Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'umask.2' command

$ man umask.2

UMASK(2) Linux Programmer's Manual UMASK(2)

NAME

umask - set file mode creation mask

SYNOPSIS

#include <sys/types.h>

#include <sys/stat.h>

mode_t umask(mode_t mask);

DESCRIPTION

umask() sets the calling process's file mode creation mask (umask) to mask & 0777 (i.e., only the file permission bits of mask are used), and returns the previous value of the mask.

The umask is used by open(2), mkdir(2), and other system calls that create files to modify the permissions placed on newly created files or directories. Specifically, permissions in the umask are turned off from the mode argument to open(2) and mkdir(2).

Alternatively, if the parent directory has a default ACL (see acl(5)), the umask is ignored, the default ACL is inherited, the permission bits are set based on the inherited ACL, and permission bits absent in the mode argument are turned off. For example, the following default ACL is equivalent to a umask of 022:

u::rwx,g::r-x,o::r-x

Combining the effect of this default ACL with a mode argument of 0666 (rw-rw-rw-), the resulting file permissions would be 0644 (rw-r--r--).

The constants that should be used to specify mask are described in in?
ode(7).

The typical default value for the process umask is S_IWGRP | S_IWOTH (octal 022). In the usual case where the mode argument to open(2) is specified as:

S_IRUSR | S_IWUSR | S_IRGRP | S_IWGRP | S_IROTH | S_IWOTH

(octal 0666) when creating a new file, the permissions on the resulting file will be:

S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH

(because 0666 & ~022 = 0644; i.e., rw-r--r--).

RETURN VALUE

This system call always succeeds and the previous value of the mask is returned.

CONFORMING TO

POSIX.1-2001, POSIX.1-2008, SVr4, 4.3BSD.

NOTES

A child process created via fork(2) inherits its parent's umask. The umask is left unchanged by execve(2).

It is impossible to use umask() to fetch a process's umask without at the same time changing it. A second call to umask() would then be needed to restore the umask. The nonatomicity of these two steps pro? vides the potential for races in multithreaded programs.

Since Linux 4.7, the umask of any process can be viewed via the Umask field of /proc/[pid]/status. Inspecting this field in /proc/self/sta? tus allows a process to retrieve its umask without at the same time changing it.

The umask setting also affects the permissions assigned to POSIX IPC objects (mq_open(3), sem_open(3), shm_open(3)), FIFOs (mkfifo(3)), and UNIX domain sockets (unix(7)) created by the process. The umask does not affect the permissions assigned to System V IPC objects created by the process (using msgget(2), semget(2), shmget(2)).

SEE ALSO

chmod(2), mkdir(2), open(2), stat(2), acl(5)

COLOPHON

This page is part of release 5.10 of the Linux man-pages project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at https://www.kernel.org/doc/man-pages/.