NAME read – read a line from standard input

SYNOPSIS **/usr/bin/read [-r]** *var* **. . .**

sh **read** *name* . . .

csh **set** *variable* **= $<**

ksh **read** [–**prsu**[*n*]] [*name***?***prompt*] [*name* . . .]

DESCRIPTION

/usr/bin/read

The **read** utility will read a single line from standard input.

By default, unless the –**r** option is specified, backslash (\) acts as an escape character. If standard input is a terminal device and the invoking shell is interactive, **read** will prompt for a continuation line when:

- The shell reads an input line ending with a backslash, unless the –**r** option is specified.
- A here-document is not terminated after a newline character is entered.

The line will be split into fields as in the shell; the first field will be assigned to the first variable *var*, the second field to the second variable *var*, and so forth. If there are fewer *var* operands specified than there are fields, the leftover fields and their intervening separators will be assigned to the last *var*. If there are fewer fields than *var*s, the remaining *var*s will be set to empty strings.

The setting of variables specified by the *var* operands will affect the current shell execution environment. If it is called in a subshell or separate utility execution environment, such as one of the following:

```
(read foo)
nohup read ...
find . -exec read ... \;
```

it will not affect the shell variables in the caller's environment.

The standard input must be a text file.

sh

One line is read from the standard input and, using the internal field separator, **IFS** (normally space or tab), to delimit word boundaries, the first word is assigned to the first *name*, the second word to the second *name*, etc., with leftover words assigned to the last *name*. Lines can be continued using **\newline**. Characters other than **newline** can be quoted by preceding them with a backslash. These backslashes are removed before words are assigned to *names*, and no interpretation is done on the character that follows the backslash. The return code is **0**, unless an EOF is encountered.

csh

The notation

set *variable* **= $<**

loads one line of standard input as the value for *variable*. (See **csh**(1)).

ksh The shell input mechanism. One line is read and is broken up into fields using the characters in **IFS** as separators. The escape character, **(\)**, is used to remove any special meaning for the next character and for line continuation. In raw mode, **−r,** the \ character is not treated specially. The first field is assigned to the first *name*, the second field to the second *name*, etc., with leftover fields assigned to the last *name*. The **−p** option causes the input line to be taken from the input pipe of a process spawned by the shell using **|&**. If the **−s** flag is present, the input will be saved as a command in the history file. The flag **−u** can be used to specify a one digit file descriptor unit *n* to read from. The file descriptor can be opened with the **exec** special command. The default value of *n* is **0**. If *name* is omitted then **REPLY** is used as the default *name*. The exit status is **0** unless the input file is not open for reading or an end-of-file is encountered. An end-of-file with the **−p** option causes cleanup for this process so that another can be spawned. If the first argument contains a **?**, the remainder of this word is used as a *prompt* on standard error when the shell is interactive. The exit status is **0** unless an end-of-file is encountered.

OPTIONS

The following option is supported:

−r Do not treat a backslash character in any special way. Consider each backslash to be part of the input line.

OPERANDS

The following operand is supported:

var The name of an existing or non-existing shell variable.

EXAMPLES

The following example for **/usr/bin/read** prints a file with the first field of each line moved to the end of the line.

```
while read -r xx yy
do
        printf "%s %s\n" "$yy" "$xx"
done < input_file
```

ENVIRONMENT

See **environ**(5) for descriptions of the following environment variables that affect the execution of **read**: **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**.

IFS Determine the internal field separators used to delimit fields.

PS2 Provide the prompt string that an interactive shell will write to standard error when a line ending with a backslash is read and the **−r** option was not specified, or if a here-document is not terminated after a newline character is entered.

EXIT STATUS

The following exit values are returned:

0 Successful completion.

>0 End-of-file was detected or an error occurred.

SEE ALSO

csh(1), **ksh**(1), **line**(1), **set**(1), **sh**(1), **environ**(5)