NAME

system – system configuration information file

DESCRIPTION

The **system** file is used for customizing the operation of the operating system kernel. The recommended procedure is to preserve the original **system** file before modifying it.

The **system** file contains commands which are read by the kernel during initialization and used to customize the operation of your system. These commands are useful for modifying the system's treatment of its loadable kernel modules.

The syntax of the **system** file consists of a list of keyword/value pairs which are recognized by the system as valid commands. Comment lines must begin with an asterisk ('*') and end with a newline character. All commands are case-insensitive except where noted. A command line can be no more than **80** characters in length.

Commands that modify the system's operation with respect to loadable kernel modules require you to specify the module type by listing the module's namespace. The following namespaces are currently supported:

drv Modules in this namespace are device drivers.

exec Modules in this namespace are execution format modules. The following **exec** modules are currently provided by SunSoft:

SPARC system:	**aoutexec** **elfexec** **intpexec**
x86 system:	**coffexec** **elfexec** **intpexec**

fs These modules are filesystems.

sched These modules implement a process scheduling algorithm.

strmod These modules are STREAMS modules.

sys These modules implement loadable system-call modules.

misc These modules do not fit into any of the above categories, so are considered "miscellaneous" modules.

Below is a description of each of the supported commands:

exclude: <*namespace*>/<*modulename*>
Do not allow the listed loadable kernel module to be loaded. **exclude** commands are cumulative; the list of modules to **exclude** is created by combining every **exclude** entry in the **system** file.

include: <*namespace*>/<*modulename*>
Include the listed loadable kernel module. This is the system's default, so using **include** does not modify the system's operation. **include** commands are cumulative.

forceload: <*namespace*>/<*modulename*>
Force this kernel module to be loaded during kernel initialization. The default action is to automatically load the kernel module when its services are first accessed. **forceload** commands are cumulative.

rootdev: <*device name*>
Set the root device to the listed value instead of using the default root device as supplied by the boot program.

rootfs: <*root filesystem type*>
Set the root filesystem type to the listed value.

moddir: <*first module path*>[[:,]<*second ...*>]...]
Set the search path for loadable kernel modules. This command operates very much like the **PATH** shell variable. Multiple directories to search can be listed together, delimited either by blank spaces or colons.

set [<*module*>:]<*symbol*> {=, |, &} [~][-]<*value*>
Set an integer or character pointer in the kernel or in the selected kernel module to a new value. This command is used to change kernel and module parameters and thus modify the operation of your system. Assignment operations are not cumulative, whereas bitwise AND and OR operations are cumulative.

Operations that are supported for modifying integer variables are: simple assignment, inclusive bitwise OR, bitwise AND, one's complement, and negation. Variables in a specific loadable module can be targeted for modification by specifying the variable name prefixed with the kernel module name and a colon (:) separator. Values can be specified as hexadecimal (0x10), Octal (046), or Decimal (5).

The only operation supported for modifying character pointers is simple assignment. Static string data such as character arrays cannot be modified using the **set** command. Use care and ensure that the variable you are modifying is in fact a character pointer. The **set** command is very powerful, and will likely cause problems if used carelessly. The entire command, including the quoted string, cannot exceed **80** characters. The following escape sequences are supported within the quoted string:

\n **(newline)**
\t **(tab)**
\b **(backspace)**

EXAMPLES

The following is a sample **system** file.

```
* Force the ELF exec kernel module to be loaded during kernel
* initialization. Execution type modules are in the exec namespace.
forceload: exec/elfexec
```

```
* Change the root device to /sbus@1,f8000000/esp@0,800000/sd@3,0:a.
* You can derive root device names from /devices.
* Root device names must be the fully expanded Open Boot Prom
* device name.  This command is platform and configuration specific.
* This example uses the first partition (a) of the SCSI disk at
* SCSI target 3 on the esp host adapter in slot 0 (on board)
* of the SBus of the machine.
* Adapter unit-address 3,0 at sbus unit-address 0,800000.
rootdev: /sbus@1,f8000000/esp@0,800000/sd@3,0:a

* Set the filesystem type of the root to ufs. Note that
* the equal sign can be used instead of the colon.
rootfs:ufs

* Set the search path for kernel modules to look first in
* /usr/phil/mod_test for modules, then in /kernel/modules (the
* default) if not found. Useful for testing new modules.
* Note that you can delimit your module pathnames using
* colons instead of spaces: moddir:/newmodules:/kernel/modules
moddir:/usr/phil/mod_test /kernel/modules.

* Set the integer variable "maxusers" in the kernel to 16. This is a
* useful tuning parameter.
set maxusers = 16

* Turn on debugging messages in the modules mydriver. This is useful
* during driver development.
set mydriver:debug = 1

* Bitwise AND the kernel variable "moddebug" with the
* one's complement of the hex value 0x880, and set
* "moddebug" to this new value.
set moddebug & ~0x880

* Demonstrate the cumulative effect of the SET
* bitwise AND/OR operations by further modifying "moddebug"
* by ORing it with 0x40.
set moddebug | 0x40
```

WARNINGS

system file lines must be fewer than **80** characters in length.

Use care when modifying the **system** file; it modifies the operation of the kernel. If you preserved the original **system** file, you can use the **boot -a** option and supply the path to the original file, allowing the system to boot correctly.

NOTES **/etc/system** is only read once: at boot time.