NAME

if − conditional command

SYNOPSIS

```
if expr command [ arg ... ]

if expr then
        command(s)
        ...
[ else [ command ]
        ... ]
endif

test expr
```

DESCRIPTION

If evaluates the expression *expr.* In the first form above, if *expr* is true, the given *command* is executed with the given arguments. The command may be another*if* .

In the second form, if *expr* is true, the commands between the *then* and the next unmatched *else* or *endif* are executed. If*e xpr* is false, the commands after *then* are skipped, and the commands after the optional *else* are executed. Zero or one commands may be written on the same line as the*else* . In particular, *if* may be used this way. The pseudo commands*else* and *endif* (whiche ver occurs first) must not be hidden behind semicolons or other commands. This form may be nested: every *then* needs a matching *endif.*

Test is an entry to *if* that evaluates the expression and returns exit code 0 if it is true, and code 1 if it is false or in error.

The following primitives are used to construct the *expr:*

−r file true if the file exists and is readable.

−w file true if the file exists and is writable.

−s file true if the file exists and has a size greater than zero.

−f file true if the file exists and is an ordinary file.

−d file true if the file exists and is a directory.

−z s1 true if the length of string *s1* is zero.

−n s1 true if the length of string *s1* is nonzero.

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

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

n1 **−eq** n2
n1 **−ne** n2
n1 **−gt** n2
n1 **−ge** n2
n1 **−lt** n2
n1 **−le** n2 true if the stated algebraic relationship exists. The arguments *n1* and *n2* must be integers.

{ command **}** The bracketed command is executed to obtain the exit status. Status zero is considered *true.* The command must **not** be another *if.*

These primaries may be combined with the following operators:

! unary negation operator

−a binary *and* operator

−o binary *or* operator

(expr **)** parentheses for grouping.

−a has higher precedence than **−o.** Notice that all the operators and flags are separate arguments to *if* and hence must be surrounded by spaces. Notice also that parentheses are meaningful to the Shell and must be escaped.

EXIT CODES

0 − true expression, no error.
1 − false condition or error.

SEE ALSO

exit(I), goto(I), sh(I), switch(I), while(I), exit(II)

DIAGNOSTICS

if:missing endif
if:syntax error: value
if:non-numeric arg: value
if:no command: name
else:missing endif

Test may issue any of the *if* messages above, except the first.

BUGS

In general, *if, else, endif,* and *test* must not be hidden behind semicolons on a command line. Many of the effects are obtained by searching the input file and adjusting the read pointer appropriately. Thus, including any of these commands in a part of the file intended to be read by a command other than the shell may cause strange results if they are encountered while searching.
These commands ignore redirection or piping of their standard input or output. Commands executed by *if* or *test* may be affected by redirections, but this practice is undesirable and should be avoided.