

**PREGUNTA 1 .** Construya un programa en **C** que realice la labor que se describe a continuación **utilizando únicamente las llamadas al sistema tratadas en los guiones de prácticas.**

Se trata de construir una versión reducida de la orden `cat`. Llamaremos al ejecutable de este programa `micat`. Sintaxis de la línea de órdenes en que se ejecutará `micat` : (el carácter `|` indica elección entre izquierda y derecha)

```
$ micat f|- d|- ## debe tener 2 argumentos
```

donde **f** representa la ruta de un archivo que llamaremos “archivo fuente”, y **d** representa la ruta de un archivo que llamaremos “archivo destino”

Si los dos argumentos son distintos de “-” (guion ) el programa deberá leer todos los caracteres del archivo fuente **f** y escribirlos en el archivo destino **d**.

Si como argumento primero se ha especificado “-”, entenderemos que no se quiere especificar ningún archivo fuente y el programa deberá tomar como fuente la entrada estándar.

Si como argumento segundo se ha especificado “-”, entenderemos que no se quiere especificar ningún archivo destino y el programa deberá tomar como destino la salida estándar.

Estructura obligatoria que debe tener el programa: Debe realizar esta función utilizando dos procesos y cauces como mecanismos de comunicación entre los procesos (con nombre o sin nombre, decídalo usted). Uno de esos procesos leerá los datos de su fuente y los lleva a un cauce. El otro lee los datos del cauce y los lleva a su destino.

**PREGUNTA 2 .** La actuación del programa anterior queremos dividirla en dos programas fuentes en C.

Por un lado construya un programa llamado `leede0Escribeen1.c` que simplemente lee de `STDIN_FILENO` mientras no encuentre fin de archivo y escribe cada caracter leído en `STDOUT_FILENO`.

Por otro lado, en el programa `micat.c` ninguno de los dos procesos que teníamos contienen operaciones `read` ni `write`, sino que hacen un `exec` de `leede0Escribeen1` habiendo previamente manipulado adecuadamente la Tabla de Descriptores de Archivos Abiertos. A este programa le llamaremos `micatExec.c`

**PREGUNTA 3 .** Construya un programa en **C** que realice la labor que se describe a continuación **utilizando únicamente las llamadas al sistema tratadas en los guiones de prácticas.** Se trata de construir un programa que antes de realizar su trabajo central se asegure de que es la única instancia ejecutándose del archivo ejecutable. Para ello se utilizará un archivo como objeto para operar sobre él con `fcntl`.

El programa deberá tener esta estructura:

```
// Asegurarse de que no hay una instancia anterior de mi mismo ejecutándose
// mostrar mensaje indicativo de que se ha realizado la comprobacion
// hacer mi trabajo (esperar unos segundos)
// mostrar mensaje de fin
```

Realizaremos dos versiones:

- a) utilizando `F_SETLK`
- b) utilizando `F_SETLKW`