

NAME

`exprog` — perform standard Shell execute sequence

SYNOPSIS

```
exprog(argv)  
char *argv[];
```

DESCRIPTION

Exprog has been replaced by *execvp*(2) in the newer versions of the compiler. *Exprog* attempts to locate the file specified by *argv*[0] in the current directory. *Argv* should be an argument string in the format required by *execv* (see *exec*(2)). If the file does not exist, *exprog* prepends */bin/* to *argv*[0] and tries again. Upon failure it further prepends */usr* and makes one last attempt before returning with an error indication.

If the file is executable but the attempt to execute it fails (see *exec*(2) for reasons for failure) *exprog* passes the file to the shell for interpretation as a command file.

In all cases all arguments given to *exprog* in the argument vector are passed to the program or shell.

DIAGNOSTICS

A -1 is returned if there is no UNIX Shell. Otherwise if *exprog* returns, it returns the global system error number (*errno*) which describes why the execute was unsuccessful.

BUGS

Exprog uses the default command look-up strategy employed by the shell; however, if you have specified an alternate look-up sequence, *exprog* will continue to use the default strategy. See *sh*(1) for details on the shell look-up. Only 100 arguments may be passed to the shell by *exprog*, a generous but unnecessary restriction.