

# SR02 – TD03

## 1

### 1.1

```
def rouler(n):
    save_contexte(PCB[actif]);

    for i in range(n):
        while RE != 1:
            wait();
            RC = 1;

    load_contexte(PCB[actif]);
    load_psw(PCB[actif].psw);
```

### 1.2

```
def rouler(n):
    save_contexte(PCB[active]);
    push(blocked_queue, active);

    packet_io p;
    p.pid = PCB[active].pid;
    p.nb = n;
    p.type = 1;

    push(request_queue, p);
    new_pr = pop(ready_queue);

    load_contexte(PCB[active]);
    load_psw(PCB[active].psw);
```

```
def driver_robot():
    while(TRUE):
        if not empty(request_queue):
            p = pop(request_queue);
            while(RE != 1):
                wait();
            RCC = p.type;
            p.nb--;
            pause();
            pr = pop(blocked_queue);
            push(ready_queue, pr);
```

.

```
def robot_int_routine(p):
    save_context(PCB[active]);
    if p.nb == 0:
        # envoie d'un signal pour
        # reveiller le driver
    else:
        RC = p.type;
        p.nb--;

    load_contexte(PCB[active]);
    load_psw(PCB[active].psw);
```

## 2

### 2.1

|             |     |
|-------------|-----|
| <b>FIFO</b> | 732 |
| <b>PCTR</b> | 202 |
| <b>SCAN</b> | 258 |
| <b>LOOK</b> | 190 |

### 2.2

Risque de stagner si toutes les opérations se font au même endroit, laissant de côté celles un peu plus lointaines → phénomène de famine  
PCTR est le seul concerné

## 3

### Contiguë

**Ajout Début**

**Suppression Fin**

### Chaînage

**Ajout Début** Accès au répertoire, écriture sur le disque

**Suppression Fin** Accès au répertoire, parcours de tous les blocs

### Chaînage Indexé

**Ajout Début** Accès au répertoire, écriture sur le disque

**Suppression Fin** Accès au répertoire

### Indexé

**Ajout Début** Accès au répertoire, écriture sur le disque

**Suppression Fin** Accès au répertoire

## 4

3 accès sont nécessaires