NAME

tabs - set terminal tab stops

SYNOPSIS

tabs [*options*] [*tabstop-list*]

DESCRIPTION

The **tabs** program clears and sets tab-stops on the terminal. This uses the terminfo **clear_all_tabs** and **set_tab** capabilities. If either is absent, **tabs** is unable to clear/set tab-stops. The terminal should be configured to use hard tabs, e.g.,

```
stty tab0
```

Like **clear**(1), **tabs** writes to the standard output. You can redirect the standard output to a file (which prevents **tabs** from actually changing the tabstops), and later **cat** the file to the screen, setting tabstops at that point.

These are hardware tabs, which cannot be queried rapidly by applications running in the terminal, if at all. Curses and other full-screen applications may use hardware tabs in optimizing their output to the terminal. If the hardware tabstops differ from the information in the terminal database, the result is unpredictable. Before running curses programs, you should either reset tab-stops to the standard interval

```
tabs -8
```

or use the **reset** program, since the normal initialization sequences do not ensure that tab-stops are reset.

OPTIONS

General Options

-T*name*
Tell **tabs** which terminal type to use. If this option is not given, **tabs** will use the **$TERM** environment variable. If that is not set, it will use the *ansi+tabs* entry.

-d The debugging option shows a ruler line, followed by two data lines. The first data line shows the expected tab-stops marked with asterisks. The second data line shows the actual tab-stops, marked with asterisks.

-n This option tells **tabs** to check the options and run any debugging option, but not to modify the terminal settings.

-V reports the version of *ncurses* which was used in this program, and exits.

The **tabs** program processes a single list of tab stops. The last option to be processed which defines a list is the one that determines the list to be processed.

Implicit Lists

Use a single number as an option, e.g., "**-5**" to set tabs at the given interval (in this case 1, 6, 11, 16, 21, etc.). Tabs are repeated up to the right margin of the screen.

Use "**-0**" to clear all tabs.

Use "**-8**" to set tabs to the standard interval.

Explicit Lists

An explicit list can be defined after the options (this does not use a "**-**"). The values in the list must be in increasing numeric order, and greater than zero. They are separated by a comma or a blank, for example,

```
tabs 1,6,11,16,21
tabs 1 6 11 16 21
```

Use a "+" to treat a number as an increment relative to the previous value, e.g.,

```
tabs 1,+5,+5,+5,+5
```

which is equivalent to the 1,6,11,16,21 example.

Predefined Tab Stops

POSIX defines several predefined lists of tab stops.

-a Assembler, IBM S/370, first format
1,10,16,36,72

-a2 Assembler, IBM S/370, second format
1,10,16,40,72

-c COBOL, normal format
1,8,12,16,20,55

-c2 COBOL compact format

1,6,10,14,49

-c3 COBOL compact format extended
1,6,10,14,18,22,26,30,34,38,42,46,50,54,58,62,67

-f FORTRAN
1,7,11,15,19,23

-p PL/I
1,5,9,13,17,21,25,29,33,37,41,45,49,53,57,61

-s SNOBOL
1,10,55

-u UNIVAC 1100 Assembler
1,12,20,44

Margins

A few terminals expose a means of changing their left and right margins. **tabs** supports this feature with an option.

+m *margin*
The effect depends on whether the terminal has the margin capabilities:

- If the terminal provides the capability for setting the left margin, **tabs** uses this, and adjusts the available tab stop widths.

- If the terminal does not provide the margin capabilities, **tabs** imitates their effect, putting tab stops at appropriate places on each line. The terminal's left margin is not modified.

If the *margin* parameter is omitted, the default is 10. Use **+m0** to reset the left margin, that is, to make it the left edge of the terminal's display. Before setting a left margin, **tabs** resets the margin to reduce problems that might arise from moving the cursor to the left of the current left margin.

When setting or resetting the left margin, **tabs** may also reset the right margin.

FILES

/usr/share/tabset
tab stop initialization database

PORTABILITY

IEEE Std 1003.1/The Open Group Base Specifications Issue 7 (POSIX.1-2008) describes a **tabs** utility. However,

- this standard describes a **+m** option to set a terminal’s left margin. Very few of the entries in the terminal database provide the **set_left_margin** (**smgl**) or **set_left_margin_parm** (**smglp**) capabilities needed to support the feature.

- There is no counterpart in X/Open Curses Issue 7 for this utility, unlike **tput**(1).

The **-d** (debug) and **-n** (no-op) options are *ncurses* extensions not provided by other implementations.

HISTORY

A **tabs** utility appeared in PWB/Unix 1.0 (1977). A reduced version shipped in Seventh Edition Unix (early 1979) and in 3BSD (later the same year); it supported a "-n" option to set the first tab stop at the left margin. That option is not specified by POSIX.

The PWB/Unix **tabs** utility returned in System III (1980), and used built-in tables to support a half-dozen hardcopy terminal (printer) types. It also had logic to support setting the left margin, as well as a feature for copying the tab settings from a file.

Versions of the program in later releases of AT&T Unix, such as SVr4, added support for the terminal database, but retained the tables to support the printers. By this time, System V **tput** had incorporated the tab stop initialization feature of BSD’s **tset** from 1982, but employed the *terminfo* database to do so.

The **+m** option was documented in the POSIX Base Specifications Issue 5 (Unix98, 1997), then omitted in Issue 6 (Unix03, 2004) without express motivation, though an introductory comment "and optionally adjusts the margin" remains, overlooked in the removal. The **tabs** utility documented in Issues 6 and later has no mechanism for setting margins. The **+m** option in *ncurses* **tabs** differs from the SVr4 feature by using terminal capabilities rather than built-in tables.

POSIX documents no limit on the number of tab stops. Other implementations impose one; the limit is 20 in PWB/Unix’s **tabs** utility. While some terminals may not accept an arbitrary number of tab stops, *ncurses* **tabs** attempts to set tab stops up to the right margin if the list thereof is sufficiently long.

The "Rationale" section of the Issue 6 **tabs** reference page details how the committee considered redesigning the **tabs** and **tput** utilities, without settling on an improved solution. It claims that

> "no known historical version of *tabs* supports the capability of setting arbitrary tab stops."

The feature described in subsection "Explicit Lists" above was implemented in PWB/Unix, and permitted the setting of abitrary tab stops nevertheless.

SEE ALSO

infocmp(1M), **tset**(1), **curses**(3X), **terminfo**(5)