Linux/UNIX system programming training

umask(3p) — Linux manual page

[   ] Search online pages

UMASK(3P)   POSIX Programmer's Manual   UMASK(3P)

PROLOG   top

     This manual page is part of the POSIX Programmer's Manual. The
     Linux implementation of this interface may differ (consult the
     corresponding Linux manual page for details of Linux behavior),
     or the interface may not be implemented on Linux.

NAME   top

     umask — set and get the file mode creation mask

SYNOPSIS   top

     #include <sys/stat.h>

     mode_t umask(mode_t *cmask*);

DESCRIPTION   top

The *umask*() function shall set the file mode creation mask of the
process to *cmask* and return the previous value of the mask. Only
the file permission bits of *cmask* (see *<sys/stat.h>*) are used;
the meaning of the other bits is implementation-defined.

The file mode creation mask of the process is used to turn off
permission bits in the *mode* argument supplied during calls to the
following functions:

 *   *open*(), *openat*(), *creat*(), *mkdir*(), *mkdirat*(), *mkfifo*(), and
     mkfifoat()

 *   *mknod*(), *mknodat*()

 *   *mq_open*()

 *   *sem_open*()

Bit positions that are set in *cmask* are cleared in the mode of
the created file.

RETURN VALUE   top

The file permission bits in the value returned by *umask*() shall
be the previous value of the file mode creation mask. The state
of any other bits in that value is unspecified, except that a
subsequent call to *umask*() with the returned value as *cmask* shall
leave the state of the mask the same as its state before the
first call, including any unspecified use of those bits.

ERRORS   top

No errors are defined.

The following sections are informative.

EXAMPLES   top

None.

APPLICATION USAGE   top

None.

RATIONALE   top

Unsigned argument and return types for *umask*() were proposed. The
return type and the argument were both changed to **mode_t.**

Historical implementations have made use of additional bits in
cmask for their implementation-defined purposes. The addition of
the text that the meaning of other bits of the field is
implementation-defined permits these implementations to conform
to this volume of POSIX.1-2017.

FUTURE DIRECTIONS   top

None.

SEE ALSO   top

creat(3p), exec(1p), mkdir(3p), mkfifo(3p), mknod(3p),
mq_open(3p), open(3p), sem_open(3p)

The Base Definitions volume of POSIX.1-2017, sys_stat.h(0p),
sys_types.h(0p)

COPYRIGHT   top

       Portions of this text are reprinted and reproduced in electronic
       form from IEEE Std 1003.1-2017, Standard for Information
       Technology -- Portable Operating System Interface (POSIX), The
       Open Group Base Specifications Issue 7, 2018 Edition, Copyright
       (C) 2018 by the Institute of Electrical and Electronics
       Engineers, Inc and The Open Group. In the event of any
       discrepancy between this version and the original IEEE and The
       Open Group Standard, the original IEEE and The Open Group
       Standard is the referee document. The original Standard can be
       obtained online at http://www.opengroup.org/unix/online.html .

       Any typographical or formatting errors that appear in this page
       are most likely to have been introduced during the conversion of
       the source files to man page format. To report such errors, see
       https://www.kernel.org/doc/man-pages/reporting_bugs.html .

IEEE/The Open Group   2017   UMASK(3P)

Pages that refer to this page: sys_stat.h(0p), mkdir(1p), sh(1p), umask(1p), exec(3p),
mkdir(3p), mkfifo(3p), mknod(3p), open(3p), posix_typed_mem_open(3p), shm_open(3p)