NAME

while − shell iteration command

SYNOPSIS

while expr
commands... (may include *break* or *continue*)
end

DESCRIPTION

While evaluates the expression *expr,* which is similar to (and a superset of) the expression described in *if(I).* If the expression is true, *while* does nothing, permitting the command(s) on following lines to be read and executed by the Shell. If the expression is false, the input file is effectively searched for the matching *end* command, and the Shell resumes execution of the command(s) on the line following the *end.* The *while-end* grouping may be nested up to three levels deep.

In addition to the type of expression permitted by *if, while* treats a single, nonnull argument as a true expression, and treats a single null argument or lack of arguments as a false expression.

The *break* command terminates the nearest enclosing *while-end* group, causing execution to resume after the nearest succeeding unmatched *end.* Exit from *n* levels is obtained by writing *n break* commands on the same line.

The *continue* command causes execution to resume at a preceding *while,* i.e., the *while* that begins the smallest loop containing the *continue.*

A common loop is that of processing arguments one at a time: see *shift(I).*

The following is a shell procedure that is also a filter. It reads a line at a time from the standard input that existed when the procedure was invoked, exiting on end-of-file.

```
while 1
        = a <-- || exit
        commands using $a ...
end
```

SEE ALSO

goto(I), if(I), onintr(I), sh(I), shift(I), switch(I)

DIAGNOSTICS

while: missing end
while: >3 levels
while: syntax errors like those of if.
break: missing end
break: used outside loop
continue: used outside loop
end: used outside loop

BUGS

A *goto* may be used to terminate one or more *while-end* groupings. Those who use it to branch into a loop will receive appropriately peculiar results. When an interrupt is caught and transfer to a label caused by use of *onintr(I),* all currently effective *while-end* loops are cancelled, i.e., the *onintr* performs a *goto* that breaks all loops. Neither*while* nor *end* may be hidden behind semicolons or used within other commands.