NAME

p2open, p2close – open, close pipes to and from a command

SYNOPSIS

cc [*flag* ...] *file* ... **–lgen** [*library* ...]

#include <libgen.h>

int p2open(const char **cmd*, **FILE** **fp*[2]);

int p2close(FILE **fp*[2]);

MT-LEVEL

Unsafe

DESCRIPTION

p2open( ) forks and execs a shell running the command line pointed to by *cmd*. On return, **fp[0]** points to a **FILE** pointer to write the command's standard input and **fp[1]** points to a **FILE** pointer to read from the command's standard output. In this way the program has control over the input and output of the command.

The function returns **0** if successful; otherwise, it returns **–1**.

p2close( ) is used to close the file pointers that **p2open( )** opened. It waits for the process to terminate and returns the process status. It returns **0** if successful; otherwise, it returns **–1**.

RETURN VALUES

A common problem is having too few file descriptors. **p2close( )** returns **–1** if the two file pointers are not from the same **p2open( )**.

EXAMPLES

```
#include <stdio.h>
#include <libgen.h>

main(argc,argv)
int argc;
char **argv;
{
	FILE *fp[2];
	pid_t pid;
	char buf[16];

	pid=p2open("/usr/bin/cat", fp);
	if ( pid == –1 ) {
		fprintf(stderr, "p2open failed\n");
		exit(1);
	}
	write(fileno(fp[0]),"This is a test\n", 16);
	if(read(fileno(fp[1]), buf, 16) <=0)
		fprintf(stderr, "p2open failed\n");
	else
		write(1, buf, 16);
	(void)p2close(fp);
}
```

SEE ALSO

fclose(3S), **popen**(3S), **setbuf**(3S)

NOTES

Buffered writes on **fp[0]** can make it appear that the command is not listening. Judiciously placed **fflush( )** calls or unbuffering **fp[0]** can be a big help; see **fclose**(3S).

Many commands use buffered output when connected to a pipe. That, too, can make it appear as if things are not working.

Usage is not the same as for **popen( )**, although it is closely related.