NAME

if - conditional command

SYNOPSIS

if [*expression* [*command* [*arg* ...]]]

DESCRIPTION

If evaluates the specified *expression*, and if its value is true, returns a zero exit status or executes the specified *command* with the given arguments. Otherwise, it returns a non-zero exit status. When *expression* is not specified, **if** also returns a non-zero exit status.

The following primaries are used to construct *expression*:

{ *command* [*arg* ...] **}**
The specified *command* is executed with the given arguments to obtain its exit status. A zero status is *true*; a non-zero status is *false*.

-d *file* True if *file* exists and is a directory.

-e *file* True if *file* exists.

-f *file* True if *file* exists and is a regular file.

-h *file* True if *file* exists and is a symbolic link.

-n *string* True if the length of *string* is non-zero.

-r *file* True if *file* exists and is readable.

-s *file* True if *file* exists and has a size greater than zero bytes.

-t *fildes* True if the file whose file descriptor number is *fildes* is open and associated with a terminal device. *Fildes* must be a decimal digit (0 - 9).

-w *file* True if *file* exists and is writable.

-x *file* True if *file* exists and is executable, or if *file* is a searchable directory.

-z *string* True if the length of *string* is zero.

file1 **-ef** *file2*
: True if *file1* and *file2* both exist and refer to the same file (same device, same inode).

file1 **-nt** *file2*
: True if *file1* and *file2* both exist and last data-modification time of *file1* is newer than that of *file2*.

file1 **-ot** *file2*
: True if *file1* and *file2* both exist and last data-modification time of *file1* is older than that of *file2*.

s1 **!=** *s2* True if the strings *s1* and *s2* are not equal.

s1 == *s2* Is a synonym for =.

s1 = *s2* True if the strings *s1* and *s2* are equal.

s1 < *s2* True if the string *s1* comes before *s2* according to their ASCII character values.

s1 > *s2* True if the string *s1* comes after *s2* according to their ASCII character values.

n1 **-eq** *n2* True if the integers *n1* and *n2* are algebraically equal.

n1 **-ne** *n2* True if the integers *n1* and *n2* are not algebraically equal.

n1 **-gt** *n2* True if the integer *n1* is algebraically greater than the integer *n2*.

n1 **-ge** *n2* True if the integer *n1* is algebraically greater than or equal to the integer *n2*.

n1 **-lt** *n2* True if the integer *n1* is algebraically less than the integer *n2*.

n1 **-le** *n2* True if the integer *n1* is algebraically less than or equal to the integer *n2*.

These primaries may also be combined with the following operators:

! *expression*
: unary *negation* operator

expression1 **-a** *expression2*
: binary *and* operator

expression1 **-o** *expression2*
binary *or* operator

(*expression* **)**
parentheses for grouping

-a has higher precedence than **-o**. Notice that all of the operators and flags are separate arguments to **if**. Notice also that parentheses are meaningful to the shell and must be escaped.

Symbolic links are followed for all *file*-related primaries except **-h**.

EXIT STATUS

The **if** command exits with one of the following values:

0 The expression was true (see below).

1 The expression was false or was not specified.

2 An error was detected.

125 The specified command was found but did not begin with the proper magic number or a '#!shell' sequence, and a valid shell was not specified by EXECSHELL with which to execute it.

126 The specified command was found but could not be executed.

127 The specified command was not found.

If the expression is true and if *command* is specified and executed, the exit status is that of the executed *command*.

ENVIRONMENT

Notice that the concept of 'user environment' was not defined in Sixth Edition (V6) UNIX. Thus, use of the following environment variables by this port of the conditional command is an enhancement:

EXECSHELL
If set to a non-empty string, the value of this variable is taken as the path name of the shell which is invoked to execute the specified command when it does not begin with the proper magic number or a '#!shell' sequence.

PATH

If set to a non-empty string, the value of this variable is taken as the sequence of directories which is used to search for the specified command. Notice that the conditional command from Sixth Edition (V6) UNIX always used the equivalent of '.:/bin:/usr/bin', not PATH.

SEE ALSO

goto(1), etsh(1), tsh(1), test(1)

Etsh home page: https://etsh.io/

COMPATIBILITY

The **if** command from Sixth Edition (V6) UNIX does not support the ==, <, >, **-d**, **-e**, **-f**, **-h**, **-n**, **-s**, **-t**, **-x**, **-z**, **-ef**, **-nt**, **-ot**, **-eq**, **-ne**, **-gt**, **-ge**, **-lt**, and **-le** operators.

In addition to supporting the above operators, this port also differs from the original in that the exit status returned varies according to whether the expression is true or false, as is the case with *test*(1).

HISTORY

An **if** command appeared as */bin/if* in Third Edition UNIX.

AUTHORS

This port of the **if** command is derived from Sixth Edition (V6) UNIX /usr/source/s1/if.c. Presumably, Ken Thompson of Bell Labs wrote it. Jeffrey Allen Neitzel <*jan@etsh.io*> ported and currently maintains it as *if*(1).

LICENSE

See either the LICENSE file which is distributed with **etsh** or https://etsh.io/license/ for full details.