Linux内核源码：为了追踪Linux内核，你需要自己编译一个新的Linux内核,内核的具体编译方式在Linux内核源码压缩包中README文件有介绍

（3）镜像文件：下载一个ubuntu的iso文件（或者其它Linux发行版的iso文件），使用qemu创建一个ubuntu镜像，创建过程为

a）qemu-img create -f qcow2 ubuntu.img 10G

b）qemu-i386 -hda ubuntu.img -cdrom ~/Documents/ubuntu-14.10-desktop-i386.iso

2、启动qemu追踪内核

（1）启动qemu模拟器：qemu-system-i386 -kernel linux-build/arch/i386/boot/bzImage -hda ubuntu.img -append "root=/dev/sda1" -S

（2）启动gdbserver：切换到qemu模拟器，alt+ctrl+2， 启动gdbserver，侦听端口1234（gdbserver tcp::1234）

（3）启动gdb：gdb ./vmlinux ,然后在gdb命令条件下链接上gdbserver， target remote localhost:1234

3、开始追踪内核

使用gdb的基本常用命令，step， break等（其他具体的命令可以参考gdb手册）。这时你可以查看系统运行的一些内核参数及系统状态信息，结合源码，查看系统的运行状态

二：内核切换（快速起动机制）

快速启动机制:允许通过已经运行的Linux内核的上下文启动另一个Linux内核，不需要经过BIOS。BIOS可能会消耗很多时间，特别是带有众多数量的外设的大型服务器。这种办法可以为经常启动机器的开发者节省很多时间。

1. 使用该机制要满足两个基本条件

1）内核版本必需为2.6以上，因为自该版本起，linux内核中加入了kexec system call模块。

2）系统需要安装 kexec-tools工具，提供用户空间的kexec命令。

$sudo apt-get install kexec-tools

三：内核优化:

内核优化是Linux管理员进阶的必经之路。特别是某些专业需求的服务器，内核经过定制优化后，可以提升30%-40%性能。内核优化之前要根据需求来定制，再就是准备好后悔药，优化后出现问题的几率很高，特别是生产系统出了问题就麻烦大了。（要搞明白为什么要优化，性能瓶颈在应用还是再系统内核，不适当的优化是问题的根源）

参数优化：由于默认的Linux内核参数考虑的是最通用的场景，这明显不符合用于支持高并发访问的Web服务器的定义，所以需要修改Linux内核参数，使得Nginx可以拥有更高的性能。在优化内核时，可以做的事件很多，不过，我们通常会根据业务特点来进行调整，当Nginx作为静态Web内容服务器、反向代理服务器或是提供图片缩略功能（实时压缩图片）的服务器时，其内核参数的调整都是不同的。

四：内核修复

把Ubuntu内核升级后多余的旧内核删除掉,可是不小心连同当前使用的内核一起删除了。这下系统没有了内核，更新后的grub启动列表里也没有了Ubuntu。解决方法应该是不关机，立刻安装新的内核。可是在那个瞬间我突然很二地想到，我要关机看看会出现什么情况。于是关机、重启，然后发现根本没有任何办法进入Ubuntu。在grub中执行命令，引导系统。现在可以正常进入系统了。然后安装新的内核

五：内核测试

◆ 通过内核提供的各种系统调用，测试内核功能的完整性;

　　◆ 把测试程序作为内核的一部分来测试，例如把测试程序写成内核模块，插入到内核中动态运行；也可以直接把测试程序编译到内核中，然后直接运行带有测试程序的内核;

　　◆ 通过内核和用户态的接口来进行交互测试，例如proc文件系统，sys文件系统;

　　◆ 通过内核调试程序进行内核测试

六：内核线程

内核线程是直接由内核本身启动的进程。内核线程实际上是将内核函数委托给独立的进程，与系统中其他进程“并行”执行（实际上，也并行于内核自身的执行），内核线程经常被称为内核“守护进程”。它们主要用于执行下列任务：

a 周期性地将修改的内存页与页来源块设备同步。

b 如果内存页很少使用，则写入交换区。

c 管理延时动作

d 实现文件系统的事务日志。

内核线程可以通过两种方式实现：

1 将一个函数传递给kernel\_thread，该函数接下来负责帮助内核调用daemonize已转换为守护进程，具体包括下列操作：

a 该函数释放其父进程的所有资源，不然这些资源会一直锁定直到线程结束。

b 阻塞信号的接收。

c 将init用作守护进程的父进程。

2． 创建内核更常用的方法是辅助函数kthread\_create，该函数创建一个新的内核线程。最初线程是停止的，需要使用wake\_up\_process启动它。或使用kthread\_run，与kthread\_create不同的是，其创建新线程后立即唤醒它。

七：内核移植

内核，即操作系统。它为底层的可编程部件提供服务，为上层应用程序提供执行环境。内核裁剪就是对这些功能进行裁剪，选取满足特定平台和需求的功能。不同的硬件平台对内核要求也不同，因此从一个平台到另一个平台需要对内核进行重新配置和编译。操作系统从一个平台过渡到另一个平台称为移植

Linux内核移植步骤

Linux内核移植是嵌入式开发只至关重要的一步，不同的嵌入式应用需要不同的嵌入式平台。内核的可裁剪行也可以在内核移植中体现出来，对于特定的硬件环境，我们可以只保留和硬件相关的内核代码，从而达到缩小内核代码体积，减少代码所占存储空间。

1、修改硬件平台的支持

2、修改平台时钟

3、修改机器码

下面就是配置内核了，配置内核有三种方式：

1、运行命令make config或make oldconfig

2、运行命令make menuconfig

3、运行命令make xconfig

1是命令行的形式，2是基于文本的图形界面，3是需要x window system支持，前两种不支持鼠标，第三种支持鼠标。

等到所有配置已经完成，开始编译内核，内核编译需要很长时间。

intel

一：内核性能测试

我们的目标是通过与社区合作，进一步提高性能的内核。在这个网站上提供的数据允许社区成员密切跟踪每一个版本的内核的性能。最终，我们希望这些测试数据导致性能的提升。

近年来，操作系统内核的性能和可扩展性一直是成功的关键部分，改进和测试的内核性能就显得尤为重要。

我们正在运行一个大规模的基准测试，覆盖内核核心组件（虚拟内存管理，I / O子系统，进程调度程序，文件系统，网络，设备驱动程序等）。基准测试是运行在各种平台测试每一周，莱纳斯的Git开发树最新的快照。从我们的测试的综合性能数据将在这里托管方便访问。

FIO是一种工具，将产生大量的线程或进程做一个特定类型的由用户指定的IO操作。该采取了一些全局参数，分别继承了由线程除非另有参数给他们重写设置了，运行该工具测试磁盘的io

ffsb是一个文件系统性能测量工具，这是一个多线程应用程序，它不同于其他的文件系统基准，用户可以提供一个配置文件来创建自定义的工作负载，而大多数其他的文件系统基准测试使用一组固定的工作负载。

oltp：系统在负载量大的情况下运行数据库（mysql）时对系统参数测试。

swap（Linux系统中的交换分区基准测试），在一个内存使用紧张的系统中测量系统的性能。