

sutie linux Home

我的思想飞飞飞！

首页 | 博文目录 | 关于我



sutie

博客访问：90641
博文数量：38
博客积分：2000
博客等级：大尉
技术积分：375
用户组：普通用户
注册时间：2005-02-13 22:18

加关注 短消息
论坛 加好友

文章分类

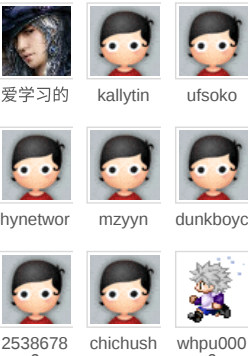
- 全部博文 (38)
- 科学相关 (7)
- 政治实事 (9)
- 诗文 (7)
- 硬件设备相关 (8)
- FC2学习笔记 (7)
- 未分配的博文 (0)

文章存档

- 2006年 (19)
- 2005年 (19)

我的朋友

最近访客



Linux下Modules的概念及使用详解[转贴] 2005-05-04 09:18:21

分类： LINUX

一、什么是 modules？

modules 的字面意思就是模块，在此指的是 kernel modules；简单来说，一个模块提供了一个功能，如 isofs、minix、nfs、lp 等等。传统来讲，模块化有两个方法解决：设计者可以把各项功能分离到单独的叫做线程的处理中去，或者是将内核以包含/排除一些功能的方式重新编译。如果把 功能分离到线程中去，那么内核就叫做“微内核”(micro-kernel)，这种解决方法增加了线程间协调工作的通信开销。就象名字暗示的那样，这种解 决方案的优点在于内核的大小。

Linux的解决方案是包含内核模块，这些模块是可以按需要随时装入和卸下的。这样做可以使得内核的大小和通信量都达到最小。将模块从内核中独立出 来，不必预先『绑』在 kernel codes 中。这样做有三种优点： 第一， 将来修改 kernel 时，不必全部重新compile，可节省不少时间；第二， 若需要安装新的 modules ，不必重新 compile kernel，只要插入 (通过insmode指令) 对应的 modules 即可；第三，减少内核对系统资源的占用，内核可以集中精力做最基本的事情，把一些扩展功能都交由modules实现。

模块也可以用来尝试新的内核代码而不需要每次都创建和重激活内核。但是，这样做带来的问题是：使用内核模块通常会轻微的增加性能和内存开支。一个可加 载模块肯定会产生更多的代码，这种代码和额外的数据结构会占用更多一点的内存。另外因为间接访问内核资源也让模块的效率轻微降低。

模块化的思想已经被广泛接受，主要的原因在于它可以扩展系统的功能，用户可以灵活的配置系统。Apache也采取了这种功能扩展方式，在本文中主要讨论是内核的模块安装与卸载，Apache模块的安装请参照Apapce的相关文档。

二、如何加载模块？

加载内核模块的方法有两种。第一种使用insmod命令手工把它插入到内核。另一个更智能的方法是在需要的时候加载这个模块：这叫做按需加载（demand loading）。当内核发现需要一个模块的时候，例如当用户安装一个不在内核的文件系统的时候，内核会请求内核守护进程（kerneld）试图加载合适 的模块。说到这里就不能不提到内核守护进程kerneld了，它非常的聪明，能够主动的把您需要的modules自动插入 kernel，将没用到的 module 从kernel中清退。Kerneld由两个独立的部分构成：一部分工作于Linux的内核，负责向daemon发送请求；另一部分工作于系统的用户数据 区，负责调入由内核请求指定的modules。若少了这个kerneld，就只能通过手工的方式,用insmode或modeprobe命令进行加载。

三、modules的相关命令介绍

与modules有关的命令有：

引用：

- lsmode :列出已经被内核调入的模块
- insmode:将某个module插入到内核中
- rmmod： 将某个module从内核中卸载
- depmod: 生成依赖文件，告诉将来的 insmod 要从哪儿调入 modules。这个依赖文件就在/lib/modules/[您的kernel版本]/modules.dep。
- Kerneld：负责自动的将模块调入内核和把模块从内核中卸载。

微信关注



IT168企业级官微

微信号：IT168qiye

系统架构师大会



系统架构师大会

微信号：SACC2013

订阅

推荐博文

·xshell可以ping通但是不能远...

·nginx设置泛域名解析的https...

·kvm性能优化

·nginx 1.12 stream 日志设置...

·静态库和动态库的生成和加载...

·MySQL中的derived table(r12...

·ORACLE利用STANDBY端RMAN备...

·Oracle并行计算

·Oracle 12CR2 dbca -silent -...

·MySQL中的反连接(r12笔记第45...

热词专题

·lua编译(linux)

四、编译一个最小的Linux内核

模块一般用来支持那些不经常使用的功能。例如，通常情况下你仅使用拨号网络，因此网络功能并不是任何时候都需要的，那么就应该使用可装入的模块来提供 这个功能。仅在你进行拨号联接的时候，该模块才被装入。而在你断掉连接的时候它会被自动卸下。这样会使内核使用内存的量最小，减小系统的负荷。

当然，那些象硬盘访问这样时时刻刻都需要的功能，则必须作在内核里。如果你搭一台网络工作站或web服务器，那么网络功能是时刻都需要的，你就应该考 虑把网络功能编译到内核里。另外一个方法是在启动的时候就装入网络模块。这种方法的优点是你不需要重新编译内核。而缺点是网络功能不能特别高效。

按照以上的原则，我们首先列出一张清单，看看 kernel 中哪些选项是非有不可的，也就是说，这些东西是必须被编译到内核中的。将那些非必需的模块剔除到内核以外。

第一个是root所在的硬盘配置。哪果您的硬盘是IDE接口，就把 ide 的选项标记下来。如果是SCSI接口，请把您的接口参数及 SCSI id 记标下来。

第二个是选择使用哪一个文件系统。Linux的默认文件系统是 ext2 ，那么就一定要把它标记下来。如果机器中还其它的操作系统，如win98或windows NT，您还会可能选择FAT32或NTFS的支持，不过后面你可以通过手工加载的方式来加入新的模块支持。

第三个是选择Linux所支持的可执行文件格式。这里有两种格式可供选择：

elf：这是当前Linux普遍支持的可执行文件格式，必须编译到内核中 。

a.out： 这是旧版的Linux的可执行文件各函数库的格式，如果你确认肯定用不到这种格式的可执行文件，那么就可以不把它编译到内核当中。

以上这些内容，是必须要编译到内核中的。其它的内容凡是所有选项中m提示的，都选择m，这样可以通过手工的方式添加该模块。

代码：

```
** Loadable module support*Enable loadable module support (CONFIG_MODULES) [Y/n/?]Set version information on all symbols for modules (CONFIG_MODVERSIONS) [N/y/?]Kernel daemon support (e.g. autoload of modules) (CONFIG_KERNELD) [Y/n/?]
```

分别回答 Y,N,Y 。其中 CONFIG_KERNELD 的 default 值是 N， 所以要注意选择Y。

make config 完后，仍旧是

代码：

```
make dep; make clean。
```

接下来要 make zliilo 或 make zImage。

然后 make modules ; make modules_install 。完成之后，就编译出一个没有调入多余模块的一个“干净的”内核映像文件了。

五、如何手工加载Modules？

如果要以手工的方式加载模块, 建议最好使用 modprobe, 因为它可以解决模块之间的依赖性问题，以声卡的部分来说，以sound blaster 为例其总共有以下模块：

引用：

```
sb 33652 0 (autoclean)
uart401 6160 0 (autoclean) [sb]
```

```
sound 56492 0 (autoclean) [sb uart401]
soundcore 2372 5 (autoclean) [sb sound]
```

这些模块都要加载上来, 整个声卡才能工作, 而且它们之间是有依赖性关系的。最核心的 soundcore 必须首先装入, 最后装入sb。但一般人是不知道其先后顺序的。因此, modprobe 就是用来解决这个问题用的。

通常我们只要 :modprobe sb

它就会自动的找出 sb 用到的所有的模块, 将它们一一 的加载进来, 故一般使用者就不用去伤脑筋了。

那么内核是怎么知道这些模块间的依赖性关系的呢? 原来, 在系统启动脚本里有一条'depmod -a'命令, 会给系统中的所有可用的模块创建一个依赖关系的列表。而 'modprobe module-name'会使用这个列表, 在装入指定的

模块前先装入那些事先装入的模块。如果在这个从属列表中找到'module-name'的话, 它会给出相应的出错信息。

但若使用 insmod, 它可不会自动完成其它模块的调入。比如说, 我们要加入PPP模块, 用这个命令:

代码:

```
root/root>insmod ppp
root/root>
```

如果操作成功, 系统出现操作提示符。如果没有成功, 可能出现下列信息:

引用:

```
/lib/modules/2.2.10/net/ppp.o: unresolved symbol slhc_init_Rsmp_1ca65fca
/lib/modules/2.2.10/net/ppp.o: unresolved symbol slhc_compress_Rsmp_cfd3a418
/lib/modules/2.2.10/net/ppp.o: unresolved symbol slhc_free_Rsmp_b99033d9
/lib/modules/2.2.10/net/ppp.o: unresolved symbol slhc_toss_Rsmp_a152cec0
/lib/modules/2.2.10/net/ppp.o: unresolved symbol slhc_remember_Rsmp_07972313
/lib/modules/2.2.10/net/ppp.o: unresolved symbol slhc_uncompress_Rsmp_3bb36b01
[root /root]#
```

这说明, PPP模块没有加载成功, 错误提示中的unresolved symbol说明, PPP模块所需要的一些模块还没有载入。错误提示第一行的内容是: slhc_init_Rsmp_1ca65fca, 这是哪个模块? 这其中可能需要一些经验来做判断, 它是以slhc开头的, 就试试slhc吧。

代码:

```
root/root>insmod slhc 一切正常, 然后我们再加加载PPP模块
root/root>insmod ppp
root/root>
```

这回没有什么返回信息, 说明PPP模块加载成功了。

六、从内存中卸载一个Modules

要卸载一个模块, 首先用lsmod看看该模块是否确实已经加载上来, 然后再做操作。除此之外, 在碰到有依赖关系的模块时, 从内核中卸载模块的过程与载入的过程恰好相反, 它遵循“first in last out”的准则, 即在一系列有依赖关系的模块中, 必须先卸载最后加载进来的模块, 最后卸载最先加载进来的模块。比如: 如果要用 rmmod 移除正在使用中的模块(如上例, 要卸载slhc, 但仍有PPP模块在使用它)会出现错误提示: Device or resource busy。所以, 在将PPP模块从内存中卸载后, 才可能将slhc模块从内存中卸载。

总之, 在卸载模块时, 对于可能出现的模块间依赖性问题, Linux会给你提示足够的信息, 仔细查看这些信息, 是能够为你采取相应的操作并最终解决问题提供帮助的。

阅读(2302) | 评论(0) | 转发(0) |

上一篇：曾经回答的几个问题
下一篇：偶得一诗

0

相关热门文章

- linux 常见服务端口
- linux dhcp peizhi roc
- xmanager 2.0 for linux配置
- 关于Unix文件的软链接
- 【ROOTFS搭建】busybox的httpd...
- 求教这个命令什么意思，我是新...
- openwrt中luci学习笔记
- sed -e "/grep/d" 是什么意思...
- Linux里如何查找文件内容...
- 谁能够帮我解决LINUX 2.6 10...

给主人留下些什么吧！~~

评论热议

登录后评论。
[登录](#) [注册](#)



[天津巨成房地产经纪](#)
先落户 后买房 热线 400-1055-157 ,一站式为您服务



[搬家上58同城先行赔](#)
提供各类车型价格统一 专业单位/住宅搬家,24小时



[天津巨成房地产经纪](#)
400-1055-157