NAME

popen, **pclose** - process I/O

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <stdio.h>

*FILE **
popen(*const char *command, const char *type*);

int
pclose(*FILE *stream*);

DESCRIPTION

The **popen**() function "opens" a process by creating a bidirectional pipe forking, and invoking the shell. Any streams opened by previous **popen**() calls in the parent process are closed in the new child process. Historically, **popen**() was implemented with a unidirectional pipe; hence many implementations of **popen**() only allow the *type* argument to specify reading or writing, not both. Since **popen**() is now implemented using a bidirectional pipe, the *type* argument may request a bidirectional data flow. The *type* argument is a pointer to a null-terminated string which must be ‘r’ for reading, ‘w’ for writing, or ‘r+’ for reading and writing.

A letter ‘e’ may be appended to that to request that the underlying file descriptor be set close-on-exec.

The *command* argument is a pointer to a null-terminated string containing a shell command line. This command is passed to */bin/sh* using the **-c** flag; interpretation, if any, is performed by the shell.

The return value from **popen**() is a normal standard I/O stream in all respects save that it must be closed with **pclose**() rather than **fclose**(). Writing to such a stream writes to the standard input of the command; the command’s standard output is the same as that of the process that called **popen**(), unless this is altered by the command itself. Conversely, reading from a "popened" stream reads the command’s standard output, and the command’s standard input is the same as that of the process that called **popen**().

Note that output **popen**() streams are fully buffered by default.

The **pclose**() function waits for the associated process to terminate and returns the exit status of the command as returned by wait4(2).

RETURN VALUES

The **popen**() function returns NULL if the fork(2) or pipe(2) calls fail, or if it cannot allocate memory.

The **pclose**() function returns -1 if *stream* is not associated with a "popened" command, if *stream* already "pclosed", or if wait4(2) returns an error.

ERRORS

The **popen**() function does not reliably set *errno*.

SEE ALSO

sh(1), fork(2), pipe(2), wait4(2), fclose(3), fflush(3), fopen(3), stdio(3), system(3)

HISTORY

A **popen**() and a **pclose**() function appeared in Version 7 AT&T UNIX.

Bidirectional functionality was added in FreeBSD 2.2.6.

BUGS

Since the standard input of a command opened for reading shares its seek offset with the process that called **popen**(), if the original process has done a buffered read, the command's input position may not be as expected. Similarly, the output from a command opened for writing may become intermingled with that of the original process. The latter can be avoided by calling fflush(3) before **popen**().

Failure to execute the shell is indistinguishable from the shell's failure to execute command, or an immediate exit of the command. The only hint is an exit status of 127.

The **popen**() function always calls sh(1), never calls csh(1).