NAME

apply - apply a command to a set of arguments

SYNOPSIS

apply [**-a** *c*] [**-d**] [-*#*] *command argument* ...

DESCRIPTION

The **apply** utility runs the named *command* on each argument *argument* in turn.

Character sequences of the form "%d" in *command*, where ‘d’ is a digit from 1 to 9, are replaced by the d’th following unused *argument*. In this case, the largest digit number of arguments are discarded for each execution of *command*.

The options are as follows:

-*#* Normally arguments are taken singly; the optional number # specifies the number of arguments to be passed to *command*. If the number is zero, *command* is run, without arguments, once for each *argument*.

If any sequences of "%d" occur in *command*, the **-#** option is ignored.

-a *c* The use of the character ‘%’ as a magic character may be changed with the **-a** option.

-d Display the commands that would have been executed, but do not actually execute them.

ENVIRONMENT

The following environment variable affects the execution of **apply**:

SHELL Pathname of shell to use. If this variable is not defined, the Bourne shell is used.

FILES

/bin/sh default shell

EXAMPLES

```
apply echo *
          is similar to ls(1);
apply -2 cmp a1 b1 a2 b2 a3 b3
          compares the ‘a’ files to the ‘b’ files;
apply -0 who 1 2 3 4 5
          runs who(1) 5 times; and
```

```
apply ’ln %1 /usr/joe’ *
```

links all files in the current directory to the directory */usr/joe*.

HISTORY

The **apply** command appeared in 4.2BSD.

AUTHORS

Rob Pike

BUGS

Shell metacharacters in *command* may have bizarre effects; it is best to enclose complicated commands in single quotes (’’).

The **apply** utility does not recognize multibyte characters.