NAME

fmt — break, squeeze, and fill lines

SYNOPSIS

fmt [**-**{**m**|**w**} *max*|**-***max*] [**-g** *goal*] [*file*]...

DESCRIPTION

Breaks and fills lines from *file*s (standard input stream if "**-**", the default) such that their width approximates *goal* without going over *max* to the standard output stream. Lines may be indented by blanks, and different indentation levels are filled separately.

"Words" are delimited by white-space; inter-word white-space is squeezed into a single space.
Blank lines are unfilled and normalised to empty ones.

Invalid and non-printable characters are taken to have null width.

OPTIONS

-w, **-m**, **--width**=*max*, **-***max* Maximum filled line width (unless a single word is longer). Defaults to **75**.

-g, **--goal**=*goal* Fill words to get closest to the *goal* width. Defaults to .87*width* ≡ 65.

EXIT STATUS

1 if a *file* couldn't be opened or read.

EXAMPLES

```
$ sed -n 19,31p reply
Since I first posted this, procps free(1) started using MemAvailable to evaluate f

A Hurd image from 2021 I have (bullseye branding)
and  the  2023  release  (bookworm branding)
don't have     MemAvailable,  neither does
kFreeBSD 10 (from the 2017 installer ISO;
 appears to be the latest from
 http://wiki.debian.org/Debian_GNU/kFreeBSD).

I've updated the Salsa revision and
am including an updated patch here,
which overrides MemFree with
MemAvailable if available.
$ sed -n 19,31p reply | fmt
Since I first posted this, procps free(1) started using MemAvailable
to evaluate free/used, so sure. I don't feel strongly either way.

A Hurd image from 2021 I have (bullseye branding) and the 2023
release (bookworm branding) don't have MemAvailable, neither does
kFreeBSD 10 (from the 2017 installer ISO;
 appears to be the latest from http://wiki.debian.org/Debian_GNU/kFreeBSD).

I've updated the Salsa revision and am including an updated patch
here, which overrides MemFree with MemAvailable if available.
```

SEE ALSO

fold(1) **-s**, iswspace(3), mbrtowc(3), wcwidth(3)

In vi(1), running **!**}**fmt** will re-flow (format) the paragraph starting at the current line, **!***3*}**fmt** three, and **!**G**fmt** everything. Indeed, making mail palatable before sending is the primary use of **fmt**.

STANDARDS

Only the

fmt [**−w** *max*|**−***max*] < *file*

usage is globally portable, since some implementations take *goal* and *max* as positional arguments before *file*.

Deriving of *goal* from *max* (or vice versa) varies: on the GNU system, *max* = **75** and *goal* = .93*max*, rounded normally; on AT&T System V Release 4 UNIX *max* = 72 (and no separate *goal*); on OpenBSD and FreeBSD, *goal* = 65 and *max* = *goal* + 10, but if **−w** then *goal* = *max* (no **−g**); on NetBSD *goal* = 65 and *max* = 75.

BSD-derived implementations handle backspaces and don't format lines starting with dots '**.**'. Most implementations don't squeeze white-space by default.

HISTORY

Appeared fully-formed in 2BSD as `fmt`(UCB) ("fmt − simple text formatter") with a **SYNOPSIS** of

fmt [name ...]

filling until **72** columns. Present-day indentation behaviour (incl. expanding and unexpanding start-of-line tabs) is present, "interword spacing" is preserved (even though inter-word tabs aren't). For the purposes of line width, the backspace (**0x8**) is accepted as being *−1* wide (and not going past the beginning of the line). Other non-printable bytes except the tab (**0x9**) are discarded. If a word ends in a '**.**', '**:**', or '**!**' it's followed by two spaces.

*From_* lines are unfilled (and only split if they overrun **72** columns), and the the clear intent is that the following two (three?) lines, *if* they start with "**To**", "**Cc**", or "**Subj**" behave the same, but the string-prefix check is broken, and instead matches lines starting with '**T**' or '**C**'. This corresponds to a "Designed for use with Mail ~|" comment at the start of the implementation and a "*Fmt* is meant to format mail messages prior to sending" note in the manual.

The **BUGS** note that "The program was designed to be simple and fast − for more complex operations, the standard text processors are likely to be more appropriate.", and **SEE ALSO** bears `nroff`(1) and `roff`(1) — lines that start with dots ('**.**') are also just split.

3BSD more specifically matches "Subject" instead and doesn't break lines on whitespace "escaped" by backslashes (but, since word breaking happens after expansion, escaping works only if a tab yielded a single space). Lines that start with "From " but aren't valid *From_* lines emit warnings to the standard error stream. Thus, **echo** `'From Regents Regents Regents Regents Regents Regents Regent\ aaaa'` | **fmt** yields

```
"From Regents Regents Regents Regents Regents Regents Regents Regent\ aaaa"
not a header because Date field not legal date
From Regents Regents Regents Regents Regents Regents Regents
Regent\ aaaa
```

`fmt`(1) gains a use-**!**}**fmt**-in-`ex`(1)-visual-mode stanza, much like present-day.

4BSD no longer spews errors.

4.2BSD also follows '**?**' with two spaces, and treats any arguments starting with a '**-**' as **−***width* (interspersed with *file*s; **fmt −***width* continues to read the standard input stream), constrained to [**1**, **1021**]. This is not documented.

4.3BSD−Tahoe instead sees 3BSD **fmt** patched by "LIZ@UOM 6/18/85", taking

```
Syntax : fmt [ goal [ max ] ] [ name ... ]
```

where *goal* and *max* are consumed, in that order, if they start with a(n optional '**−**' and a) digit. The only constraint is that *max* > *goal*; thus, output lines exceeding the aforementioned overrun a buffer and segfault. The goal-/max-width word allocation strategy is equivalent to this implementation's. The defaults match this implementation's **65** and **75**. None of this makes it to the manual.

4.3BSD−Reno copies lines starting with dots verbatim (thus actually ensuring `roff`(1) input is intact), and propagates the new usage to the **SYNOPSIS** as

fmt [*goal* [*maximum*]] [name ...]

The 4.2BSD-based SunOS carries its **fmt** (though branded as having come from 4.1BSD).

SunOS 2 sees a **SYNOPSIS** of

fmt [-**width**] [-**c**] [*filename* ...]

documenting **-***width* for the first time, and inventing **-c** as the "Crown margin mode - the first two lines following an empty line retain their indention. Subsequent non-blank lines are aligned with the second.". Thus:

```
$ fmt -25 -c
      The quick
    brown fox
  jumps over
the lazy dog.
^D
      The quick brown
    fox jumps over the
    lazy dog.
```

SunOS 3 sees an equivalent-but-reformatted **SYNOPSIS** of

fmt [**−***width*] [**-c**] [*input-file* ...]

and says that "*Fmt* does not fill lines beginning with **.** or **From:** for compatibility with *nroff*(1) and *mail*(1).". While it's impossible to validate whether the '**.**' behaviour changed to 4.3BSD−Reno-like to make mentioning this note-worthy, the parsed headers are unchanged; it's most likely that whoever was updating the manual mis-took the **Mail** *From_* line parsing as the "**From:**" header (especially since the actual handled "headers" are not mentioned).

SunOS 4 (Solaris) sees a **SYNOPSIS** of

fmt [**−cs**] [**−***width*] [*inputfile*...]

-s makes **fmt** behave like present-day **fold -s** — breaking at whitespace for *width* without filling — though it's the only available spelling for that behaviour.

The don't-fill-after-*From_*-line prefix list gains "**cc**", "**Bcc**", and "**bcc**", but more importantly fixes the comparison to actually work. Up to six "header" lines are recognised "because some users like to put cc's on multiple lines".

A **fmt_mail** ("for full compatibility with mail(1)") shares the **SYNOPSIS** and bears the same one (though for some reason formatted differently) — this is wrong, since its actual usage is

fmt_mail [*recipients*]... < *file*

with *recipients* passed directly to sendmail(8). It's better suited for processing actual mail, because it doesn't touch any lines before the first empty one, then passes the rest of the input to **fmt -s**. Thus, both the *From_* line and any and all headers are left alone, including multi-line ones. Passing **-s** is safest since it "prevents sample lines of code, and other such "formatted" text, from being unduly combined", but is nevertheless sub-optimal unless you write everything in a single line per paragraph.

The input *bytes* are classified according to the locale, and non-printables are deleted if they're non-printables in the current locale. This works if the locale is single-byte, but not otherwise.

AT&T System V Release 4 UNIX includes SunOS 4 (Solaris)' **fmt** sans the locale changes, except, undocumentedly-except-in-the-usage-string, re-defining the usage as

```
usage: fmt [-c] [-s] [-w width | -width] [inputfile...]
```

parsing the arguments strictly before *inputfile*s, and erroring out for invalid **-***width*s instead of stopping parsing at the first non-digit, and, thus having "**-S**" be **-***0*. The manual notes that **-***width* is "acceptable for BSD compatibility, but it may go away in future releases". The illumos gate carries this to this day, 33 years later.

SunOS 5 (Solaris 2), now based on AT&T System V Release 4 UNIX, nevertheless derives its **fmt** from SunOS 4 (Solaris)', but actually reading the input as characters and accepting **-w** *width* (still interspersed).