NAME

strfile − create a random access file for storing strings
unstr − dump strings in pointer order

SYNOPSIS

strfile [**−iorsx**] [**−c** *char*] *sourcefile* [*outputfile*]
unstr [**−c** *char*] *datafile*[*.ext*] [*outputfile*]

DESCRIPTION

strfile reads a file containing groups of lines separated by a line containing a single percent '%' sign (or other specified delimiter character) and creates a data file which contains a header structure and a table of file offsets for each group of lines. This allows random access of the strings.

The output file, if not specified on the command line, is named *sourcefile.dat*.

The purpose of **unstr** is to undo the work of **strfile**. It prints out the strings contained in the sourcefile, which is *datafile.ext* without its extension, or *datafile* if no extension is specified (in this case, the extension *.dat* is added to the name of the datafile) in the order that they are listed in the header file *datafile*. If no *outputfile* is specified, it prints to standard output; otherwise it prints to the file specified. **unstr** can also universally change the delimiter character in a strings file. It is possible to create sorted versions of input files by using **strfile −o** and then using **unstr** to dump them out in the table order.

Options

The options are as follows:

−c *char* Change the delimiting character from the percent sign to *char*. This option is available for both **strfile** and **unstr**.

−i Ignore case when ordering the strings.

−o Order the strings in alphabetical order. The offset table will be sorted in the alphabetical order of the groups of lines referenced. Any initial non-alphanumeric characters are ignored. This option causes the STR_ORDERED bit in the header *str_flags* field to be set. (It also now really does sort! It didn't used to).

−r Randomize access to the strings. Entries in the offset table will be randomly ordered. This option causes the STR_RANDOM bit in the header *str_flags* field to be set. (And really does randomize)

−s Run silently; don't give a summary message when finished.

−x Note that each alphabetic character in the groups of lines is rotated 13 positions in a simple caesar cypher. This option causes the STR_ROTATED bit in the header *str_flags* field to be set. Note that it **does not** rotate the strings--that operation must be performed separately.

Header

The format of the header is:

```
#define VERSION 1
unsigned long str_version;  /* version number */
unsigned long str_numstr;   /* # of strings in the file */
unsigned long str_longlen;  /* length of longest string */
unsigned long str_shortlen; /* shortest string length */
#define STR_RANDOM    0x1   /* randomized pointers */
#define STR_ORDERED   0x2   /* ordered pointers */
#define STR_ROTATED   0x4   /* rot-13'd text */
unsigned long str_flags;    /* bit field for flags */
char str_delim;             /* delimiting character */
```

All fields are written in network byte order.

BUGS

Fewer now, one hopes. However, fortunes (text strings) beginning with a blank line appear to be sorted between random letters. This includes ASCII art that contains no letters, and first lines that are solely non-

alphanumeric, apparently. I've no idea why this should be.

OTHER USES

What can you do with this besides printing sarcastic and obscene messages to the screens of lusers at login or logout?

There **are** some other possibilities. Source code for a sample program, **randstr**, is included with this distribution: randstr splits the difference between **unstr** and **fortune**. It reads a single, specified file, and randomly selects a single text string.

1 Include *strfile.h* into a news reading/posting program, to generate random signatures. **T in**(1) does something similar, in a much more complex manner.

2 Include it in a game. While strfile doesn't support 'fields' or 'records', there's no reason that the text strings can't be consistent: first line, a die roll; second line, a score; third and subsequent lines, a text message.

3 Use it to store your address book. Hell, some of the guys I know would be as well off using it to decide who to call on Friday nights (and for some, it wouldn't matter whether there were phone numbers in it or not).

4 Use it in 'lottery' situations. If you're an ISP, write a script to store login names and GECOS from */etc/passwd* in strfile format, write another to send 'congratulations, you've won' to the lucky login selected. The prize might be a month's free service, or if you're AOL, a month free on a real service provider.

SEE ALSO

byteorder(3), **fortune**(6)

HISTORY

The **strfile** utility first appeared in 4.4BSD. This version was heavily modified, much of it in ways peculiar to Linux. Work has since been done to make the code more generic, and has so far been tested to work with SunOS 4.x. More platforms are expected to be supported as work continues.