NAME

test – condition evaluation command

SYNOPSIS

test *expression*

[*expression*]

DESCRIPTION

test evaluates the expression *expression* and if its value is true, sets a **0** (**TRUE**) exit status; otherwise, a non-zero (**FALSE**) exit status is set; **test** also sets a non-zero exit status if there are no arguments. When permissions are tested, the effective user ID of the process is used.

All operators, flags, and brackets (brackets used as shown in the second **SYNOPSIS** line) must be separate arguments to **test**. Normally these items are separated by spaces.

USAGE

Primitives

The following primitives are used to construct *expression*:

–**r** *filename*	True if *filename* exists and is readable.
–**w** *filename*	True if *filename* exists and is writable.
–**x** *filename*	True if *filename* exists and is executable.
–**f** *filename*	True if *filename* exists and is a regular file.
–**d** *filename*	True if *filename* exists and is a directory.
–**c** *filename*	True if *filename* exists and is a character special file.
–**b** *filename*	True if *filename* exists and is a block special file.
–**p** *filename*	True if *filename* exists and is a named pipe (FIFO).
–**u** *filename*	True if *filename* exists and its set-user-ID bit is set.
–**g** *filename*	True if *filename* exists and its set-group-ID bit is set.
–**k** *filename*	True if *filename* exists and its sticky bit is set.
–**s** *filename*	True if *filename* exists and has a size greater than **0**.
–**t** [*fildes*]	True if the open file whose file descriptor number is *fildes* (**1** by default) is associated with a terminal device.
–**z** *s1*	True if the length of string *s1* is **0.**
–**n** *s1*	True if the length of the string *s1* is non-zero.
s1 **=** *s2*	True if strings *s1* and *s2* are identical.
s1 **!=** *s2*	True if strings *s1* and *s2* are *not* identical.
s1	True if *s1* is *not* the null string.

n1 −**eq** *n2* True if the integers *n1* and *n2* are algebraically equal. Any of the comparisons −**ne**, −**gt**, −**ge**, −**lt**, and −**le** may be used in place of −**eq**.

Operators

These primaries may be combined with the following operators:

! Unary negation operator.

−**a** Binary *and* operator.

−**o** Binary *or* operator (−**a** has higher precedence than −**o**).

\`( *expression* )\` Parentheses for grouping. Notice also that parentheses are meaningful to the shell and, therefore, must be quoted.

SEE ALSO

find(1), **sh**(1)

NOTES

If you test a file you own (the **-r ,** **-w ,** or **-x** tests), but the permission tested does not have the *owner* bit set, a non-zero (false) exit status will be returned even though the file may have the *group* or *other* bit set for that permission. The correct exit status will be set if you are super-user.

The **=** and **!=** operators have a higher precedence than the −**r** through −**n** operators, and **=** and **!=** always expect arguments; therefore, **=** and **!=** cannot be used with the −**r** through −**n** operators.

If more than one argument follows the −**r** through −**n** operators, only the first argument is examined; the others are ignored, unless a −**a** or a −**o** is the second argument.