

“Con mis maestros he aprendido mucho, con mis colegas, más, con mis alumnos todavía más.” Proverbio hindú

*“Nuestro más elevado propósito ha de ser formar seres humanos libres, capaces de impartir desde sí mismos, propósito y dirección a sus vidas” (R. Steiner).*

### **Ejercicio de pipe**

- Realizar un CHAT entre procesos emparentados usando pipes.

### **Objetivos**

- Aprender características básicas de la programación concurrente.
- Aprender los mecanismos que usa el Sistema Operativo para la creación de procesos.
- Aprender el uso del recurso pipe para la comunicación entre procesos emparentados.

### **Planteo del ejercicio**

Crear una estructura de procesos emparentados de cinco procesos, un proceso padre creará dos procesos hijos y cada proceso hijo creará un proceso nieto. La creación de procesos se realiza con la llamada al sistema fork.

El chat se realizará entre los procesos hijos que serán los escritores y cada proceso nieto será el lector de su correspondiente escritor.

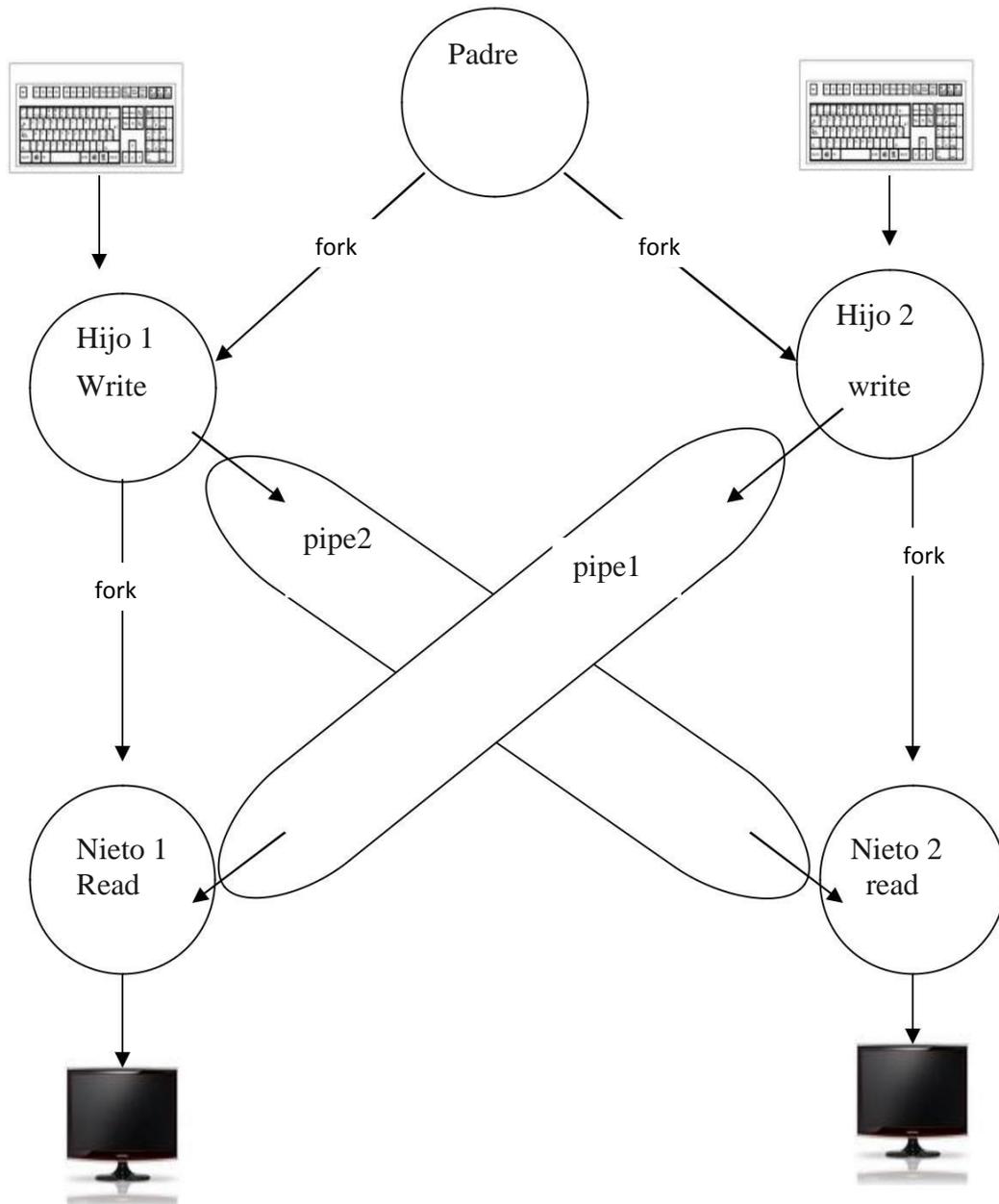
Se decide usar dos pipes como recursos de comunicación entre procesos para realizar una programación mas simple y enfocar el problema al concepto.

Se recomienda que el alumno realice el mismo ejercicio utilizando solo un pipe.

“Con mis maestros he aprendido mucho, con mis colegas, más, con mis alumnos todavía más.” Proverbio hindú

*“Nuestro más elevado propósito ha de ser formar seres humanos libres, capaces de impartir desde sí mismos, propósito y dirección a sus vidas” (R. Steiner).*

### Diagrama del Planteo



“Con mis maestros he aprendido mucho, con mis colegas, más, con mis alumnos todavía más.” Proverbio hindú

*“Nuestro más elevado propósito ha de ser formar seres humanos libres, capaces de impartir desde sí mismos, propósito y dirección a sus vidas” (R. Steiner).*

### Descripción del Planteo

**Padre:** Crea los dos recursos pipe, el pipe pipe1 y el pipe pipe2, comienza la creación de la estructura de procesos, especifica que no será escritor ni lector de ninguno de los dos pipes y espera la terminación del hijo 1 y del hijo 2 y termina.

Crea los recursos pipe

```
pipe(pipe1);  
pipe(pipe2);
```

Creación de la estructura de procesos

```
//padre  
if ( fork( ) )  
{  
    //padre  
    if ( fork( ) )  
    {  
        // padre  
        // Especifica que no será lector  
        close(pipe1[0]);  
        close(pipe2[0]);  
        // Especifica que no será escritor  
        close(pipe1[1]);  
        close(pipe2[1]);  
    }  
    else  
    {  
        //hijo 2  
        if ( fork( ) )  
            //hijo 2  
        else  
            //nieta 2  
    }  
}  
else  
{  
    //hijo 1  
    if ( fork( ) )  
        //hijo 1  
    else  
        //nieta 1  
}
```

Espera la terminación del hijo 1 y del hijo 2 y termina.

“Con mis maestros he aprendido mucho, con mis colegas, más, con mis alumnos todavía más.” Proverbio hindú

*“Nuestro más elevado propósito ha de ser formar seres humanos libres, capaces de impartir desde sí mismos, propósito y dirección a sus vidas” (R. Steiner).*

```
while(wait(NULL) != -1);  
exit(0);
```

**Hijo 1:** Será únicamente escritor del pipe pipe2, el mensaje que escribe en el pipe pipe2, lo obtiene de la entrada estándar. La lectura de la entrada estándar termina cuando obtiene el string “chau” y se queda esperando la terminación del Nieto 1.

```
// Especifica que no será lector  
close(pipe1[0]);  
close(pipe2[0]);  
// Especifica que no será escritor del pipe2  
close(pipe1[1]);
```

**Hijo 2:** Será únicamente escritor del pipe pipe1, el mensaje que escribe en el pipe pipe1, lo obtiene de la entrada estándar. La lectura de la entrada estándar termina cuando obtiene el string “chau” y se queda esperando la terminación del Nieto 2.

```
// Especifica que no será lector  
close(pipe1[0]);  
close(pipe2[0]);  
// Especifica que no será escritor del pipe1  
close(pipe2[1]);
```

**Nieto 1:** Será únicamente lector del pipe pipe1, la lectura del pipe la escribe en la salida estándar, y termina cuando lee del pipe “chau”.

```
// Especifica que no será lector del pipe1  
close(pipe2[0]);  
// Especifica que no será escritor  
close(pipe1[1]);  
close(pipe2[1]);
```

**Nieto 2:** Será únicamente lector del pipe pipe2, la lectura del pipe la escribe en la salida estándar, y termina cuando lee del pipe “chau”.

```
// Especifica que no será lector del pipe2  
close(pipe1[0]);  
// Especifica que no será escritor  
close(pipe1[1]);  
close(pipe2[1]);
```

“Con mis maestros he aprendido mucho, con mis colegas, más, con mis alumnos todavía más.” Proverbio hindú

*“Nuestro más elevado propósito ha de ser formar seres humanos libres, capaces de impartir desde sí mismos, propósito y dirección a sus vidas” (R. Steiner).*

### **Implementación**

Se pide la implementación del ejercicio en un entorno operativo Linux, utilizando el compilador del lenguaje C.

Juan Carlos Romero  
Mg. Guillermo Cherencio  
UTN FRD SO