

2. Ejemplo PIPE

Una vez que sabemos cómo leer y escribir en ficheros, veamos algunos ejemplos usando *pipes*.

En el primer ejemplo se crea un proceso hijo con *fork()*. El proceso hijo envía al proceso padre mediante el uso de *pipes* el mensaje "Hola papi" en el descriptor para escritura *fd[1]*, el proceso padre mediante el descriptor *fd[0]* lee los datos enviados por el hijo:

```
#include <stdio .h>  
#include <stdlib .h>  
#include <unistd.h>
```

```
int main ()  
{  
    int fd[2];  
    char buffer[30];  
    pid_t pid;
```

```
pipe(fd); //se crea el pipe
```

```
pid=fork(); //se crea el proceso hijo
```

```
switch (pid) {
```

```
    case -1: //ERROR
```

```
        printf("NO SE HA PODIDO CREAR HIJO ... ");
```

```
        exit(-1);
```

```
        break;
```

case 0 : //HIJO

printf("El HIJO escribe en el pipe ... \n ");

write(fd[l], "Hola papi", 10);

break;

```
default : //PADRE
```

```
    wait(NULL); //espera que finalice proceso hijo
```

```
    printf("El PADRE lee del pipe ... \n");
```

```
    read(fd[0], buffer, 10);
```

```
    printf("\tMensaje leído: %s\n",buffer);
```

```
    break;
```

```
}
```

```
}
```

La compilación y ejecución muestra la siguiente salida:

```
administrador@ubuntu:~$ gcc ejemploPipel.c  
-o ejemploPipel
```

```
administrador@ubuntu:~$ ./ejemploPipel
```


El HIJO escribe en el pipe ...

El PADRE lee del pipe ...

Mensaje leído: Hola papi

administrador@ubuntu:~\$

Primero se crea la tubería con *pipe()* y a continuación el proceso hijo. Recordemos que cuando se crea un proceso hijo con *fork()*, recibe una copia de todos los descriptores de ficheros del proceso padre, incluyendo copia de los descriptores de ficheros del pipe (*fd[0]* y *fd[1]*).

Esto permite que el proceso hijo mande datos al extremo de escritura del pipe *fd[1]*, y el padre los reciba del extremo de lectura *fd[0]*.

