NAME

strings − print the sequences of printable characters in files

SYNOPSIS

strings [**−afovV**] [**−***min-len*]
[**−n** *min-len*] [**−−bytes=***min-len*]
[**−t** *radix*] [**−−radix=***radix*]
[**−e** *encoding*] [**−−encoding=***encoding*]
[**−**] [**−−all**] [**−−print−file−name**]
[**−T** *bfdname*] [**−−target=***bfdname*]
[**−w**] [**−−include−all−whitespace**]
[**−s**] [**−−output−separator***sep_string*]
[**−−help**] [**−−version**] *file...*

DESCRIPTION

For each *file* given, GNU **strings** prints the printable character sequences that are at least 4 characters long (or the number given with the options below) and are followed by an unprintable character.

Depending upon how the strings program was configured it will default to either displaying all the printable sequences that it can find in each file, or only those sequences that are in loadable, initialized data sections. If the file type in unrecognizable, or if strings is reading from stdin then it will always display all of the printable sequences that it can find.

For backwards compatibility any file that occurs after a command-line option of just **−** will also be scanned in full, regardless of the presence of any **−d** option.

strings is mainly useful for determining the contents of non-text files.

OPTIONS

−a
−−all
− Scan the whole file, regardless of what sections it contains or whether those sections are loaded or initialized. Normally this is the default behaviour, but strings can be configured so that the **−d** is the default instead.

The **−** option is position dependent and forces strings to perform full scans of any file that is mentioned after the **−** on the command line, even if the **−d** option has been specified.

−d
−−data
Only print strings from initialized, loaded data sections in the file. This may reduce the amount of garbage in the output, but it also exposes the strings program to any security flaws that may be present in the BFD library used to scan and load sections. Strings can be configured so that this option is the default behaviour. In such cases the **−a** option can be used to avoid using the BFD library and instead just print all of the strings found in the file.

−f
−−print−file−name
Print the name of the file before each string.

−−help
Print a summary of the program usage on the standard output and exit.

−*min-len*
−n *min-len*
−−bytes=*min-len*
Print sequences of characters that are at least *min-len* characters long, instead of the default 4.

−o Like **−t o**. Some other versions of **strings** have **−o** act like **−t d** instead. Since we can not be compatible with both ways, we simply chose one.

−t *radix*
−−radix=*radix*
Print the offset within the file before each string. The single character argument specifies the radix of the offset−−−**o** for octal, **x** for hexadecimal, or **d** for decimal.

−e *encoding*
−−encoding=*encoding*
Select the character encoding of the strings that are to be found. Possible values for *encoding* are: **s** = single−7−bit−byte characters (ASCII, ISO 8859, etc., default), **S** = single−8−bit−byte characters, **b** = 16−bit bigendian, **l** = 16−bit littleendian, **B** = 32−bit bigendian, **L** = 32−bit littleendian. Useful for finding wide character strings. (**l** and **b** apply to, for example, Unicode UTF−16/UCS−2 encodings).

−T *bfdname*
−−target=*bfdname*
Specify an object code format other than your system's default format.

−v
−V
−−version
Print the program version number on the standard output and exit.

−w
−−include−all−whitespace
By default tab and space characters are included in the strings that are displayed, but other whitespace characters, such a newlines and carriage returns, are not. The **−w** option changes this so that all whitespace characters are considered to be part of a string.

−s
−−output−separator
By default, output strings are delimited by a new-line. This option allows you to supply any string to be used as the output record separator. Useful with −−include−all−whitespace where strings may contain new-lines internally.

@*file*
Read command-line options from *file*. The options read are inserted in place of the original @*file* option. If *file* does not exist, or cannot be read, then the option will be treated literally, and not removed.

Options in *file* are separated by whitespace. A whitespace character may be included in an option by surrounding the entire option in either single or double quotes. Any character (including a backslash) may be included by prefixing the character to be included with a backslash. The *file* may itself contain additional @*file* options; any such options will be processed recursively.

SEE ALSO

ar (1), **nm** (1), **objdump** (1), **ranlib** (1), **readelf** (1) and the Info entries for *binutils*.

COPYRIGHT