IEEE P1003.2a Draft 8 – December 1991

5.32 unexpand — Convert spaces to tabs

5.32.1 Synopsis

`unexpand` [`–a` | `–t` *tablist*] [*file* . . .]

5.32.2 Description

The `unexpand` utility shall copy files or standard input to standard output, converting `<blank>`s at the beginning of each line into the maximum number of `<tab>`s followed by the minimum number of `<space>`s needed to fill the same column positions originally filled by the translated `<blank>`s. By default, tabstops shall be set at every eighth column position. Each `<backspace>` character shall be copied to the output, and shall cause the column position count for tab calculations to be decremented; the count shall never be decremented to a value less than one.

5.32.3 Options

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

The following option shall be supported by the implementation:

`–a` In addition to translating `<blank>`s at the beginning of each line, translate all sequences of two or more `<blank>`s immediately preceding a tab stop to the maximum number of `<tab>`s followed by the minimum number of `<space>`s needed to fill the same column positions originally filled by the translated `<blank>`s.

`–t` *tablist* Specify the tab stops. The option-argument *tablist* shall be a sin- 8
gle argument consisting of a single positive decimal integer or 8
multiple positive decimal integers, separated by `<blank>`s or commas, in ascending order. If a single number is given, tabs shall be set *tablist* column positions apart instead of the default 8. If multiple numbers are given, the tabs shall be set at those specific column positions.

Each tab-stop position *N* shall be an integer value greater than zero, and the list shall be in strictly ascending order. This shall be taken to mean that, from the start of a line of output, tabbing to position *N* shall cause the next character output to be in the (*N*+1)th column position on that line. When the `–t` option is not specified, the default shall be the equivalent of specifying `-t 8` (except for the interaction with `–a`, described below).

No `<space>`-to-`<tab>` conversions shall occur for characters at positions beyond the last of those specified in a multiple tab-stop list.

When `-t` is specified, the presence or absence of the `-a` option shall be ignored; conversion shall not be limited to the processing of leading `<blank>`s.

5.32.4 Operands

The following operand shall be supported by the implementation:

file A pathname of a text file to be used as input.

5.32.5 External Influences

5.32.5.1 Standard Input

See Input Files.

5.32.5.2 Input Files

The input files shall be text files.

5.32.5.3 Environment Variables

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

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 interpretation of sequences of bytes of text data as characters (e.g., single- versus multibyte characters in arguments and input files), the processing of `<tab>`s and `<space>`s, and for the determination of the width in column positions each character would occupy on a constant-width-font output device.

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

5.32.5.4 Asynchronous Events

Default.

5.32.6 External Effects

5.32.6.1 Standard Output

The standard output is equivalent to the input files with the specified `<space>` to `<tab>` conversions.

5.32.6.2 Standard Error

Used only for diagnostic messages.

5.32.6.3 Output Files

None.

5.32.7 Extended Description

None.

5.32.8 Exit Status

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

- 0 Successful completion.
- >0 An error occurred.

5.32.9 Consequences of Errors

Default.

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

Usage, Examples

None.

History of Decisions Made

On several occasions, consideration was given to adding a `−t` option to the `unexpand` utility to complement the `−t` in `expand` (see 5.11). The historical intent of `unexpand` was to translate multiple `<blank>`s into tab stops, where tab stops were a multiple of eight column positions on most UNIX systems. A previous draft omitted `–t` because it seemed outside the UPE scope; it was not described in any of the Base Documents. However, a good case was made that hard-coding tab stops every eight columns was not suitable for the international community and broke historical precedents for some vendors in the FORTRAN community, so `–t` was restored in conjunction with the list of valid extension categories considered by the working group. Thus, `unexpand` is now the logical converse of `expand`, as one might expect.

One nonintuitive aspect of `unexpand` is its restriction to leading spaces when `–a` (and now, `–t`) are not specified. This is historical practice and there is no compelling reason to change it. Users who desire converting all spaces in the file can easily alias `unexpand` to use the `–a` or `-t 8` option.

5.33 uudecode — Decode a binary file

5.33.1 Synopsis

`uudecode` [*file*]

5.33.2 Description

The `uudecode` utility shall read a file, or standard input if no file is specified, that includes data created by the `uuencode` utility (see 5.34). The `uudecode` utility shall scan the input file, searching for data compatible with the format specified in 5.34.6.1 and attempt to create or overwrite the file described by the data. The pathname, file access permission bits, and contents for the file to be produced shall all be contained in that data. The mode bits of the created file shall be set from the file access permission bits contained in the data; i.e., other attributes of the mode, including the file mode creation mask (see `umask` in 4.67), 7
shall not affect the file being produced. 7

If the pathname of the file to be produced exists, and the user does not have write permission on that file, `uudecode` shall terminate with an error. If the pathname of the file to be produced exists, and the user has write permission on that file, the existing file shall be overwritten.

If the input data was produced by `uuencode` on a system with a different number of bits per byte than on the target system, the results of `uudecode` are unspecified.