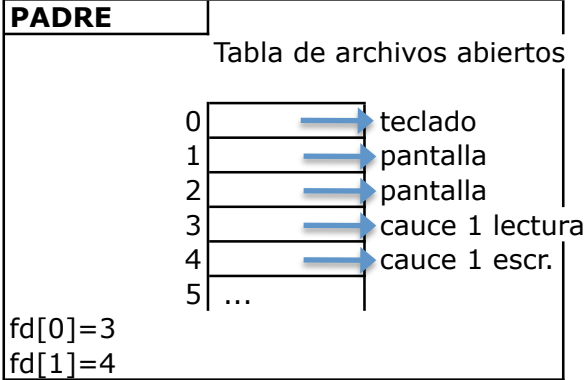


1	/*	tarea7nueva.c
2		
3	Programa ilustrativo del uso de pipes y la redireccion de entrada y	
4	salida estándar: "mils rutadirectorio   wc" */	
5	#include<sys/types.h>	
6	#include<fcntl.h>	
7	#include<unistd.h>	
8	#include<stdio.h>	
9	#include<stdlib.h>	
10	#include<errno.h>	
11	int main(int argc, char *argv[])	
12	{	
13	int fd[2];	
14	pid_t PID;	
15	pipe(fd); // Llamada al sistema para crear un pipe	
16	if ( (PID= fork())<0) {	
17	perror("fork");	
18	exit(1);	
19	}	
20	if(PID == 0) { // ----- hijo	
21	close(fd[0]); //cierro lo que no uso	
22		
23	<b>dup2 ( fd[1] , STDOUT_FILENO);</b>	
24	execlp("./mils","mils",argv[1], NULL);	
25	}	
26	else {	
27	close(fd[1]); //cierro lo que no uso	
28		
29	<b>dup2 ( fd[0] , STDIN_FILENO );</b>	
30	execlp("wc","wc",NULL);	
31	}	
32	return(0);	
33	}	

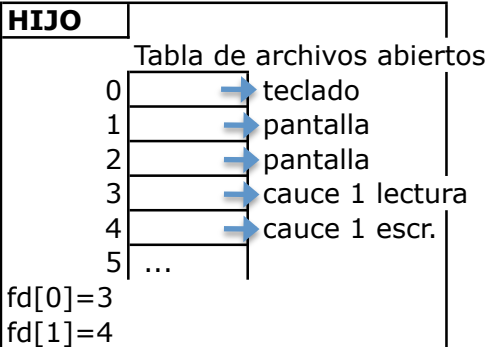
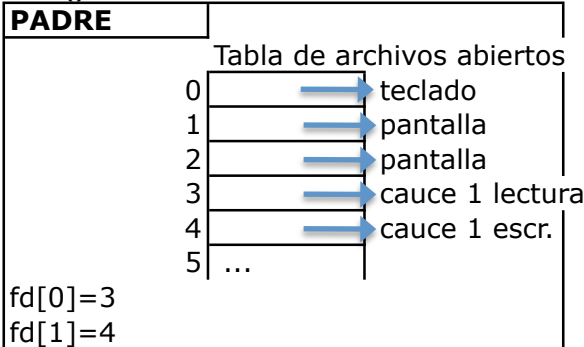
#### Archivo mils.c:

```
// programa que muestra los nombres de los hijos dle directorio dado como argumento
int main( int argc, char *argv[] )
{
    char mensajeerror[]="Debe dar 1 argumento: ruta de un directorio\n";
    DIR *dir;
    struct dirent *mi_dirent;
    if( argc != 2 )
    {
        write(STDERR_FILENO, mensajeerror, strlen(mensajeerror));
        exit(EXIT_FAILURE);
    }
    if( (dir = opendir( argv[1] )) == NULL )
    {
        perror( "opendir" );
        exit( -1 );
    }
    while( (mi_dirent = readdir( dir )) != NULL ){
        if (strcmp(mi_dirent->d_name,".")==0) continue;
        if (strcmp(mi_dirent->d_name,"..")==0) continue;
        printf( "%s\n", mi_dirent->d_name);/// escribe en 1 (salida estandar)
    }
    closedir( dir );
    exit(EXIT_SUCCESS);
}
```

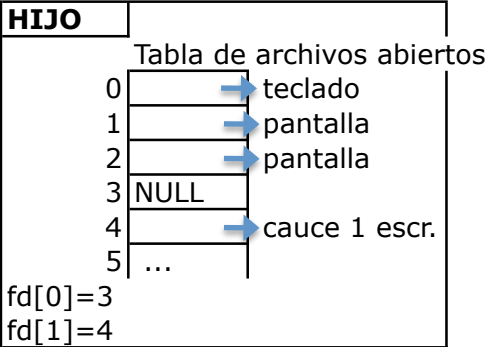
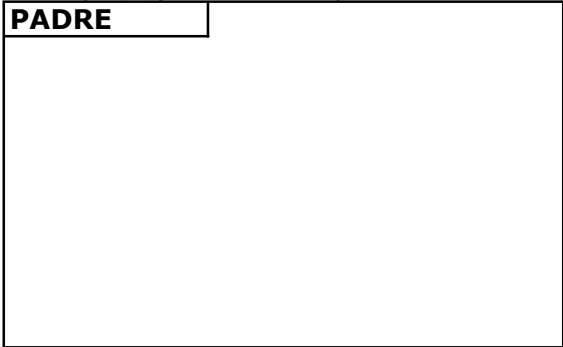
Tras 15: pipe (fd);



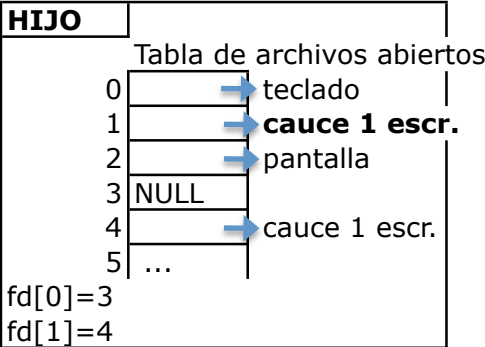
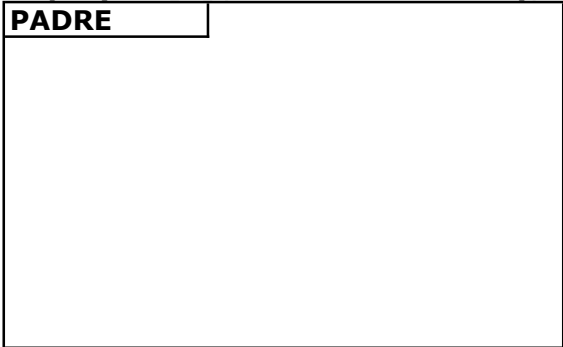
Tras 16: fork()



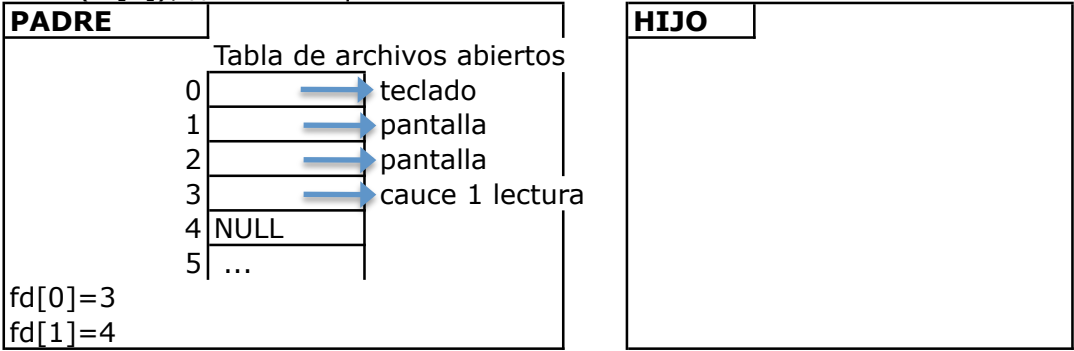
Tras 21: close(fd[0]); //cierro lo que no uso



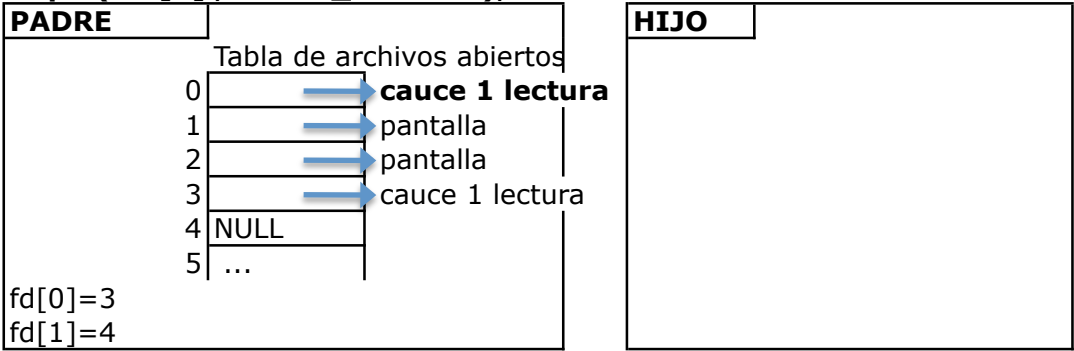
Tras 23: dup2 ( fd[1] , STDOUT\_FILENO );



Tras 27: `close(fd[1]); //cierro lo que no uso`



Tras 29: `dup2 ( fd[0] , STDIN_FILENO );`



COMO RESULTADO DE LO ANTERIOR:

- \* Si en el hijo se escribe en 1, va a ir realmente al cauce
- \* Si en el padre se lee de 0, se lee realmente del cauce

Con ello hemos conseguido redirigir la salida estandar del hijo, y la entrada estandar del padre.  
Sin cambiar el código de los programas que se lanzan con `exec` conseguimos que escriban "en otro sitio" o lean "en otro sitio"