Fedora Draft
Documentation
System Administrator's Reference Guide

A Reference Guide for Administrators of Fedora

Jaromír Hradílek
Douglas Silas
Martin Prpič
Stephen Wadeley
Fedora Draft Documentation System Administrator's Reference Guide
A Reference Guide for Administrators of Fedora Edition 20.0.1

Copyright © 2014 Red Hat, Inc. and others.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at http://creativecommons.org/licenses/by-sa/3.0/. The original authors of this document, and Red Hat, designate the Fedora Project as the "Attribution Party" for purposes of CC-BY-SA. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

For guidelines on the permitted uses of the Fedora trademarks, refer to https://fedoraproject.org/wiki/Legal:Trademark_guidelines.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

All other trademarks are the property of their respective owners.

The System Administrator's Reference Guide provides reference material for Fedora 20, and is oriented towards system administrators with a basic understanding of the system. For information regarding the deployment, configuration, and administration of this system, refer to the Fedora System Administrator's Guide.
# Preface

1. Document Conventions ................................................................. v
   1.1. Typographic Conventions ...................................................... v
   1.2. Pull-quote Conventions ......................................................... vi
   1.3. Notes and Warnings ............................................................... vii
2. We Need Feedback! ..................................................................... vii

# 1. Network Interfaces

1.1. Network Configuration Files ..................................................... 1
1.2. Interface Configuration Files ................................................... 2
   1.2.1. Ethernet Interfaces .......................................................... 2
   1.2.2. Specific ifcfg Options for Linux on System z ...................... 7
   1.2.3. Required ifcfg Options for Linux on System z .................. 7
   1.2.4. Alias and Clone Files ....................................................... 8
   1.2.5. Dialup Interfaces ............................................................ 9
   1.2.6. Other Interfaces ............................................................. 10
1.3. Interface Control Scripts ......................................................... 11
1.4. Network Function Files .......................................................... 13
1.5. Ethtool .................................................................................. 13
1.6. Additional Resources ............................................................. 19
   1.6.1. Installed Documentation .................................................. 19
   1.6.2. Useful Websites ............................................................. 19

# 2. The proc File System

2.1. A Virtual File System .............................................................. 21
   2.1.1. Viewing Virtual Files ....................................................... 21
   2.1.2. Changing Virtual Files ..................................................... 22
2.2. Top-level Files Within the proc File System .............................. 22
   2.2.1. /proc/buddyinfo ............................................................. 23
   2.2.2. /proc/cmdline ................................................................. 23
   2.2.3. /proc/cpuinfo ............................................................... 23
   2.2.4. /proc/crypto ................................................................. 24
   2.2.5. /proc/devices ............................................................... 24
   2.2.6. /proc/dma ................................................................. 25
   2.2.7. /proc/execdomains ....................................................... 25
   2.2.8. /proc/fb .......................................................... 26
   2.2.9. /proc/filesystems ......................................................... 26
   2.2.10. /proc/interrupts .......................................................... 26
   2.2.11. /proc/iomem .............................................................. 27
   2.2.12. /proc/ioports ............................................................ 28
   2.2.13. /proc/kcore .............................................................. 28
   2.2.14. /proc/kmsg .............................................................. 28
   2.2.15. /proc/loadavg ........................................................... 28
   2.2.16. /proc/locks .............................................................. 29
   2.2.17. /proc/mdstat ............................................................ 29
   2.2.18. /proc/meminfo .......................................................... 30
   2.2.19. /proc/misc .............................................................. 31
   2.2.20. /proc/modules ......................................................... 31
   2.2.21. /proc/mounts ............................................................ 32
   2.2.22. /proc/mtrr .............................................................. 33
   2.2.23. /proc/partitions ......................................................... 33
   2.2.24. /proc/slabinfo ........................................................... 33
   2.2.25. /proc/stat ............................................................... 34
   2.2.26. /proc/swaps ............................................................. 35
   2.2.27. /proc/sysrq-trigger .................................................... 35
3. The sysconfig Directory

3.1. Files in the /etc/sysconfig/ Directory

3.1.1. /etc/sysconfig/arpwatch ................................................................. 57
3.1.2. /etc/sysconfig/authconfig ............................................................. 57
3.1.3. /etc/sysconfig/autofs ................................................................. 60
3.1.4. /etc/sysconfig/clock ................................................................. 62
3.1.5. /etc/sysconfig/dhcpd ................................................................. 62
3.1.6. /etc/sysconfig/firstboot ............................................................. 62
3.1.7. /etc/sysconfig/i18n ................................................................. 63
3.1.8. /etc/sysconfig/init ................................................................. 63
3.1.9. /etc/sysconfig/ip6tables-config .................................................. 64
3.1.10. /etc/sysconfig/keyboard .......................................................... 66
3.1.11. /etc/sysconfig/ldap ................................................................. 66
3.1.12. /etc/sysconfig/named ............................................................... 67
3.1.13. /etc/sysconfig/network ............................................................ 68
3.1.14. /etc/sysconfig/ntpd ................................................................. 68
3.1.15. /etc/sysconfig/quagga .............................................................. 69
3.1.16. /etc/sysconfig/radvd ............................................................... 70
3.1.17. /etc/sysconfig/samba .............................................................. 70
3.1.18. /etc/sysconfig/saslauthd .......................................................... 70
3.1.19. /etc/sysconfig/selinux ............................................................. 71
3.1.20. /etc/sysconfig/sendmail .......................................................... 71
3.1.21. /etc/sysconfig/spamassassin ...................................................... 72
3.1.22. /etc/sysconfig/squid ............................................................... 72
3.1.23. /etc/sysconfig/system-config-users ........................................... 72
3.1.24. /etc/sysconfig/vncservers ......................................................... 73
3.1.25. /etc/sysconfig/xinetd ............................................................... 73

3.2. Directories in the /etc/sysconfig/ Directory

3.3. Additional Resources

2.4. Using the sysctl Command

2.5. Additional Resources

A. Revision History

Index
Preface

1. Document Conventions
This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the Liberation Fonts set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

1.1. Typographic Conventions
Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

Mono-spaced Bold
Used to highlight system input, including shell commands, file names and paths. Also used to highlight keycaps and key combinations. For example:

To see the contents of the file my_next_bestselling_novel in your current working directory, enter the cat my_next_bestselling_novel command at the shell prompt and press Enter to execute the command.

The above includes a file name, a shell command and a keycap, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from keycaps by the hyphen connecting each part of a key combination. For example:

Press Enter to execute the command.

Press Ctrl+Alt+F2 to switch to the first virtual terminal. Press Ctrl+Alt+F1 to return to your X-Windows session.

The first paragraph highlights the particular keycap to press. The second highlights two key combinations (each a set of three keycaps with each set pressed simultaneously).

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in mono-spaced bold. For example:

File-related classes include filesystem for file systems, file for files, and dir for directories. Each class has its own associated set of permissions.

Proportional Bold
This denotes words or phrases encountered on a system, including application names; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

Choose System → Preferences → Mouse from the main menu bar to launch Mouse Preferences. In the Buttons tab, click the Left-handed mouse check box and click

1 https://fedorahosted.org/liberation-fonts/
Close to switch the primary mouse button from the left to the right (making the mouse
suitable for use in the left hand).

To insert a special character into a gedit file, choose Applications → Accessories
→ Character Map from the main menu bar. Next, choose Search → Find… from the
Character Map menu bar, type the name of the character in the Search field and click
Next. The character you sought will be highlighted in the Character Table. Double-
click this highlighted character to place it in the Text to copy field and then click the
Copy button. Now switch back to your document and choose Edit → Paste from the
gedit menu bar.

The above text includes application names; system-wide menu names and items; application-specific
menu names; and buttons and text found within a GUI interface, all presented in proportional bold and
all distinguishable by context.

**Mono-spaced Bold Italic or Proportional Bold Italic**

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or
variable text. Italics denotes text you do not input literally or displayed text that changes depending on
circumstance. For example:

To connect to a remote machine using ssh, type `ssh username@domain.name` at
a shell prompt. If the remote machine is `example.com` and your username on that
machine is john, type `ssh john@example.com`.

The `mount -o remount file-system` command remounts the named file
system. For example, to remount the `/home` file system, the command is `mount -o
remount /home`.

To see the version of a currently installed package, use the `rpm -q package`
command. It will return a result as follows: `package-version-release`.

Note the words in bold italics above — username, domain.name, file-system, package, version and
release. Each word is a placeholder, either for text you enter when issuing a command or for text
displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and
important term. For example:

Publican is a DocBook publishing system.

### 1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in mono-spaced roman and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in mono-spaced roman but add syntax highlighting as follows:

```
package org.jboss.book.jca.ex1;
import javax.naming.InitialContext;
public class ExClient
```
public static void main(String args[])
throws Exception
{
    InitialContext iniCtx = new InitialContext();
    Object ref = iniCtx.lookup("EchoBean");
    EchoHome home = (EchoHome) ref;
    Echo echo = home.create();

    System.out.println("Created Echo");

    System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
}

1.3. Notes and Warnings
Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: http://bugzilla.redhat.com/bugzilla/ against the product Fedora Documentation.

When submitting a bug report, be sure to mention the manual's identifier: system-administrator's-reference-guide

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.
Network Interfaces

Under Fedora, all network communications occur between configured software *interfaces* and physical *networking devices* connected to the system.

The configuration files for network interfaces are located in the `/etc/sysconfig/network-scripts/` directory. The scripts used to activate and deactivate these network interfaces are also located here. Although the number and type of interface files can differ from system to system, there are three categories of files that exist in this directory:

1. *Interface configuration files*
2. *Interface control scripts*
3. *Network function files*

The files in each of these categories work together to enable various network devices.

This chapter explores the relationship between these files and how they are used.

### 1.1. Network Configuration Files

The primary network configuration files are as follows:

**/etc/hosts**

The main purpose of this file is to resolve host names that cannot be resolved any other way. It can also be used to resolve host names on small networks with no DNS server. Regardless of the type of network the computer is on, this file should contain a line specifying the IP address of the loopback device (127.0.0.1) as `localhost.localdomain`. For more information, see the `hosts(5)` man page.

**/etc/hostname**

There are three separate classes of host names in use on a given system. The pretty host name is the high level host name often presented to users by their desktop environment or shell. The static host name is used by the kernel at boot, and is usually the system's fully qualified domain name. A system may also have a transient host name assigned by a dhcp server. The `hostnamectl` tool is provided for administering these host names. For more information on host names, see man `hostname(5)` and man `hostnamectl(1)`.

**/etc/resolv.conf**

This file specifies the IP addresses of DNS servers and the search domain. Unless configured to do otherwise, the network initialization scripts populate this file. For more information about this file, refer to the `resolv.conf(5)` man page.

**/etc/sysconfig/network**

This file specifies routing and host information for all network interfaces. It is used to contain directives which are to have global effect and not to be interface specific. For more information about this file and the directives it accepts, see Section 3.1.13, “/etc/sysconfig/network”.

**/etc/sysconfig/network-scripts/ifcfg-interface-name**

For each network interface, there is a corresponding interface configuration script. Each of these files provide information specific to a particular network interface. See Section 1.2, “Interface Configuration Files” for more information on this type of file and the directives it accepts.
Network interface names

Network interface names may be different on different hardware types. See the Fedora Networking Guide for more information on device naming.

1.2. Interface Configuration Files

Interface configuration files control the software interfaces for individual network devices. As the system boots, it uses these files to determine what interfaces to bring up and how to configure them. These files are usually named ifcfg-name, where name refers to the name of the device that the configuration file controls.

1.2.1. Ethernet Interfaces

One of the most common interface files is /etc/sysconfig/network-scripts/ifcfg-eth0, which controls the first Ethernet network interface card or NIC in the system. In a system with multiple NICs, there are multiple ifcfg-ethX files (where X is a unique number corresponding to a specific interface). Because each device has its own configuration file, an administrator can control how each interface functions individually.

The following is a sample ifcfg-eth0 file for a system using a fixed IP address:

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
NETMASK=255.255.255.0
IPADDR=10.0.1.27
USERCTL=no
```

The values required in an interface configuration file can change based on other values. For example, the ifcfg-eth0 file for an interface using DHCP looks different because IP information is provided by the DHCP server:

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

Below is a listing of the configurable parameters in an Ethernet interface configuration file:

- **BONDING_OPTS=parameters**
  sets the configuration parameters for the bonding device, and is used in /etc/sysconfig/network-scripts/ifcfg-bondN. These parameters are identical to those used for bonding devices in /sys/class/net/bonding_device/bonding, and the module parameters for the bonding driver as described in bonding Module Directives.

This configuration method is used so that multiple bonding devices can have different configurations. It is highly recommended to place all of your bonding options after the BONDING_OPTS directive in ifcfg-name. Do not specify options for the bonding device in /etc/modprobe.d/bonding.conf, or in the deprecated /etc/modprobe.conf file.
**BOOTPROTO**=`protocol`

where `protocol` is one of the following:

- **none** — No boot-time protocol should be used.
- **bootp** — The BOOTP protocol should be used.
- **dhcp** — The DHCP protocol should be used.

**BROADCAST**=`address`

where `address` is the broadcast address. This directive is deprecated, as the value is calculated automatically with `ipcalc`.

**DEVICE**=`name`

where `name` is the name of the physical device (except for dynamically-allocated PPP devices where it is the logical name).

**DHCP_HOSTNAME**=`name`

where `name` is a short host name to be sent to the DHCP server. Use this option only if the DHCP server requires the client to specify a host name before receiving an IP address.

**DHCPV6C**=`answer`

where `answer` is one of the following:

- **yes** — Use DHCP to obtain an IPv6 address for this interface.
- **no** — Do not use DHCP to obtain an IPv6 address for this interface. This is the default value.

An IPv6 link-local address will still be assigned by default. The link-local address is based on the MAC address of the interface as per RFC 4862

**DHCPV6C_OPTIONS**=`answer`

where `answer` is one of the following:

- **-P** — Enable IPv6 prefix delegation.
- **-S** — Use DHCP to obtain stateless configuration only, not addresses, for this interface.
- **-N** — Restore normal operation after using the **-T** or **-P** options.
- **-T** — Use DHCP to obtain a temporary IPv6 address for this interface.
- **-D** — Override the default when selecting the type of DHCP Unique Identifier (DUID) to use.

By default, the DHCPv6 client (dhclient) creates a DHCP Unique Identifier (DUID) based on the link-layer address (DUID-LL) if it is running in stateless mode (with the **-S** option, to not request an address), or it creates an identifier based on the link-layer address plus a timestamp (DUID-LLT) if it is running in stateful mode (without **-S**, requesting an address). The **-D** option overrides this default, with a value of either LL or LLT.

**DNS[1,2]**=`address`

where `address` is a name server address to be placed in `/etc/resolv.conf` if the PEERDNS directive is set to **yes**.

---

1 [http://www.rfc-editor.org/info/rfc4862](http://www.rfc-editor.org/info/rfc4862)
ETHTOOL_OPTS=\textit{options}\n
where \textit{options} are any device-specific options supported by \texttt{ethtool}. For example, if you wanted to force 100Mb, full duplex:

\[ \text{ETHTOOL_OPTS="autoneg off speed 100 duplex full"} \]

Instead of a custom initscript, use \texttt{ETHTOOL_OPTS} to set the interface speed and duplex settings. Custom initscripts run outside of the network init script lead to unpredictable results during a post-boot network service restart.

\begin{center}
\textbf{Set “autoneg off” before changing speed or duplex settings}
\end{center}

Changing speed or duplex settings almost always requires disabling auto-negotiation with the \texttt{autoneg off} option. This option needs to be stated first, as the option entries are order-dependent.

Refer to Section 1.5, “Ethtool” for more \texttt{ethtool} options.

GATEWAY=\textit{address}\n
where \textit{address} is the IP address of the network router or gateway device (if any). Note that if this option is present in an interface configuration file it will take precedence over the \texttt{GATEWAY} option in the \texttt{/etc/sysconfig/network} file. It is therefore recommended that you chose between setting this option for global effect in \texttt{/etc/sysconfig/network} or per-interface.

HOTPLUG=\textit{answer}\n
where \textit{answer} is one of the following:

- \texttt{yes} — This device should be activated when it is hot-plugged (this is the default option).
- \texttt{no} — This device should not be activated when it is hot-plugged.

The \texttt{HOTPLUG=no} option can be used to prevent a channel bonding interface from being activated when a bonding kernel module is loaded.

HWADDR=\textit{MAC-address}\n
where \textit{MAC-address} is the hardware address of the Ethernet device in the form \texttt{AA:BB:CC:DD:EE:FF}. This directive must be used in machines containing more than one NIC to ensure that the interfaces are assigned the correct device names regardless of the configured load order for each NIC’s module. This directive should \textbf{not} be used in conjunction with \texttt{MACADDR}. See the Fedora Networking Guide for more information on device naming.

IPADDR=\textit{address}\n
where \textit{address} is the IPv4 address.

IPV6ADDR=\textit{address}\n
where \textit{address} is the first static, or primary, IPv6 address on an interface.

The format is Address/Prefix-length. If no prefix length is specified, \texttt{/64} is assumed. Note that this setting depends on \texttt{IPV6INIT} being enabled.

IPV6ADDR\_SECONDARIES=\textit{address}\n
where \textit{address} is one or more, space separated, additional IPv6 addresses.
The format is Address/Prefix-length. If no prefix length is specified, /64 is assumed. Note that this setting depends on IPV6INIT being enabled.

**IPV6INIT=answer**

where *answer* is one of the following:

- **yes** — Initialize this interface for IPv6 addressing.
- **no** — Do not initialize this interface for IPv6 addressing. This is the default value.

This setting is required for IPv6 static and DHCP assignment of IPv6 addresses. It does not affect IPv6 Stateless Address Autoconfiguration (SLAAC) as per RFC 4862\(^2\).

See Section 3.1.13, “/etc/sysconfig/network” for information on disabling IPv6.

**IPV6_AUTOCONF=answer**

where *answer* is one of the following:

- **yes** — Enable IPv6 autoconf configuration for this interface.
- **no** — Disable IPv6 autoconf configuration for this interface.

If enabled, an IPv6 address will be requested using Neighbor Discovery (ND) from a router running the radvd daemon.

Note that the default value of IPV6_AUTOCONF depends on IPV6FORWARDING as follows:

- If IPV6FORWARDING=**yes**, then **IPV6_AUTOCONF** will default to **no**.
- If IPV6FORWARDING=**no**, then **IPV6_AUTOCONF** will default to **yes** and **IPV6_ROUTER** has no effect.

**IPV6_MTU=value**

where *value* is an optional dedicated MTU for this interface.

**IPV6_PRIVACY=rfc3041**

where **rfc3041** optionally sets this interface to support RFC 3041 Privacy Extensions for Stateless Address Autoconfiguration in IPv6\(^3\). Note that this setting depends on IPV6INIT option being enabled.

The default is for RFC 3041\(^4\) support to be disabled. Stateless Autoconfiguration will derive addresses based on the MAC address, when available, using the modified EUI-64 method. The address is appended to a prefix but as the address is normally derived from the MAC address it is globally unique even when the prefix changes. In the case of a link-local address the prefix is fe80::/64 as per RFC 2462 IPv6 Stateless Address Autoconfiguration\(^5\).

**LINKDELAY=time**

where *time* is the number of seconds to wait for link negotiation before configuring the device.

**MACADDR=MAC-address**

where **MAC-address** is the hardware address of the Ethernet device in the form AA:BB:CC:DD:EE:FF.

\(^2\) [http://www.rfc-editor.org/info/rfc4862](http://www.rfc-editor.org/info/rfc4862)

\(^3\) [http://www.rfc-editor.org/info/rfc3041](http://www.rfc-editor.org/info/rfc3041)

\(^4\) [http://www.rfc-editor.org/info/rfc3041](http://www.rfc-editor.org/info/rfc3041)

\(^5\) [http://www.rfc-editor.org/info/rfc2462](http://www.rfc-editor.org/info/rfc2462)
This directive is used to assign a MAC address to an interface, overriding the one assigned to the physical NIC. This directive should not be used in conjunction with the `HWADDR` directive.

\textbf{MASTER=\textit{bond-interface}}

where `bond-interface` is the channel bonding interface to which the Ethernet interface is linked.

This directive is used in conjunction with the \textbf{SLAVE} directive.

\textbf{NETMASK=\textit{mask}}

where `mask` is the netmask value.

\textbf{NETWORK=\textit{address}}

where `address` is the network address. This directive is deprecated, as the value is calculated automatically with `ipcalc`.

\textbf{NM\_CONTROLLED=\textit{answer}}

where `answer` is one of the following:

\begin{itemize}
  \item \texttt{yes} — NetworkManager is permitted to configure this device. This is the default behavior and can be omitted.
  \item \texttt{no} — NetworkManager is not permitted to configure this device.
\end{itemize}

\textbf{Note}

The \textbf{NM\_CONTROLLED} directive is dependent on the \textbf{NM\_BOND\_VLAN\_ENABLED} directive in `/etc/sysconfig/network`. If and only if that directive is present and is one of \texttt{yes}, \texttt{y}, or \texttt{true}, will NetworkManager detect and manage bonding and VLAN interfaces.

\textbf{NOZEROCONF=\textit{answer}}

where `answer` is one of the following:

\begin{itemize}
  \item \texttt{yes} — Do not set a route for dynamic IPv6 link-local addresses.
  \item \texttt{no} — Allow setting a route for dynamic IPv6 link-local addresses.
\end{itemize}

\textbf{ONBOOT=\textit{answer}}

where `answer` is one of the following:

\begin{itemize}
  \item \texttt{yes} — This device should be activated at boot-time.
  \item \texttt{no} — This device should not be activated at boot-time.
\end{itemize}

\textbf{PEERDNS=\textit{answer}}

where `answer` is one of the following:

\begin{itemize}
  \item \texttt{yes} — Modify `/etc/resolv.conf` if the DNS directive is set. If using DHCP, then \texttt{yes} is the default.
  \item \texttt{no} — Do not modify `/etc/resolv.conf`.
\end{itemize}

\textbf{SLAVE=\textit{answer}}

where `answer` is one of the following:
• **yes** — This device is controlled by the channel bonding interface specified in the `MASTER` directive.

• **no** — This device is not controlled by the channel bonding interface specified in the `MASTER` directive.

This directive is used in conjunction with the `MASTER` directive.

**SRCADDR**=`address`

where `address` is the specified source IP address for outgoing packets.

**USERCTL**=`answer`

where `answer` is one of the following:

• **yes** — Non-root users are allowed to control this device.

• **no** — Non-root users are not allowed to control this device.

### 1.2.2. Specific ifcfg Options for Linux on System z

**SUBCHANNELS**=`<read_device_bus_id>, <write_device_bus_id>, <data_device_bus_id>`

where `<read_device_bus_id>, <write_device_bus_id>, and <data_device_bus_id>` are the three device bus IDs representing a network device.

**PORTNAME**=`myname;`

where `myname` is the Open Systems Adapter (OSA) portname or LAN Channel Station (LCS) portnumber.

**CTCPROT**=`answer`

where `answer` is one of the following:

• **0** — Compatibility mode, TCP/IP for Virtual Machines (used with non-Linux peers other than IBM S/390 and IBM System z operating systems). This is the default mode.

• **1** — Extended mode, used for Linux-to-Linux Peers.

• **3** — Compatibility mode for S/390 and IBM System z operating systems.

This directive is used in conjunction with the `NETTYPE` directive. It specifies the CTC protocol for `NETTYPE='ctc'`. The default is 0.

**OPTION**=`'answer'`

where `'answer'` is a quoted string of any valid sysfs attributes and their value. The Fedora installer currently uses this to configure the layer mode, (layer2), and the relative port number, (portno), of QETH devices. For example:

```
OPTIONS='layer2=1 portno=0'
```

### 1.2.3. Required ifcfg Options for Linux on System z

**NETTYPE**=`answer`

where `answer` is one of the following:

• **ctc** — Channel-to-Channel communication. For point-to-point TCP/IP or TTY.
• **lcs** — LAN Channel Station (LCS).

• **qeth** — QETH (QDIO Ethernet). This is the default network interface. It is the preferred installation method for supporting real or virtual OSA cards and HiperSockets devices.

### 1.2.4. Alias and Clone Files

Two lesser-used types of interface configuration files are *alias* and *clone* files. As the `ip` command of the `iproute` package now supports assigning multiple address to the same interface it is no longer necessary to use this method of binding multiple addresses to the same interface.

**Note**

At the time of writing, *NetworkManager* does not detect IP aliases in *ifcfg* files. For example, if `ifcfg-eth0` and `ifcfg-eth0:1` files are present, *NetworkManager* creates two connections, which will cause confusion.

For new installations, users should select the **Manual** method on the IPv4 or IPv6 tab in *NetworkManager* to assign multiple IP address to the same interface. For more information on using this tool, see the *Fedora Networking Guide*.

Alias interface configuration files, which are used to bind multiple addresses to a single interface, use the `ifcfg-if-name:alias-value` naming scheme. Do not confuse alias *ifcfg* file and interface names with VLAN *ifcfg* file and interface names which take the form: `ifcfg-if-name.vlan-id` and `ethX.vlan-id` respectively.

For example, an `ifcfg-eth0:0` file could be configured to specify `DEVICE=eth0:0` and a static IP address of `10.0.0.2`, serving as an alias of an Ethernet interface already configured to receive its IP information via DHCP in `ifcfg-eth0`. Under this configuration, `eth0` is bound to a dynamic IP address, but the same physical network card can receive requests via the fixed, `10.0.0.2` IP address.

**Warning**

Alias interfaces do not support DHCP.

A clone interface configuration file should use the following naming convention: `ifcfg-if-name-clone-name`. While an alias file allows multiple addresses for an existing interface, a clone file is used to specify additional options for an interface. For example, a standard DHCP Ethernet interface called `eth0`, may look similar to this:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
```

Since the default value for the **USERCTL** directive is **no** if it is not specified, users cannot bring this interface up and down. To give users the ability to control the interface, create a clone by copying `ifcfg-eth0` to `ifcfg-eth0-user` and add the following line to `ifcfg-eth0-user`:
This way a user can bring up the eth0 interface using the /sbin/ifup eth0-user command because the configuration options from ifcfg-eth0 and ifcfg-eth0-user are combined. While this is a very basic example, this method can be used with a variety of options and interfaces.

It is no longer possible to create alias and clone interface configuration files using a graphical tool. However, as explained at the beginning of this section, it is no longer necessary to use this method as it is now possible to directly assign multiple IP address to the same interface. For new installations, users should select the Manual method on the IPv4 or IPv6 tab in NetworkManager to assign multiple IP address to the same interface. For more information on using this tool, see the Fedora Networking Guide.

1.2.5. Dialup Interfaces

If you are connecting to the Internet via a dialup connection, a configuration file is necessary for the interface.

PPP interface files are named using the following format:

ifcfg-pppX

where X is a unique number corresponding to a specific interface.

The PPP interface configuration file is created automatically when wvdial, or Kppp is used to create a dialup account. It is also possible to create and edit this file manually.

The following is a typical /etc/sysconfig/network-scripts/ifcfg-ppp0 file:

```
DEVICE=ppp0
NAME=test
WVDIALSELECT=test
MODEMPORT=/dev/modem
LINESPEED=115200
PAPNAME=test
USERCTL=true
ONBOOT=no
PERSIST=no
DEFROUTE=yes
PEERDNS=yes
DEMAND=no
IDLETIMEOUT=600
```

Serial Line Internet Protocol (SLIP) is another dialup interface, although it is used less frequently. SLIP files have interface configuration file names such as ifcfg-sl0.

Other options that may be used in these files include:

```
DEFROUTE=answer

where answer is one of the following:

- yes — Set this interface as the default route.
- no — Do not set this interface as the default route.
```

```
DEMAND=answer

where answer is one of the following:
```

---

USERCTL=yes
Chapter 1. Network Interfaces

• yes — This interface allows pppd to initiate a connection when someone attempts to use it.

• no — A connection must be manually established for this interface.

IDLETIMEOUT=value
where value is the number of seconds of idle activity before the interface disconnects itself.

INITSTRING=string
where string is the initialization string passed to the modem device. This option is primarily used in conjunction with SLIP interfaces.

LINESPEED=value
where value is the baud rate of the device. Possible standard values include 57600, 38400, 19200, and 9600.

MODEMPORT=device
where device is the name of the serial device that is used to establish the connection for the interface.

MTU=value
where value is the Maximum Transfer Unit (MTU) setting for the interface. The MTU refers to the largest number of bytes of data a frame can carry, not counting its header information. In some dialup situations, setting this to a value of 576 results in fewer packets dropped and a slight improvement to the throughput for a connection.

NAME=name
where name is the reference to the title given to a collection of dialup connection configurations.

PAPNAME=name
where name is the username given during the Password Authentication Protocol (PAP) exchange that occurs to allow connections to a remote system.

PERSIST=answer
where answer is one of the following:

• yes — This interface should be kept active at all times, even if deactivated after a modem hang up.

• no — This interface should not be kept active at all times.

REMIP=address
where address is the IP address of the remote system. This is usually left unspecified.

WVDIALSECT=name
where name associates this interface with a dialer configuration in /etc/wvdial.conf. This file contains the phone number to be dialed and other important information for the interface.

1.2.6. Other Interfaces

Other common interface configuration files include the following:

ifcfg-lo
A local loopback interface is often used in testing, as well as being used in a variety of applications that require an IP address pointing back to the same system. Any data sent to the loopback device is immediately returned to the host's network layer.
Do not manually edit the ifcfg-lo script

The loopback interface script, /etc/sysconfig/network-scripts/ifcfg-lo, should never be edited manually. Doing so can prevent the system from operating correctly.

ifcfg-irlan0

An infrared interface allows information between devices, such as a laptop and a printer, to flow over an infrared link. This works in a similar way to an Ethernet device except that it commonly occurs over a peer-to-peer connection.

ifcfg-plip0

A Parallel Line Interface Protocol (PLIP) connection works much the same way as an Ethernet device, except that it utilizes a parallel port.

Interface configuration files for Linux on System z include the following:

ifcfg-hsiN

A HiperSockets interface is an interface for high-speed TCP/IP communication within and across z/VM guest virtual machines and logical partitions (LPARs) on an IBM System z mainframe.

1.3. Interface Control Scripts

The interface control scripts activate and deactivate system interfaces. There are two primary interface control scripts that call on control scripts located in the /etc/sysconfig/network-scripts/ directory: /sbin/ifdown and /sbin/ifup.

The ifup and ifdown interface scripts are symbolic links to scripts in the /sbin/ directory. When either of these scripts are called, they require the value of the interface to be specified, such as:

```
ifup eth0
```

Use the ifup and ifdown interface scripts

The ifup and ifdown interface scripts are the only scripts that the user should use to bring up and take down network interfaces.

The following scripts are described for reference purposes only.

Two files used to perform a variety of network initialization tasks during the process of bringing up a network interface are /etc/rc.d/init.d/functions and /etc/sysconfig/network-scripts/network-functions. Refer to Section 1.4, "Network Function Files" for more information.

After verifying that an interface has been specified and that the user executing the request is allowed to control the interface, the correct script brings the interface up or down. The following are common interface control scripts found within the /etc/sysconfig/network-scripts/ directory:
ifup-aliases
Configures IP aliases from interface configuration files when more than one IP address is associated with an interface.

ifup-ippp and ifdown-ippp
Brings ISDN interfaces up and down.

ifup-ipv6 and ifdown-ipv6
Brings IPv6 interfaces up and down.

ifup-plip
Brings up a PLIP interface.

ifup-plusb
Brings up a USB interface for network connections.

ifup-post and ifdown-post
Contains commands to be executed after an interface is brought up or down.

ifup-ppp and ifdown-ppp
Brings a PPP interface up or down.

ifup-routes
Adds static routes for a device as its interface is brought up.

ifdown-sit and ifup-sit
Contains function calls related to bringing up and down an IPv6 tunnel within an IPv4 connection.

ifup-wireless
Brings up a wireless interface.

Be careful when removing or modifying network scripts!

Removing or modifying any scripts in the /etc/sysconfig/network-scripts/ directory can cause interface connections to act irregularly or fail. Only advanced users should modify scripts related to a network interface.

The easiest way to manipulate all network scripts simultaneously is to use the /sbin/service command on the network service (/etc/rc.d/init.d/network), as illustrated by the following command:

/sbin/service network action

Here, action can be either start, stop, or restart.

To view a list of configured devices and currently active network interfaces, use the following command:

/sbin/service network status
1.4. Network Function Files

Fedora makes use of several files that contain important common functions used to bring interfaces up and down. Rather than forcing each interface control file to contain these functions, they are grouped together in a few files that are called upon when necessary.

The `/etc/sysconfig/network-scripts/network-functions` file contains the most commonly used IPv4 functions, which are useful to many interface control scripts. These functions include contacting running programs that have requested information about changes in the status of an interface, setting host names, finding a gateway device, verifying whether or not a particular device is down, and adding a default route.

As the functions required for IPv6 interfaces are different from IPv4 interfaces, a `/etc/sysconfig/network-scripts/network-functions-ipv6` file exists specifically to hold this information. The functions in this file configure and delete static IPv6 routes, create and remove tunnels, add and remove IPv6 addresses to an interface, and test for the existence of an IPv6 address on an interface.

1.5. Ethtool

**Ethtool** is a utility for configuration of Network Interface Cards (NICs). This utility allows querying and changing settings such as speed, port, auto-negotiation, PCI locations and checksum offload on many network devices, especially Ethernet devices.

We present here a short selection of often used `ethtool` commands together with some useful commands that are not well known. For a full list of commands type `ethtool -h` or refer to the man page, `ethtool(8)`, for a more comprehensive list and explanation. The first two examples are information queries and show the use of the different formats of the command.

But first, the command structure:

```
ethtool [option...] devname
```

where *option* is none or more options, and *devname* is your Network Interface Card (NIC). For example eth0 or em1.

**ethtool**

The `ethtool` command with only a device name as an option is used to print the current settings of the specified device. It takes the following form:

```
ethtool devname
```

where *devname* is your NIC. For example eth0 or em1.

Some values can only be obtained when the command is run as root. Here is an example of the output when the command is run as root:

```
~]$ ethtool em1
Settings for em1:
Supported ports: [ TP ]
Supported link modes: 10baseT/Half 10baseT/Full 100baseT/Half 100baseT/Full
```
1000baseT/Full
Supported pause frame use: No
Supports auto-negotiation: Yes
Advertised link modes:  10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                        1000baseT/Full
Advertised pause frame use: No
Advertised auto-negotiation: Yes
Speed: 1000Mb/s
Duplex: Full
Port: Twisted Pair
PHYAD: 2
Transceiver: internal
Auto-negotiation: on
MDI-X: on
Supports Wake-on: pumbg
Wake-on: g
Current message level: 0x00000007 (7)
  drv probe link
Link detected: yes

Issue the following command, using the short or long form of the argument, to query the specified network device for associated driver information:

etttool -i, --driver devname

where devname is your Network Interface Card (NIC). For example eth0 or em1.

Here is an example of the output:

```
~$ ethtool -i em1
driver: e1000e
version: 2.0.0-k
firmware-version: 0.13-3
bus-info: 0000:00:19.0
supports-statistics: yes
supports-test: yes
supports-eeprom-access: yes
supports-register-dump: yes

Here follows a list of command options to query, identify or reset the device. They are in the usual - short and --long form:

--statistics
The --statistics or -S queries the specified network device for NIC and driver statistics. It takes the following form:

-S, --statistics devname

where devname is your NIC.

--identify
The --identify or -p option initiates adapter-specific action intended to enable an operator to easily identify the adapter by sight. Typically this involves blinking one or more LEDs on the specified network port. It takes the following form:

-p, --identify devname integer
where integer is length of time in seconds to perform the action,

and devname is your NIC.

--show-time-stamping
The --show-time-stamping or -T option queries the specified network device for time stamping parameters. It takes the following form:

-T, --show-time-stamping devname

where devname is your NIC.

--show-offload
The --show-features, or --show-offload, or -k option queries the specified network device for the state of protocol offload and other features. It takes the following form:

-k, --show-features, --show-offload devname

where devname is your NIC.

--test
The --test or -t option is used to perform tests on a Network Interface Card. It takes the following form:

-t, --test word devname

where word is one of the following:

- offline — Perform a comprehensive set of tests. Service will be interrupted.
- online — Perform a reduced set of tests. Service should not be interrupted.
- external_lb — Perform full set of tests including loopback tests while fitted with a loopback cable.

and devname is your NIC.

Changing some or all settings of the specified network device requires the -s or --change option. All the following options are only applied if the -s or --change option is also specified. For the sake of clarity we will omit it here.

To make these settings permanent you can make use of the ETHTOOL_OPTS directive. It can be used in interface configuration files to set the desired options when the network interface is brought up. Refer to Section 1.2.1, “Ethernet Interfaces” for more details on how to use this directive.

--offload
The --features, or --offload, or -K option changes the offload parameters and other features of the specified network device. It takes the following form:

-K, --features, --offload devname feature boolean

where feature is a built-in or kernel supplied feature,

boolean is one of ON or OFF.
and devname is your NIC.

The `ethtool(8)` man page lists most features. As the feature set is dependent on the NIC driver, you should consult the driver documentation for features not listed in the man page.

**--speed**

The `--speed` option is used to set the speed in megabits per second (Mb/s). Omitting the speed value will show the supported device speeds. It takes the following form:

```
--speed number devname
```

where `number` is the speed in megabits per second (Mb/s),

and devname is your NIC.

**--duplex**

The `--duplex` option is used to set the transmit and receive mode of operation. It takes the following form:

```
--duplex word devname
```

where `word` is one of the following:

- `half` — Sets half-duplex mode. Usually used when connected to a hub.
- `full` — Sets full-duplex mode. Usually used when connected to a switch or another host.

and devname is your NIC.

**--port**

The `--port` option is used to select the device port. It takes the following form:

```
--port value devname
```

where `value` is one of the following:

- `tp` — An Ethernet interface using Twisted-Pair cable as the medium.
- `aui` — Attachment Unit Interface (AUI). Normally used with hubs.
- `bnc` — An Ethernet interface using BNC connectors and co-axial cable.
- `mii` — An Ethernet interface using a Media Independent Interface (MII).
- `fibre` — An Ethernet interface using Optical Fibre as the medium.

and devname is your NIC.

**--autoneg**

The `--autoneg` option is used to control auto-negotiation of network speed and mode of operation (full-duplex or half-duplex mode). If auto-negotiation is enabled you can initiate re-negotiation of network speeds and mode of operation by using the `-r, --negotiate` option. You can display the auto-negotiation state using the `--a, --show-pause` option.

It takes the following form:
**--autoneg** value devname

where value is one of the following:

- **yes** — Allow auto-negotiating of network speed and mode of operation.
- **no** — Do not allow auto-negotiating of network speed and mode of operation.

and devname is your NIC.

**--advertise**

The **--advertise** option is used to set what speeds and modes of operation (duplex mode) are advertised for auto-negotiation. The argument is one or more hexadecimal values from Table 1.1, "Ethtool advertise options: speed and mode of operation".

It takes the following form:

**--advertise** option devname

where option is one or more of the hexadecimal values from the table below and devname is your NIC.

**Table 1.1. Ethtool advertise options: speed and mode of operation**

<table>
<thead>
<tr>
<th>Hex Value</th>
<th>Speed</th>
<th>Duplex Mode</th>
<th>IEEE standard?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x001</td>
<td>10</td>
<td>Half</td>
<td>Yes</td>
</tr>
<tr>
<td>0x002</td>
<td>10</td>
<td>Full</td>
<td>Yes</td>
</tr>
<tr>
<td>0x004</td>
<td>100</td>
<td>Half</td>
<td>Yes</td>
</tr>
<tr>
<td>0x008</td>
<td>100</td>
<td>Full</td>
<td>Yes</td>
</tr>
<tr>
<td>0x010</td>
<td>1000</td>
<td>Half</td>
<td>No</td>
</tr>
<tr>
<td>0x020</td>
<td>1000</td>
<td>Full</td>
<td>Yes</td>
</tr>
<tr>
<td>0x8000</td>
<td>2500</td>
<td>Full</td>
<td>Yes</td>
</tr>
<tr>
<td>0x1000</td>
<td>10000</td>
<td>Full</td>
<td>Yes</td>
</tr>
<tr>
<td>0x20000</td>
<td>20000MLD2</td>
<td>Full</td>
<td>No</td>
</tr>
<tr>
<td>0x20000</td>
<td>20000MLD2</td>
<td>Full</td>
<td>No</td>
</tr>
<tr>
<td>0x40000</td>
<td>20000KR2</td>
<td>Full</td>
<td>No</td>
</tr>
</tbody>
</table>

**--phyad**

The **--phyad** option is used to change the physical address. Often referred to as the MAC or hardware address but in this context referred to as the physical address.

It takes the following form:

**--phyad** physical_address devname

where physical_address is the physical address in hexadecimal format and devname is your NIC.

**--xcvr**

The **--xcvr** option is used to select the transceiver type. Currently only "internal" and "external" can be specified. In the future other types might be added.
Chapter 1. Network Interfaces

It takes the following form:

```
--xcvr word devname
```

where `word` is one of the following:

- **internal** — Use internal transceiver.
- **external** — Use external transceiver.

and `devname` is your NIC.

**--wol**
The `--wol` option is used to set “Wake-on-LAN” options. Not all devices support this. The argument to this option is a string of characters specifying which options to enable.

It takes the following form:

```
--wol value devname
```

where `value` is one or more of the following:

- **p** — Wake on PHY activity.
- **u** — Wake on unicast messages.
- **m** — Wake on multicast messages.
- **b** — Wake on broadcast messages.
- **g** — Wake-on-Lan; wake on receipt of a “magic packet”.
- **s** — Enable security function using password for Wake-on-Lan.
- **d** — Disable Wake-on-Lan and clear all settings.

and `devname` is your NIC.

**--sopass**
The `--sopass` option is used to set the “SecureOn” password. The argument to this option must be 6 bytes in Ethernet MAC hexadecimal format (`xx:yy:zz:aa:bb:cc`).

It takes the following form:

```
--sopass xx:yy:zz:aa:bb:cc devname
```

where `xx:yy:zz:aa:bb:cc` is the password in the same format as a MAC address and `devname` is your NIC.

**--msglvl**
The `--msglvl` option is used to set the driver message-type flags by name or number. The precise meanings of these type flags differ between drivers.

It takes the following form:

```
--msglvl message_type devname
```
where `message_type` is one of:
- message type name in plain text.
- hexadecimal number indicating the message type.

and `devname` is your NIC.

The defined message type names and numbers are shown in the table below:

<table>
<thead>
<tr>
<th>Message Type</th>
<th>Hex Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>drv</td>
<td>0x0001</td>
<td>General driver status</td>
</tr>
<tr>
<td>probe</td>
<td>0x0002</td>
<td>Hardware probing</td>
</tr>
<tr>
<td>link</td>
<td>0x0004</td>
<td>Link state</td>
</tr>
<tr>
<td>timer</td>
<td>0x0008</td>
<td>Periodic status check</td>
</tr>
<tr>
<td>ifdown</td>
<td>0x0010</td>
<td>Interface being brought down</td>
</tr>
<tr>
<td>ifup</td>
<td>0x0020</td>
<td>Interface being brought up</td>
</tr>
<tr>
<td>rx_err</td>
<td>0x0040</td>
<td>Receive error</td>
</tr>
<tr>
<td>tx_err</td>
<td>0x0080</td>
<td>Transmit error</td>
</tr>
<tr>
<td>intr</td>
<td>0x0200</td>
<td>Interrupt handling</td>
</tr>
<tr>
<td>tx_done</td>
<td>0x0400</td>
<td>Transmit completion</td>
</tr>
<tr>
<td>rx_status</td>
<td>0x0800</td>
<td>Receive completion</td>
</tr>
<tr>
<td>pktdata</td>
<td>0x1000</td>
<td>Packet contents</td>
</tr>
<tr>
<td>hw</td>
<td>0x2000</td>
<td>Hardware status</td>
</tr>
<tr>
<td>wol</td>
<td>0x4000</td>
<td>Wake-on-LAN status</td>
</tr>
</tbody>
</table>

### 1.6. Additional Resources

The following are resources which explain more about network interfaces.

#### 1.6.1. Installed Documentation

`/usr/share/doc/initscripts-version/sysconfig.txt`

A guide to available options for network configuration files, including IPv6 options not covered in this chapter.

#### 1.6.2. Useful Websites

`http://linux-ip.net/gl/ip cref/`

This document contains a wealth of information about the `ip` command, which can be used to manipulate routing tables, among other things. The information can also be found in the `ip cref.ps` file by installing the `iproute-doc` sub-package from the optional content channel.
Chapter 2.

The proc File System

The Linux kernel has two primary functions: to control access to physical devices on the computer and to schedule when and how processes interact with these devices. The /proc directory (also called the proc file system) contains a hierarchy of special files which represent the current state of the kernel, allowing applications and users to peer into the kernel's view of the system.

The /proc directory contains a wealth of information detailing system hardware and any running processes. In addition, some of the files within /proc can be manipulated by users and applications to communicate configuration changes to the kernel.

Note

Later versions of the 2.6 kernel have made the /proc/ide/ and /proc/pci/ directories obsolete. The /proc/ide/ file system is now superseded by files in sysfs; to retrieve information on PCI devices, use lspci instead. For more information on sysfs or lspci, refer to their respective man pages.

2.1. A Virtual File System

Linux systems store all data as files. Most users are familiar with the two primary types of files: text and binary. But the /proc directory contains another type of file called a virtual file. As such, /proc is often referred to as a virtual file system.

Virtual files have unique qualities. Most of them are listed as zero bytes in size, but can still contain a large amount of information when viewed. In addition, most of the time and date stamps on virtual files reflect the current time and date, indicative of the fact they are constantly updated.

Virtual files such as /proc/interrupts, /proc/meminfo, /proc/mounts, and /proc/partitions provide an up-to-the-moment glimpse of the system's hardware. Others, like the /proc/filesystems file and the /proc/sys/ directory provide system configuration information and interfaces.

For organizational purposes, files containing information on a similar topic are grouped into virtual directories and sub-directories. Process directories contain information about each running process on the system.

2.1.1. Viewing Virtual Files

Most files within /proc files operate similarly to text files, storing useful system and hardware data in human-readable text format. As such, you can use cat, more, or less to view them. For example, to display information about the system's CPU, run cat /proc/cpuinfo. This will return output similar to the following:

```
processor : 0
vendor_id : AuthenticAMD
cpu family : 5
model : 9
model name : AMD-K6(tm) 3D+
Processor stepping : 1 cpu
MHz : 400.919
cache size : 256 KB
```
fdiv_bug           : no  
hlt_bug            : no  
f00f_bug           : no  
coma_bug           : no  
fpu                : yes  
fpu_exception      : yes  
cpuid_level        : 1  
wps                : yes  
flags              : fpu vme de pse tsc msr mce cx8 pge mmx syscall 3dnow k6_mtrr  
bogomips           : 799.53

Some files in /proc contain information that is not human-readable. To retrieve information from such files, use tools such as lspci, apm, free, and top.

**Note**

Some of the virtual files in the /proc directory are readable only by the root user.

### 2.1.2. Changing Virtual Files

As a general rule, most virtual files within the /proc directory are read-only. However, some can be used to adjust settings in the kernel. This is especially true for files in the /proc/sys/ subdirectory.

To change the value of a virtual file, use the following command:

```
echo value > /proc/file_name
```

For example, to change the host name on the fly, run:

```
echo www.example.com > /proc/sys/kernel/hostname
```

Other files act as binary or Boolean switches. Typing `cat /proc/sys/net/ipv4/ip_forward` returns either a 0 (off or false) or a 1 (on or true). A 0 indicates that the kernel is not forwarding network packets. To turn packet forwarding on, run:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

**Note**

Another command used to alter settings in the /proc/sys/ subdirectory is /sbin/sysctl. For more information on this command, refer to Section 2.4, “Using the sysctl Command”

For a listing of some of the kernel configuration files available in the /proc/sys/ subdirectory, refer to Section 2.3.9, “/proc/sys/”.

### 2.2. Top-level Files Within the proc File System

Below is a list of some of the more useful virtual files in the top-level of the /proc directory.
Note

In most cases, the content of the files listed in this section are not the same as those installed on your machine. This is because much of the information is specific to the hardware on which Fedora is running for this documentation effort.

2.2.1. /proc/buddyinfo

This file is used primarily for diagnosing memory fragmentation issues. Using the buddy algorithm, each column represents the number of pages of a certain order (a certain size) that are available at any given time. For example, for zone direct memory access (DMA), there are 90 of $2^{(0\times\text{PAGE\_SIZE})}$ chunks of memory. Similarly, there are 6 of $2^{(1\times\text{PAGE\_SIZE})}$ chunks, and 2 of $2^{(2\times\text{PAGE\_SIZE})}$ chunks of memory available.

The DMA row references the first 16 MB on a system, the HighMem row references all memory greater than 4 GB on a system, and the Normal row references all memory in between.

The following is an example of the output typical of /proc/buddyinfo:

<table>
<thead>
<tr>
<th>Node 0, zone</th>
<th>DMA</th>
<th>90</th>
<th>6</th>
<th>2</th>
<th>1</th>
<th>1</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node 0, zone</td>
<td>Normal</td>
<td>1650</td>
<td>310</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>Node 0, zone</td>
<td>HighMem</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

2.2.2. /proc/cmdline

This file shows the parameters passed to the kernel at the time it is started. A sample /proc/cmdline file looks like the following:

```
ro root=/dev/VolGroup00/LogVol00 rhgb quiet 3
```

This tells us that the kernel is mounted read-only (signified by (ro)), located on the first logical volume (LogVol00) of the first volume group (/dev/VolGroup00). LogVol00 is the equivalent of a disk partition in a non-LVM system (Logical Volume Management), just as /dev/VolGroup00 is similar in concept to /dev/hda1, but much more extensible.

For more information on LVM used in Fedora, refer to [http://www.tldp.org/HOWTO/LVM-HOWTO/index.html](http://www.tldp.org/HOWTO/LVM-HOWTO/index.html).

Next, rhgb signals that the rhgb package has been installed, and graphical booting is supported, assuming /etc/inittab shows a default runlevel set to id:5:initdefault:.

Finally, quiet indicates all verbose kernel messages are suppressed at boot time.

2.2.3. /proc/cpuinfo

This virtual file identifies the type of processor used by your system. The following is an example of the output typical of /proc/cpuinfo:

```
processor      : 0
vendor_id      : GenuineIntel
cpu family     : 15
model          : 2
model name     : Intel(R) Xeon(TM) CPU 2.40GHz
```
2.2.4. /proc/crypto
This file lists all installed cryptographic ciphers used by the Linux kernel, including additional details for each. A sample /proc/crypto file looks like the following:

<table>
<thead>
<tr>
<th>name</th>
<th>sha1</th>
</tr>
</thead>
<tbody>
<tr>
<td>module</td>
<td>kernel</td>
</tr>
<tr>
<td>type</td>
<td>digest</td>
</tr>
<tr>
<td>blocksize</td>
<td>64</td>
</tr>
<tr>
<td>digestsize</td>
<td>20</td>
</tr>
</tbody>
</table>

2.2.5. /proc/devices
This file displays the various character and block devices currently configured (not including devices whose modules are not loaded). Below is a sample output from this file:

<table>
<thead>
<tr>
<th>name</th>
<th>sha1</th>
</tr>
</thead>
<tbody>
<tr>
<td>module</td>
<td>kernel</td>
</tr>
<tr>
<td>type</td>
<td>digest</td>
</tr>
<tr>
<td>blocksize</td>
<td>64</td>
</tr>
<tr>
<td>digestsize</td>
<td>16</td>
</tr>
</tbody>
</table>
Character devices:

1. mem
2. /dev/vc/0
3. tty
4. ttyS
5. /dev/tty
6. /dev/console
7. vcs
8. misc
9. input
10. fb
11. netlink
12. ptm
13. pts
14. usb
15. vcs
16. misc
17. input
18. fb
19. netlink
20. ptm
21. pts
22. usb

Block devices:

1. ramdisk
2. ide0
3. md
4. ide1
5. device-mapper
6. mdp

The output from `/proc/devices` includes the major number and name of the device, and is broken into two major sections: **Character devices** and **Block devices**.

**Character devices** are similar to **block devices**, except for two basic differences:

1. Character devices do not require buffering. Block devices have a buffer available, allowing them to order requests before addressing them. This is important for devices designed to store information — such as hard drives — because the ability to order the information before writing it to the device allows it to be placed in a more efficient order.

2. Character devices send data with no preconfigured size. Block devices can send and receive information in blocks of a size configured per device.


### 2.2.6. /proc/dma

This file contains a list of the registered ISA DMA channels in use. A sample `/proc/dma` files looks like the following:

```
4: cascade
```

### 2.2.7. /proc/execdomains

This file lists the *execution domains* currently supported by the Linux kernel, along with the range of personalities they support.

```
0-0   Linux   [kernel]
```

Think of execution domains as the “personality” for an operating system. Because other binary formats, such as Solaris, UnixWare, and FreeBSD, can be used with Linux, programmers can change the way the operating system treats system calls from these binaries by changing the personality of
the task. Except for the `PER_LINUX` execution domain, different personalities can be implemented as dynamically loadable modules.

### 2.2.8. /proc/fb

This file contains a list of frame buffer devices, with the frame buffer device number and the driver that controls it. Typical output of `/proc/fb` for systems which contain frame buffer devices looks similar to the following:

```
0  VESA  VGA
```

### 2.2.9. /proc/filesystems

This file displays a list of the file system types currently supported by the kernel. Sample output from a generic `/proc/filesystems` file looks similar to the following:

```
nodev   sysfs
nodev   rootfs
nodev   bdev
nodev   proc
nodev   sockfs
nodev   binfmt_misc
nodev   usbfs
nodev   usbdevfs
nodev   futexfs
nodev   tmpfs
nodev   sockfs
nodev   proc
nodev   rootfs
nodev   bdev
nodev   sysfs
```

The first column signifies whether the file system is mounted on a block device. Those beginning with `nodev` are not mounted on a device. The second column lists the names of the file systems supported.

The `mount` command cycles through the file systems listed here when one is not specified as an argument.

### 2.2.10. /proc/interrupts

This file records the number of interrupts per IRQ on the x86 architecture. A standard `/proc/interrupts` looks similar to the following:

```
CPU0
0:  80448940  XT-PIC  timer
1:  174412    XT-PIC  keyboard
2:          0    XT-PIC  cascade
8:          1    XT-PIC  rtc
10:         410964  XT-PIC  eth0
12:         68330  XT-PIC  PS/2 Mouse
14:        1314121  XT-PIC  ide0
15:        5195422  XT-PIC  ide1
```
### /proc/iomem

For a multi-processor machine, this file may look slightly different:

<table>
<thead>
<tr>
<th>CPU0</th>
<th>CPU1</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0: 1366814704</td>
<td>0</td>
<td>XT-PIC</td>
<td>timer</td>
</tr>
<tr>
<td>1: 128</td>
<td>340</td>
<td>IO-APIC-edge</td>
<td>keyboard</td>
</tr>
<tr>
<td>2: 0</td>
<td>0</td>
<td>XT-PIC</td>
<td>cascade</td>
</tr>
<tr>
<td>8: 0</td>
<td>1</td>
<td>IO-APIC-edge</td>
<td>rtc</td>
</tr>
<tr>
<td>12: 5323</td>
<td>5793</td>
<td>IO-APIC-edge</td>
<td>PS/2 Mouse</td>
</tr>
<tr>
<td>13: 1</td>
<td>0</td>
<td>XT-PIC</td>
<td>fpu</td>
</tr>
<tr>
<td>16: 11184204</td>
<td>15940594</td>
<td>IO-APIC-level</td>
<td>Intel EtherExpress Pro 10/100 Ethernet</td>
</tr>
<tr>
<td>20: 8450843</td>
<td>11120093</td>
<td>IO-APIC-level</td>
<td>megaraid</td>
</tr>
<tr>
<td>30: 10432</td>
<td>10722</td>
<td>IO-APIC-level</td>
<td>aic7xxx</td>
</tr>
<tr>
<td>31: 23</td>
<td>22</td>
<td>IO-APIC-level</td>
<td>aic7xxx</td>
</tr>
</tbody>
</table>

The first column refers to the IRQ number. Each CPU in the system has its own column and its own number of interrupts per IRQ. The next column reports the type of interrupt, and the last column contains the name of the device that is located at that IRQ.

Each of the types of interrupts seen in this file, which are architecture-specific, mean something different. For x86 machines, the following values are common:

- **XT-PIC** — This is the old AT computer interrupts.
- **IO-APIC-edge** — The voltage signal on this interrupt transitions from low to high, creating an edge, where the interrupt occurs and is only signaled once. This kind of interrupt, as well as the **IO-APIC-level** interrupt, are only seen on systems with processors from the 586 family and higher.
- **IO-APIC-level** — Generates interrupts when its voltage signal is high until the signal is low again.

#### 2.2.11. /proc/iomem

This file shows you the current map of the system's memory for each physical device:

```
00000000-00000000 : System RAM
00000000-00000000 : reserved
00000000-00000000 : Video RAM area
00000000-00000000 : Video ROM
00000000-00000000 : System ROM
00000000-00000000 : System RAM
00000000-00000000 : Video RAM area
00000000-00000000 : Video ROM
00000000-00000000 : System ROM
00000000-00000000 : System RAM
00000000-00000000 : reserved
00000000-00000000 : reserved
```

The first column displays the memory registers used by each of the different types of memory. The second column lists the kind of memory located within those registers and displays which memory registers are used by the kernel within the system RAM or, if the network interface card has multiple Ethernet ports, the memory registers assigned for each port.
2.2.12. /proc/ioports

The output of /proc/ioports provides a list of currently registered port regions used for input or output communication with a device. This file can be quite long. The following is a partial listing:

<table>
<thead>
<tr>
<th>Address Range</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-001f</td>
<td>dma1</td>
</tr>
<tr>
<td>0020-003f</td>
<td>pic1</td>
</tr>
<tr>
<td>0040-005f</td>
<td>timer</td>
</tr>
<tr>
<td>0060-006f</td>
<td>keyboard</td>
</tr>
<tr>
<td>0070-007f</td>
<td>rtc</td>
</tr>
<tr>
<td>0080-008f</td>
<td>dma page reg</td>
</tr>
<tr>
<td>00a0-00bf</td>
<td>pic2</td>
</tr>
<tr>
<td>00c0-00df</td>
<td>dma2</td>
</tr>
<tr>
<td>00f0-00ff</td>
<td>fpu</td>
</tr>
<tr>
<td>0170-0177</td>
<td>ide1</td>
</tr>
<tr>
<td>01f0-01f7</td>
<td>ide0</td>
</tr>
<tr>
<td>02f8-02ff</td>
<td>serial(auto)</td>
</tr>
<tr>
<td>0376-0376</td>
<td>ide1</td>
</tr>
<tr>
<td>03c0-03df</td>
<td>vga+</td>
</tr>
<tr>
<td>03f6-03f6</td>
<td>ide0</td>
</tr>
<tr>
<td>03f8-03ff</td>
<td>serial(auto)</td>
</tr>
<tr>
<td>0cf8-0cff</td>
<td>PCI conf1</td>
</tr>
<tr>
<td>d000-dfff</td>
<td>PCI Bus #01</td>
</tr>
<tr>
<td>e000-e00f</td>
<td>VIA Technologies, Inc. Bus Master IDE</td>
</tr>
<tr>
<td>e000-e007</td>
<td>ide0</td>
</tr>
<tr>
<td>e008-e00f</td>
<td>ide1</td>
</tr>
<tr>
<td>e800-e87f</td>
<td>Digital Equipment Corporation DECchip 21140 [FasterNet]</td>
</tr>
<tr>
<td>e800-e87f</td>
<td>tulip</td>
</tr>
</tbody>
</table>

The first column gives the I/O port address range reserved for the device listed in the second column.

2.2.13. /proc/kcore

This file represents the physical memory of the system and is stored in the core file format. Unlike most /proc files, kcore displays a size. This value is given in bytes and is equal to the size of the physical memory (RAM) used plus 4 KB.

The contents of this file are designed to be examined by a debugger, such as gdb, and is not human readable.

⚠️ Warning

Do not view the /proc/kcore virtual file. The contents of the file scramble text output on the terminal. If this file is accidentally viewed, press Ctrl+C to stop the process and then type reset to bring back the command line prompt.

2.2.14. /proc/kmsg

This file is used to hold messages generated by the kernel. These messages are then picked up by other programs, such as /sbin/klogd or /bin/dmesg.

2.2.15. /proc/loadavg

This file provides a look at the load average in regard to both the CPU and IO over time, as well as additional data used by uptime and other commands. A sample /proc/loadavg file looks similar to the following:
0.20 0.18 0.12 1/80 11206

The first three columns measure CPU and IO utilization of the last one, five, and 15 minute periods. The fourth column shows the number of currently running processes and the total number of processes. The last column displays the last process ID used.

In addition, load average also refers to the number of processes ready to run (i.e. in the run queue, waiting for a CPU share).

2.2.16. /proc/locks

This file displays the files currently locked by the kernel. The contents of this file contain internal kernel debugging data and can vary tremendously, depending on the use of the system. A sample /proc/locks file for a lightly loaded system looks similar to the following:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>POSIX</td>
<td>ADVISORY</td>
<td>WRITE</td>
<td>3568</td>
</tr>
<tr>
<td>2</td>
<td>FLOCK</td>
<td>ADVISORY</td>
<td>WRITE</td>
<td>3517</td>
</tr>
<tr>
<td>3</td>
<td>POSIX</td>
<td>ADVISORY</td>
<td>WRITE</td>
<td>3452</td>
</tr>
<tr>
<td>4</td>
<td>POSIX</td>
<td>ADVISORY</td>
<td>WRITE</td>
<td>3443</td>
</tr>
<tr>
<td>5</td>
<td>POSIX</td>
<td>ADVISORY</td>
<td>WRITE</td>
<td>3326</td>
</tr>
<tr>
<td>6</td>
<td>POSIX</td>
<td>ADVISORY</td>
<td>WRITE</td>
<td>3175</td>
</tr>
<tr>
<td>7</td>
<td>POSIX</td>
<td>ADVISORY</td>
<td>WRITE</td>
<td>3056</td>
</tr>
</tbody>
</table>

Each lock has its own line which starts with a unique number. The second column refers to the class of lock used, with FLOCK signifying the older-style UNIX file locks from a flock system call and POSIX representing the newer POSIX locks from the lockf system call.

The third column can have two values: ADVISORY or MANDATORY. ADVISORY means that the lock does not prevent other people from accessing the data; it only prevents other attempts to lock it. MANDATORY means that no other access to the data is permitted while the lock is held. The fourth column reveals whether the lock is allowing the holder READ or WRITE access to the file. The fifth column shows the ID of the process holding the lock. The sixth column shows the ID of the file being locked, in the format of MAJOR-DEVICE:MINOR-DEVICE:INODE-NUMBER. The seventh and eighth column shows the start and end of the file's locked region.

2.2.17. /proc/mdstat

This file contains the current information for multiple-disk, RAID configurations. If the system does not contain such a configuration, then /proc/mdstat looks similar to the following:

Personalities : read_ahead not set unused devices: <none>

This file remains in the same state as seen above unless a software RAID or md device is present. In that case, view /proc/mdstat to find the current status of mdX RAID devices.

The /proc/mdstat file below shows a system with its md0 configured as a RAID 1 device, while it is currently re-syncing the disks:

Personalities : [linear] [raid1] read_ahead 1024 sectors
md0: active raid1 sda2[1] sdb2[0] 9940 blocks [2/2] [UU] resync=1% finish=12.3min algorithm 2 [3/3] [UUU] unused devices: <none>
### 2.2.18. /proc/meminfo

This is one of the more commonly used files in the /proc directory, as it reports a large amount of valuable information about the systems RAM usage.

The following sample /proc/meminfo virtual file is from a system with 256 MB of RAM and 512 MB of swap space:

| MemTotal:       | 256908 kB |
| MemFree:        | 69936 kB  |
| Buffers:        | 15812 kB  |
| Cached:         | 115124 kB |
| SwapCached:     | 0 kB      |
| Active:         | 92700 kB  |
| Inactive:       | 63792 kB  |
| HighTotal:      | 0 kB      |
| HighFree:       | 0 kB      |
| LowTotal:       | 255908 kB |
| LowFree:        | 69936 kB  |
| SwapTotal:      | 524280 kB |
| SwapFree:       | 524280 kB |
| Dirty:          | 4 kB      |
| Writeback:      | 0 kB      |
| Mapped:         | 42236 kB  |
| Slab:           | 25912 kB  |
| Committed_AS:   | 118680 kB |
| PageTables:     | 1236 kB   |
| VmallocTotal:   | 3874808 kB|
| VmallocUsed:    | 1416 kB   |
| VmallocChunk:   | 3872908 kB|
| HugePages_Total:| 0         |
| HugePages_Free: | 0         |
| Hugepagesize:   | 4096 kB   |

Much of the information here is used by the `free`, `top`, and `ps` commands. In fact, the output of the `free` command is similar in appearance to the contents and structure of /proc/meminfo. But by looking directly at /proc/meminfo, more details are revealed:

- **MemTotal** — Total amount of physical RAM, in kilobytes.
- **MemFree** — The amount of physical RAM, in kilobytes, left unused by the system.
- **Buffers** — The amount of physical RAM, in kilobytes, used for file buffers.
- **Cached** — The amount of physical RAM, in kilobytes, used as cache memory.
- **SwapCached** — The amount of swap, in kilobytes, used as cache memory.
- **Active** — The total amount of buffer or page cache memory, in kilobytes, that is in active use. This is memory that has been recently used and is usually not reclaimed for other purposes.
- **Inactive** — The total amount of buffer or page cache memory, in kilobytes, that are free and available. This is memory that has not been recently used and can be reclaimed for other purposes.
- **HighTotal** and **HighFree** — The total and free amount of memory, in kilobytes, that is not directly mapped into kernel space. The HighTotal value can vary based on the type of kernel used.
- **LowTotal** and **LowFree** — The total and free amount of memory, in kilobytes, that is directly mapped into kernel space. The LowTotal value can vary based on the type of kernel used.
- **SwapTotal** — The total amount of swap available, in kilobytes.
• **SwapFree** — The total amount of swap free, in kilobytes.

• **Dirty** — The total amount of memory, in kilobytes, waiting to be written back to the disk.

• **Writeback** — The total amount of memory, in kilobytes, actively being written back to the disk.

• **Mapped** — The total amount of memory, in kilobytes, which have been used to map devices, files, or libraries using the `mmap` command.

• **Slab** — The total amount of memory, in kilobytes, used by the kernel to cache data structures for its own use.

• **Committed_AS** — The total amount of memory, in kilobytes, estimated to complete the workload. This value represents the worst case scenario value, and also includes swap memory.

• **PageTables** — The total amount of memory, in kilobytes, dedicated to the lowest page table level.

• **VMallocTotal** — The total amount of memory, in kilobytes, of total allocated virtual address space.

• **VMallocUsed** — The total amount of memory, in kilobytes, of used virtual address space.

• **VMallocChunk** — The largest contiguous block of memory, in kilobytes, of available virtual address space.

• **HugePages_Total** — The total number of hugepages for the system. The number is derived by dividing **Hugepagesize** by the megabytes set aside for hugepages specified in `/proc/sys/vm/hugetlb_pool`. *This statistic only appears on the x86, Itanium, and AMD64 architectures.*

• **HugePages_Free** — The total number of hugepages available for the system. *This statistic only appears on the x86, Itanium, and AMD64 architectures.*

• **Hugepagesize** — The size for each hugepages unit in kilobytes. By default, the value is 4096 KB on uniprocessor kernels for 32 bit architectures. For SMP, hugemem kernels, and AMD64, the default is 2048 KB. For Itanium architectures, the default is 262144 KB. *This statistic only appears on the x86, Itanium, and AMD64 architectures.*

2.2.19. `/proc/misc`

This file lists miscellaneous drivers registered on the miscellaneous major device, which is device number 10:

```
63 device-mapper 175 agpgart 135 rtc 134 apm_bios
```

The first column is the minor number of each device, while the second column shows the driver in use.

2.2.20. `/proc/modules`

This file displays a list of all modules loaded into the kernel. Its contents vary based on the configuration and use of your system, but it should be organized in a similar manner to this sample `/proc/modules` file output:
Note

This example has been reformatted into a readable format. Most of this information can also be viewed via the `/sbin/lsmod` command.

```
nfs       170109  0 -          Live 0x129b0000
lockd     51593   1 nfs,       Live 0x128b0000
nls_utf8   1729    0 -          Live 0x12830000
vfat       12097   0 -          Live 0x12823000
fat        38881   1 vfat,      Live 0x1287b000
autofs4    28293   2 -          Live 0x1284f000
sunrpc    140453  3 nfs,lockd, Live 0x12954000
3c59x      33257   0 -          Live 0x12871000
uhci_hcd   28377   0 -          Live 0x12869000
md5        3777    1 -          Live 0x1282c000
ipv6       211845 16 -          Live 0x128de000
ext3       92585   2 -          Live 0x12886000
jbd        65625   1 ext3,      Live 0x12857000
dm_mod     46677   3 -          Live 0x12833000
```

The first column contains the name of the module.

The second column refers to the memory size of the module, in bytes.

The third column lists how many instances of the module are currently loaded. A value of zero represents an unloaded module.

The fourth column states if the module depends upon another module to be present in order to function, and lists those other modules.

The fifth column lists what load state the module is in: `Live`, `Loading`, or `Unloading` are the only possible values.

The sixth column lists the current kernel memory offset for the loaded module. This information can be useful for debugging purposes, or for profiling tools such as `oprofile`.

### 2.2.21. `/proc/mounts`

This file provides a list of all mounts in use by the system:

```
rootfs / rootfs rw 0 0
/proc /proc proc rw,nodiratime 0 0 none
/dev ramfs rw 0 0
/dev/mapper/VolGroup00-LogVol00 / ext3 rw 0 0 none /dev ramfs rw 0 0
/proc /proc proc rw,nodiratime 0 0
/sys /sys sysfs rw 0 0 none /dev/pts devpts rw 0 0
usbdevfs /proc/bus/usb usbdevfs rw 0 0
/dev/hda1 /boot ext3 rw 0 0 none /dev/shm tmpfs rw 0 0
none /proc/sys/fs/binfmt_misc binfmt_misc rw 0 0
sunrpc /var/lib/nfs/rpc_pipefs rpc_pipefs rw 0 0
```

The output found here is similar to the contents of `/etc/mtab`, except that `/proc/mounts` is more up-to-date.
The first column specifies the device that is mounted, the second column reveals the mount point, and the third column tells the file system type, and the fourth column tells you if it is mounted read-only (ro) or read-write (rw). The fifth and sixth columns are dummy values designed to match the format used in /etc/mtab.

2.2.22. /proc/mtrr

This file refers to the current Memory Type Range Registers (MTRRs) in use with the system. If the system architecture supports MTRRs, then the /proc/mtrr file may look similar to the following:

```plaintext
reg00: base=0x00000000 (   0MB), size= 256MB: write-back, count=1
reg01: base=0xe8000000 (3712MB), size=  32MB: write-combining, count=1
```

MTRRs are used with the Intel P6 family of processors (Pentium II and higher) and control processor access to memory ranges. When using a video card on a PCI or AGP bus, a properly configured /proc/mtrr file can increase performance more than 150%.

Most of the time, this value is properly configured by default. More information on manually configuring this file can be found in the /usr/share/doc/kernel-doc-<kernel_version>/Documentation/<arch>/mtrr.txt file.

2.2.23. /proc/partitions

This file contains partition block allocation information. A sampling of this file from a basic system looks similar to the following:

```plaintext
major minor  #blocks  name
3     0   19531250 hda
3     1     104391 hda1
3     2   19422585 hda2
253     0   22708224 dm-0
253     1     524288 dm-1
```

Most of the information here is of little importance to the user, except for the following columns:

- **major** — The major number of the device with this partition. The major number in the /proc/partitions, (3), corresponds with the block device ide0, in /proc/devices.
- **minor** — The minor number of the device with this partition. This serves to separate the partitions into different physical devices and relates to the number at the end of the name of the partition.
- **#blocks** — Lists the number of physical disk blocks contained in a particular partition.
- **name** — The name of the partition.

2.2.24. /proc/slabinfo

This file gives full information about memory usage on the slab level. Linux kernels greater than version 2.2 use slab pools to manage memory above the page level. Commonly used objects have their own slab pools.

Instead of parsing the highly verbose /proc/slabinfo file manually, the /usr/bin/slabtop program displays kernel slab cache information in real time. This program allows for custom configurations, including column sorting and screen refreshing.

A sample screen shot of /usr/bin/slabtop usually looks like the following example:
Chapter 2. The proc File System

Active / Total Objects (% used) : 133629 / 147300 (90.7%)
Active / Total Slabs (% used) : 11492 / 11493 (100.0%)
Active / Total Caches (% used) : 77 / 121 (63.6%)
Active / Total Size (% used) : 41739.83K / 44081.89K (94.7%)
Minimum / Average / Maximum Object : 0.01K / 0.30K / 128.00K

<table>
<thead>
<tr>
<th>OBJS</th>
<th>ACTIVE</th>
<th>USE</th>
<th>OBJ SIZE</th>
<th>SLABS</th>
<th>OBJ/SLAB</th>
<th>CACHE SIZE</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>44814</td>
<td>43159</td>
<td>96%</td>
<td>0.62K</td>
<td>7469</td>
<td>6</td>
<td>29876K</td>
<td>ext3_inode_cache</td>
</tr>
<tr>
<td>36900</td>
<td>34614</td>
<td>93%</td>
<td>0.05K</td>
<td>492</td>
<td>75</td>
<td>1968K</td>
<td>buffer_head</td>
</tr>
<tr>
<td>35213</td>
<td>33124</td>
<td>94%</td>
<td>0.16K</td>
<td>1531</td>
<td>23</td>
<td>6124K</td>
<td>dentry_cache</td>
</tr>
<tr>
<td>7364</td>
<td>6463</td>
<td>87%</td>
<td>0.27K</td>
<td>526</td>
<td>14</td>
<td>2104K</td>
<td>radix_tree_node</td>
</tr>
<tr>
<td>2585</td>
<td>1781</td>
<td>68%</td>
<td>0.05K</td>
<td>73</td>
<td>31</td>
<td>292K</td>
<td>size-128</td>
</tr>
<tr>
<td>1904</td>
<td>1125</td>
<td>50%</td>
<td>0.03K</td>
<td>16</td>
<td>119</td>
<td>64K</td>
<td>size-32</td>
</tr>
<tr>
<td>1666</td>
<td>768</td>
<td>46%</td>
<td>0.02K</td>
<td>14</td>
<td>119</td>
<td>56K</td>
<td>anon_vma</td>
</tr>
<tr>
<td>1512</td>
<td>1482</td>
<td>98%</td>
<td>0.44K</td>
<td>168</td>
<td>9</td>
<td>672K</td>
<td>inode_cache</td>
</tr>
<tr>
<td>1464</td>
<td>1640</td>
<td>71%</td>
<td>0.06K</td>
<td>24</td>
<td>61</td>
<td>96K</td>
<td>size-64</td>
</tr>
<tr>
<td>1320</td>
<td>820</td>
<td>62%</td>
<td>0.19K</td>
<td>66</td>
<td>20</td>
<td>264K</td>
<td>filp</td>
</tr>
<tr>
<td>678</td>
<td>587</td>
<td>86%</td>
<td>0.02K</td>
<td>3</td>
<td>226</td>
<td>12K</td>
<td>dm_io</td>
</tr>
<tr>
<td>576</td>
<td>574</td>
<td>99%</td>
<td>0.47K</td>
<td>72</td>
<td>8</td>
<td>288K</td>
<td>proc_inode_cache</td>
</tr>
<tr>
<td>528</td>
<td>514</td>
<td>97%</td>
<td>0.50K</td>
<td>66</td>
<td>8</td>
<td>264K</td>
<td>size-512</td>
</tr>
<tr>
<td>492</td>
<td>372</td>
<td>75%</td>
<td>0.09K</td>
<td>12</td>
<td>41</td>
<td>48K</td>
<td>bio</td>
</tr>
<tr>
<td>465</td>
<td>314</td>
<td>67%</td>
<td>0.25K</td>
<td>31</td>
<td>15</td>
<td>124K</td>
<td>size-256</td>
</tr>
<tr>
<td>452</td>
<td>331</td>
<td>73%</td>
<td>0.02K</td>
<td>2</td>
<td>226</td>
<td>8K</td>
<td>biovec-1</td>
</tr>
<tr>
<td>420</td>
<td>420</td>
<td>100%</td>
<td>0.19K</td>
<td>21</td>
<td>20</td>
<td>84K</td>
<td>skbuff_head_cache</td>
</tr>
<tr>
<td>395</td>
<td>256</td>
<td>83%</td>
<td>0.06K</td>
<td>5</td>
<td>61</td>
<td>20K</td>
<td>biovec-4</td>
</tr>
<tr>
<td>290</td>
<td>4</td>
<td>1%</td>
<td>0.01K</td>
<td>1</td>
<td>290</td>
<td>4K</td>
<td>revoke_table</td>
</tr>
<tr>
<td>264</td>
<td>264</td>
<td>100%</td>
<td>4.00K</td>
<td>264</td>
<td>1</td>
<td>1056K</td>
<td>size-4096</td>
</tr>
<tr>
<td>260</td>
<td>256</td>
<td>98%</td>
<td>0.19K</td>
<td>13</td>
<td>20</td>
<td>52K</td>
<td>biovec-16</td>
</tr>
<tr>
<td>260</td>
<td>256</td>
<td>98%</td>
<td>0.75K</td>
<td>52</td>
<td>5</td>
<td>200K</td>
<td>biovec-64</td>
</tr>
</tbody>
</table>

Some of the more commonly used statistics in /proc/slabinfo that are included into /usr/bin/slabtop include:

- **OBJS** — The total number of objects (memory blocks), including those in use (allocated), and some spares not in use.
- **ACTIVE** — The number of objects (memory blocks) that are in use (allocated).
- **USE** — Percentage of total objects that are active. ((ACTIVE/OBJS)(100))
- **OBJ SIZE** — The size of the objects.
- **SLABS** — The total number of slabs.
- **OBJ/SLAB** — The number of objects that fit into a slab.
- **CACHE SIZE** — The cache size of the slab.
- **NAME** — The name of the slab.

For more information on the /usr/bin/slabtop program, refer to the slabtop man page.

2.2.25. /proc/stat

This file keeps track of a variety of different statistics about the system since it was last restarted. The contents of /proc/stat, which can be quite long, usually begins like the following example:

```
cpu  259246 7001 60190 34250993 137517 772 0
cpu0 259246 7001 60190 34250993 137517 772 0
intr 354133732 347209999 2272 0 4 4 0 0 3 1 1249247 0 0 80143 0 422626 5169433
ctxt 12547729
```
Some of the more commonly used statistics include:

- **cpu** — Measures the number of jiffies (1/100 of a second for x86 systems) that the system has been in user mode, user mode with low priority (nice), system mode, idle task, I/O wait, IRQ (hardirq), and softirq respectively. The IRQ (hardirq) is the direct response to a hardware event. The IRQ takes minimal work for queuing the “heavy” work up for the softirq to execute. The softirq runs at a lower priority than the IRQ and therefore may be interrupted more frequently. The total for all CPUs is given at the top, while each individual CPU is listed below with its own statistics. The following example is a 4-way Intel Pentium Xeon configuration with multi-threading enabled, therefore showing four physical processors and four virtual processors totaling eight processors.

- **page** — The number of memory pages the system has written in and out to disk.

- **swap** — The number of swap pages the system has brought in and out.

- **intr** — The number of interrupts the system has experienced.

- **btime** — The boot time, measured in the number of seconds since January 1, 1970, otherwise known as the epoch.

### 2.2.26. /proc/swaps

This file measures swap space and its utilization. For a system with only one swap partition, the output of `/proc/swaps` may look similar to the following:

<table>
<thead>
<tr>
<th>Filename</th>
<th>Type</th>
<th>Size</th>
<th>Used</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/mapper/VolGroup00-LogVol01</td>
<td>partition</td>
<td>524280</td>
<td>0</td>
<td>-1</td>
</tr>
</tbody>
</table>

While some of this information can be found in other files in the `/proc` directory, `/proc/swap` provides a snapshot of every swap file name, the type of swap space, the total size, and the amount of space in use (in kilobytes). The priority column is useful when multiple swap files are in use. The lower the priority, the more likely the swap file is to be used.

### 2.2.27. /proc/sysrq-trigger

Using the `echo` command to write to this file, a remote root user can execute most System Request Key commands remotely as if at the local terminal. To `echo` values to this file, the `/proc/sys/`
Section 2.3.9.3, “/proc/sys/kernel/”.

Although it is possible to write to this file, it cannot be read, even by the root user.

### 2.2.28. /proc/uptime

This file contains information detailing how long the system has been on since its last restart. The output of /proc/uptime is quite minimal:

```
350735.47 234388.90
```

The first value represents the total number of seconds the system has been up. The second value is the sum of how much time each core has spent idle, in seconds. Consequently, the second value may be greater than the overall system uptime on systems with multiple cores.

### 2.2.29. /proc/version

This file specifies the version of the Linux kernel, the version of gcc used to compile the kernel, and the time of kernel compilation. It also contains the kernel compiler's user name (in parentheses).

```
Linux version 2.6.8-1.523 (user@foo.redhat.com) (gcc version 3.4.1 20040714 \  (Red Hat Enterprise Linux 3.4.1-7)) #1 Mon Aug 16 13:27:03 EDT 2004
```

This information is used for a variety of purposes, including the version data presented when a user logs in.

### 2.3. Directories within /proc

Common groups of information concerning the kernel are grouped into directories and subdirectories within the /proc directory.

#### 2.3.1. Process Directories

Every /proc directory contains a number of directories with numerical names. A listing of them may be similar to the following:

```
dr-xr-xr-x 3 root    root    0 Feb 13 01:28  1
dr-xr-xr-x 3 root    root    0 Feb 13 01:28 1010
dr-xr-xr-x 3 xfs     xfs     0 Feb 13 01:28 1087
dr-xr-xr-x 3 daemon  daemon  0 Feb 13 01:28 1123
dr-xr-xr-x 3 root    root    0 Feb 13 01:28 11307
dr-xr-xr-x 3 apache  apache  0 Feb 13 01:28 13660
dr-xr-xr-x 3 rpc     rpc     0 Feb 13 01:28 637
dr-xr-xr-x 3 rpcuser rpcuser 0 Feb 13 01:28 666
```

These directories are called **process directories**, as they are named after a program's process ID and contain information specific to that process. The owner and group of each process directory is set to the user running the process. When the process is terminated, its /proc process directory vanishes.

Each process directory contains the following files:

- **cmdline** — Contains the command issued when starting the process.
- **cwd** — A symbolic link to the current working directory for the process.
• **environ** — A list of the environment variables for the process. The environment variable is given in all upper-case characters, and the value is in lower-case characters.

• **exe** — A symbolic link to the executable of this process.

• **fd** — A directory containing all of the file descriptors for a particular process. These are given in numbered links:

```
total 0
lrwx------ 1 root root 64 May 8 11:31 0 -> /dev/null
lrwx------ 1 root root 64 May 8 11:31 1 -> /dev/null
lrwx------ 1 root root 64 May 8 11:31 2 -> /dev/null
lrwx------ 1 root root 64 May 8 11:31 3 -> /dev/ptmx
lrwx------ 1 root root 64 May 8 11:31 4 -> socket:[7774817]
lrwx------ 1 root root 64 May 8 11:31 5 -> /dev/ptmx
lrwx------ 1 root root 64 May 8 11:31 6 -> socket:[7774829]
lrwx------ 1 root root 64 May 8 11:31 7 -> /dev/ptmx
```

• **maps** — A list of memory maps to the various executables and library files associated with this process. This file can be rather long, depending upon the complexity of the process, but sample output from the `sshd` process begins like the following:

```
08048000-08086000 r-xp 00000000 03:03 391479 /usr/sbin/sshd
08086000-08088000 rw-p 0003e000 03:03 391479 /usr/sbin/sshd
08088000-08095000 rwxp 00000000 00:00 0
40000000-40013000 r-xp 00000000 03:03 293205 /lib/ld-2.2.5.so
40013000-40014000 rw-p 00013000 03:03 293205 /lib/ld-2.2.5.so
40031000-40038000 r-xp 00000000 03:03 293282 /lib/libpam.so.0.75
40038000-40039000 rw-p 00006000 03:03 293282 /lib/libpam.so.0.75
40039000-4003a000 rw-p 00000000 00:00 0
4003a000-4003c000 r-xp 00000000 03:03 293218 /lib/libdl-2.2.5.so
4003c000-4003d000 rw-p 00001000 03:03 293218 /lib/libdl-2.2.5.so
```

• **mem** — The memory held by the process. This file cannot be read by the user.

• **root** — A link to the root directory of the process.

• **stat** — The status of the process.

• **statm** — The status of the memory in use by the process. Below is a sample `/proc/statm` file:

```
263 210 210 5 0 205 0
```

The seven columns relate to different memory statistics for the process. From left to right, they report the following aspects of the memory used:

1. Total program size, in kilobytes.
2. Size of memory portions, in kilobytes.
3. Number of pages that are shared.
4. Number of pages that are code.
5. Number of pages of data/stack.
6. Number of library pages.
7. Number of dirty pages.
• **status** — The status of the process in a more readable form than **stat** or **statm**. Sample output for **sshd** looks similar to the following:

```text
Name: sshd
State: S (sleeping)
Tgid: 797
Pid: 797
PPid: 1
TracerPid: 0
Uid: 0 0 0 0
Gid: 0 0 0 0
FDSize: 32
Groups:
  VmSize: 3072 kB
  VmLck: 0 kB
  VmRSS: 840 kB
  VmData: 104 kB
  VmStk: 12 kB
  VmExe: 300 kB
  VmLib: 2528 kB
  SigPnd: 0000000000000000
  SigBlk: 0000000000000000
  SigIgn: 8000000000001000
  SigCgt: 0000000000014005
  CapInh: 0000000000000000
  CapPrm: 00000000fffffeff
  CapEff: 00000000fffffeff
```

The information in this output includes the process name and ID, the state (such as **S (sleeping)** or **R (running)**), user/group ID running the process, and detailed data regarding memory usage.

### 2.3.1. /proc/self/

The **/proc/self/** directory is a link to the currently running process. This allows a process to look at itself without having to know its process ID.

Within a shell environment, a listing of the **/proc/self/** directory produces the same contents as listing the process directory for that process.

### 2.3.2. /proc/bus/

This directory contains information specific to the various buses available on the system. For example, on a standard system containing PCI and USB buses, current data on each of these buses is available within a subdirectory within **/proc/bus/** by the same name, such as **/proc/bus/pci/**.

The subdirectories and files available within **/proc/bus/** vary depending on the devices connected to the system. However, each bus type has at least one directory. Within these bus directories are normally at least one subdirectory with a numerical name, such as **001**, which contain binary files.

For example, the **/proc/bus/usb/** subdirectory contains files that track the various devices on any USB buses, as well as the drivers required for them. The following is a sample listing of a **/proc/bus/usb/** directory:

```text
total 0
dr-xr-xr-x 1 root root 0 May 3 16:25 001
-r--r--r-- 1 root root 0 May 3 16:25 devices
-r--r--r-- 1 root root 0 May 3 16:25 drivers
```

The **/proc/bus/usb/001/** directory contains all devices on the first USB bus and the **devices** file identifies the USB root hub on the motherboard.
The following is an example of a `/proc/bus/usb/devices` file:

```
T:  Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#=  1 Spd=12  MxCh=  2
B:  Alloc=  0/900 us ( 0%), #Int=  0, #Iso=  0
D:  Ver= 1.00 Cls=09(hub ) Sub=00 Prot=00 MxPS= 8 #Cfgs=  1
P:  Vendor=0000 ProdID=0000 Rev= 0.00
S:  Product=USB UHCI Root Hub
S:  SerialNumber=d400
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr= 0mA
I:  If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub
E:  Ad=81(I) Atr=03(Int.) MxP=   8 Ivl=255ms
```

### 2.3.3. `/proc/bus/pci`

Later versions of the 2.6 Linux kernel have obsoleted the `/proc/pci` directory in favor of the `/proc/bus/pci` directory. Although you can get a list of all PCI devices present on the system using the command `cat /proc/bus/pci/devices`, the output is difficult to read and interpret.

For a human-readable list of PCI devices, run the following command:

```
~# /sbin/lspci -vb
```

```
00:00.0 Host bridge: Intel Corporation 82X38/X48 Express DRAM Controller
  Subsystem: Hewlett-Packard Company Device 1308
  Flags: bus master, fast devsel, latency 0
  Capabilities: [e0] Vendor Specific Information <?>
  Kernel driver in use: x38_edac
  Kernel modules: x38_edac

00:01.0 PCI bridge: Intel Corporation 82X38/X48 Express Host-Primary PCI Express Bridge
  (prog-if 00 [Normal decode])
  Flags: bus master, fast devsel, latency 0
  Bus: primary=00, secondary=01, subordinate=01, sec-latency=0
  I/O behind bridge: 00001000-00001fff
  Memory behind bridge: f0000000-f2ffffff
  Capabilities: [88] Subsystem: Hewlett-Packard Company Device 1308
  Capabilities: [80] Power Management version 3
  Capabilities: [90] MSI: Enable+ Count=1/1 Maskable- 64bit-
  Capabilities: [a0] Express Root Port (Slot+), MSI 00
  Capabilities: [100] Virtual Channel <?>
  Capabilities: [140] Root Complex Link <?>
  Kernel driver in use: pcieport
  Kernel modules: shpchp

00:1a.0 USB Controller: Intel Corporation 82801I (ICH9 Family) USB UHCI Controller #4 (rev 02)
  (prog-if 00 [UHCI])
  Subsystem: Hewlett-Packard Company Device 1308
  Flags: bus master, medium devsel, latency 0, IRQ 5
  I/O ports at 2100
  Capabilities: [50] PCI Advanced Features
  Kernel driver in use: uhci_hcd
```

The output is a sorted list of all IRQ numbers and addresses as seen by the cards on the PCI bus instead of as seen by the kernel. Beyond providing the name and version of the device, this list also gives detailed IRQ information so an administrator can quickly look for conflicts.

### 2.3.4. `/proc/driver/`

This directory contains information for specific drivers in use by the kernel.
Chapter 2. The proc File System

A common file found here is `rtc` which provides output from the driver for the system's Real Time Clock (RTC), the device that keeps the time while the system is switched off. Sample output from `/proc/driver/rtc` looks like the following:

```
rtc_time        : 16:21:00
rtc_date        : 2004-08-31
rtc_epoch       : 1900
alarm           : 21:16:27
DST_enable      : no
BCD             : yes
24hr            : yes
square_wave     : no
alarm_IRQ       : no
update_IRQ      : no
periodic_IRQ    : no
periodic_freq   : 1024
batt_status     : okay
```

For more information about the RTC, refer to the `/usr/share/doc/kernel-doc-<kernel_version>/Documentation/rtc.txt` file.

2.3.5. `/proc/fs`

This directory shows which file systems are exported. If running an NFS server, typing `cat /proc/fs/nfsd/exports` displays the file systems being shared and the permissions granted for those file systems. For more on file system sharing with NFS, refer to the Network File System (NFS) chapter of the Storage Administration Guide.

2.3.6. `/proc/irq/`

This directory is used to set IRQ to CPU affinity, which allows the system to connect a particular IRQ to only one CPU. Alternatively, it can exclude a CPU from handling any IRQs.

Each IRQ has its own directory, allowing for the individual configuration of each IRQ. The `/proc/irq/prof_cpu_mask` file is a bitmask that contains the default values for the `smp_affinity` file in the IRQ directory. The values in `smp_affinity` specify which CPUs handle that particular IRQ.

For more information about the `/proc/irq/` directory, refer to the `/usr/share/doc/kernel-doc-<kernel_version>/Documentation/filesystems/proc.txt`.

2.3.7. `/proc/net/`

This directory provides a comprehensive look at various networking parameters and statistics. Each directory and virtual file within this directory describes aspects of the system's network configuration. Below is a partial list of the `/proc/net/` directory:

- **arp** — Lists the kernel's ARP table. This file is particularly useful for connecting a hardware address to an IP address on a system.

- **atm/** directory — The files within this directory contain Asynchronous Transfer Mode (ATM) settings and statistics. This directory is primarily used with ATM networking and ADSL cards.

- **dev** — Lists the various network devices configured on the system, complete with transmit and receive statistics. This file displays the number of bytes each interface has sent and received, the number of packets inbound and outbound, the number of errors seen, the number of packets dropped, and more.

- **dev_mcast** — Lists Layer2 multicast groups on which each device is listening.
• **igmp** — Lists the IP multicast addresses which this system joined.

• **ip_conntrack** — Lists tracked network connections for machines that are forwarding IP connections.

• **ip_tables_names** — Lists the types of **iptables** in use. This file is only present if **iptables** is active on the system and contains one or more of the following values: **filter**, **mangle**, or **nat**.

• **ip_mr_cache** — Lists the multicast routing cache.

• **ip_mr_vif** — Lists multicast virtual interfaces.

• **netstat** — Contains a broad yet detailed collection of networking statistics, including TCP timeouts, SYN cookies sent and received, and much more.

• **psched** — Lists global packet scheduler parameters.

• **raw** — Lists raw device statistics.

• **route** — Lists the kernel's routing table.

• **rt_cache** — Contains the current routing cache.

• **snmp** — List of Simple Network Management Protocol (SNMP) data for various networking protocols in use.

• **sockstat** — Provides socket statistics.

• **tcp** — Contains detailed TCP socket information.

• **tr_rif** — Lists the token ring RIF routing table.

• **udp** — Contains detailed UDP socket information.

• **unix** — Lists UNIX domain sockets currently in use.

• **wireless** — Lists wireless interface data.

### 2.3.8. /proc/scsi/

The primary file in this directory is **/proc/scsi/scsi**, which contains a list of every recognized SCSI device. From this listing, the type of device, as well as the model name, vendor, SCSI channel and ID data is available.

For example, if a system contains a SCSI CD-ROM, a tape drive, a hard drive, and a RAID controller, this file looks similar to the following:

```
Attached devices:
Host: scsi1
Channel: 00
Id: 05
Lun: 00
Vendor: NEC
Model: CD-ROM DRIVE:466
Rev: 1.06
Type: CD-ROM
ANSI SCSI revision: 02
Host: scsi1
Channel: 00
Id: 06
```
Each SCSI driver used by the system has its own directory within `/proc/scsi/`, which contains files specific to each SCSI controller using that driver. From the previous example, `aic7xxx/` and `megaraid/` directories are present, since two drivers are in use. The files in each of the directories typically contain an I/O address range, IRQ information, and statistics for the SCSI controller using that driver. Each controller can report a different type and amount of information. The Adaptec AIC-7880 Ultra SCSI host adapter's file in this example system produces the following output:

Adaptec AIC7xxx driver version: 5.1.20/3.2.4
Compile Options:
TCQ Enabled By Default : Disabled
AIC7XXX_PROC_STATS     : Enabled
AIC7XXX_RESET_DELAY    : 5
Adapter Configuration:
SCSI Adapter: Adaptec AIC-7880 Ultra SCSI host adapter
Ultra Narrow Controller     PCI MMAPed
I/O Base: 0xfcffe000
Adapter SEEPROM Config: SEEPROM found and used.
Adaptec SCSI BIOS: Enabled
IRQ: 30
SCBs: Active 0, Max Active 1, Allocated 15, HW 16, Page 255
Interrupts: 33726
BIOS Control Word: 0x18a6
Adapter Control Word: 0x1c5f
Extended Translation: Enabled
Disconnect Enable Flags: 0x00ff
Ultra Enable Flags: 0x0020
Tag Queue Enable Flags: 0x0000
Ordered Queue Tag Flags: 0x0000
Default Tag Queue Depth: 8
Tagged Queue By Device array for aic7xxx
Actual queue depth per device for aic7xxx host instance 1:
{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}
Statistics:

(scsl1:0:5:0) Device using Narrow/Sync transfers at 20.0 MByte/sec, offset 15
Transinfo settings: current(12/15/0/0), goal(12/15/0/0), user(12/15/0/0)
Total transfers 0 (0 reads and 0 writes)
  < 2K  2K+  4K+  8K+  16K+  32K+  64K+  128K+
Reads:  0  0  0  0  0  0  0  0
This output reveals the transfer speed to the SCSI devices connected to the controller based on channel ID, as well as detailed statistics concerning the amount and sizes of files read or written by that device. For example, this controller is communicating with the CD-ROM at 20 megabytes per second, while the tape drive is only communicating at 10 megabytes per second.

### 2.3.9. /proc/sys/

The /proc/sys/ directory is different from others in /proc because it not only provides information about the system but also allows the system administrator to immediately enable and disable kernel features.

**Warning**

Use caution when changing settings on a production system using the various files in the /proc/sys/ directory. Changing the wrong setting may render the kernel unstable, requiring a system reboot.

For this reason, be sure the options are valid for that file before attempting to change any value in /proc/sys/.

A good way to determine if a particular file can be configured, or if it is only designed to provide information, is to list it with the -l option at the shell prompt. If the file is writable, it may be used to configure the kernel. For example, a partial listing of /proc/sys/fs looks like the following:

```
-r--r--r-- 1 root  root 0 May 10 16:14 dentry-state
-rw-r--r-- 1 root  root 0 May 10 16:14 dir-notify-enable
-rw-r--r-- 1 root  root 0 May 10 16:14 file-max
-r--r--r-- 1 root  root 0 May 10 16:14 file-nr
```

In this listing, the files dir-notify-enable and file-max can be written to and, therefore, can be used to configure the kernel. The other files only provide feedback on current settings.

Changing a value within a /proc/sys/ file is done by echoing the new value into the file. For example, to enable the System Request Key on a running kernel, type the command:

```
echo 1 > /proc/sys/kernel/sysrq
```

This changes the value for syscall from 0 (off) to 1 (on).

A few /proc/sys/ configuration files contain more than one value. To correctly send new values to them, place a space character between each value passed with the echo command, such as is done in this example:

```
echo 4 2 45 > /proc/sys/kernel/acct
```
Note

Any configuration changes made using the `echo` command disappear when the system is restarted. To make configuration changes take effect after the system is rebooted, refer to Section 2.4, “Using the sysctl Command”.

The `/proc/sys/` directory contains several subdirectories controlling different aspects of a running kernel.

### 2.3.9.1. /proc/sys/dev/

This directory provides parameters for particular devices on the system. Most systems have at least two directories, `cdrom/` and `raid/`. Customized kernels can have other directories, such as `parport/`, which provides the ability to share one parallel port between multiple device drivers.

The `cdrom/` directory contains a file called `info`, which reveals a number of important CD-ROM parameters:

```
CD-ROM information, Id: cdrom.c 3.20 2003/12/17
  drive name: hdc
  drive speed: 48
  drive # of slots: 1
  Can close tray: 1
  Can open tray: 1
  Can lock tray: 1
  Can change speed: 1
  Can select disk: 0
  Can read multisession: 1
  Can read MCN: 1
  Reports media changed: 1
  Can play audio: 1
  Can write CD-R: 0
  Can write CD-RW: 0
  Can read DVD: 0
  Can write DVD-R: 0
  Can write DVD-RAM: 0
  Can read MRW: 0
  Can write MRW: 0
  Can write RAM: 0
```

This file can be quickly scanned to discover the qualities of an unknown CD-ROM. If multiple CD-ROMs are available on a system, each device is given its own column of information.

Various files in `/proc/sys/dev/cdrom`, such as `autoclose` and `checkmedia`, can be used to control the system's CD-ROM. Use the `echo` command to enable or disable these features.

If RAID support is compiled into the kernel, a `/proc/sys/dev/raid/` directory becomes available with at least two files in it: `speed_limit_min` and `speed_limit_max`. These settings determine the acceleration of RAID devices for I/O intensive tasks, such as resyncing the disks.

### 2.3.9.2. /proc/sys/fs/

This directory contains an array of options and information concerning various aspects of the file system, including quota, file handle, inode, and dentry information.
The `binfmt_misc/` directory is used to provide kernel support for miscellaneous binary formats.

The important files in `/proc/sys/fs/` include:

- **dentry-state** — Provides the status of the directory cache. The file looks similar to the following:

  | 57411 | 52939 | 45  | 0   | 0   | 0   |

  The first number reveals the total number of directory cache entries, while the second number displays the number of unused entries. The third number tells the number of seconds between when a directory has been freed and when it can be reclaimed, and the fourth measures the pages currently requested by the system. The last two numbers are not used and display only zeros.

- **file-max** — Lists the maximum number of file handles that the kernel allocates. Raising the value in this file can resolve errors caused by a lack of available file handles.

- **file-nr** — Lists the number of allocated file handles, used file handles, and the maximum number of file handles.

- **overflowgid** and **overflowuid** — Defines the fixed group ID and user ID, respectively, for use with file systems that only support 16-bit group and user IDs.

### 2.3.9.3. /proc/sys/kernel/

This directory contains a variety of different configuration files that directly affect the operation of the kernel. Some of the most important files include:

- **acct** — Controls the suspension of process accounting based on the percentage of free space available on the file system containing the log. By default, the file looks like the following:

  | 4   | 2   | 30  |

  The first value dictates the percentage of free space required for logging to resume, while the second value sets the threshold percentage of free space when logging is suspended. The third value sets the interval, in seconds, that the kernel polls the file system to see if logging should be suspended or resumed.

- **ctrl-alt-del** — Controls whether **Ctrl+Alt+Delete** gracefully restarts the computer using **init** (0) or forces an immediate reboot without syncing the dirty buffers to disk (1).

- **domainname** — Configures the system domain name, such as **example.com**.

- **exec-shield** — Configures the Exec Shield feature of the kernel. Exec Shield provides protection against certain types of buffer overflow attacks.

  There are two possible values for this virtual file:

  - **0** — Disables Exec Shield.
  - **1** — Enables Exec Shield. This is the default value.
Important

If a system is running security-sensitive applications that were started while Exec Shield was
disabled, these applications must be restarted when Exec Shield is enabled in order for Exec
Shield to take effect.

• **hostname** — Configures the system host name, such as [www.example.com](http://www.example.com).

• **hotplug** — Configures the utility to be used when a configuration change is detected by the
system. This is primarily used with USB and Cardbus PCI. The default value of `/sbin/hotplug`
should not be changed unless testing a new program to fulfill this role.

• **modprobe** — Sets the location of the program used to load kernel modules. The default value is `/sbin/modprobe` which means `kmod` calls it to load the module when a kernel thread calls `kmod`.

• **msgmax** — Sets the maximum size of any message sent from one process to another and is
set to 8192 bytes by default. Be careful when raising this value, as queued messages between
processes are stored in non-swappable kernel memory. Any increase in `msgmax` would increase
RAM requirements for the system.

• **msgmnb** — Sets the maximum number of bytes in a single message queue. The default is **16384**.

• **msgmni** — Sets the maximum number of message queue identifiers. The default is **4008**.

• **osrelease** — Lists the Linux kernel release number. This file can only be altered by changing the
kernel source and recompiling.

• **ostype** — Displays the type of operating system. By default, this file is set to Linux, and this value
 can only be changed by changing the kernel source and recompiling.

• **overflowgid** and **overflowuid** — Defines the fixed group ID and user ID, respectively, for use
with system calls on architectures that only support 16-bit group and user IDs.

• **panic** — Defines the number of seconds the kernel postpones rebooting when the system
experiences a kernel panic. By default, the value is set to **0**, which disables automatic rebooting
after a panic.

• **printk** — This file controls a variety of settings related to printing or logging error messages. Each
error message reported by the kernel has a loglevel associated with it that defines the importance of
the message. The loglevel values break down in this order:

  • 0 — Kernel emergency. The system is unusable.

  • 1 — Kernel alert. Action must be taken immediately.

  • 2 — Condition of the kernel is considered critical.

  • 3 — General kernel error condition.

  • 4 — General kernel warning condition.

  • 5 — Kernel notice of a normal but significant condition.
• 6 — Kernel informational message.

• 7 — Kernel debug-level messages.

Four values are found in the printk file:

| 6 | 4 | 1 | 7 |

Each of these values defines a different rule for dealing with error messages. The first value, called the console loglevel, defines the lowest priority of messages printed to the console. (Note that, the lower the priority, the higher the loglevel number.) The second value sets the default loglevel for messages without an explicit loglevel attached to them. The third value sets the lowest possible loglevel configuration for the console loglevel. The last value sets the default value for the console loglevel.

• random/ directory — Lists a number of values related to generating random numbers for the kernel.

• sem — Configures semaphore settings within the kernel. A semaphore is a System V IPC object that is used to control utilization of a particular process.

• shmall — Sets the total amount of shared memory that can be used at one time on the system, in bytes. By default, this value is 2097152.

• shmmmax — Sets the largest shared memory segment size allowed by the kernel. By default, this value is 33554432. However, the kernel supports much larger values than this.

• shmmni — Sets the maximum number of shared memory segments for the whole system. By default, this value is 4096.

• sysrq — Activates the System Request Key, if this value is set to anything other than zero (0), the default.

The System Request Key allows immediate input to the kernel through simple key combinations. For example, the System Request Key can be used to immediately shut down or restart a system, sync all mounted file systems, or dump important information to the console. To initiate a System Request Key, type \textit{Alt}+\textit{SysRq}+\textit{system request code}. Replace \textit{system request code} with one of the following system request codes:

• r — Disables raw mode for the keyboard and sets it to XLATE (a limited keyboard mode which does not recognize modifiers such as Alt, Ctrl, or Shift for all keys).

• k — Kills all processes active in a virtual console. Also called \textit{Secure Access Key (SAK)}, it is often used to verify that the login prompt is spawned from \textit{init} and not a trojan copy designed to capture usernames and passwords.

• b — Reboots the kernel without first unmounting file systems or syncing disks attached to the system.

• c — Crashes the system without first unmounting file systems or syncing disks attached to the system.

• o — Shuts off the system.

• s — Attempts to sync disks attached to the system.
Chapter 2. The proc File System

- **u** — Attempts to unmount and remount all file systems as read-only.
- **p** — Outputs all flags and registers to the console.
- **t** — Outputs a list of processes to the console.
- **m** — Outputs memory statistics to the console.
- **0** through **9** — Sets the log level for the console.
- **e** — Kills all processes except **init** using SIGTERM.
- **i** — Kills all processes except **init** using SIGKILL.
- **1** — Kills all processes using SIGKILL (including **init**). *The system is unusable after issuing this System Request Key code.*
- **h** — Displays help text.

This feature is most beneficial when using a development kernel or when experiencing system freezes.

**Warning**
The System Request Key feature is considered a security risk because an unattended console provides an attacker with access to the system. For this reason, it is turned off by default.

Refer to the `/usr/share/doc/kernel-doc-kernel_version/Documentation/sysrq.txt` file for more information about the System Request Key.

- **tainted** — Indicates whether a non-GPL module is loaded.
  - **0** — No non-GPL modules are loaded.
  - **1** — At least one module without a GPL license (including modules with no license) is loaded.
  - **2** — At least one module was force-loaded with the command `insmod -f`.
- **threads-max** — Sets the maximum number of threads to be used by the kernel, with a default value of **2048**.
- **version** — Displays the date and time the kernel was last compiled. The first field in this file, such as **#3**, relates to the number of times a kernel was built from the source base.

### 2.3.9.4. /proc/sys/net/

This directory contains subdirectories concerning various networking topics. Various configurations at the time of kernel compilation make different directories available here, such as `ethernet/`, `ipv4/`, `ipx/`, and `ipv6/`. By altering the files within these directories, system administrators are able to adjust the network configuration on a running system.

Given the wide variety of possible networking options available with Linux, only the most common `/proc/sys/net/` directories are discussed.
The `/proc/sys/net/core/` directory contains a variety of settings that control the interaction between the kernel and networking layers. The most important of these files are:

- **message_burst** — Sets the amount of time in tenths of a second required to write a new warning message. This setting is used to mitigate Denial of Service (DoS) attacks. The default setting is 10.

- **message_cost** — Sets a cost on every warning message. The higher the value of this file (default of 5), the more likely the warning message is ignored. This setting is used to mitigate DoS attacks.

The idea of a DoS attack is to bombard the targeted system with requests that generate errors and fill up disk partitions with log files or require all of the system's resources to handle the error logging. The settings in `message_burst` and `message_cost` are designed to be modified based on the system's acceptable risk versus the need for comprehensive logging.

- **netdev_max_backlog** — Sets the maximum number of packets allowed to queue when a particular interface receives packets faster than the kernel can process them. The default value for this file is 1000.

- **optmem_max** — Configures the maximum ancillary buffer size allowed per socket.

- **rmem_default** — Sets the receive socket buffer default size in bytes.

- **rmem_max** — Sets the receive socket buffer maximum size in bytes.

- **wmem_default** — Sets the send socket buffer default size in bytes.

- **wmem_max** — Sets the send socket buffer maximum size in bytes.

The `/proc/sys/net/ipv4/` directory contains additional networking settings. Many of these settings, used in conjunction with one another, are useful in preventing attacks on the system or when using the system to act as a router.

**Warning**

An erroneous change to these files may affect remote connectivity to the system.

The following is a list of some of the more important files within the `/proc/sys/net/ipv4/` directory:

- **icmp_echo_ignore_all** and **icmp_echo_ignore_broadcasts** — Allows the kernel to ignore ICMP ECHO packets from every host or only those originating from broadcast and multicast addresses, respectively. A value of 0 allows the kernel to respond, while a value of 1 ignores the packets.

- **ip_default_ttl** — Sets the default Time To Live (TTL), which limits the number of hops a packet may make before reaching its destination. Increasing this value can diminish system performance.

- **ip_forward** — Permits interfaces on the system to forward packets to one other. By default, this file is set to 0. Setting this file to 1 enables network packet forwarding.

- **ip_local_port_range** — Specifies the range of ports to be used by TCP or UDP when a local port is needed. The first number is the lowest port to be used and the second number specifies the highest port. Any systems that expect to require more ports than the default 1024 to 4999 should use a range from 32768 to 61000.
• **tcp_syn_retries** — Provides a limit on the number of times the system re-transmits a SYN packet when attempting to make a connection.

• **tcp_retries1** — Sets the number of permitted re-transmissions attempting to answer an incoming connection. Default of 3.

• **tcp_retries2** — Sets the number of permitted re-transmissions of TCP packets. Default of 15.


A number of other directories exist within the `/proc/sys/net/ipv4/` directory and each covers a different aspect of the network stack. The `/proc/sys/net/ipv4/conf/` directory allows each system interface to be configured in different ways, including the use of default settings for unconfigured devices (in the `/proc/sys/net/ipv4/conf/default/` subdirectory) and settings that override all special configurations (in the `/proc/sys/net/ipv4/conf/all/` subdirectory).

The `/proc/sys/net/ipv4/neigh/` directory contains settings for communicating with a host directly connected to the system (called a network neighbor) and also contains different settings for systems more than one hop away.

Routing over IPv4 also has its own directory, `/proc/sys/net/ipv4/route/`. Unlike conf/ and neigh/, the `/proc/sys/net/ipv4/route/` directory contains specifications that apply to routing with any interfaces on the system. Many of these settings, such as `max_size`, `max_delay`, and `min_delay`, relate to controlling the size of the routing cache. To clear the routing cache, write any value to the `flush` file.

Additional information about these directories and the possible values for their configuration files can be found in the `/usr/share/doc/kernel-doc-kernel_version/Documentation/filesystems/proc.txt` file.

### 2.3.9.5. /proc/sys/vm/

This directory facilitates the configuration of the Linux kernel's virtual memory (VM) subsystem. The kernel makes extensive and intelligent use of virtual memory, which is commonly referred to as swap space.

The following files are commonly found in the `/proc/sys/vm/` directory:

• **block_dump** — Configures block I/O debugging when enabled. All read/write and block dirtying operations done to files are logged accordingly. This can be useful if diagnosing disk spin up and spin downs for laptop battery conservation. All output when block_dump is enabled can be retrieved via `dmesg`. The default value is 0.

  **Note**

  If block_dump is enabled at the same time as kernel debugging, it is prudent to stop the klogd daemon, as it generates erroneous disk activity caused by block_dump.

• **dirty_background_ratio** — Starts background writeback of dirty data at this percentage of total memory, via a pdflush daemon. The default value is 10.
• **dirty_expire_centisecs** — Defines when dirty in-memory data is old enough to be eligible for writeout. Data which has been dirty in-memory for longer than this interval is written out next time a pdflush daemon wakes up. The default value is 3000, expressed in hundredths of a second.

• **dirty_ratio** — Starts active writeback of dirty data at this percentage of total memory for the generator of dirty data, via pdflush. The default value is 20.

• **dirty_writeback_centisecs** — Defines the interval between pdflush daemon wakeups, which periodically writes dirty in-memory data out to disk. The default value is 500, expressed in hundredths of a second.

• **laptop_mode** — Minimizes the number of times that a hard disk needs to spin up by keeping the disk spun down for as long as possible, therefore conserving battery power on laptops. This increases efficiency by combining all future I/O processes together, reducing the frequency of spin ups. The default value is 0, but is automatically enabled in case a battery on a laptop is used.

This value is controlled automatically by the acpid daemon once a user is notified battery power is enabled. No user modifications or interactions are necessary if the laptop supports the ACPI (Advanced Configuration and Power Interface) specification.

For more information, refer to the `/usr/share/doc/kernel-doc-kernel_version/Documentation/laptop-mode.txt` file.

• **max_map_count** — Configures the maximum number of memory map areas a process may have. In most cases, the default value of 65536 is appropriate.

• **min_free_kbytes** — Forces the Linux VM (virtual memory manager) to keep a minimum number of kilobytes free. The VM uses this number to compute a `pages_min` value for each `lowmem` zone in the system. The default value is in respect to the total memory on the machine.

• **nr_hugepages** — Indicates the current number of configured hugetlb pages in the kernel.

For more information, refer to the `/usr/share/doc/kernel-doc-kernel_version/Documentation/vm/hugetlbpage.txt` file.

• **nr_pdflush_threads** — Indicates the number of pdflush daemons that are currently running. This file is read-only, and should not be changed by the user. Under heavy I/O loads, the default value of two is increased by the kernel.

• **overcommit_memory** — Configures the conditions under which a large memory request is accepted or denied. The following three modes are available:

  • **0** — The kernel performs heuristic memory over commit handling by estimating the amount of memory available and failing requests that are blatantly invalid. Unfortunately, since memory is allocated using a heuristic rather than a precise algorithm, this setting can sometimes allow available memory on the system to be overloaded. This is the default setting.

  • **1** — The kernel performs no memory over commit handling. Under this setting, the potential for memory overload is increased, but so is performance for memory intensive tasks (such as those executed by some scientific software).

  • **2** — The kernel fails any request for memory that would cause the total address space to exceed the sum of the allocated swap space and the percentage of physical RAM specified in `/proc/sys/vm/overcommit_ratio`. This setting is best for those who desire less risk of memory overcommitment.
Note

This setting is only recommended for systems with swap areas larger than physical memory.

- **overcommit_ratio** — Specifies the percentage of physical RAM considered when `/proc/sys/vm/overcommit_memory` is set to 2. The default value is 50.

- **page-cluster** — Sets the number of pages read in a single attempt. The default value of 3, which actually relates to 16 pages, is appropriate for most systems.

- **swappiness** — Determines how much a machine should swap. The higher the value, the more swapping occurs. The default value, as a percentage, is set to 60.

All kernel-based documentation can be found in the `/usr/share/doc/kernel-doc-kernel_version/Documentation/` directory, which contains additional information.

### 2.3.10. /proc/sysvipc/

This directory contains information about System V IPC resources. The files in this directory relate to System V IPC calls for messages (`msg`), semaphores (`sem`), and shared memory (`shm`).

### 2.3.11. /proc/tty/

This directory contains information about the available and currently used `tty` devices on the system. Originally called teletype devices, any character-based data terminals are called `tty` devices.

In Linux, there are three different kinds of `tty` devices. **Serial devices** are used with serial connections, such as over a modem or using a serial cable. **Virtual terminals** create the common console connection, such as the virtual consoles available when pressing `Alt+<F-key>` at the system console. **Pseudo terminals** create a two-way communication that is used by some higher level applications, such as XFree86. The `drivers` file is a list of the current `tty` devices in use, as in the following example:

<table>
<thead>
<tr>
<th>serial</th>
<th>/dev/cua</th>
<th>5</th>
<th>64-127</th>
<th>serial:callout</th>
</tr>
</thead>
<tbody>
<tr>
<td>serial</td>
<td>/dev/ttyS</td>
<td>4</td>
<td>64-127</td>
<td>serial</td>
</tr>
<tr>
<td>pty_slave</td>
<td>/dev/pts</td>
<td>136</td>
<td>0-255</td>
<td>pty:slave</td>
</tr>
<tr>
<td>pty_master</td>
<td>/dev/ptm</td>
<td>128</td>
<td>0-255</td>
<td>pty:master</td>
</tr>
<tr>
<td>pty_slave</td>
<td>/dev/tty</td>
<td>3</td>
<td>0-255</td>
<td>pty:slave</td>
</tr>
<tr>
<td>pty_master</td>
<td>/dev/pty</td>
<td>2</td>
<td>0-255</td>
<td>pty:master</td>
</tr>
<tr>
<td>/dev/vc/0</td>
<td>/dev/vc/0</td>
<td>4</td>
<td>0</td>
<td>system:vtmaster</td>
</tr>
<tr>
<td>/dev/ptmx</td>
<td>/dev/ptmx</td>
<td>5</td>
<td>2</td>
<td>system</td>
</tr>
<tr>
<td>/dev/console</td>
<td>/dev/console</td>
<td>5</td>
<td>1</td>
<td>system:console</td>
</tr>
<tr>
<td>/dev/tty</td>
<td>/dev/tty</td>
<td>5</td>
<td>0</td>
<td>system:/dev/tty</td>
</tr>
<tr>
<td>unknown</td>
<td>/dev/vc/%d</td>
<td>4</td>
<td>1-63</td>
<td>console</td>
</tr>
</tbody>
</table>

The `/proc/tty/driver/serial` file lists the usage statistics and status of each of the serial `tty` lines.

In order for `tty` devices to be used as network devices, the Linux kernel enforces **line discipline** on the device. This allows the driver to place a specific type of header with every block of data transmitted over the device, making it possible for the remote end of the connection to treat a block of data as just
one in a stream of data blocks. SLIP and PPP are common line disciplines, and each are commonly used to connect systems to one other over a serial link.

### 2.3.12. `/proc/PID/`

Out of Memory (OOM) refers to a computing state where all available memory, including swap space, has been allocated. When this situation occurs, it will cause the system to panic and stop functioning as expected. There is a switch that controls OOM behavior in `/proc/sys/vm/panic_on_oom`. When set to `1` the kernel will panic on OOM. A setting of `0` instructs the kernel to call a function named `oom_killer` on an OOM. Usually, `oom_killer` can kill rogue processes and the system will survive.

The easiest way to change this is to echo the new value to `/proc/sys/vm/panic_on_oom`.

```bash
# cat /proc/sys/vm/panic_on_oom
1
# echo 0 > /proc/sys/vm/panic_on_oom
# cat /proc/sys/vm/panic_on_oom
0
```

It is also possible to prioritize which processes get killed by adjusting the `oom_killer` score. In `/proc/PID/` there are two tools labeled `oom_adj` and `oom_score`. Valid scores for `oom_adj` are in the range -16 to +15. To see the current `oom_killer` score, view the `oom_score` for the process. `oom_killer` will kill processes with the highest scores first.

This example adjusts the `oom_score` of a process with a `PID` of 12465 to make it less likely that `oom_killer` will kill it.

```bash
# cat /proc/12465/oom_score
79872
# echo -5 > /proc/12465/oom_adj
# cat /proc/12465/oom_score
78
```

There is also a special value of `-17`, which disables `oom_killer` for that process. In the example below, `oom_score` returns a value of 0, indicating that this process would not be killed.

```bash
# cat /proc/12465/oom_score
78
# echo -17 > /proc/12465/oom_adj
# cat /proc/12465/oom_score
0
```

A function called `badness()` is used to determine the actual score for each process. This is done by adding up 'points' for each examined process. The process scoring is done in the following way:

1. The basis of each process's score is its memory size.
2. The memory size of any of the process's children (not including a kernel thread) is also added to the score.
3. The process's score is increased for 'niced' processes and decreased for long running processes.
4. Processes with the CAP_SYS_ADMIN and CAP_SYS_RAWIO capabilities have their scores reduced.

5. The final score is then bitshifted by the value saved in the oom_adj file.

Thus, a process with the highest oom_score value will most probably be a non-privileged, recently started process that, along with its children, uses a large amount of memory, has been 'niced', and handles no raw I/O.

### 2.4. Using the sysctl Command

The /sbin/sysctl command is used to view, set, and automate kernel settings in the /proc/sys/ directory.

For a quick overview of all settings configurable in the /proc/sys/ directory, type the /sbin/sysctl -a command as root. This creates a large, comprehensive list, a small portion of which looks something like the following:

```
net.ipv4.route.min_delay = 2 kernel.sysrq = 0 kernel.sem = 250 32000 32 128
```

This is the same information seen if each of the files were viewed individually. The only difference is the file location. For example, the /proc/sys/net/ipv4/route/min_delay file is listed as net.ipv4.route.min_delay, with the directory slashes replaced by dots and the proc.sys portion assumed.

The `sysctl` command can be used in place of `echo` to assign values to writable files in the /proc/sys/ directory. For example, instead of using the command `echo 1 > /proc/sys/kernel/sysrq`

```bash
sysctl -w kernel.sysrq="1"
```

While quickly setting single values like this in /proc/sys/ is helpful during testing, this method does not work as well on a production system as special settings within /proc/sys/ are lost when the machine is rebooted. To preserve custom settings, add them to the /etc/sysctl.conf file.

Each time the system boots, the init program runs the /etc/rc.d/rc.sysinit script. This script contains a command to execute `sysctl` using /etc/sysctl.conf to determine the values passed to the kernel. Any values added to /etc/sysctl.conf therefore take effect each time the system boots.

### 2.5. Additional Resources

Below are additional sources of information about proc file system.

#### Installed Documentation

Some of the best documentation about the proc file system is installed on the system by default.

- /usr/share/doc/kernel-doc-kernel_version/Documentation/filesystems/proc.txt — Contains assorted, but limited, information about all aspects of the /proc directory.
• `/usr/share/doc/kernel-doc-kernel_version/Documentation/sysrq.txt` — An overview of System Request Key options.

• `/usr/share/doc/kernel-doc-kernel_version/Documentation/sysctl/` — A directory containing a variety of `sysctl` tips, including modifying values that concern the kernel (`kernel.txt`), accessing file systems (`fs.txt`), and virtual memory use (`vm.txt`).


**Online Documentation**

• [http://www.linuxhq.com/](http://www.linuxhq.com/) — This website maintains a complete database of source, patches, and documentation for various versions of the Linux kernel.
The sysconfig Directory

This chapter outlines some of the files and directories found in the `/etc/sysconfig/` directory, their function, and their contents. The information in this chapter is not intended to be complete, as many of these files have a variety of options that are only used in very specific or rare circumstances.

**Note**

The actual content of your `/etc/sysconfig/` directory depends on the programs you have installed on your machine. To find the name of the package the configuration file belongs to, type the following at a shell prompt as `root`:

```bash
yum provides /etc/sysconfig/file_name
```

For more information on how to install new packages in Fedora 20, refer to the *Fedora System Administrator's Guide*.

### 3.1. Files in the `/etc/sysconfig/` Directory

The following sections offer descriptions of files normally found in the `/etc/sysconfig/` directory.

#### 3.1.1. `/etc/sysconfig/arpwatch`

The `/etc/sysconfig/arpwatch` file is used to pass arguments to the `arpwatch` daemon at boot time. By default, it contains the following option:

```
OPTIONS=value
```

Additional options to be passed to the `arpwatch` daemon. For example:

```
OPTIONS="-u arpwatch -e root -s 'root (Arpwatch)"
```

#### 3.1.2. `/etc/sysconfig/authconfig`

The `/etc/sysconfig/authconfig` file sets the authorization to be used on the host. By default, it contains the following options:

**USEMKHOMEDIR=boolean**

A Boolean to enable (yes) or disable (no) creating a home directory for a user on the first login. For example:

```
USEMKHOMEDIR=no
```

**USEPAMACCESS=boolean**

A Boolean to enable (yes) or disable (no) the PAM authentication. For example:

```
USEPAMACCESS=no
```

**USESSSDAUTH=boolean**

A Boolean to enable (yes) or disable (no) the SSSD authentication. For example:
### USESSDAUTH = boolean
A Boolean to enable (yes) or disable (no) shadow passwords. For example:

```
USESSDAUTH=no
```

### USESHADOW = boolean
A Boolean to enable (yes) or disable (no) shadow passwords. For example:

```
USESHADOW=yes
```

### USEWINBIND = boolean
A Boolean to enable (yes) or disable (no) using Winbind for user account configuration. For example:

```
USEWINBIND=no
```

### USEDB = boolean
A Boolean to enable (yes) or disable (no) the FAS authentication. For example:

```
USEDB=no
```

### USEFPRINTD = boolean
A Boolean to enable (yes) or disable (no) the fingerprint authentication. For example:

```
USEFPRINTD=yes
```

### FORCESMARTCARD = boolean
A Boolean to enable (yes) or disable (no) enforcing the smart card authentication. For example:

```
FORCESMARTCARD=no
```

### PASSWDALGORITHM = value
The password algorithm. The value can be bigcrypt, descrypt, md5, sha256, or sha512. For example:

```
PASSWDALGORITHM=sha512
```

### USELDAPAUTH = boolean
A Boolean to enable (yes) or disable (no) the LDAP authentication. For example:

```
USELDAPAUTH=no
```

### USELOCAUTHORIZE = boolean
A Boolean to enable (yes) or disable (no) the local authorization for local users. For example:

```
USELOCAUTHORIZE=yes
```

### USECRACKLIB = boolean
A Boolean to enable (yes) or disable (no) using the CrackLib. For example:

```
USECRACKLIB=yes
```
<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>USEWINBINDAUTH=boolean</td>
<td>A Boolean to enable (yes) or disable (no) the Winbind authentication. For example:</td>
<td>USEWINBINDAUTH=no</td>
</tr>
<tr>
<td>USESMARTCARD=boolean</td>
<td>A Boolean to enable (yes) or disable (no) the smart card authentication. For example:</td>
<td>USESMARTCARD=no</td>
</tr>
<tr>
<td>USELDAP=boolean</td>
<td>A Boolean to enable (yes) or disable (no) using LDAP for user account configuration. For example:</td>
<td>USELDAP=no</td>
</tr>
<tr>
<td>USENIS=boolean</td>
<td>A Boolean to enable (yes) or disable (no) using NIS for user account configuration. For example:</td>
<td>USENIS=no</td>
</tr>
<tr>
<td>USEKERBEROS=boolean</td>
<td>A Boolean to enable (yes) or disable (no) the Kerberos authentication. For example:</td>
<td>USEKERBEROS=no</td>
</tr>
<tr>
<td>USESYSNETAUTH=boolean</td>
<td>A Boolean to enable (yes) or disable (no) authenticating system accounts with network services. For example:</td>
<td>USESYSNETAUTH=no</td>
</tr>
<tr>
<td>USESMBAUTH=boolean</td>
<td>A Boolean to enable (yes) or disable (no) the SMB authentication. For example:</td>
<td>USESMBAUTH=no</td>
</tr>
<tr>
<td>USESSSD=boolean</td>
<td>A Boolean to enable (yes) or disable (no) using SSSD for obtaining user information. For example:</td>
<td>USESSSD=no</td>
</tr>
<tr>
<td>USEHESIOD=boolean</td>
<td>A Boolean to enable (yes) or disable (no) using the Hesoid name service. For example:</td>
<td>USEHESIOD=no</td>
</tr>
</tbody>
</table>
For more information on how to configure system authentication on Fedora 20, refer to the *Fedora System Administrator's Guide*.

### 3.1.3. /etc/sysconfig/autofs

The */etc/sysconfig/autofs* file defines custom options for the automatic mounting of devices. This file controls the operation of the automount daemons, which automatically mount file systems when you use them and unmount them after a period of inactivity. File systems can include network file systems, CD-ROM drives, diskettes, and other media.

By default, it contains the following options:

**MASTER\_MAP\_NAME=value**

The default name for the master map. For example:

```
MASTER_MAP_NAME="auto.master"
```

**TIMEOUT=value**

The default mount timeout. For example:

```
TIMEOUT=300
```

**NEGATIVE\_TIMEOUT=value**

The default negative timeout for unsuccessful mount attempts. For example:

```
NEGATIVE_TIMEOUT=60
```

**MOUNT\_WAIT=value**

The time to wait for a response from `mount`. For example:

```
MOUNT_WAIT=-1
```

**UMOUNT\_WAIT=value**

The time to wait for a response from `umount`. For example:

```
UMOUNT_WAIT=12
```

**BROWSE\_MODE=boolean**

A Boolean to enable (yes) or disable (no) browsing the maps. For example:

```
BROWSE_MODE="no"
```

**MOUNT\_NFS\_DEFAULT\_PROTOCOL=value**

The default protocol to be used by `mount.nfs`. For example:

```
MOUNT_NFS_DEFAULT_PROTOCOL=4
```

**APPEND\_OPTIONS=boolean**

A Boolean to enable (yes) or disable (no) appending the global options instead of replacing them. For example:
APPEND_OPTIONS="yes"

**LOGGING=value**
The default logging level. The value has to be either **none**, **verbose**, or **debug**. For example:

```
LOGGING="none"
```

**LDAP_URI=value**
A space-separated list of server URIs in the form of `protocol://server`. For example:

```
LDAP_URI="ldaps://ldap.example.com/
```

**LDAP_TIMEOUT=value**
The synchronous API calls timeout. For example:

```
LDAP_TIMEOUT=-1
```

**LDAP_NETWORK_TIMEOUT=value**
The network response timeout. For example:

```
LDAP_NETWORK_TIMEOUT=8
```

**SEARCH_BASE=value**
The base Distinguished Name (DN) for the map search. For example:

```
SEARCH_BASE=""
```

**AUTH_CONF_FILE=value**
The default location of the SASL authentication configuration file. For example:

```
AUTH_CONF_FILE="/etc/autofs_ldap_auth.conf"
```

**MAP_HASH_TABLE_SIZE=value**
The hash table size for the map cache. For example:

```
MAP_HASH_TABLE_SIZE=1024
```

**USE_MISC_DEVICE=boolean**
A Boolean to enable (yes) or disable (no) using the autofs miscellaneous device. For example:

```
USE_MISCDEVICE="yes"
```

**OPTIONS=value**
Additional options to be passed to the LDAP daemon. For example:

```
OPTIONS=""
```
3.1.4. /etc/sysconfig/clock

The /etc/sysconfig/clock file controls the interpretation of values read from the system hardware clock. It is used by the Date/Time Properties tool, and should not be edited by hand. By default, it contains the following option:

ZONE=value

The time zone file under /usr/share/zoneinfo that /etc/localtime is a copy of. For example:

ZONE="Europe/Prague"

For more information on date and time configuration, refer to the Fedora System Administrator’s Guide.

3.1.5. /etc/sysconfig/dhcpd

The /etc/sysconfig/dhcpd file is used to pass arguments to the dhcpd daemon at boot time. By default, it contains the following options:

DHCPDARGS=value

Additional options to be passed to the dhcpd daemon. For example:

DHCPDARGS=

For more information on DHCP and its usage, refer to the Fedora Networking Guide.

3.1.6. /etc/sysconfig/firstboot

The /etc/sysconfig/firstboot file defines whether to run the firstboot utility. By default, it contains the following option:

RUN_FIRSTBOOT=boolean

A Boolean to enable (YES) or disable (NO) running the firstboot program. For example:

RUN_FIRSTBOOT=NO

The first time the system boots, the init program calls the /etc/rc.d/init.d/firstboot script, which looks for the /etc/sysconfig/firstboot file. If this file does not contain the RUN_FIRSTBOOT=NO option, the firstboot program is run, guiding a user through the initial configuration of the system.

Note

To start the firstboot program the next time the system boots, change the value of RUN_FIRSTBOOT option to YES, and type the following at a shell prompt as root:

systemctl enable firstboot-graphical.service
3.1.7. /etc/sysconfig/i18n

The `/etc/sysconfig/i18n` configuration file defines the default language, any supported languages, and the default system font. By default, it contains the following options:

**LANG=value**

The default language. For example:

```
LANG="en_US.UTF-8"
```

**SUPPORTED=value**

A colon-separated list of supported languages. For example:

```
SUPPORTED="en_US.UTF-8:en_US:en"
```

**SYSFONT=value**

The default system font. For example:

```
SYSFONT="latarcyrheb-sun16"
```

3.1.8. /etc/sysconfig/init

The `/etc/sysconfig/init` file controls how the system appears and functions during the boot process. By default, it contains the following options:

**BOOTUP=value**

The bootup style. The value has to be either color (the standard color boot display), verbose (an old style display which provides more information), or anything else for the new style display, but without ANSI formatting. For example:

```
BOOTUP=color
```

**RES_COL=value**

The number of the column in which the status labels start. For example:

```
RES_COL=60
```

**MOVE_TO_COL=value**

The terminal sequence to move the cursor to the column specified in `RES_COL` (see above). For example:

```
MOVE_TO_COL="echo -en \033[${RES_COL}G"
```

**SETCOLOR_SUCCESS=value**

The terminal sequence to set the success color. For example:

```
SETCOLOR_SUCCESS="echo -en \033[0;32m"
```

**SETCOLOR_FAILURE=value**

The terminal sequence to set the failure color. For example:
Chapter 3. The sysconfig Directory

3.1.9. /etc/sysconfig/ip6tables-config

The /etc/sysconfig/ip6tables-config file stores information used by the kernel to set up IPv6 packet filtering at boot time or whenever the ip6tables service is started. Note that you should not modify it unless you are familiar with ip6tables rules. By default, it contains the following options:

**IP6TABLES_MODULES=value**
A space-separated list of helpers to be loaded after the firewall rules are applied. For example:

```
IP6TABLES_MODULES=
```
IP6TABLES_MODULES="ip_nat_ftp ip_nat_irc"

**IP6TABLES_MODULES_UNLOAD=boolean**
A Boolean to enable *(yes)* or disable *(no)* module unloading when the firewall is stopped or restarted. For example:

```bash
IP6TABLES_MODULES_UNLOAD="yes"
```

**IP6TABLES_SAVE_ON_STOP=boolean**
A Boolean to enable *(yes)* or disable *(no)* saving the current firewall rules when the firewall is stopped. For example:

```bash
IP6TABLES_SAVE_ON_STOP="no"
```

**IP6TABLES_SAVE_ON_RESTART=boolean**
A Boolean to enable *(yes)* or disable *(no)* saving the current firewall rules when the firewall is restarted. For example:

```bash
IP6TABLES_SAVE_ON_RESTART="no"
```

**IP6TABLES_SAVE_COUNTER=boolean**
A Boolean to enable *(yes)* or disable *(no)* saving the rule and chain counters. For example:

```bash
IP6TABLES_SAVE_COUNTER="no"
```

**IP6TABLES_STATUS_NUMERIC=boolean**
A Boolean to enable *(yes)* or disable *(no)* printing IP addresses and port numbers in a numeric format in the status output. For example:

```bash
IP6TABLES_STATUS_NUMERIC="yes"
```

**IP6TABLES_STATUS_VERBOSE=boolean**
A Boolean to enable *(yes)* or disable *(no)* printing information about the number of packets and bytes in the status output. For example:

```bash
IP6TABLES_STATUS_VERBOSE="no"
```

**IP6TABLES_STATUS_LINENUMBERS=boolean**
A Boolean to enable *(yes)* or disable *(no)* printing line numbers in the status output. For example:

```bash
IP6TABLES_STATUS_LINENUMBERS="yes"
```
Note

You can create the rules manually using the `ip6tables` command. Once created, type the following at a shell prompt as root:

```
service ip6tables save
```

This will add the rules to the `/etc/sysconfig/ip6tables` file. Once this file exists, any firewall rules saved in it persist through a system reboot or a service restart.

3.1.10. `/etc/sysconfig/keyboard`

The `/etc/sysconfig/keyboard` file controls the behavior of the keyboard. By default, it contains the following options:

**KEYTABLE=value**

The name of a keytable file. The files that can be used as keytables start in the `/lib/kbd/keymaps/i386/` directory, and branch into different keyboard layouts from there, all labeled `value.kmap.gz`. The first file name that matches the `KEYTABLE` setting is used. For example:

```
KEYTABLE="us"
```

**MODEL=value**

The keyboard model. For example:

```
MODEL="pc105+inet"
```

**LAYOUT=value**

The keyboard layout. For example:

```
LAYOUT="us"
```

**KEYBOARDTYPE=value**

The keyboard type. Allowed values are `pc` (a PS/2 keyboard), or `sun` (a Sun keyboard). For example:

```
KEYBOARDTYPE="pc"
```

3.1.11. `/etc/sysconfig/ldap`

The `/etc/sysconfig/ldap` file holds the basic configuration for the LDAP server. By default, it contains the following options:

**SLAPD_OPTIONS=value**

Additional options to be passed to the `slapd` daemon. For example:

```
SLAPD_OPTIONS="-4"
```
SLURPD_OPTIONS=value
Additional options to be passed to the slurpd daemon. For example:

```
SLURPD_OPTIONS=""
```

SLAPD_LDAP=boolean
A Boolean to enable (yes) or disable (no) using the LDAP over TCP (that is, ldap:///). For example:

```
SLAPD_LDAP="yes"
```

SLAPD_LDAPI=boolean
A Boolean to enable (yes) or disable (no) using the LDAP over IPC (that is, ldapi:///). For example:

```
SLAPD_LDAPI="no"
```

SLAPD_LDAPS=boolean
A Boolean to enable (yes) or disable (no) using the LDAP over TLS (that is, ldaps:///). For example:

```
SLAPD_LDAPS="no"
```

SLAPD_URLS=value
A space-separated list of URLs. For example:

```
SLAPD_URLS="ldapi:///var/lib/ldap_root/ldapi ldapi:/// ldaps:///
```

SLAPD_SHUTDOWN_TIMEOUT=value
The time to wait for slapd to shut down. For example:

```
SLAPD_SHUTDOWN_TIMEOUT=3
```

SLAPD_ULIMIT_SETTINGS=value
The parameters to be passed to ulimit before the slapd daemon is started. For example:

```
SLAPD_ULIMIT_SETTINGS=""
```

For more information on OpenLDAP and its configuration, refer to the Fedora System Administrator’s Guide.

### 3.1.12. /etc/sysconfig/named

The /etc/sysconfig/named file is used to pass arguments to the named daemon at boot time. By default, it contains the following options:

ROOTDIR=value
The chroot environment under which the named daemon runs. The value has to be a full directory path. For example:
Chapter 3. The sysconfig Directory

ROOTDIR="/var/named/chroot"

Note that the chroot environment has to be configured first (type `info chroot` at a shell prompt for more information).

OPTIONS=`value`
Additional options to be passed to `named`. For example:

OPTIONS="-6"

Note that you should not use the `-t` option. Instead, use `ROOTDIR` as described above.

KEYTAB_FILE=`value`
The keytab file name. For example:

KEYTAB_FILE="/etc/named.keytab"

For more information on the BIND and DNS servers in general, refer to the *Fedora Networking Guide*.

### 3.1.13. /etc/sysconfig/network

The `/etc/sysconfig/network` file is used to specify information about the desired network configuration. By default, it contains the following options:

NETWORKING=`boolean`
A Boolean to enable (yes) or disable (no) the networking. For example:

NETWORKING=yes

HOSTNAME=`value`
The hostname of the machine. For example:

HOSTNAME=penguin.example.com

GATEWAY=`value`
The IP address of the network's gateway. For example:

GATEWAY=192.168.1.0

**Warning**

Do not use custom init scripts to configure network settings. When performing a post-boot network service restart, custom init scripts configuring network settings that are run outside of the network init script lead to unpredictable results.

### 3.1.14. /etc/sysconfig/ntpd

The `/etc/sysconfig/ntpd` file is used to pass arguments to the `ntpd` daemon at boot time. By default, it contains the following option:
OPTIONS=value
Additional options to be passed to ntpd. For example:

```
OPTIONS="-u ntp:ntp -p /var/run/ntpd.pid -g"
```

For more information on NTP and how to configure it, refer to the *Fedora System Administrator's Guide*.

### 3.1.15. /etc/sysconfig/quagga

The /etc/sysconfig/quagga file holds the basic configuration for Quagga daemons. By default, it contains the following options:

**QCONFDIR=value**
The directory with the configuration files for Quagga daemons. For example:

```
QCONFDIR="/etc/quagga"
```

**BGPD_OPTS=value**
Additional options to be passed to the bgpd daemon. For example:

```
BGPD_OPTS="-A 127.0.0.1 -f ${QCONFDIR}/bgpd.conf"
```

**OSPF6D_OPTS=value**
Additional options to be passed to the ospf6d daemon. For example:

```
OSPF6D_OPTS="-A ::1 -f ${QCONFDIR}/ospf6d.conf"
```

**OSPFD_OPTS=value**
Additional options to be passed to the ospfd daemon. For example:

```
OSPFD_OPTS="-A 127.0.0.1 -f ${QCONFDIR}/ospfd.conf"
```

**RIPD_OPTS=value**
Additional options to be passed to the ripd daemon. For example:

```
RIPD_OPTS="-A 127.0.0.1 -f ${QCONFDIR}/ripd.conf"
```

**RIPNGD_OPTS=value**
Additional options to be passed to the ripngd daemon. For example:

```
RIPNGD_OPTS="-A ::1 -f ${QCONFDIR}/ripngd.conf"
```

**ZEBRA_OPTS=value**
Additional options to be passed to the zebra daemon. For example:

```
ZEBRA_OPTS="-A 127.0.0.1 -f ${QCONFDIR}/zebra.conf"
```

**ISISD_OPTS=value**
Additional options to be passed to the isisd daemon. For example:
Chapter 3. The sysconfig Directory

ISISD_OPTS="-A ::1 -f ${QCONFDIR}/isisd.conf"

WATCH_OPTS=value
Additional options to be passed to the watchquagga daemon. For example:

WATCH_OPTS="-Az -b_ -r/sbin/service_%s_restart -s/sbin/service_%s_start -k/sbin/service_%s_stop"

WATCH_DAEMONS=value
A space separated list of monitored daemons. For example:

WATCH_DAEMONS="zebra bgpd ospfd ospf6d ripd ripngd"

3.1.16. /etc/sysconfig/radvd
The /etc/sysconfig/radvd file is used to pass arguments to the radvd daemon at boot time. By default, it contains the following option:

OPTIONS=value
Additional options to be passed to the radvd daemon. For example:

OPTIONS="-u radvd"

3.1.17. /etc/sysconfig/samba
The /etc/sysconfig/samba file is used to pass arguments to the Samba daemons at boot time. By default, it contains the following options:

SMBDOPTIONS=value
Additional options to be passed to smbd. For example:

SMBDOPTIONS="-D"

NMBDOPTIONS=value
Additional options to be passed to nmbd. For example:

NMBDOPTIONS="-D"

WINBINDOPTIONS=value
Additional options to be passed to winbindd. For example:

WINBINDOPTIONS=""

For more information on Samba and its configuration, refer to the Fedora System Administrator’s Guide.

3.1.18. /etc/sysconfig/saslauthd
The /etc/sysconfig/saslauthd file is used to control which arguments are passed to saslauthd, the SASL authentication server. By default, it contains the following options:
**SOCKETDIR=value**
The directory for the saslauthd's listening socket. For example:

```bash
SOCKETDIR=/var/run/saslauthd
```

**MECH=value**
The authentication mechanism to use to verify user passwords. For example:

```bash
MECH=pam
```

**DAEMONOPTS=value**
Options to be passed to the daemon() function that is used by the `/etc/rc.d/init.d/saslauthd` init script to start the saslauthd service. For example:

```bash
DAEMONOPTS="--user saslauth"
```

**FLAGS=value**
Additional options to be passed to the saslauthd service. For example:

```bash
FLAGS=
```

### 3.1.19. /etc/sysconfig/selinux

The `/etc/sysconfig/selinux` file contains the basic configuration options for SELinux. It is a symbolic link to `/etc/selinux/config`, and by default, it contains the following options:

**SELINUX=value**
The security policy. The `value` can be either enforcing (the security policy is always enforced), permissive (instead of enforcing the policy, appropriate warnings are displayed), or disabled (no policy is used). For example:

```bash
SELINUX=enforcing
```

**SELINUNXTYPE=value**
The protection type. The `value` can be either targeted (the targeted processes are protected), or mls (the Multi Level Security protection). For example:

```bash
SELINUNXTYPE=targeted
```

### 3.1.20. /etc/sysconfig/sendmail

The `/etc/sysconfig/sendmail` file is used to set the default values for the Sendmail application. By default, it contains the following values:

**DAEMON=boolean**
A Boolean to enable (yes) or disable (no) running `sendmail` as a daemon. For example:

```bash
DAEMON=yes
```

**QUEUE=value**
The interval at which the messages are to be processed. For example:
For more information on Sendmail and its configuration, refer to the *Fedora System Administrator's Guide*.

### 3.1.21. /etc/sysconfig/spamassassin

The `/etc/sysconfig/spamassassin` file is used to pass arguments to the `spamd` daemon (a daemonized version of *SpamAssassin*) at boot time. By default, it contains the following option:

**SPAMDOPTIONS=** *value*

Additional options to be passed to the `spamd` daemon. For example:

```
SPAMDOPTIONS="-d -c -m5 -H"
```

For more information on SpamAssassin and its configuration, refer to the *Fedora System Administrator's Guide*.

### 3.1.22. /etc/sysconfig/squid

The `/etc/sysconfig/squid` file is used to pass arguments to the `squid` daemon at boot time. By default, it contains the following options:

**SQUID_OPTS=** *value*

Additional options to be passed to the `squid` daemon. For example:

```
SQUID_OPTS=""
```

**SQUID_SHUTDOWN_TIMEOUT=** *value*

The time to wait for `squid` daemon to shut down. For example:

```
SQUID_SHUTDOWN_TIMEOUT=100
```

**SQUID_CONF=** *value*

The default configuration file. For example:

```
SQUID_CONF="/etc/squid/squid.conf"
```

### 3.1.23. /etc/sysconfig/system-config-users

The `/etc/sysconfig/system-config-users` file is the configuration file for the *User Manager* tool, and should not be edited by hand. By default, it contains the following options:

**FILTER=** *boolean*

A Boolean to enable (true) or disable (false) filtering of system users. For example:

```
FILTER=true
```

**ASSIGN_HIGHEST_UID=** *boolean*

A Boolean to enable (true) or disable (false) assigning the highest available UID to newly added users. For example:
ASSIGN_HIGHEST_GID = boolean
A Boolean to enable (true) or disable (false) assigning the highest available GID to newly added groups. For example:

ASSIGN_HIGHEST_GID = true

PREFER_SAME_UID_GID = boolean
A Boolean to enable (true) or disable (false) using the same UID and GID for newly added users when possible. For example:

PREFER_SAME_UID_GID = true

For more information on the User Manager tool and its usage, refer to the Fedora System Administrator’s Guide.

3.1.24. /etc/sysconfig/vncservers
The /etc/sysconfig/vncservers file configures the way the Virtual Network Computing (VNC) server starts up. By default, it contains the following options:

VNCSESSIONS = value
A list of space separated display:username pairs. For example:

VNCSESSIONS = "2:myusername"

VNCSERVERARGS[display] = value
Additional arguments to be passed to the VNC server running on the specified display. For example:

VNCSERVERARGS[2] = "-geometry 800x600 -nolisten tcp -localhost"

3.1.25. /etc/sysconfig/xinetd
The /etc/sysconfig/xinetd file is used to pass arguments to the xinetd daemon at boot time. By default, it contains the following options:

EXTRAOPTIONS = value
Additional options to be passed to xinetd. For example:

EXTRAOPTIONS = ""

XINETD_LANG = value
The locale information to be passed to every service started by xinetd. Note that to remove locale information from the xinetd environment, you can use an empty string (""") or none. For example:

XINETD_LANG = "en_US"
For more information on how to configure the `xinetd` services, refer to the *Fedora System Administrator's Guide*.

### 3.2. Directories in the `/etc/sysconfig/` Directory

The following directories are normally found in `/etc/sysconfig/`.

**/etc/sysconfig/cbq/**

This directory contains the configuration files needed to do *Class Based Queuing* for bandwidth management on network interfaces. CBQ divides user traffic into a hierarchy of classes based on any combination of IP addresses, protocols, and application types.

**/etc/sysconfig/networking/**

This directory is used by the now deprecated *Network Administration Tool* (`.system-config-network`), and its contents should not be edited manually. For more information about configuring network interfaces using graphical configuration tools, refer to the *Fedora Networking Guide*.

**/etc/sysconfig/network-scripts/**

This directory contains the following network-related configuration files:

- Network configuration files for each configured network interface, such as `ifcfg-eth0` for the `eth0` Ethernet interface.
- Scripts used to bring network interfaces up and down, such as `ifup` and `ifdown`.
- Scripts used to bring ISDN interfaces up and down, such as `ifup-isdn` and `ifdown-isdn`.
- Various shared network function scripts which should not be edited directly.

For more information on the `/etc/sysconfig/network-scripts/` directory, refer to the *Fedora Networking Guide*.

### 3.3. Additional Resources

This chapter is only intended as an introduction to the files in the `/etc/sysconfig/` directory. The following source contains more comprehensive information.

**Installed Documentation**

- `/usr/share/doc/initscripts-version/sysconfig.txt` — A more authoritative listing of the files found in the `/etc/sysconfig/` directory and the configuration options available for them.
Appendix A. Revision History

Revision 0.0-0  Fri Jan 18 2013  Jaromír Hradílek  jhradilek@redhat.com
Initial creation of the book.
Index

E
Ethernet (see network)
Ethtool
command
devname , 13
option
--advertise , 17
--autoneg , 16
--duplex , 16
--features , 15
--identify , 14
--msglvl , 18
--phyad , 17
--port , 16
--show-features , 15
--show-time-stamping , 15
--sopass , 18
--speed , 16
--statistics , 14
--test , 15
--wol , 18
--xcvr , 17

F
feedback
contact information for this manual, vii

I
ifdown, 11
ifup, 11

N
network
additional resources, 19
commands
/sbin/ifdown, 11
/sbin/ifup, 11
/sbin/service network, 11
configuration, 2
configuration files, 1
functions, 13
interface configuration files, 2
interfaces
alias, 8
clone, 8
dialup, 9
Ethernet, 2
ethtool, 13
scripts, 1

S
sysconfig directory
/etc/sysconfig/network-scripts/ directory, 1