

1) Définitions sur les éléments d'un STR:

A) Scheduling: Ordonnancement
 → un programme qui fait partie du STR

- On appelle Scheduler la partie du OS qui répartie le temps d'utilisation du processeur (CPU) entre différentes tâches. → Ceci permet d'augmenter la rentabilité du processeur.

→ le Scheduler doit allouer le processeur aux différentes tâches selon des priorités prédéfinies (choisies d'avance).

* un ordonnanceur doit assurer deux états

- 1) exécution des tâches
- 2) le Wait: blockage d'une tâche + le moment où l'exécution et le Wait + moment du Wait de chaque tâche sont définis dans la table de chaque tâche.

Ordonnanceur ⇒ un algorithme d'ordonnancement.

- 1) un algorithme d'ordonnancement est la stratégie utilisée par l'ordonnanceur pour organiser l'exécution des tâches.

Cet algorithme est basé sur la connaissance de la nature des tâches (priorité, ..., tâche périodique ou aperiodique - préemption ou non - OS à priorité fixe ou dynamique.

③

STR $\Leftrightarrow$ ordonnanceur

- quand on parle de STR $\Rightarrow$ il existe un ordonnanceur
- quand on parle de ordonnanceur $\Rightarrow$ notre Système d'exploitation (OS) est un STR.

1b) Ordonnancement Préemptif / non préemptif:

* ordonnancement ^{non} préemptif:

- dans cet ordonnancement on n'interrompt jamais l'exécution d'une tâche en cours au profit d'une autre tâche.

* Ordonnancement préemptif

- une tâche en cours d'exécution peut perdre involontairement (sans que un wait soit mentionné dans sa table) le processeur au profit d'une autre tâche jugée plus urgente.

par exemple si une tâche monopolise le CPU sur une trop longue période, le Système-OS peut décider de la suspendre. elle est placée en état runnable, on dit que la tâche a été préemptée. $\rightarrow$ STR préemptif.

7

S M T W T F S

DATE

2) Le noyau d'un OS:

- Le noyau d'un OS est la partie principale du Système d'exploitation. Le noyau comporte donc des éléments indispensables pour le fonctionnement d'un OS.
- En tant que partie du Système d'exploitation, le noyau fournit des mécanismes de gestion du matériel, notamment de la mémoire, du processus et des échanges d'informations entre logiciels et périphériques matériels.

3) Les interruptions:

* Supposons que nous avons un STR contenant deux tâches à exécuter l'une après l'autre:

La tâche 1 appelée T_1 sera exécutée pendant un temps t_1 (temps d'exécution de T_1). ensuite lorsque la tâche T_1 sera finie, la tâche T_2 sera exécutée pendant un temps t_2 . Les deux temps t_1 et t_2 sont appelés des "Timing".

• Dans ce cours nous allons expliquer

l'exécution des tâches T_1 et T_2 par le STR avec plus de détails que les cours passés. Pour cela on va citer les éléments suivants

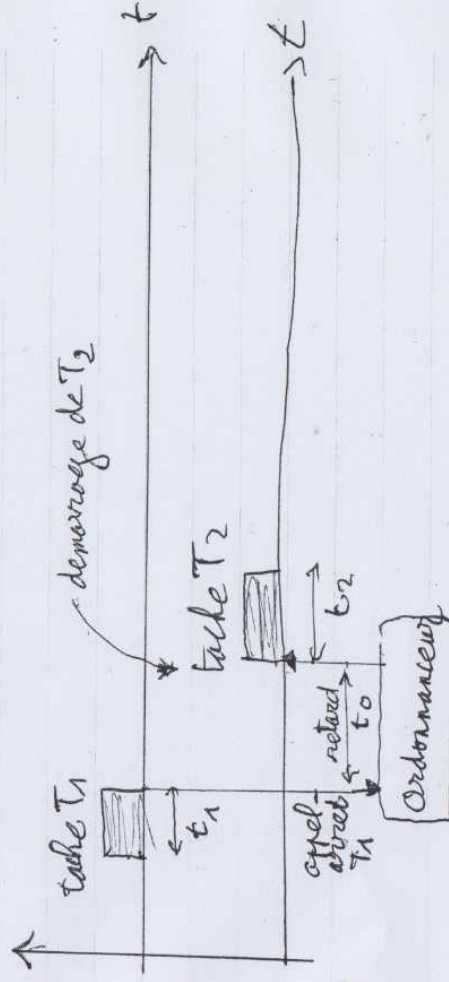
- L'ordonnanceur (déjà cité)
- les retards d'exécution des tâches (*)
- les timings des tâches
- les interruptions (*)
- les états d'une tâche.

a) L'ordonnanceur est la partie du STR qui sélectionne à chaque moment la prochaine tâche à exécuter grâce à des "interruptions" qui donnent le C_{n+1} à la n^{e} tâche. ...

b) le retard d'exécution t_0 :

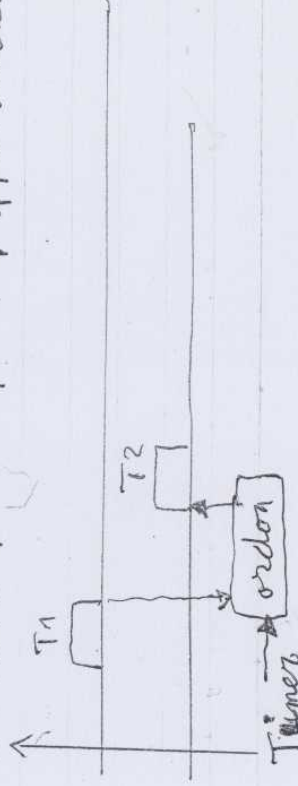
⑤

- En premier, la tâche T_1 appelle l'ordonnanceur pour arrêter l'exécution de la tâche T_1 .
- L'ordonnanceur va faire la commutation du processeur de la tâche 1 vers la tâche 2 après un retard t_0 .



3) les timing :

la tâche T_1 doit finir son temps d'exécution (timing) pour passer la main (laisser le processeur) à la tâche T_2 . Mais si le T_1 peut arrêter avant t_1



(10)

DATE

S M T W T F S

* pour réaliser les deux timings T_1 et T_2 l'ordonnanceur devrait être connecté à un Timer (compteur).

d) des interruptions !

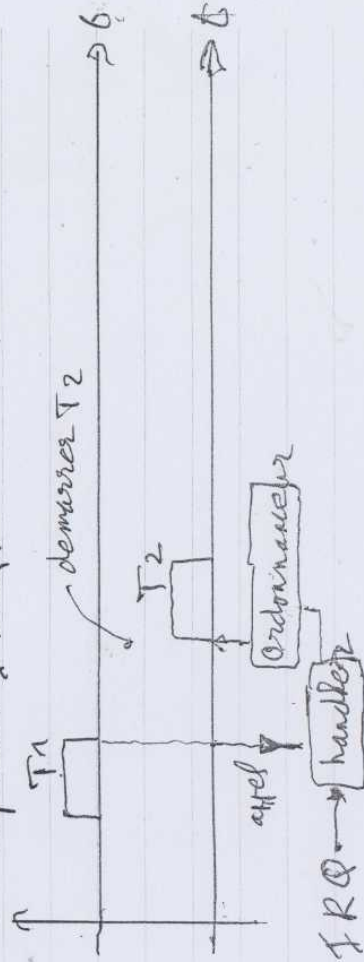
* Le passage de la tâche à une autre tâche se fait en fonction de plusieurs critères :

- priorité de la tâche (dans le tableau)
- tâche prête à être exécutée ou non
- ...

→ on va voir dans (2)

d) les interruptions :

Pour que l'ordonnanceur fait passer la tâche T_2 qui était prête à être exécutée (après la tâche T_1), l'ordonnanceur a besoin d'une "interruption" IRQ.



handler = programme
"à exécuter"

(11)
* les interruptions sont activées par l'horloge^H de l'OS. (STR)

* il n'y a pas de STR sans ordonnanceur et il n'y a pas de commutation entre les tâches sans interruptions $\Rightarrow$ pas de STR sans interruptions

- les interruptions dans un OS permettent de réagir aux événements donc à être (le processeur) en phase avec l'environnement OS.

Il y a deux types d'interruptions :

1) interruption matérielle :

provoque création de tâche immédiate par rapport à la file d'attente de l'ordonnanceur

[par exemple STR fonctionne et une clé USB branchée]

2) interruption logicielle : [notre cas pour activer ordonnanceur]

provoque une commutation de tâches différentes.