NAME

case, switch, select – shell built-in functions to choose from among a list of actions

SYNOPSIS

sh

case *word* **in** [*pattern* [| *pattern*] **)** *actions* **;;**] . . . **esac**

csh

switch (*expression***)**
    case *comparison1***:**
        actions
        breaksw
    case *comparison2***:**
        actions
        breaksw
    . . .
    default:
endsw

ksh

case *word* **in** [*pattern* [| *pattern*] **)** *actions* **;;**] . . . **esac**
select *identifier* [**in** *word* . . .] **; do** *list* **; done**

DESCRIPTION

sh

A **case** command executes the *actions* associated with the first *pattern* that matches *word*. The form of the patterns is the same as that used for file-name generation except that a slash, a leading dot, or a dot immediately following a slash need not be matched explicitly.

csh

The c-shell uses the **switch** statement, in which each *comparison* is successively matched, against the specified *expression*, which is first command and filename expanded. The file metacharacters *, **?** and **[**. . .**]** may be used in the case comparison, which are variable expanded. If none of the comparisons match before a “default” comparison is found, execution begins after the default comparison. Each **case** statement and the **default** statement must appear at the beginning of a line. The command **breaksw** continues execution after the **endsw**. Otherwise control falls through subsequent **case** and **default** statements as with C. If no comparison matches and there is no default, execution continues after the **endsw**.

case *comparison***:** A compared-expression in a **switch** statement.

default: If none of the preceeding *comparisons* match *expression*, then this is the default case in a **switch** statement. The default should come after all **case** comparisons. Any remaining commands on the command line are first executed.

breaksw exits from a **switch**, resuming after the **endsw**.

ksh

A **case** command executes the *actions* associated with the first *pattern* that matches *word*. The form of the patterns is the same as that used for file-name generation (see File Name Generation in **ksh**(1)).

A **select** command prints to standard error (file descriptor 2), the set of *word*s, each preceded by a number. If **in** *word* . . . is omitted, then the positional parameters are used instead. The **PS3** prompt is printed and a line is read from the standard input. If this line consists of the number of one of the listed *word*s, then the value of the variable *identifier* is set to the *word* corresponding to this number. If this line is empty the selection list is printed again. Otherwise the value of the variable *identifier* is set to **NULL**. The contents of the line read from standard input is saved in the shell variable **REPLY**. The *list* is executed for each selection until a **break** or *end-of-file* is encountered. If the **REPLY** variable is set to **NULL** by the execution of *list*, then the selection list is printed before displaying the **PS3** prompt for the next selection.

EXAMPLES

sh

```
STOPLIGHT=green
case $STOPLIGHT in
        red)            echo "STOP" ;;
        orange)         echo "Go with caution; prepare to stop" ;;
        green)          echo "you may GO" ;;
        blue | brown)   echo "invalid stoplight colors" ;;
esac
```

csh

In the C-shell, you must add NEWLINE characters as below.

```
set STOPLIGHT = green
switch ($STOPLIGHT)
        case red:
                        echo "STOP"
                        breaksw
        case orange:
                        echo "Go with caution; prepare to stop"
                        breaksw
        case green:
                        echo "you may GO"
                        endsw
endsw
```

ksh

```
STOPLIGHT=green
case $STOPLIGHT in
        red)            echo "STOP" ;;
        orange)         echo "Go with caution; prepare to stop" ;;
        green)          echo "you may GO" ;;
        blue | brown)   echo "invalid stoplight colors" ;;
esac
```

SEE ALSO

break(1), **csh**(1), **ksh**(1), **sh**(1)