



PRACTICA 7

⌚ Created	@January 15, 2021 10:31 AM
📅 Date	@November 23, 2020 → December 6, 2020
🏷️ Tags	
📝 Tema	

Tema i què es fa

Gestion entrada/salida. Entender como funcionan los device drivers: crear, eliminar, cargar y descargar dispositivos del Kernel. Redireccionar la entrada y salida de un proceso desde la Shell mediante el uso de < y >

Sessió i codis

▼ es1_v2.c

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

// Este código escribe todos los bytes que lee de la entrada std en la salida stda

main(){
char read_buffer[256]="";
char *buf="fin ejecución\n";
char buffer[1024];
int ret;
// USO
sprintf(buffer,".....\n");
write(2,buffer,strlen(buffer));
sprintf(buffer,"Este programa escribe por la salida std todo lo que lee de la entrada std. Si no has redirigido la salida, lo que
write(2,buffer,strlen(buffer));
sprintf(buffer,"Para acabar CtrlD\n");
write(2,buffer,strlen(buffer));
sprintf(buffer,".....\n");
write(2,buffer,strlen(buffer));

// Leemos del canal 0 (entrada std), 1 byte
ret=read(0,&read_buffer,sizeof(read_buffer));
// Cuando el read devuelve 0 significa que se ha acabado la
// entrada de datos --> acabamos el bucle de lectura
while(ret>0){
    // Escribimos en el canal 1 (salida std) 1 byte
    write(1,&read_buffer,sizeof(read_buffer));
    ret=read(0,&read_buffer,sizeof(read_buffer));
}
write(1,buf,strlen(buf));
}
```

▼ es6_v2.c

```

#include <unistd.h>
#include <stdio.h>
#include <errno.h>
#include <signal.h>
#include <string.h>

void func(int s) {
    char buff[128] = "recibido\n";
    write(1, &buff, strlen(buff));
}

int main(){
    struct sigaction sa;

    sa.sa_handler = &func;
    //sa.sa_flags = SA_RESTART;
    sigfillset(&sa.sa_mask);
    sigaction(SIGINT, &sa, NULL);

    char buf[256];
    char c;

    int ret = read(0, &c, sizeof(char));

    if (ret >= 0) {
        sprintf(buf, "read correcto\n");
        write(1,buf,strlen(buf));
    } else {
        if (errno == EINTR) {
            sprintf(buf, "read interrumpido por signal\n");
            write(1,buf,strlen(buf));
        } else {
            sprintf(buf, "read con error\n");
            write(1,buf,strlen(buf));
        }
    }
}

```

Manual

? mknod - Crea fichero especial

? insmod - Inserta un modulo en el Kernel

? lsmod - Muestra los modulos cargados en Kernel