# SR02 – TD01

# 1

## 1.1 Mono-Programmé

**Un seul programme en mémoire**

— Pas besoin d'ordonnaceur qui gère les priorités
— Mémoire non fragmenté

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CPU** | $p_1$ | $p_1$ | $p_1$ | | | $p_1$ | $p_1$ | | $p_1$ | $p_1$ | $p_1$ | $p_2$ | $p_2$ | | | $p_2$ | | |
| **Impression** | | | | $p_1$ | $p_1$ | | | | | | | | | | | | $p_2$ | $p_2$ |
| **Écran** | | | | | | | | $p_1$ | | | | | | $p_2$ | $p_2$ | | | |
| **Attente** | | $p_2$ | $p_2$ | $p_2$ | $p_2$ | $p_2$ | $p_2$ | $p_2$ | $p_2$ | $p_2$ | $p_2$ | | | | | | | |

**Taux Occupation Processeur** $\frac{11}{18} \rightarrow 61\%$

## 1.2 Multi-Programmé

**Plusieurs programmes en mémoire**

— Entrées/Sorties pas prises en compte par le processeur
— Besoin d'un gestionnaire de mémoire

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **CPU** | $p_1$ | $p_1$ | $p_1$ | $p_2$ | $p_2$ | $p_1$ | $p_1$ | $p_2$ | $p_1$ | $p_1$ | $p_1$ |
| **Impression** | | | | $p_1$ | $p_1$ | | | | $p_2$ | $p_2$ | |
| **Écran** | | | | | | $p_2$ | $p_2$ | $p_1$ | | | |
| **Attente** | | $p_2$ | $p_2$ | | | | | | | | |

**Taux Occupation Processeur** $\frac{11}{11} \rightarrow 100\%$

## 1.3 Temps Partagé

**Quantum de 1s (unité de temps allouée à chaque processus)**

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CPU** | $p_1$ | $p_2$ | $p_1$ | $p_2$ | $p_1$ | | $p_2$ | $p_1$ | $p_1$ | | $p_1$ | $p_1$ | $p_1$ |
| **Impression** | | | | | | $p_1$ | $p_1$ | $p_2$ | $p_2$ | | | | |
| **Écran** | | | | | $p_2$ | $p_2$ | | | | $p_1$ | | | |
| **Attente** | | $p_1$ | $p_2$ | $p_2$ | | | | | | | | | |

**Taux Occupation Processeur** $\frac{11}{13} \rightarrow 84\%$

# 2

**2.1** États possibles du processus :
- Prêt : Attente du processeur
- En Exécution :
- Bloqué : Attente IO

**2.2** Plusieurs programmes coexistent à l'état d'attente (prêt ou bloqué), il faut donc deux files.

**2.3** Favoriser les processus demandant de l'I/O en augmentant la priorité du processus lors de l'accès à l'I/O (puis de le réduire si il n'en a pas fait depuis un certain temps)

## 2.4 PCB

— PID
— État
  · AX, BX, CX, DX, EX
  · CO
  · FLAGS
  · DS, CS
— Cause du blocage
— Priorité, quantum

## 2.5

```
# Program 1
10 MOV DX, 0
11 MOV EX, 2

label:
12 ADD DX, 3
13 MOV DX, CX
14 SUB EX, 1
15 MOV EX, CX
16 JNZ label
17 OUT
```

**PCBs** PID
state
CO = 0, DS = 10, CS = 10
AX, BX, CX, DX, EX = 0
quantum = 2, priority = 0

Alors en vrai après il a filé un papier avec la correction parce que c'était chiant

```
# Program 2
30 MOV @0, 3
31 IN
32 MOV @1, DX
34 MOV AX, @0
35 MOV BX, @1
36 ADD AX, BX
37 MOV DX, CX
38 OUT
```

# 3

On représente RM avec

| **EX** | **A** | **B** | **C** | **D** | **H** |
|---|---|---|---|---|---|
| x | x | x | x | x | x |

Chaque interruption à un masque propre afin de représenter la priorité :

| | | | | | | |
|---|---|---|---|---|---|---|
| **EX** | 0 | 0 | 0 | 0 | 0 | 0 |
| **A** | 1 | 0 | 0 | 0 | 0 | 0 |
| **B** | 1 | 1 | 0 | 0 | 0 | 0 |
| **C** | 1 | 1 | 1 | 0 | 0 | 0 |
| **D** | 1 | 1 | 1 | 1 | 0 | 0 |
| **H** | 1 | 1 | 1 | 1 | 1 | 0 |

```
Routine_i:
    STM     PCB.masque
    MSK     masque[i]
    MOV     PCB.co, [0]
    STO     PCB.accumulateur
    ACT

    ; Routine

    DOR
    CLA     PCB.accumulateur
    MSK     PCB.co
    ACT
    TRA     @PCB.co
```

A l'initialisation :

| **EX** | **A** | **B** | **C** | **D** | **H** |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 |

**On met dans la case mémoire I(1(EX), 2(A), 3(B), 4(C), 5(D), 6(H))**
**TRA adresse-routine-i pour charger l'adresse de la routine**

# 4

SVC emit_horloge

$$\frac{10^{-3} \times 100}{10^{-6} \times 5} = 20.000$$

```
routine_int_clock:
    SAV     PCB                 ; Sauvegarde contexte
    SVC     prelever_mesures
    SVC     afficher_mesures
    SVC     init_clock, 20000
    RES     PCB                 ; Restore contexte
```

```
svc_fin:
    DEL     PCB                 ; Detruire PCB
    SVC     moniteur


routine_int_clock:
    SIG     delai_garde         ; Alerter(delai_garde)
    DEL     PCB                 ; Detruire PCB
    SVC     moniteur


svc_moniteur:
    MOV     PCB, load_process
    SVC     init_clock, delai_garde
    MPSW    PCB, PSW
```