

NAME

fortune – print a random, hopefully interesting, adage

SYNOPSIS

fortune [**-acefilosu**w] [**-n** *length*] [**-m** *pattern*] [[*n*%] *file/dir/all*]

DESCRIPTION

When **fortune** is run with no arguments it prints out a random epigram. Epigrams are divided into several categories, where each category is sub-divided into those which are potentially offensive and those which are not.

Options

The options are as follows:

- a** Choose from all lists of maxims, both offensive and not. (See the **-o** option for more information on offensive fortunes.)
- c** Show the cookie file from which the fortune came.
- e** Consider all fortune files to be of equal size (see discussion below on multiple files).
- f** Print out the list of files which would be searched, but don't print a fortune.
- l** Long dictums only. See **-n** on how "long" is defined in this sense.
- m** *pattern*
Print out all fortunes which match the basic regular expression *pattern*. The syntax of these expressions depends on how your system defines **re_comp(3)** or **regcomp(3)**, but it should nevertheless be similar to the syntax used in **grep(1)**.

The fortunes are output to standard output, while the names of the file from which each fortune comes are printed to standard error. Either or both can be redirected; if standard output is redirected to a file, the result is a valid fortunes database file. If standard error is *also* redirected to this file, the result is *still valid*, **but there will be "bogus" fortunes**, i.e. the filenames themselves, in parentheses. This can be useful if you wish to remove the gathered matches from their original files, since each filename-record will precede the records from the file it names.

-n *length*

Set the longest fortune length (in characters) considered to be "short" (the default is 160). All fortunes longer than this are considered "long". Be careful! If you set the length too short and ask for short fortunes, or too long and ask for long ones, fortune goes into a never-ending thrash loop.

- o** Choose only from potentially offensive aphorisms. The **-o** option is ignored if a fortune directory is specified.

Please, please, please request a potentially offensive fortune if and only if you believe, deep in your heart, that you are willing to be offended. (And that you'll just quit using **-o rather than give us grief about it, okay?)**

... let us keep in mind the basic governing philosophy of The Brotherhood, as handsomely summarized in these words: we believe in healthy, hearty laughter — at the expense of the whole human race, if needs be. Needs be.

—H. Allen Smith, "Rude Jokes"

- s** Short apothegms only. See **-n** on which fortunes are considered "short".
- i** Ignore case for **-m** patterns.
- w** Wait before termination for an amount of time calculated from the number of characters in the message. This is useful if it is executed as part of the logout procedure to guarantee that the message can be read before the screen is cleared.

-u Don't translate UTF-8 fortunes to the locale when searching or translating.

The user may specify alternate sayings. You can specify a specific file, a directory which contains one or more files, or the special word *all* which says to use all the standard databases. Any of these may be preceded by a percentage, which is a number *n* between 0 and 100 inclusive, followed by a *%*. If it is, there will be a *n* percent probability that an adage will be picked from that file or directory. If the percentages do not sum to 100, and there are specifications without percentages, the remaining percent will apply to those files and/or directories, in which case the probability of selecting from one of them will be based on their relative sizes.

As an example, given two databases *funny* and *not-funny*, with *funny* twice as big (in number of fortunes, not raw file size), saying

```
fortune funny not-funny
```

will get you fortunes out of *funny* two-thirds of the time. The command

```
fortune 90% funny 10% not-funny
```

will pick out 90% of its fortunes from *funny* (the "10% not-funny" is unnecessary, since 10% is all that's left).

The **-e** option says to consider all files equal; thus

```
fortune -e funny not-funny
```

is equivalent to

```
fortune 50% funny 50% not-funny
```

FILES

Note: these are the defaults as defined at compile time.

/usr/share/games/fortunes

Directory for inoffensive fortunes.

/usr/share/games/fortunes/off

Directory for offensive fortunes.

If a particular set of fortunes is particularly unwanted, there is an easy solution: delete the associated *.dat* file. This leaves the data intact, should the file later be wanted, but since **fortune** no longer finds the pointers file, it ignores the text file.

BUGS

The supplied fortune databases have been attacked, in order to correct orthographical and grammatical errors, and particularly to reduce redundancy and repetition and redundancy. But especially to avoid repetitiousness. This has not been a complete success. In the process, some fortunes may also have been lost.

The fortune databases are now divided into a larger number of smaller files, some organized by format (poetry, definitions), and some by content (religion, politics). There are parallel files in the main directory and in the offensive files directory (e.g., *fortunes/definitions* and *fortunes/off/definitions*). Not all the potentially offensive fortunes are in the offensive fortunes files, nor are all the fortunes in the offensive files potentially offensive, probably, though a strong attempt has been made to achieve greater consistency. Also, a better division might be made.

When passing files to fortune, directories must be specified by absolute pathnames, and filenames starting with a dot are ignored. See: <http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=241888>

HISTORY

This version of fortune is based on the NetBSD fortune 1.4, but with a number of bug fixes and enhancements.

The original fortune/strfile format used a single file; strfile read the text file and converted it to null-delimited strings, which were stored after the table of pointers in the .dat file. By NetBSD fortune 1.4, this had changed to two separate files: the .dat file was only the header (the table of pointers, plus flags; see *strfile.h*), and the text strings were left in their own file. The potential problem with this is that text file and header file may get out of synch, but the advantage is that the text files can be easily edited without resorting to unstr, and there is a potential savings in disk space (on the assumption that the sysadmin kept both .dat file with strings and the text file).

Many of the enhancements made over the NetBSD version assumed a Linux system, and thus caused it to fail under other platforms, including BSD. The source code has since been made more generic, and currently works on SunOS 4.x as well as Linux, with support for more platforms expected in the future. Note that some bugs were inadvertently discovered and fixed during this process.

At a guess, a great many people have worked on this program, many without leaving attributions.

SEE ALSO

re_comp(3), **regcomp(3)**, **strfile(1)**, **unstr(1)**