

# Ejercicio de Drivers Solución

Sistemas Operativos

1er cuatrimestre 2023

## 1. Lector

### 1.1. Operaciones

- read: lee el código de barras y lo devuelve

### 1.2. Registros

- CODIGO: código de barras leído

### 1.3. Código del driver

```
1 read(char* buffer){
2     int codigo = IN(CODIGO)
3     copy_to_user(buffer, &codigo, sizeof(codigo));
4
5     return sizeof(codigo);
6 }
```

## 2. Brazo

### 2.1. Operaciones

- write: indica al brazo hacia dónde moverse, y qué hacer con la mano

### 2.2. Registros

- POSICION: 0, 1,...,10, 0 para el robot, 1 a 10 para la estantería, 11 para la cinta
- MANO: ABRIR, CERRAR: indica qué debe hacer cuando llegue a la posición

### 2.3. Código del driver

```
1  write(char* buffer){
2      brazo_operacion op;
3      copy_from_user(&op, buffer, sizeof(op));
4
5      OUT(POSICION, op.posicion);
6      OUT(MANO, op.mano);
7
8      return sizeof(op);
9  }
```

## 3. Movimiento

### 3.1. Operaciones

- write: indica al robot hacia dónde moverse; se bloquea mientras se mueve y retorna cuando haya llegado a destino

### 3.2. Registros

- DEST\_X: indica la posición X
- DEST\_Y: indica la posición Y
- CONTROL: START para empezar a moverse

### 3.3. Interrupciones

- INT\_MOV\_OK: se activa cuando el robot llega a destino

### 3.4. Código del driver

```
1  sem sem = sem_init(0);
2
3  handler() {
4      sem.signal();
5  }
6
7  init() {
8      map_irq(INT_MOV_OK, handler)
9  }
10
11 write(char* buffer){
12     ubicacion u;
13     copy_from_user(&u, buffer, sizeof(u));
14
15     OUT(DEST_X, u.x);
16     OUT(DEST_Y, u.y);
17     OUT(CONTROL, START);
18     sem.wait();
19
20     return sizeof(u);
21 }
```

## 4. Código del usuario

```
1  struct paquete {
2      int x;
3      int y;
4      int codigo;
5  }
6
7  struct brazo_operacion {
8      int posicion;
9      enum mano {ABRIR, CERRAR};
10 }
11
12 struct ubicacion {
13     int x;
14     int y;
15 }
16
17 void tomar_objeto(int i) {
18     op = {posicion=i, mano=CERRAR};
19     write(brazo, op, sizeof(op));
20 }
21
22 void main() {
23     int lector = open("/dev/lector");
24     int brazo = open("/dev/brazo");
25     int movimiento = open("/dev/movimiento");
26     brazo_operacion op;
27
28     while(1) {
29         paquete p = siguiente_paquete();
30         ubicacion u = {x=p.x, y=p.y};
31         write(movimiento, u, sizeof(u)); // se bloquea hasta que llegue
32
33         // llegó a la estantería
34         for(int i=1; i<=10; i++) {
35             // Tomar el objeto de la estantería
36             tomar_objeto(i);
37
38             // Acercarlo al robot y leerlo
39             tomar_objeto(0);
40
41             int codigo;
42             read(lector, &codigo, sizeof(codigo));
43             if(codigo==p.codigo) {
44                 break;
45             }
46
47             // Devolver el objeto a la estantería
48             soltar_objeto(i);
49         }
50
51         // Al salir del for, el objeto está en la mano del robot
52         // Moverse hasta la cinta de entrega, y soltar el paquete
53         ubicacion cinta = {x=0, y=0};
54         write(movimiento, cinta, sizeof(p)); // se bloquea hasta que llegue
55         op = {posicion=11, mano=ABRIR};
56         write(brazo, op, sizeof(op));
57     }
58 }
```