https://blog.csdn.net/ocelotandsnake/article/details/83881858

# Linux如何在系统启动时自动加载内核模块

out of date：

centos7.9: /etc/sysconfig/modules/

ubuntu20.04: /etc/ initramfs-tools/modules, /etc/modules

rc.sysinit中有这样的一段代码：

# Load other user-defined modules

for file in /etc/sysconfig/modules/\*.modules ; do

[ -x $file ] && $file

done

# Load modules (for backward compatibility with VARs)

if [ -f /etc/rc.modules ]; then

/etc/rc.modules

fi

可见只需要配置两个地方的任何一个就可以了(以加载fuse内核模块为例)

1. 在/etc/sysconfig/modules/下面创建\*.modules文件，例如：

echo “modprobe fuse” >> /etc/sysconfig/modules/my.modules

chmod 755 my.modules

(2) 或者在/etc/rc.modules里面加上modprobe fuse，没有的话创建该文件。

然后reboot，lsmod | grep fuse验证一下就OK了。

==================================================================

要了解如何在系统启动时自动加载模块，就得先了解系统的启动顺序，这些启动操作都有哪些文件和脚本控制。

1. 整个开机流程是

(1) 载入BIOS的硬件信息，并取得第一个开机装置的代号

(2)读取第一个开机装置的MBR的boot Loader (grub)的开机信息

(3)载入OS Kernel信息，解压Kernel，尝试驱动硬件

(4) Kernel执行init程序并获得run-lebel信息(如3或5)

(5) init执行/etc/rc.d/rc.sysinit

(6)启动内核外挂模块(/etc/modprobe.conf)

(7) init执行run-level的各种Scripts，启动服务

(8) init执行/etc/rc.d/rc.local

(9)执行/bin/login，等待用户Login

(10)Login后进入Shell

网上有人建议把 modprobe modulename写入rc.local，却不曾想rc.local的执行被放在整个启动顺序的很后面，而启动init.d下面定义的服务却在 rc.local前面，那么如果某个服务要用这个模块就不行了。

看来正确的方式是把需要加载的模块放在(5)或(6)，但正如网络上很多人的尝试，修改modprobe.conf都没有成功(例如在 modprobe.conf中增加install ip\_vs…)。于是我修改了/etc/rc.d/rc.sysinit就成功加载了。

初步尝试在rc.sysinit最后增加 modprobe.conf ip\_vs，重启后lsmod | grep ip\_vs，发现成功自动加载了。于是仿效rc.sysinit中其他模块的加载方法，扩展改脚本文件，在最后增加下来一段：

# load LVS IPVS modules

if [ -d /lib/modules/$unamer/kernel/net/ipv4/ipvs ]; then

for module in /lib/modules/$unamer/kernel/net/ipv4/ipvs/\* ; do

module=${module##\*/}

module=${module%.ko}

modprobe $module >/dev/null 2>&1

done

fi

其中：

module=${module##\*/} ：其中##表示从前面删除字符，\*/表示删除到最后一个/，如果一个#就表示只删除到第一个/。如果变量后面接##，表示在##后面的字符串取最长的(一直到最后面)，如果接#，表示取最小的一段。

module=${module%.ko}：表示从后面删除.ko。如果变量后面接%%，表示在%%后面的字符串取最长的(一直到最前面)，如果接%，表示取最小的一段。

这样多module的两次修改就得到了模块名，就是文件名不带路径和.ko后缀。

modprobe $module >/dev/null 2>&1：加载模块，输出都指向空设备