## NAME

pexec − path search and execute a file

## SYNOPSIS

**char pathstr[128];**
**char shellnam[16];**
**pexec(name, argv)**
**char *name, *argv[];**
**pexinit( )**

## DESCRIPTION

*Pexec* provides an interface to the *execv* function that duplicates the shell's actions in searching for an executable file in a list of directories, as specified in the user's '.path' file.

*Pexinit* investigates the external arrays *pathstr* and *shellnam.* If either array is non-null, it leaves that array alone. If*pathstr* is null, it attempts to open the user' s '.path' file and place the first line found there into *pathstr,* to be used later as a list of directories to be searched. If '.path' cannot be opened, it uses:

| | |
|---|---|
| /bin:/etc:/ | for super-user |
| :/bin:/usr/bin | for anyone else |

If a second line is found in the '.path' file, it is taken as the name of the shell to be executed to interpret a shell procedure. If none is found, '/bin/sh' is used. *Pexinit* returns 0 to show successful completion, guaranteeing both arrays filled, and −1 otherwise.

*Pexec* first calls *pexinit,* then searches for the named file and executes it. The existence of two functions permits *pexinit* to be called once, followed by many *fork/pexec* pairs.

This function is kept in the **−lPW** library.

## SEE ALSO

sh(I), exec(II), fork(II)

## DIAGNOSTICS

Items in parentheses refer to error names in *intro*(II).
''cannot read .path''
''.path too long'' (more than 128+16 = 144 bytes long)
''No shell!'' (real trouble, cannot execute shell)
''too large'' (ENOMEM)
''arg list too long'' (E2BIG)
''file not executable'' (EACCES, no x bits set in file mode)
''not found'' (name could not be found at all)
''text busy'' (ETXTBSY, should be very rare)

## BUGS

A pathname generated by the search mechanism may not exceed 47 characters in length.