

Sistemas Operativos

Redireccionamiento

Redireccionamiento

- Procesos FILTRO
- Redirección entrada/salida/errores
- Ejemplos
- ¿Cómo hace el *Shell* para redireccionar?

Procesos FILTRO



FILTRO: Proceso que LEE datos de la entrada estándar (*stdin*), los TRANSFORMA y ESCRIBE los datos transformados en la salida estándar (*stdout*, *stderr*)



Esa convención inicial se puede cambiar REDIRECCIONANDO

Redirección

- Desde el intérprete de comandos:
 - Redirección de la salida:
who > quien_hay.tmp
 - Redirección de la entrada:
sort < quien_hay.tmp
 - Redirección de la salida de errores:
cc -o ej10 ej10.c **2>** error_10
 - Redirección de la entrada y la salida:
sort < quien_hay.tmp > quien_hay

/* ej110.c */

```
#include "error.h"
#include <fcntl.h>
#include <stdio.h>
```

who > quien_hay.tmp

```
main() {
    int fd;
    if( (fd = creat( "quien_hay.tmp", 0640) ) == -1)
        if( errno == EACCES ){
            printf( "fichero existe sin permiso escritura\n" );
            exit( 0 );
        }
        else syserr( "creat" );
    close( fd );
    close( 1 );
    fd = open( "quien_hay.tmp", O_WRONLY );
    if( fd == 1 )
        fprintf( stderr, "Salida Redireccionada\n" );
    else{
        fprintf( stderr, "Virus\n" );
        exit( 1 );
    }
    execl( "/bin/who", "who", 0 );
    syserr( "execl" );
}
```

se puede mejorar?

Ejercicios

- ej111.c:

Programa que realice las siguientes redirecciones:

```
sort < quien_hay.tmp > quien_hay
```

(quien_hay.tmp es el fichero generado en ej110.c)

- ej112.c:

Programa que ejecute el comando:

```
date >> quien_hay
```

(recordar: >> añade la salida del comando date al final del fichero quien_hay. Usar open(... O_APPEND) o lseek())

¿Cómo hace el shell para redireccionar?

```
. . .
switch ( fork() )
{
    . . .
    . . .
    case 0:                /* HIJO */
        if ( r_in )        redireccionar_entrada();
        if ( r_out )        redireccionar_salida();
        if ( r_out_app)    redireccionar_salida_append();

        exec(. . . . .);
    default:
        . . .
        . . .
}
```

hereda TDF y por lo tanto las redirecciones...

La separación de fork() y exec() simplifica el tratamiento de señales y los redireccionamientos. Además, permite que el código quede perfectamente localizado