

**NAME**

**xargs** - construct argument list(s) and execute utility

**SYNOPSIS**

**xargs** [-**0**opt] [-**E** *eofstr*] [-**I** *replstr* [-**R** *replacements*] [-**S** *replsize*]] [-**J** *replstr*] [-**L** *number*] [-**n** *number* [-**x**]] [-**P** *maxprocs*] [-**s** *size*] [*utility* [*argument* ...]]

**DESCRIPTION**

The **xargs** utility reads space, tab, newline and end-of-file delimited strings from the standard input and executes *utility* with the strings as arguments.

Any arguments specified on the command line are given to *utility* upon each invocation, followed by some number of the arguments read from the standard input of **xargs**. This is repeated until standard input is exhausted.

Spaces, tabs and newlines may be embedded in arguments using single (‘ ’) or double (‘ ’’) quotes or backslashes (‘\’). Single quotes escape all non-single quote characters, excluding newlines, up to the matching single quote. Double quotes escape all non-double quote characters, excluding newlines, up to the matching double quote. Any single character, including newlines, may be escaped by a backslash.

The options are as follows:

**-0, --null**

Change **xargs** to expect NUL (‘\0’) characters as separators, instead of spaces and newlines. This is expected to be used in concert with the **-print0** function in `find(1)`.

**-E eofstr**

Use *eofstr* as a logical EOF marker.

**-I replstr**

Execute *utility* for each input line, replacing one or more occurrences of *replstr* in up to *replacements* (or 5 if no **-R** flag is specified) arguments to *utility* with the entire line of input. The resulting arguments, after replacement is done, will not be allowed to grow beyond *replsize* (or 255 if no **-S** flag is specified) bytes; this is implemented by concatenating as much of the argument containing *replstr* as possible, to the constructed arguments to *utility*, up to *replsize* bytes. The size limit does not apply to arguments to *utility* which do not contain *replstr*, and furthermore, no replacement will be done on *utility* itself. Implies **-x**.

**-J replstr**

If this option is specified, **xargs** will use the data read from standard input to replace the first occurrence of *replstr* instead of appending that data after all other arguments. This option will not affect how many arguments will be read from input (**-n**), or the size of the command(s) **xargs** will generate (**-s**). The option just moves where those arguments will be placed in the command(s) that are executed. The *replstr* must show up as a distinct *argument* to **xargs**. It will not be recognized if, for instance, it is in the middle of a quoted string. Furthermore, only the first occurrence of the *replstr* will be replaced. For example, the following command will copy the list of files and directories which start with an uppercase letter in the current directory to *destdir*:

```
/bin/ls -ld [A-Z]* | xargs -J % cp -Rp % destdir
```

**-L *number***

Call *utility* for every *number* lines read. If EOF is reached and fewer lines have been read than *number* then *utility* will be called with the available lines.

**-n *number*, --max-args=*number***

Set the maximum number of arguments taken from standard input for each invocation of *utility*. An invocation of *utility* will use less than *number* standard input arguments if the number of bytes accumulated (see the **-s** option) exceeds the specified *size* or there are fewer than *number* arguments remaining for the last invocation of *utility*. The current default value for *number* is 5000.

**-o** Reopen stdin as */dev/tty* in the child process before executing the command. This is useful if you want **xargs** to run an interactive application.

**-P *maxprocs*, --max-procs=*maxprocs***

Parallel mode: run at most *maxprocs* invocations of *utility* at once. If *maxprocs* is set to 0, **xargs** will run as many processes as possible.

**-p, --interactive**

Echo each command to be executed and ask the user whether it should be executed. An affirmative response, 'y' in the POSIX locale, causes the command to be executed, any other response causes it to be skipped. No commands are executed if the process is not attached to a terminal.

**-r, --no-run-if-empty**

Compatibility with GNU **xargs**. The GNU version of **xargs** runs the *utility* argument at least once, even if **xargs** input is empty, and it supports a **-r** option to inhibit this behavior. The FreeBSD version of **xargs** does not run the *utility* argument on empty input, but it supports the **-r**

option for command-line compatibility with GNU **xargs**, but the **-r** option does nothing in the FreeBSD version of **xargs**.

**-R** *replacements*

Specify the maximum number of arguments that **-I** will do replacement in. If *replacements* is negative, the number of arguments in which to replace is unbounded.

**-S** *replsize*

Specify the amount of space (in bytes) that **-I** can use for replacements. The default for *replsize* is 255.

**-s** *size*, **--max-chars=***size*

Set the maximum number of bytes for the command line length provided to *utility*. The sum of the length of the utility name, the arguments passed to *utility* (including NULL terminators) and the current environment will be less than or equal to this number. The current default value for *size* is ARG\_MAX - 4096.

**-t**, **--verbose**

Echo the command to be executed to standard error immediately before it is executed.

**-x**, **--exit**

Force **xargs** to terminate immediately if a command line containing *number* arguments will not fit in the specified (or default) command line length.

If *utility* is omitted, `echo(1)` is used.

Undefined behavior may occur if *utility* reads from the standard input.

If a command line cannot be assembled, or cannot be invoked, or if an invocation of *utility* is terminated by a signal, or an invocation of *utility* exits with a value of 255, the **xargs** utility stops processing input and exits after all invocations of *utility* finish processing.

## EXIT STATUS

The **xargs** utility exits with a value of 0 if no error occurs. If *utility* cannot be found, **xargs** exits with a value of 127, otherwise if *utility* cannot be executed, **xargs** exits with a value of 126. If any other error occurs, **xargs** exits with a value of 1.

## EXAMPLES

Create a 3x3 matrix with numbers from 1 to 9. Every `echo(1)` instance receives three lines as arguments:

```
$ seq 1 9 | xargs -L3 echo
1 2 3
4 5 6
7 8 9
```

Duplicate every line from standard input:

```
$ echo -e "one\ntwo\nthree" | xargs -I % echo % %
one one
two two
three three
```

Execute at most 2 concurrent instances of `find(1)` every one of them using one of the directories from the standard input:

```
echo -e "/usr/ports\n/etc\n/usr/local" | xargs -J % -P2 -n1 find % -name file
```

## SEE ALSO

`echo(1)`, `find(1)`, `execvp(3)`

## STANDARDS

The **xargs** utility is expected to be IEEE Std 1003.2 ("POSIX.2") compliant. The **-J**, **-o**, **-P**, **-R** and **-S** options are non-standard FreeBSD extensions which may not be available on other operating systems.

## HISTORY

The **xargs** utility appeared in PWB UNIX.

## BUGS

If *utility* attempts to invoke another command such that the number of arguments or the size of the environment is increased, it risks `execvp(3)` failing with E2BIG.

The **xargs** utility does not take multibyte characters into account when performing string comparisons for the **-I** and **-J** options, which may lead to incorrect results in some locales.