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All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change.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]

4.8 chown — Change file ownership

4.8.1 Synopsis

chown [–R] *owner* [: *group*] *file* ...

4.8.2 Description

The chown utility shall set the user ID of the file named by each *file* operand to the user ID specified by the *owner* operand.

For each *file* operand, it shall perform actions equivalent to the POSIX.1 {8} *chown* () function, called with the following arguments:

(1) The *file* operand shall be used as the *path* argument.

(2) The user ID indicated by the *owner* portion of the first operand shall be used as the *owner* argument.

(3) If the *group* portion of the first operand is given, the group ID indicated by it shall be used as the *group* argument; otherwise, the group ID of the file shall be used as the *group* argument.

4.8.3 Options

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

The following option shall be supported by the implementation:

–R Recursively change file user IDs, and if the *group* operand is specified, group IDs. For each *file* operand that names a directory, chown changes the user and group ID of the directory and all files in the file hierarchy below it.

4.8.4 Operands

The following operands shall be supported by the implementation:

owner [: *group*]
A user ID and optional group ID to be assigned to file. The *owner* portion of this operand shall be a user name from the user database or a numeric user ID. Either specifies a user ID to be given to each file named by one of the *file* operands. If a numeric *owner* operand exists in the user database as a user name, the user ID number associated with that user name is used as the user ID. Similarly, if the *group* portion of this operand is present, it shall be a group name from the group database or a numeric group ID. Either specifies a group ID to be given to each file. If a numeric

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

2840 group operand exists in the group database as a group name, the
2841 group ID number associated with that group name shall be used
2842 as the group ID.

2843 *file* A pathname of a file whose user ID is to be modified.

2844 ### 4.8.5 External Influences

2845 #### 4.8.5.1 Standard Input

2846 None.

2847 #### 4.8.5.2 Input Files

2848 None.

2849 #### 4.8.5.3 Environment Variables

2850 The following environment variables shall affect the execution of `chown`:

2851 **LANG** This variable shall determine the locale to use for the
2852 locale categories when both **LC_ALL** and the correspond-
2853 ing environment variable (beginning with **LC_**) do not
2854 specify a locale. See 2.6.

2855 **LC_ALL** This variable shall determine the locale to be used to over-
2856 ride any values for locale categories specified by the set-
2857 tings of **LANG** or any environment variables beginning
2858 with **LC_**.

2859 **LC_CTYPE** This variable shall determine the locale for the interpreta-
2860 tion of sequences of bytes of text data as characters (e.g.,
2861 single- versus multibyte characters in arguments).

2862 **LC_MESSAGES** This variable shall determine the language in which mes-
2863 sages should be written.

2864 #### 4.8.5.4 Asynchronous Events

2865 Default.

2866 ### 4.8.6 External Effects

2867 #### 4.8.6.1 Standard Output

2868 None.

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

4.8.6.2 Standard Error

Used only for diagnostic messages.

4.8.6.3 Output Files

None.

4.8.7 Extended Description

None.

4.8.8 Exit Status

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

0 The utility executed successfully and all requested changes were made.

>0 An error occurred.

4.8.9 Consequences of Errors

If, when invoked with the `-R` option, `chown` attempts but fails to change the user ID and/or, if the *group* operand is specified, group ID, of a particular file in a specified file hierarchy, it shall continue to process the remaining files in the hierarchy.

If `chown` cannot read or search a directory within a hierarchy, it shall continue to process the other parts of the hierarchy that are accessible.

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

Examples, Usage

The System V and BSD versions use different exit status codes. Some implementations used the exit status as a count of the number of errors that occurred; this practice is unworkable since it can overflow the range of valid exit status values. These are masked by specifying only 0 and >0 as exit values.

The functionality of `chown` is described substantially through references to functions in POSIX.1 {8}. In this way, there is no duplication of effort required for describing the interactions of permissions, multiple groups, etc.

For implementations on which symbolic links are supported, actual use of the *chown* () function to implement this utility might not be the appropriate, depending on the implementation.

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

History of Decisions Made

The 4.3 BSD method of specifying both owner and group was included in this standard because:

(1) There are cases where the desired end condition could not be achieved using the `chgrp` and `chown` (that only changed the user ID) utilities. [If the current owner is not a member of the desired group and the desired owner is not a member of the current group, the *chown* () function could fail unless both owner and group are changed at the same time.]

(2) Even if they could be changed independently, in cases where both are being changed, there is a 100 percent performance penalty caused by being forced to invoke both utilities.

The BSD syntax *user*`[.`*group*`]` was changed to *user*`[:`*group*`]` in POSIX.2 because the period is a valid character in login names (as specified by POSIX.1 {8}, login names consist of characters in the portable filename character set). The colon character was chosen as the replacement for the period character because it would never be allowed as a character in a user name or group name on traditional implementations.

The `–R` option is considered by some observers as an undesirable departure from the traditional UNIX system tools approach; since a tool, `find`, already exists to recurse over directories, there was felt to be no good reason to require other tools to have to duplicate that functionality. However, the `–R` option was deemed an important user convenience, is far more efficient than forking a separate process for each element of the directory hierarchy, and is in widespread historical use.

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

4.9 cksum — Write file checksums and sizes

4.9.1 Synopsis

```
cksum [file ...]
```

4.9.2 Description

The cksum utility shall calculate and write to standard output a cyclic redundancy check (CRC) for each input file, and also write to standard output the number of octets in each file. The CRC used is based on the polynomial used for CRC error checking in the networking standard ISO 8802-3 {B7}.

The CRC checksum shall be obtained in the following way:

The encoding is defined by the generating polynomial:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

Mathematically, the CRC value corresponding to a given file shall be defined by the following procedure:

(1) The n bits to be evaluated are considered to be the coefficients of a mod 2 polynomial $M(x)$ of degree $n-1$. These n bits are the bits from the file, with the most significant bit being the most significant bit of the first octet of the file and the last bit being the least significant bit of the last octet, padded with zero bits (if necessary) to achieve an integral number of octets, followed by one or more octets representing the length of the file as a binary value, least significant octet first. The smallest number of octets capable of representing this integer shall be used.

(2) $M(x)$ is multiplied by x^{32} (i.e., shifted left 32 bits) and divided by $G(x)$ using mod 2 division, producing a remainder $R(x)$ of degree ≤ 31.

(3) The coefficients of $R(x)$ are considered to be a 32-bit sequence.

(4) The bit sequence is complemented and the result is the CRC.

4.9.3 Options

None.

4.9.4 Operands

The following operand shall be supported by the implementation:

file A pathname of a file to be checked. If no *file* operands are specified, the standard input is used.

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