

NAME

find – find files

SYNOPSIS**find** pathname expression**DESCRIPTION**

Find recursively descends the directory hierarchy from *pathname* seeking files that match a boolean *expression* written in the primaries given below. In the descriptions, the argument *n* is used as a decimal integer where *+n* means more than *n*, *-n* means less than *n* and *n* means exactly *n*.

- name** filename True if the *filename* argument matches the current file name. Normal *Shell* argument syntax may be used if escaped (watch out for '[', '?' and '*').
- perm** onum True if the file permission flags exactly match the octal number *onum* (see *chmod*(I)). If *onum* is prefixed by a minus sign, more flag bits (017777, see *stat*(II)) become significant and the flags are compared: *(flags&onum)==onum*.
- type** *c* True if the type of the file is *c*, where *c* is **b**, **c**, **d** or **f** for block special file, character special file, directory or plain file.
- links** *n* True if the file has *n* links.
- user** *uname* True if the file belongs to the user *uname*.
- group** *gname* As it is for **-user** so shall it be for **-group** (someday).
- size** *n* True if the file is *n* blocks long (512 bytes per block).
- atime** *n* True if the file has been accessed in *n* days.
- mtime** *n* True if the file has been modified in *n* days.
- exec** command True if the executed command returns exit status zero (most commands do). The end of the command is punctuated by an escaped semicolon. A command argument '{ }' is replaced by the current pathname.
- ok** command Like **-exec** except that the generated command line is printed with a question mark first, and is executed only if the user responds **y**.
- print** Always true; causes the current pathname to be printed.

The primaries may be combined with these operators (ordered by precedence):

- !** prefix *not*
- a** infix *and*, second operand evaluated only if first is true
- o** infix *or*, second operand evaluated only if first is false
- (expression) parentheses for grouping. (Must be escaped.)

To remove files named 'a.out' and '*.o' not accessed for a week:

```
find / "(" -name a.out -o -name "*.o" ")" -a -atime +7 -a -exec rm { } ";"
```

FILES

/etc/passwd

SEE ALSO

sh (I), if(I), file system (V)

BUGS

There is no way to check device type.
Syntax should be reconciled with *if*.