Dept. de Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación: Área de Telemática (GSyC)

Redirecciones y *pipes*

-dup, dup2, pipe-

Katia Leal Algara
katia.leal@urjc.es
https://gsyc.urjc.es/~katia/

dup: duplica un descriptor de fichero

```
#include <unistd.h>

int dup(int oldfd);
int dup2(int oldfd, int newfd);
```

- Se utiliza para convertir una ruta en un **descriptor de fichero** (número entero no negativo más pequeño no abierto actualmente para el proceso).
- Ambas llamadas al sistema crean una copia de un descriptor de fichero.
- En el caso de *dup*, se usa el descriptor de fichero más bajo que se encuentre libre de la tabla de descriptores de ficheros.
- dup2 hace que **newfd** sea la copia de **oldfd** y además cierra **newfd**:
  - Si **oldfd** no es un descriptor de fichero valido, falla la llamada (-1) y **newfd** no es cerrado.
  - Si **oldfd** y **newfd** tienen el mismo valor, dup2 no hace nada y devuelve **newfd**.

dup y dup2

- Hay que tener en cuenta que los procesos recién creados (por ejemplo, con execv) sólo entienden de:
  - Entrada estándar (0)
  - Salida estándar (1)
  - Salida error estándar (2)
- Es por ello que dup y dup2 ayudan a redireccionar la salida estándar, entrada estándar o salida de error estándar a otro descriptor de fichero que necesitemos.

pipe - crea una tubería o interconexión

```
#include <unistd.h>

int pipe(int descf[2]);
```

- pipe crea un par de descriptores de ficheros, que apuntan a un nodo-í de una tubería, y los pone en el vector de dos elementos apuntado por **descf**.
- **descf[0]** es para lectura y **descf[1]** es para escritura.
- En caso de éxito, se devuelve cero. En caso de error se devuelve -1 y se pone un valor apropiado en **errno**.