File System Space Monitoring [6]

The file system space monitoring capability improves the usability and reliability of the system. Space monitoring observes the amount of free space on the mounted file systems and takes remedial action if warning or critical thresholds are reached.

This section describes the space monitoring feature, its installation, and use of the commands `fsdaemon`(8), the monitor daemon, and `fsmon`(8), the daemon interface.

6.1 Operation of the Space Monitor

There are three major components to the space monitor: the monitor daemon (`fsdaemon`), the daemon interface (`fsmon`), and the operator's interface (`msgdaemon`).

The daemon may be started by the `/etc/rc` script (see `brc`(8)) in the area of the script reserved for starting daemons, or it can be started manually by using the `fsdaemon` command. Once the daemon starts and configures its monitor tables, it begins monitoring file system free space on a timed cycle. It remains running until stopped by an operator command or a system shutdown.

If the operator wants to continuously examine the current status of the monitored file systems or to alter some aspect of the monitoring process, the operator enters a command through the `msgdaemon` operator interface command. The `fsmon` command must be defined in the list of valid operator commands in the configuration file `$HOME/.operrc`. The operator interface executes `fsmon` and the output is returned to the operator's display screen.

The daemon interface may also be called by a privileged user. In this case, it returns its output to the `stdout` file.

The purpose of the daemon is to take action when the free space on one or more monitored file systems reaches either a warning or a critical free space threshold. When this occurs, the daemon may perform the following actions based on its configuration instructions:

- Send a message to the operator through `msgdaemon`(8)
- Initiate a shell script or command

6.2 Interprocess Communication

Communication between fsdaemon(8) and fsmon(8) occurs by way of named pipes. When it initializes, fsdaemon establishes a request pipe with a known name; all requests to it are made through that pipe.

When fsmon executes, it opens the fsdaemon request pipe and includes the name of the reply pipe along with the request it writes into the request pipe. fsdaemon replies to each request using the reply pipe name specified with the request. This allows multiple instances of fsmon to run without conflict.

Communication between fsdaemon and msgdaemon(8) is used only for warnings and critical operator messages. It works in much the same way as the fsdaemon to fsmon mechanism previously described. In this case, however, fsdaemon is the originator rather than the recipient of requests. An include file (msg.h) provided with the operator message daemon (msgdaemon) ensures that the interface is structured properly. The form of operator messages indicates that an operator response is expected even though the actual response is discarded. This is merely to ensure that operator messages are displayed and stay displayed until the operator responds or they are canceled by fsdaemon.

When an fsmon command is given to the operator interface, the parameters are passed on the command line. Output from fsmon is sent back to the operator interface through the stdout file. This interface is not interactive but allows fsmon to be run either as a normal or an operator command.

Communication between fsdaemon and administrator-specified commands or scripts occurs when fsdaemon detects that a warning or critical threshold has been reached for a particular file system. fsdaemon appends the name of the file system, preceded by a space, to the specified command or script, executes a fork(2) system call, and runs the command or script with ksh(1). The command or script is initiated with the daemon's environment (the stdin file is closed and the stdout file and stderr file are open on /dev/null) and, once started, fsdaemon pays no attention to the command and does not wait for it to complete. No other command or script is executed for the affected file system until a reset request has been received, or until a critical threshold has been reached.

6.3 The Monitor Daemon, fsdaemon

This section describes the monitor daemon capabilities, both functionally and from a user's point of view.

Except when initializing and terminating, `fsdaemon`(8) always evaluates the current state of file systems and watches for requests from the interface program. The monitoring process is controlled by these parameters:

- Names of the file systems to monitor
- Warning and critical threshold values for each file system (default values are used if specific values are not provided)
- A Monitoring enable switch for each file system
- Operator message enable switches for warning and critical thresholds
- Critical and warning command execution flags
- A command (if any) to run if a warning or critical threshold occurs
- Length of time between each monitor cycle
- Priority (nice value) of the daemon
- `plock`(2) state of the daemon
- Ability to add monitored file systems at any time

6.3.1 File System Monitoring

When the daemon starts, it does not know which file systems you want it to monitor or the thresholds you want to assign to each file system. The file systems are identified and their thresholds set with the `fsmon` command. The following paragraphs describe the monitoring process after the daemon has been started and configured. See Section 6.3.3, page 273, for more information about configuring the daemon by using the `fsmon` command.

Once each cycle, `fsdaemon` collects usage information for each file system enabled in the monitor list and takes the following steps:

1. Each entry is examined to determine whether a threshold has been reached (a threshold is reached when usage is greater than or equal to the threshold). If not, the daemon waits for the next cycle time and repeats the process. If a threshold is reached, the next stage of processing is started.
2. If a new critical threshold has occurred and a critical command is not already running, the daemon starts the critical threshold command (if one has been defined using the `fsdaemon` command), issues the critical operator log messages for this file system (unless operator messages have been disabled with `fsmon -n`), marks that this entry has been processed,

and records the time the condition was detected. Then it moves to the next enabled entry.

3. If a new warning threshold has occurred, and a critical or warning command is not already running, the daemon starts the warning threshold command (if one has been defined), issues the warning operator log messages for this file system, marks that this entry has been processed, and records the time the condition was detected. Then it moves to the next enabled entry.

Once a file system has reached the warning or critical threshold, the messages have been logged, and the command started, the daemon does not attempt to start another command of the same kind until a reset request is processed.

6.3.2 Critical and Warning Command Processing

In addition to starting the daemon, the `fsdaemon` command can be used to specify commands to be run when critical or warning thresholds are reached. Full or relative path names for the commands must be specified. Options to the commands and command redirection can also be specified; if this is desired, the command must be enclosed in quotation marks. You start the command by using the `exec`(2) system call to execute `ksh`(1) with the command as the `-c` option, and so you should initialize the environment variables and working directory as needed for execution.

When a critical or warning command is started, all files except the `stdin` file, the `stdout` file, and the `stderr` file have been closed. These special files are initiated as follows:

- The `stdin` file is closed.
- The `stdout` file and the `stderr` file are open. In normal mode, they are assigned to `/usr/spool/fsmonitor/Fd.fd12`. In test mode, they are assigned to `./Fd.fd12`.

This means that, if the command needs to access any of these files, they must be redirected properly.

In test mode (`-t` option on the `fsdaemon` command), the file `Fd.fd12` is created in the directory the daemon inherits when it is started. In normal mode, the file is in the directory `/usr/spool/fsmonitor` unless you specify a different directory using the `-p` option directory. In normal mode, the daemon's home directory also defaults to `/usr/spool/fsmonitor`, unless the `-p` option specifies another directory. The command always starts with the home directory set to `/`.

See the `fsdaemon`(8) man page for a detailed description of the `fsdaemon` command and the options you can use to specify critical or warning commands.

6.3.3 Request Processing

The `fsdaemon` program includes a number of requests designed to configure the daemon, control the monitoring process, and allow the operator to view the current state of the file systems. Requests are created from the options specified with the `fsmon` command. Access to the request processor is through the request pipe (using `fsmon`).

The following paragraphs provide a brief overview of `fsmon` command options. For a detailed description of each option, see the `fsmon` man page.

The `-a` option lets you add a file system to the table of monitored file systems. The command lines for this function are as follows:

```
fsmon -a [-c nnn] [-i [c][w]|-n [c][w]] [-w nnn] [-e] filesystems
fsmon -a [-c nnn] [-w nnn] [-d] filesystems
```

The `-m` option allows you to change entries in the table of monitored file systems. The command lines are as follows:

```
fsmon -m [-c nnn] [-i cw | -n cw] [-w nnn] [-e] [-f cw] filesystems
fsmon -m [-c nnn] [-w nnn] [-d] [-f cw] filesystems
```

You can use the optional options with the `-a` or `-m` option to do the following:

- Set critical and warning threshold percentages.
- Enable or disable operator messages.
- Enable or disable monitoring for the specified file system.
- Manually set the critical or warning condition on the specified file system. (This function is intended for testing use.)

The *filesystems* operand can be a name, an ordinal, or the special name `all`. An *ordinal* is defined as the number that appears in the `n` column of the file system status display for a particular entry in the table. (See Example 1, page 275.) Ordinals allow you to indicate a particular file system without typing the full name. (Ordinals are not fixed but depend on the specific configuration and can change whenever an entry is added to the table of monitored file systems. Ordinals have meaning only to `fsdaemon`, as shown on the status display, and have no connection with any kind of file system ordinals UNICOS may use internally.) If you use the file system name `all` with the `-m` option, it alters every entry in the table.

New entries are inserted in alphabetical order as determined by the result of a `strcmp` () comparison, which causes the ordinals of all entries beyond the new one to increase by one.

The `-q` option causes `fsdaemon` to close and rename the log file and terminate. This option is available only if `fsdaemon` had been started with the `-q` option. The command line is as follows:

```
fsmon -q
```

The `-r` option signals the daemon that the critical or warning command has completed on the specified file systems by causing the "command is running" state to be removed from the selected entries. In effect, this reenables the monitoring of the specified file system. The command lines are as follows (the first is for the completion of a critical command, the second for the completion of warning command):

```
fsmon -r c filesystems
fsmon -r w filesystems
```

A command of the same type (critical or warning) is not restarted until the occupancy of the file system falls below and then crosses above the threshold again. This mechanism prevents more than one instance of a warning or critical command from being active on the same file system, but allows a critical command to be started if the warning command has not yet completed its operation. To prevent the daemon from immediately restarting the command (in the event that the actual state of the file system has not been changed enough to remove the threshold condition), the occupancy of the file system must fall below the threshold and then rise through it again before the command will start.

If appropriate for your site, you may also specify the `-r` option with `cw` to indicate that both critical and warning commands have been completed. The special file system name `all` resets every file system in the table. Use this name carefully because it could allow more than one command to be active on the file system at the same time.

The `-R` option resets file system monitoring by clearing the internal flags that indicate whether warning or critical commands are executing, and most state information from the table of monitored file systems. The command line is as follows:

```
fsmon -R filesystems
```

This option is intended to be used when corrective action has been completed following a warning or critical event and you want to enable critical and warning monitoring. The special name `all` resets every file system in the table. Use this option carefully because it could cause multiple commands to be active in the same file system. `fsdaemon` evaluates the state of the file system after this request is processed. A warning or critical event could occur immediately.

The `-s` option lets you display file system status by sending a status block in addition to the normal reply to a request. `fsmon` formats the information for the operator display as shown in Example 1, page 275. The command line is as follows:

`fsmon [-s cdew] [`*filesystems*`]`

The `-s` option with possible arguments `c`, `d`, `e`, and `w` limits the class of table entries to display from the list. `c` is critical, `d` is disabled, `e` is enabled, and `w` is warning. Any combination may be set, but you must specify at least one type.

The `-t` option suppresses display headers and informative messages. The following example displays all critical file systems, but suppresses the header shown in Example 1, page 275:

```
fsmon -t c
```

Currently this is the only directive that has a header, but `-t` is allowed with all directives.

The `-p` option specifies an alternate path to the daemon request pipe and log file. Communication between the daemon and `fsmon` uses named pipes and consists of mutually defined structures containing mixed ASCII and binary data.

6.3.4 Status Display

Example 1, page 275 is an example of a file system status display. This section describes the contents of this display.

Example 1:

```
n    File System    Status       Use    Warn    Crit   Time encountered
1    /              E-------     71.6%  85.0%   95.0%
2    /a             E-X----      95.2%  85.0%   95.0%   10:22:53 06/23
3    /arch          E-------     5.7%   85.0%   95.0%
4    /b             E-X----      97.6%  85.0%   95.0%   10:22:53 06/23
5    /bnch          E---W-X-     92.6%  85.0%   95.0%   10:22:53 06/23
6    /c             E---W-X-     92.3%  85.0%   95.0%   10:22:53 06/23
```

```
7    /core          E-------     84.6%  85.0%   95.0%
8    /d             E-------     83.4%  85.0%   95.0%
9    /drop          E-------     2.3%   85.0%   95.0%
10   /e             E-X----      96.3%  85.0%   95.0%   10:22:53 06/23
11   /ea            E-------     50.3%  85.0%   95.0%
12   /ea/usr        E-------     31.7%  85.0%   95.0%
13   /ea/usr/src    E-------     65.1%  85.0%   95.0%
14   /g             E---W-X-     89.3%  85.0%   95.0%   10:22:53 06/23
15   /h             E-X----      97.1%  85.0%   95.0%   10:22:53 06/23
16   /j             E---W-X-     92.9%  85.0%   95.0%   10:22:53 06/23
17   /l             E---W-X-     90.2%  85.0%   95.0%   10:22:53 06/23
18   /n             E-------     84.1%  85.0%   95.0%
19   /p             E-X----      99.5%  85.0%   95.0%   10:22:53 06/23
```

The `n` column holds the file system ordinal that can be used as a synonym for the file system name. The `File System` column holds the name of the file system. In order to fit the entire display line on an 80-column window, very long names may cause the remainder of the status line to appear below the name rather than to the right of it. The `Status` column has eight status flags. A position with `-` means that the status does not apply to the file system. Position 1 always has either `D`, `E`, or `*` visible.

The following list describes the status flags:

Status	Meaning
`E-------`	Monitoring is enabled
`D-------`	Monitoring is disabled
`*-------`	Monitoring error
`-C------`	Critical threshold detected
`--c-----`	Manual critical forced
`---X----`	Critical command executing
`----W---`	Warning threshold detected
`-----w--`	Manual warning forced
`------X-`	Warning command executing
`-------?`	Internal error

The `Use` column is the current usage level of the file system expressed as percentage. A file system that is disabled or in the error state does not have a value displayed. The `Warn` column is the warning percentage threshold currently set. The `Crit` column is the critical percentage threshold currently set. The `Time encountered` column is the time, in *hh*:*mm*:*ss mm*/*dd* format, when the warning or critical threshold level was reached. Critical conditions take precedence over warnings. If there is a monitoring error, a message is displayed in this column.

6.3.5 Using the `fsdaemon` and `fsmon` Commands

The following example shows how you can use the `fsdaemon` and `fsmon` commands to start and configure the file system monitor. In this example, the following commands are executed as part of the startup file when the system is brought up:

```
/etc/fsdaemon -w /admin/scripts/warning/ -c /admin/scripts/critical
/etc/fsmon -a -w 93 -c 97 / /usr /usr/tmp /tmp
/etc/fsmon -a -w 88 -c 90 /core
```

The `fsdaemon` command line starts the daemon and configures it to execute a warning script named `/admin/scripts/warning` when a warning threshold is reached and to execute a critical script named `/admin/scripts/critical` when a critical threshold is reached. What the scripts do (or if they exist) depends on the needs of your site. Because the daemon does not attempt to start another script of the same kind until a reset request is processed, the warning and critical scripts must end with a `fsmon` command to perform the reset request. See Section 6.3.3, page 273, for a description of `fsmon` reset options.

The `fsmon` command configures the daemon. The first `fsmon` command line (using the `-a` option) adds the file systems named `/`, `/usr`, `/usr/tmp`, and `/tmp` to the table of monitored file systems, and sets the warning threshold (using the `-w` option) to 93% and the critical threshold (using the `-c` option) to 97% for each of those file systems.

The second `fsmon` command line adds the file system named `/core` with a warning threshold of 88% and a critical threshold of 90%. You can use as many `fsmon` command lines as you need to add the file systems you want to monitor. All the file systems with the same threshold values can be added with one command.

With the daemon set up using the previous example, a display of the current status might look like the following:

```
n   File System   Status      Use   Warn   Crit  Time encountered

1   /             E-------    88.2% 93.0%  97.0%
2   /core         E-------    81.4% 88.0%  90.0%
```

```
3   /tmp          E-------     5.7% 93.0%  97.0%
4   /usr          E-------    61.3% 93.0%  97.0%
5   /usr/tmp      E-------     1.6% 93.0%  97.0%
```

If you want to change the critical threshold on `/tmp` to 95% and the warning threshold to 80%, you could use the following command:

```
/etc/fsmon -m -c 95 -w 80 /tmp
```

Because the `n` value shown on the display for `/tmp` is `3`, you could also use the following command:

```
/etc/fsmon -m -c 95 -w 80 3
```

After either of these commands has been executed, a display of the current status would look like the following:

```
n   File System   Status      Use   Warn   Crit  Time encountered

1   /             E-------    88.2% 93.0%  97.0%
2   /core         E-------    81.4% 88.0%  90.0%
3   /tmp          E-------     5.7% 80.0%  95.0%
4   /usr          E-------    61.3% 93.0%  97.0%
5   /usr/tmp      E-------     1.6% 93.0%  97.0%
```

6.3.6 The Log File

The log file records events that occur during the monitor daemon operation, and are intended for operations personnel. The recorded events include automatic actions taken by the monitor, operator requests that cause a change in the operation of the daemon, and error events detected by the daemon. An example of log file messages follows:

```
06/22  16:05:34  (06)  Fsmonitor initiated.
06/22  16:05:34  (06)  Critical command: 'critical_command'

06/22  16:05:34  (06)  Warning command: 'warning_command'
06/22  16:05:34  (06)  Entered Main loop
06/22  16:06:12  (06)  Added: /a -c  95.0% -w  85.0% -d
06/22  16:06:12  (06)  Added: /b -c  95.0% -w  85.0% -d
06/22  16:06:12  (06)  Added: /c -c  95.0% -w  85.0% -d
06/22  16:06:12  (06)  Added: /d -c  95.0% -w  85.0% -d
06/22  16:06:12  (06)  Added: /e -c  95.0% -w  85.0% -d
06/22  16:06:12  (06)  Added: /f -c  95.0% -w  85.0% -d
06/22  16:07:05  (06)  Changed: /c -c  98.0% -w  91.0% -e
```

```
06/22  16:07:05  (06)  Pid 28083 running: 'warning_command /c'
06/22  16:07:05  (06)  WARNING THRESHOLD ON /c
06/22  16:07:06  (06)  Changed: /f -c  98.0% -w  91.0% -e
06/22  16:07:07  (06)  Changed: /g -c  98.0% -w  91.0% -e
06/22  16:08:13  (06)  Daemon terminated; exit(0) "Stopped by quit"
```

The sample log shows the configuration of six file systems (`/a` through `/f`) with default levels of critical (`-c`) and warning (`-w`) thresholds. All file systems were initially disabled, but later (at `16:07:05`) a request changed the threshold values and enabled threshold detection for file system `/c`. As soon as monitoring was enabled on `/c`, a warning was detected, causing process ID `28083` to run the warning command. The last line of the log shows a normal termination caused by the quit request. When `fsdaemon` terminates, the name of the log file is changed to include the process ID. If the daemon is restarted, a new log file is opened.

6.4 Informative and Error Messages

Messages seen by a user of `fsmon` may come directly from `fsmon` or by way of the reply pipe from `fsdaemon`. This section lists the most important messages from both sources. All messages indicate errors, unless specifically stated otherwise, and cause an exit value of 1; informative messages cause an exit value of 0.

Usage messages that are not explained here adequately describe the error without additional explanation.

Message   Description

`Daemon did not respond in 30 seconds`

`fsmon` waits for a reply from the daemon for a limited period of time before indicating that it did not respond. This could be caused by system load or scheduling problems, but also may indicate that `fsdaemon` is present but not responding.

`File system monitor daemon may not be running`

When `fsmon` has accepted its options, they must be translated and passed on to `fsdaemon` for processing. This message indicates that communication was impossible, possibly because the daemon was not running. This message also occurs when the `-p` option is given the wrong directory tree, or when the user does not have permission to access the daemon's request pipe.

`fsnames required for -a, -m or -r`

The file system name is mandatory with these options.

`File system` *xxx* `exists in monitor list`

The named entry cannot be added because it already exists.

`File system` *xxx* `not in monitor list`

The named file system is not currently in the monitor list.

`Monitor list empty`

This is an error only when the `-m` option has been used, but it can occur as an informative message as well.

`Monitor list full`

Another file system may not be added to the monitor list. Because the monitor list size is set to twice the maximum number of mounted file systems when the daemon is started, this should not be a problem unless some sort of configuration error occurs.

`No entries selected`

This informative message states that no monitor table entries matched the type criteria. Select `-s cdew` to see everything in the table.

`Ordinal` *nn* `not in monitor list`

The file system ordinal is 0 or greater than the current maximum value.

6.5 Installation and Operation Information

This section describes some useful information about the monitor's operational characteristics.

6.5.1 Installation

The commands `fsmon`(8) and `fsdaemon`(8) are installed by default in the `/etc` directory because they are intended as operator or administrator commands. The log file and command pipe are found in the `/usr/spool/fsmonitor` directory. Except for the file `Fd.fd12`, other files used during operation are named using `tempnam` (see `tmpnam`(3)) and so reside in the preferred temporary directory. None of the temporary files are intended to exist for long periods of time. File `Fd.fd12` is created in the directory described in Section 6.3.2, page 272, each time a command runs. This file holds all non-redirected standard output and standard error text from a command until the next command is started by the daemon. This file is intended for debugging or monitoring and is not intended as a log or other recording mechanism.

6.5.2 Operation

You must start the monitor daemon with critical and warning commands specified on its command line if you want any action to occur when a threshold is reached. You cannot change these command names without terminating and restarting `fsdaemon`. You can circumvent this restriction by writing critical and warning scripts that start the desired warning or critical operations. You can then alter or switch these initiation scripts during operation to provide whatever degree of flexibility is necessary.

Getting the monitor configured correctly is the most important issue following installation. Once `fsdaemon` is running and all the file systems are mounted, the easiest way to establish monitoring of all mounted file systems is to execute the following command line:

```
fsmon -a -c nnn -w nnn all
```

You must decide what critical threshold values (`-c` *nnn*) and warning values (`-w` *nnn*) are appropriate. (The defaults are 95 for critical and 85 for warning.) When starting `fsdaemon`, the critical and warning commands are the most important options. You should select which command to execute if either of the threshold conditions occurs.

After the monitor is configured and running, you may ask for displays and change its configuration by using the `fsmon`(8) options described previously. Because the monitor keeps track of old log files (it renames them before termination so the old file does not interfere with the next initiation), you should develop a procedure to clean up the old files from time to time. Do not allow the log to become very large under normal circumstances.

6.5.3 Permissions

Who uses the fsdaemon commands is determined by who can write to the command pipe. The daemon creates the command input pipe with owner read/write and group write permissions. This limits access to the owner and group that started the daemon. This may be changed by altering the value of IN PIPE MODE in the dmparams.h file in the fsdaemon_source directory. The directory in which the special files reside is created by the installation process to be owned by root and to belong to the operator group. This may also be changed if necessary. If permissions are changed, think about the implications, because other users could disable threshold monitoring. This could affect data migration, because in many environments threshold detection is used to activate the data migration feature.

6.5.4 Testing

To do any testing without disrupting a running version of the monitor, the path option (-p) is available. This makes it possible to run the monitor with its files and pipes in a directory separate from the default, so that two or more versions can run at the same time. The -t option with fsdaemon is also convenient for testing, because it prevents the process from detaching itself and running without a controlling terminal. The following command line is an example of testing fsdaemon:

```
fsdaemon -t -q -p /tmp/fsd.test -w warncmd -c critical &
```

6.5.5 Related Files

The running log file is usually named as follows:

```
/usr/spool/fsmonitor/Fd.log
```

When fsdaemon is terminated, the running log is renamed with the process ID of the daemon appended (Fd.log becomes Fd.log.12345, if the process ID is 12345). If for some reason the daemon does not rename the log file (for instance, if the system crashed), a new invocation of the daemon appends its output to the existing file.

Temporary files are created for the reply pipe (managed by the requesting side, not fsdaemon) and the reply pipe from the message daemon. The file names are prefixed with Fs. None of the temporary files should be left around when the daemon is correctly terminated.