IEEE P1003.2 Draft 11.2 – September 1991

Copyright © 1991 by the
Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street
New York, NY 10017, USA
All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change. The publication, distribution, or copying of this draft, as well as all derivative works based on this draft, is expressly prohibited except as set forth below.

Permission is hereby granted for IEEE Standards Committee participants to reproduce this document for purposes of IEEE standardization activities only, and subject to the restrictions contained herein.

Permission is hereby also granted for member bodies and technical committees of ISO and IEC to reproduce this document for purposes of developing a national position, subject to the restrictions contained herein.

Permission is hereby also granted to the preceding entities to make limited copies of this document in an electronic form only for the stated activities.

The following restrictions apply to reproducing or transmitting the document in any form: 1) all copies or portions thereof must identify the document's IEEE project number and draft number, and must be accompanied by this entire notice in a prominent location; 2) no portion of this document may be redistributed in any modified or abridged form without the prior approval of the IEEE Standards Department.

Other entities seeking permission to reproduce this document, or any portion thereof, for standardization or other activities, must contact the IEEE Standards Department for the appropriate license.

Use of information contained in this unapproved draft is at your own risk.

IEEE Standards Department
Copyright and Permissions
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA
+1 (908) 562-3800
+1 (908) 562-1571 [FAX]

13521 # 4.67 umask — Get or set the file mode creation mask

13522 ## 4.67.1 Synopsis

13523 `umask [-S][mask ]`

13524 ## 4.67.2 Description

13525 The umask utility shall set the file mode creation mask of the current shell execu-
13526 tion environment (see 3.12) to the value specified by the *mask* operand. This
13527 mask shall affect the initial value of the file permission bits of subsequently
13528 created files.

13529 If the *mask* operand is not specified, the umask utility shall write to standard out-
13530 put the value of the invoking process's file mode creation mask.

13531 ## 4.67.3 Options

13532 The umask utility shall conform to the utility argument syntax guidelines
13533 described in 2.10.2.

13534 The following option shall be supported by the implementation:

13535 -S Produce symbolic output.

13536 The default output style is unspecified, but shall be recognized on a subsequent
13537 invocation of umask on the same system as a *mask* operand to restore the previ-
13538 ous file mode creation mask.

13539 ## 4.67.4 Operands

13540 The following operand shall be supported by the implementation:

13541 *mask* A string specifying the new file mode creation mask. The string
13542 is treated in the same way as the *mode* operand described in 4.7.7
13543 (chmod Extended Description).

13544 For a *symbolic_mode* value, the new value of the file mode crea-
13545 tion mask shall be the logical complement of the file permission
13546 bits portion of the file mode specified by the *symbolic_mode*
13547 string.

13548 In a *symbolic_mode* value, the permissions *op* characters + and -
13549 shall be interpreted relative to the current file mode creation
13550 mask; + shall cause the bits for the indicated permissions to be
13551 cleared in the mask; - shall cause the bits for the indicated per-
13552 missions to be set in the mask.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

The interpretation of *mode* values that specify file mode bits other than the file permission bits is unspecified.

In the obsolescent octal integer form of *mode* , the specified bits shall be set in the file mode creation mask.

The file mode creation mask shall be set to the resulting numeric value.

As in `chmod`, application use of the octal number form for the *mode* values is obsolescent.

The default output of a prior invocation of `umask` on the same system with no operand shall also be recognized as a *mask* operand. The use of an operand obtained in this way is not obsolescent, even if it is an octal number.

4.67.5 External Influences

4.67.5.1 Standard Input

None.

4.67.5.2 Input Files

None.

4.67.5.3 Environment Variables

The following environment variables shall affect the execution of `umask`:

LANG This variable shall determine the locale to use for the locale categories when both **LC_ALL** and the corresponding environment variable (beginning with **LC_**) do not specify a locale. See 2.6.

LC_ALL This variable shall determine the locale to be used to override any values for locale categories specified by the settings of **LANG** or any environment variables beginning with **LC_**.

LC_CTYPE This variable shall determine the locale for the interpretation of sequences of bytes of text data as characters (e.g., single- versus multibyte characters in arguments).

LC_MESSAGES This variable shall determine the language in which messages should be written.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

13585 #### 4.67.5.4 Asynchronous Events

13586 Default.

13587 ### 4.67.6 External Effects

13588 #### 4.67.6.1 Standard Output

13589 When the *mask* operand is not specified, the `umask` utility shall write a message
13590 to standard output that can later be used as a `umask` *mask* operand.

13591 If `-S` is specified, the message shall be in the following format:

13592 `"u=%s,g=%s,o=%s\n"`, <*owner permissions*> , <*group permissions*> ,
13593 <*other permissions*>

13594 where the three values shall be combinations of letters from the set {`r`, `w`, `x`}; the
13595 presence of a letter shall indicate that the corresponding bit is clear in the file
13596 mode creation mask.

13597 If a *mask* operand is specified, there shall be no output written to standard out-
13598 put.

13599 #### 4.67.6.2 Standard Error

13600 Used only for diagnostic messages.

13601 #### 4.67.6.3 Output Files

13602 None.

13603 ### 4.67.7 Extended Description

13604 None.

13605 ### 4.67.8 Exit Status

13606 The `umask` utility shall exit with one of the following values:

13607 0 The file mode creation mask was successfully changed, or no *mask*
13608 operand was supplied.

13609 >0 An error occurred.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

4.67.9 Consequences of Errors

Default.

4.67.10 Rationale. *(This subclause is not a part of P1003.2)*

Examples, Usage

Since `umask` affects the current shell execution environment, it is generally provided as a shell regular built-in. If it is called in a subshell or separate utility execution environment, such as one of the following:

```
(umask 002)
nohup umask ...
find . -exec umask ... \;
```

it will not affect the file mode creation mask of the caller's environment.

The table mapping octal mode values in 4.7.7 does not require that the symbolic constants have those particular values.

In contrast to the negative permission logic provided by the file mode creation mask and the octal number form of the *mask* argument, the symbolic form of the *mask* argument specifies those permissions that are left alone.

Either of the commands:

```
umask a=rx,ug+w
```

```
umask 002
```

sets the mode mask so that subsequently created files have their S_IWOTH bit cleared.

After setting the mode mask with either of the above commands, the `umask` command can be used to write out the current value of the mode mask:

```
$ umask
0002
```

(The output format is unspecified, but historical implementations use the obsolescent octal integer mode format.)

```
$ umask -S
u=rwx,g=rwx,o=rx
```

Either of these outputs can be used as the mask operand to a subsequent invocation of the `umask` utility.

Assuming the mode mask is set as above, the command:

```
umask g-w
```

sets the mode mask so that subsequently created files have their S_IWGRP , and S_IWOTH bits cleared.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

The command:

```
umask -- -w
```

sets the mode mask so that subsequently created files have all their write bits cleared. Note that *mask* operands –r, –w, –x, or anything beginning with a hyphen, must be preceded by `--` to keep it from being interpreted as an option.

History of Decisions Made

The description of the historical utility was modified to allow it to use the symbolic modes of `chmod`. The –s option used in earlier drafts was changed to –S because –s could be confused with a *symbolic_mode* form of mask referring to the S_ISUID and S_ISGID bits.

The default output style is implementation defined to permit implementors to provide migration to the new symbolic style at the time most appropriate to their users. Earlier drafts of this standard specified an –o flag to force octal mode output. This was dropped because the octal mode may not be sufficient to specify all of the information that may be present in the file mode creation mask when more secure file access permission checks are implemented.

It has been suggested that trusted systems developers might appreciate softening the requirement that the mode mask ''affects'' the file access permissions, since it seems access control lists might replace the mode mask to some degree. The wording has been changed to say that it affects the file permission bits, and leaves the details of the behavior of how they affect the file access permissions to the description in POSIX .1 {8}.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

4.68 uname — Return system name

4.68.1 Synopsis

```
uname [-amnrsv]
```

4.68.2 Description

By default, the uname utility shall write the operating system name to standard output. When options are specified, symbols representing one or more system characteristics shall be written to the standard output. The format and contents of the symbols are implementation defined. On systems conforming to POSIX.1 {8}, the symbols written shall be those supported by the POSIX.1 {8} *uname* () function.

4.68.3 Options

The uname utility shall conform to the utility argument syntax guidelines described in 2.10.2.

The following options shall be supported by the implementation:

- **-a** Behave as though all of the options -mnrsv were specified.
- **-m** Write the name of the hardware type on which the system is running to standard output.
- **-n** Write the name of this node within an implementation-specified communications network.
- **-r** Write the current release level of the operating system implementation.
- **-s** Write the name of the implementation of the operating system.
- **-v** Write the current version level of this release of the operating system implementation.

If no options are specified, the uname utility shall write the operating system name, as if the -s option had been specified.

4.68.4 Operands

None.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.