



Cours A7 : Temps Réel

Pierre.Paradinas / @ / cnam.fr

Cnam/Cedric

Systemes Enfouis et Embarqués (SEE)

Motivations

Du jour :

-  les mécanismes multitâches,

-  la gestion des priorités,

-  l'ordonnancement,

-  la gestion mémoire,

-  +/- par rapport au TR.

-  Comprendre les mécanismes systèmes mis en oeuvre dans un système temps réel

Notre système

- Gestion de plusieurs tâches.
- Pour chaque tâche :
 - Contexte: son nom, les registres d'états, etc.
 - CO (compteur ordinal) et une zone mémoire de programme.
 - PP (pointeur de pile) et données temporaires.

La gestion des tâches (1/3)

Etats des tâches :

-  courant

-  en-attente

-  prêt

Action sur les tâches (changement d'états) :

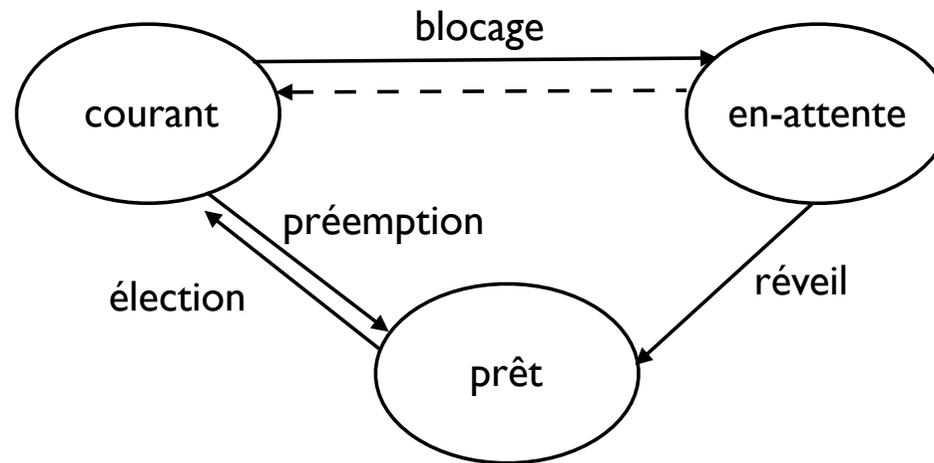
-  allocation

-  préemption

-  blocage/réveil

La gestion des tâches (1/3)

Diagramme de gestion des tâches



La gestion des tâches (1/3)

- On appelle ordonnanceur la partie du SE qui prend en charge la gestion des tâches.
 - création, initialisation, élection, destruction
 - gestion du diagramme des tâches (en-attente, courant, prêt)
 - structure de données associées
 - liste tâches prêtes
 - liste tâches en attentes
- Politique d'ordonnancement...

Les politiques d'ordonnancement

Ordonnancement statique et dynamique

-  statique : avant un flot d'exécution

-  dynamique : pendant l'exécution d'un flot les tâches sont re-prioriser en fonction des priorités

La préemption de tâches

-  pour pouvoir réorganiser les priorités des tâches dans la liste des tâches en attente et élire une tâche prioritaire il faut pouvoir arrêter la tâche courante (préemption).

Les algorithmes d'ordonnancement

- Tourniquet (round robin)
- Priorité
- Premier entré premier sorti (FIFO)

Exemple du tourniquet

 Soient les données suivantes :

Tâche	T1	T2	T3	T4
Durée	3	4	1	6

 Donner le déroulement des tâches pour le tourniquet considérant que les tâches s'effectuent pendant des durées de temps fixées.

Exemple du tourniquet

Tâche	T1	T2	T3	T4
Durée	3	4	1	6

Les tâches s'exécuteront dans cet ordre :

T1	T2	T3	T4	T1	T2	T4	T1	T2	T4	T2	T4	T4	T4
----	----	----	----	----	----	----	----	----	----	----	----	----	----

Exercice

- Donner l'algorithme de gestion des tâches dans la cas d'un tourniquet.
- Structure de données et fonctions de base (à définir et préciser) :
 - Liste_Tâche = une liste chaînée contenant les tâches prêtes.
Première_Tâche = retourne la première tâche dans Liste_Tâche.
Tâche_Suivante = permet de passer à la tâche suivante dans la liste.
TC = Tâche Courante.
DTC = Durée de TC.
Insérer_Fin_Liste_Tâche = insère la tache TC en fin de liste Liste_Tâche.

Algorithme du Tourniquet

Liste_Tâche = une liste chaînée contenant les tâches prêtes.
Première_Tâche (Liste_Tâche) = retourne la première tâche dans Liste_Tâche.
TC = Tâche Courante.
DTC = Durée de TC.
Insérer_Fin_Liste_Tâche (TC) = insère la tâche TC en fin de liste Liste_Tâche.

Tourniquet

```
FTQ Liste_Tâche NON Vide
  TC = Sortir_Première_Tâche ;
  Exécuter_Tâche (TC, 1ut)
  % Exécution pendant un quantum de temps%
  % Interruption de TC ;
  DTC = DTC - 1 ;
  Si DTC < 0 alors
    Insérer_Fin_Liste_Tâche (TC) ;
  fsi
FTQ
```

Tourniquet +/-

Avantages :

-  équité entre toutes les tâches,
-  mise en oeuvre simple.

Inconvénients :

-  ne tient pas compte des besoins applicatifs,
-  \Rightarrow notion de priorité.

Priorité

 Soient les données suivantes :

Tâche	T1	T2	T3	T4
Durée	3	4	1	6
Priorité	2	3	4	1

 Donner le déroulement des tâches pour le tourniquet considérant que 1 représente priorité la plus haute.

Priorité simple

Les tâches s'exécuteront dans cet ordre :

T4	T4	T4	T4	T4	T4	T1	T1	T1	T2	T2	T2	T2	T3
----	----	----	----	----	----	----	----	----	----	----	----	----	----

Si le tableau devient ?

Tâche	T1	T2	T3	T4
Durée	3	4	1	6
Priorité	2	2	3	1

T4	T4	T4	T4	T4	T4	?							
----	----	----	----	----	----	---	--	--	--	--	--	--	--

Exercice

- Donner l'algorithme de gestion des tâches dans la cas d'un tourniquet.
- Structure de données et fonctions de base (à définir et préciser) :
 - Liste_Tâche = une liste chaînée contenant les tâches prêtes.
Première_Tâche = retourne la première tâche dans Liste_Tâche.
Tâche_Suivante = permet de passer à la tâche suivante dans la liste.
TC = Tâche Courante.
DTC = Durée de TC.
Insérer_Fin_Liste_Tâche = insère la tache TC en fin de liste Liste_Tâche.

Algorithme Priorité simple

Priorité

```
TQ Liste_Tâche NON Vide  
    TC = Sortir_Tâche_Prioritaire ;  
    Exécuter (TC) ;  
FTQ
```

Priorité et tourniquet

Tâche	T1	T2	T3	T4
Durée	3	4	1	6
Priorité	2	2	3	1
Ordre arrivée	3	2	1	4

Les tâches s'exécuteront dans cet ordre :

T4	T4	T4	T4	T4	T4	T2	T2	T2	T2	T1	T1	T1	T3
----	----	----	----	----	----	----	----	----	----	----	----	----	----

Priorité avantages et inconvénients

Avantages :

-  Meilleur respect des contraintes applicatives.

Inconvénients :

-  Ne tient pas compte de la communication entre tâches.

Premier entré premier sorti (FIFO)

● Donner l'algorithme de cette méthode :

● **FIFO**

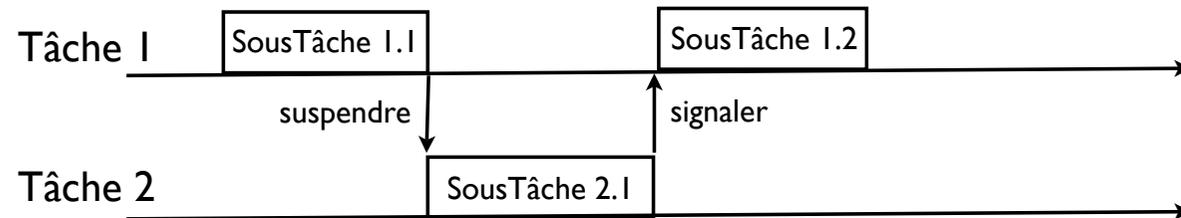
```
TQ Liste_Tâche NON Vide  
    TC = Première Tâche (Liste_Tâche);  
    Exécuter (TC) ;  
    Supprimer (TC) ;  
FTQ
```

● Discuter les pros/cons

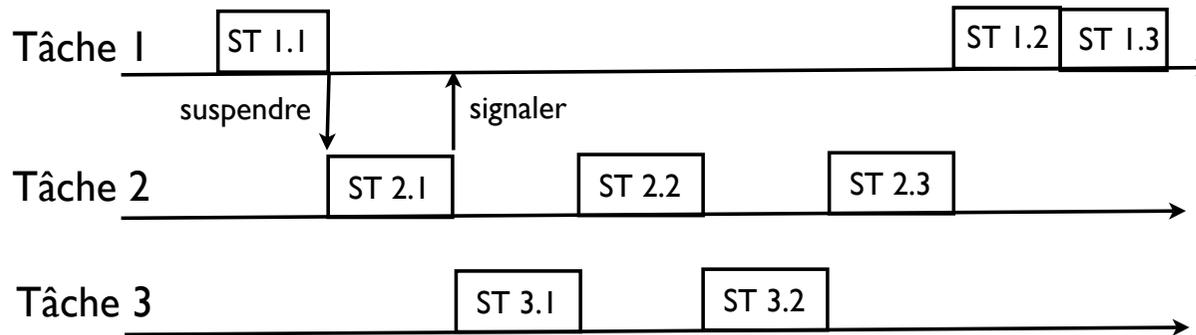
Gestion de tâches communicantes

- Renforcement des primitives de notre système:
 - besoin de suspendre une tâche en cours d'exécution : suspendre
 - besoin d'informer une tâche en-attente de la disponibilité d'une ressource signaler
 - Exemple : une tâche est mise en attente d'une E/S(suspendre), lors de la disponibilité de la ressource E/S elle sera réveillée (**signaler**)

Exemple



Exemple 2

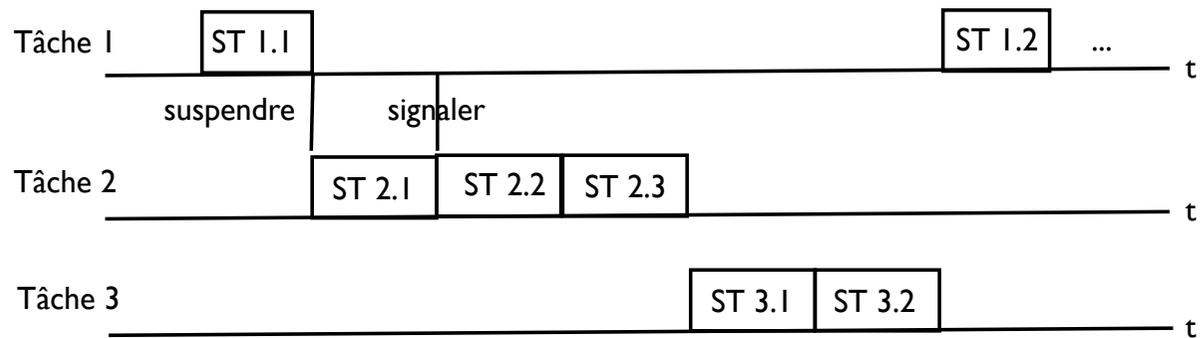


Tâche	T1	T2	T3
Durée	3	3	2
Priorité	3	2	2

Exercice

- Donner l'algorithme d'ordonnement de l'exemple 2?
- Discuter du choix ?

Exemple 2 bis



Tâche	T1	T2	T3
Durée	3	3	2
Priorité	3	2	2

Comparer les deux exemples

- Donner l'algorithme de l'exemple 2 bis
- Où il y a besoin d'un SE préemptif ?
- Les tâches prioritaires se finissent-elles les premières?

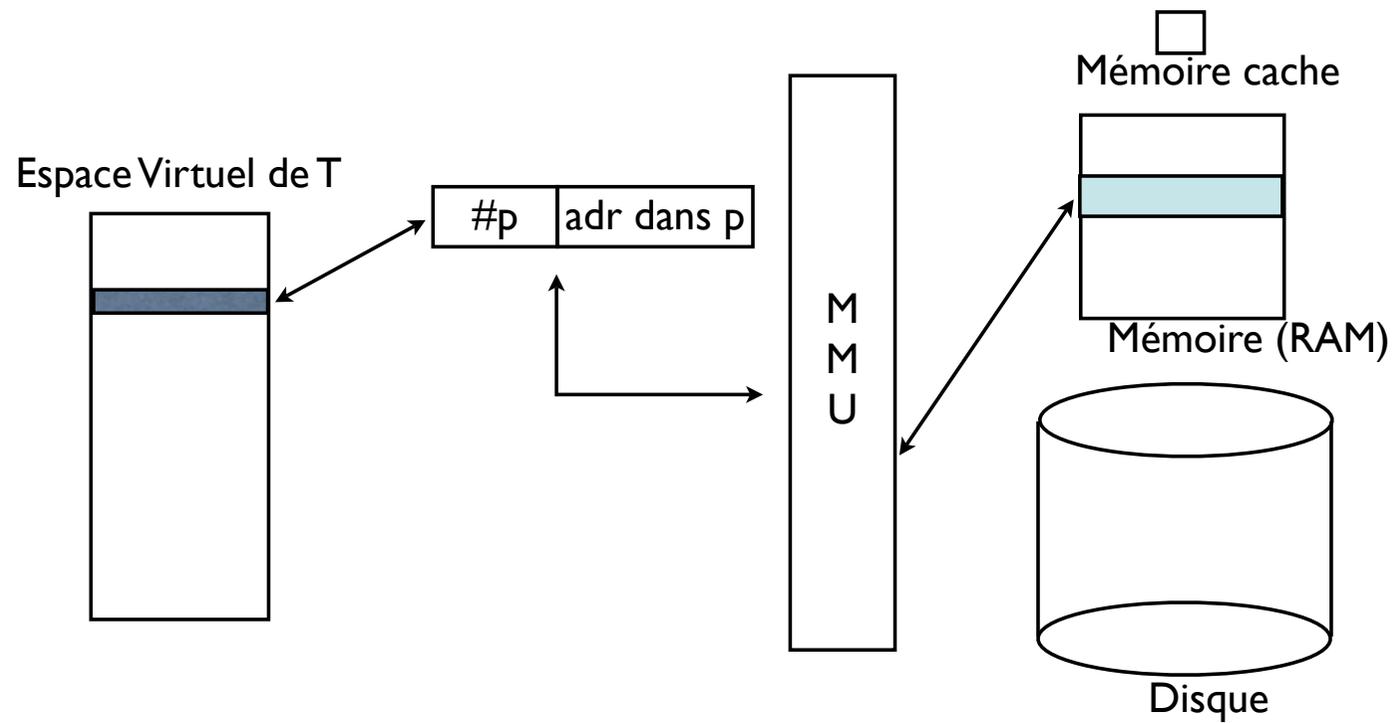
Limites de ces approches

- Unix basé sur une priorité et le partage du temps.
- Les STRs sont basés sur un ordonnancement préemptif des tâches.

Gestion mémoire (1/4)

- Dans les SEs, les adresses mémoires sont des adresses virtuelles, le SE fait la correspondance entre des adresses virtuelles et des adresses réelles.
- Trois niveau de mémoire avec des vitesses d'accès différentes :
 - mémoire cache
 - mémoire de masse
 - mémoire disque

Gestion mémoire (1/4)



Gestion mémoire (1/4)

Gestion par pages

Accès mémoire et gestion des pages :

 Demande d'accès mémoire

 Si page disponible en mémoire alors accès

 Sinon % défaut de page %

 suspendre la tâche

 dé-allouer une page %choix en fonction de la politique%

 charger nouvelle page

 signaler

fsi

Gestion mémoire (1/4)

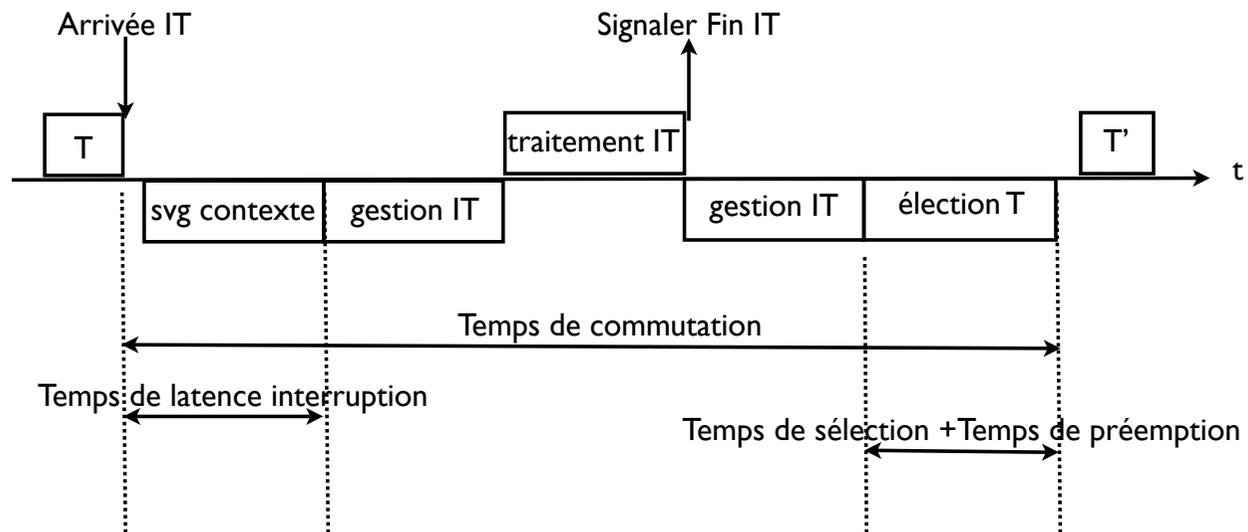
- Limite pour le TR :
 - Difficile de prédire un défaut de page :
 - Pour l'éviter des systèmes offrent des fonctions de réservation/ blocage de page.
 - Opération lourde et complexe.
 - Opération réaliser en général par interruption.

Gestion des interruptions (I/)

Les étapes de la gestion des ITs :

-  Sauvegarde du contexte d'exécution de la la tâche courante (CO, PP, registres)
-  Activation du gestionnaire d'IT
-  En fonction de la nature de l'IT et des priorités entre IT
-  Traitement de l'IT
-  Signaler fin IT
-  Reprendre tâche ou élire nouvelle tâche

Gestion des interruptions (I/)



Gestion des interruptions (I/)

Limites pour le TR

-  Pas de garantie sur les temps de latence, de traitement et de restitution du contexte d'une tâche.
-  Les STRs disposent de priorités définies finement pour gérer les interruptions et les repoussent au niveau de l'application.