Documentation for /proc/sys/fs/

kernel version 2.2.10

This file contains documentation for the sysctl files in /proc/sys/fs/ and is valid for Linux kernel version 2.2.

The files in this directory can be used to tune and monitor miscellaneous and general things in the operation of the Linux kernel. Since some of the files _can_ be used to screw up your system, it is advisable to read both documentation and source before actually making adjustments.

1. /proc/sys/fs

Currently, these files are in /proc/sys/fs:

- aio-max-nr
- aio-nr
- dentry-state
- dquot-max
- dquot-nr
- file-max
- file-nr
- inode-max
- inode-nr
- inode-state
- nr_open
- overflowuid
- overflowgid
- pipe-user-pages-hard
- pipe-user-pages-soft
- protected_fifos
- protected_hardlinks
- protected_regular
- protected_symlinks
- suid_dumpable
- super-max
- super-nr

aio-nr & aio-max-nr

aio-nr is the running total of the number of events specified on the io_setup system call for all currently active aio contexts. If aio-nr reaches aio-max-nr then io_setup will fail with EAGAIN. Note that raising aio-max-nr does not result in the pre-allocation or re-sizing of any kernel data structures.

dentry-state

From linux/include/linux/dcache.h:

```
struct dentry_stat_t dentry_stat {
      int nr_dentry;
      int nr_unused;
      int age_limit;          /* age in seconds */
      int want_pages;         /* pages requested by system */
      int nr_negative;        /* # of unused negative dentries */
      int dummy;              /* Reserved for future use */
};
```

Dentries are dynamically allocated and deallocated.

nr_dentry shows the total number of dentries allocated (active + unused). nr_unused shows the number of dentries that are not actively used, but are saved in the LRU list for future reuse.

Age_limit is the age in seconds after which dcache entries can be reclaimed when memory is short and want_pages is nonzero when shrink_dcache_pages() has been called and the dcache isn't pruned yet.

nr_negative shows the number of unused dentries that are also negative dentries which do not map to any files. Instead, they help speeding up rejection of non-existing files provided by the users.

dquot-max & dquot-nr

The file dquot-max shows the maximum number of cached disk quota entries.

The file dquot-nr shows the number of allocated disk quota entries and the number of free disk quota entries.

If the number of free cached disk quotas is very low and you have some awesome number of simultaneous system users, you might want to raise the limit.

file-max & file-nr

The value in file-max denotes the maximum number of file- handles that the Linux kernel will allocate. When you get lots of error messages about running out of file handles, you might want to increase this limit.

Historically,the kernel was able to allocate file handles dynamically, but not to free them again. The three values in file-nr denote the number of allocated file handles, the number of allocated but unused file handles, and the maximum number of file handles. Linux 2.6 always reports 0 as the number of free file handles -- this is not an error, it just means that the number of allocated file handles exactly matches the number of used file handles.

Attempts to allocate more file descriptors than file-max are reported with printk, look for "VFS: file-max limit <number> reached".

nr_open

This denotes the maximum number of file-handles a process can allocate. Default value is 1024*1024 (1048576) which should be enough for most machines. Actual limit depends on RLIMIT_NOFILE resource limit.

inode-max, inode-nr & inode-state

As with file handles, the kernel allocates the inode structures dynamically, but can't free them yet.

The value in inode-max denotes the maximum number of inode handlers. This value should be 3-4 times larger than the value in file-max, since stdin, stdout and network sockets also need an inode struct to handle them. When you regularly run out of inodes, you need to increase this value.

The file inode-nr contains the first two items from inode-state, so we'll skip to that file...

Inode-state contains three actual numbers and four dummies. The actual numbers are, in order of appearance, nr_inodes, nr_free_inodes and preshrink.

Nr_inodes stands for the number of inodes the system has allocated, this can be slightly more than inode-max because Linux allocates them one pageful at a time.

Nr_free_inodes represents the number of free inodes (?) and preshrink is nonzero when the nr_inodes > inode-max and the system needs to prune the inode list instead of allocating more.

overflowgid & overflowuid

Some filesystems only support 16-bit UIDs and GIDs, although in Linux UIDs and GIDs are 32 bits. When one of these filesystems is mounted with writes enabled, any UID or GID that would exceed 65535 is translated to a fixed value before being written to disk.

These sysctls allow you to change the value of the fixed UID and GID. The default is 65534.

pipe-user-pages-hard

Maximum total number of pages a non-privileged user may allocate for pipes. Once this limit is reached, no new pipes may be allocated until usage goes below the limit again. When set to 0, no limit is applied, which is the default setting.

pipe-user-pages-soft

Maximum total number of pages a non-privileged user may allocate for pipes before the pipe size gets limited to a single page. Once this limit is reached, new pipes will be limited to a single page in size for this user in order to limit total memory usage, and trying to increase them using fcntl() will be denied until usage goes below the limit again. The default value allows to allocate up to 1024 pipes at their default size. When set to 0, no limit is applied.

protected_fifos

The intent of this protection is to avoid unintentional writes to an attacker-controlled FIFO, where a program expected to create a regular file.

When set to "0", writing to FIFOs is unrestricted.

When set to "1" don't allow O_CREAT open on FIFOs that we don't own in world writable sticky directories, unless they are owned by the owner of the directory.

When set to "2" it also applies to group writable sticky directories.

This protection is based on the restrictions in Openwall.

protected_hardlinks

A long-standing class of security issues is the hardlink-based time-of-check-time-of-use race, most commonly seen in world-writable directories like /tmp. The common method of exploitation of this flaw is to cross privilege boundaries when following a given hardlink (i.e. a root process follows a hardlink created by another user). Additionally, on systems without separated partitions, this stops unauthorized users from "pinning" vulnerable setuid/setgid files against being upgraded by the administrator, or linking to special files.

When set to "0", hardlink creation behavior is unrestricted.

When set to "1" hardlinks cannot be created by users if they do not already own the source file, or do not have read/write access to it.

This protection is based on the restrictions in Openwall and grsecurity.

protected_regular

This protection is similar to protected_fifos, but it avoids writes to an attacker-controlled regular file, where a program expected to create one.

When set to "0", writing to regular files is unrestricted.

When set to "1" don't allow O_CREAT open on regular files that we don't own in world writable sticky directories, unless they are owned by the owner of the directory.

When set to "2" it also applies to group writable sticky directories.

protected_symlinks

A long-standing class of security issues is the symlink-based time-of-check-time-of-use race, most commonly seen in world-writable directories like /tmp. The common method of exploitation of this flaw is to cross privilege boundaries when following a given symlink (i.e. a root process follows a symlink belonging to another user). For a likely incomplete list of hundreds of examples across the years, please see: https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=/tmp

When set to "0", symlink following behavior is unrestricted.

When set to "1" symlinks are permitted to be followed only when outside a sticky world-writable directory, or when the uid of the symlink and follower match, or when the directory owner matches the symlink's owner.

This protection is based on the restrictions in Openwall and grsecurity.

suid_dumpable:

This value can be used to query and set the core dump mode for setuid or otherwise protected/tainted binaries. The modes are

0	(default)	traditional behaviour. Any process which has changed privilege levels or is execute only will not be dumped.
1	(debug)	all processes dump core when possible. The core dump is owned by the current user and no security is applied. This is intended for system debugging situations only. Ptrace is unchecked. This is insecure as it allows regular users to examine the memory contents of privileged processes.
2	(suidsafe)	any binary which normally would not be dumped is dumped anyway, but only if the "core_pattern" kernel sysctl is set to either a pipe handler or a fully qualified path. (For more details on this limitation, see CVE-2006-2451.) This mode is appropriate when administrators are attempting to debug problems in a normal environment, and either have a core dump pipe handler that knows to treat privileged core dumps with care, or specific directory defined for catching core dumps. If a core dump happens without a pipe handler or fully qualified path, a message will be emitted to syslog warning about the lack of a correct setting.

super-max & super-nr

These numbers control the maximum number of superblocks, and thus the maximum number of mounted filesystems the kernel can have. You only need to increase super-max if you need to mount more filesystems than the current value in super-max allows you to.

aio-nr & aio-max-nr

aio-nr shows the current system-wide number of asynchronous io requests. aio-max-nr allows you to change the maximum value aio-nr can grow to.

mount-max

This denotes the maximum number of mounts that may exist in a mount namespace.

2. /proc/sys/fs/binfmt_misc

Documentation for the files in /proc/sys/fs/binfmt_misc is in Documentation/admin-guide/binfmt-misc.rst.

3. /proc/sys/fs/mqueue - POSIX message queues filesystem

The "mqueue" filesystem provides the necessary kernel features to enable the creation of a user space library that implements the POSIX message queues API (as noted by the MSG tag in the POSIX 1003.1-2001 version of the System Interfaces specification.)

The "mqueue" filesystem contains values for determining/setting the amount of resources used by the file system.

/proc/sys/fs/mqueue/queues_max is a read/write file for setting/getting the maximum number of message queues allowed on the system.

/proc/sys/fs/mqueue/msg_max is a read/write file for setting/getting the maximum number of messages in a queue value. In fact it is the limiting value for another (user) limit which is set in mq_open invocation. This attribute of a queue must be less or equal then msg_max.

/proc/sys/fs/mqueue/msgsize_max is a read/write file for setting/getting the maximum message size value (it is every message queue's attribute set during its creation).

/proc/sys/fs/mqueue/msg_default is a read/write file for setting/getting the default number of messages in a queue value if attr parameter of mq_open(2) is NULL. If it exceed msg_max, the default value is initialized msg_max.

/proc/sys/fs/mqueue/msgsize_default is a read/write file for setting/getting the default message size value if attr parameter of mq_open(2) is NULL. If it exceed msgsize_max, the default value is initialized msgsize_max.

4. /proc/sys/fs/epoll - Configuration options for the epoll interface

This directory contains configuration options for the epoll(7) interface.

max_user_watches

Every epoll file descriptor can store a number of files to be monitored for event readiness. Each one of these monitored files constitutes a "watch". This configuration option sets the maximum number of "watches" that are allowed for each user. Each "watch" costs roughly 90 bytes on a 32bit kernel, and roughly 160 bytes on a 64bit one. The current default value for max_user_watches is the 1/25 (4%) of the available low memory, divided for the "watch" cost in bytes.