

Université Mohamed Khider de Biskra  
Département d'Informatique

Module: **Systemes temps réel**  
(STR)

Niveau: 1<sup>ère</sup> année Master

Assuré par: Mr. Salim Bitam

2010/2011

# Programme du module

- I - Initiation aux systèmes temps réel.
- II- Ordonnancement des tâches périodiques.
- III- Ordonnancement des tâches apériodiques.
- IV- Exécutifs temps réel.
- V- Langages évolués temps réel.

## Documentation:

1. Alain Darse Oil, Pascal Pilot, **Le temps réel en milieu industriel**, Edition DUNOD, 1991.
2. Jane W. S. Liu, **Real-time Systems**, Edition Prentice Hall, 2000.

# Chapitre 01

## Initiation aux systèmes temps réel

1. Définitions de base
2. Le système temps réel
3. Les domaines d'application
4. Les problèmes traités pour la conception d'un STR
5. Les modes de fonctionnement

# Systeme

## Définition:

ensemble d'**entités** (éléments),

**Interactions** entre les entités et avec l'environnement,

**Pour** transformer un flux d'entrées en un autre flux de sorties, selon un certain nombre de **principes** ou **règles**.

## Exemple:

### Systeme informatique:

ensemble structuré des composants logiciels et matériels, pour automatiser une tâche ou un service.

### Systeme d'exploitation:

ensemble structuré et hiérarchisé de programmes et de processus regroupés autour du noyau, qui gère les divers éléments d'un système informatique.

# Environnement

- ensemble d'éléments,
- entourent un individu ou une espèce (système),
- dont le système contribue directement à subvenir au besoins de l'environnement.

# Problématique

- Système classique (non TR): simple avec l'absence de contraintes de temps;
- Comment concevoir un système qui dépend de certains problèmes au paramètre **temps** ?
- **Exemples:**
  1. Distributeur de billets: < 5min,
  2. Avion: réponse à un arrêt brutal d'un réacteur < 3min,
  3. Détecteur de missiles: réaction < 3 sec,
  4. Système de freinage ABS: <150ms pour acquérir l'information et <1s pour réagir.

# Définition: système temps réel (1)

## Définition 01:

- Système informatique
- Réaction à l'environnement extérieur
- Certain nombre de contraintes temporelles.
- STR dépend donc:
  1. résultat attendu (réaction) -SS-
  2. moment de l'obtention du résultat -STR-

# Définition: système temps réel (2)

## Définition 02:

Dépendance de la correction du système entier:

-pas seulement des valeurs des résultats produits

-mais également des délais dans lesquels les résultats sont produits:

-> réponse exigée (pas d'absence de résultat)

-> réponse dans un **délai** spécifique



# Définition: système temps réel (3)

- En informatique:
  - un STR apparait dans l'informatique industrielle:
    - lorsque ce système informatique contrôle (ou pilote) un équipement physique à une vitesse adaptée à l'évolution de l'équipement contrôlé.
- Différenciation (STR/autres systèmes):
  - la prise en compte de **contraintes temporelles** dont le respect est aussi important que l'exactitude du résultat =
    - délivrer des résultats exacts **mais** dans des délais imposés.

# Caractéristiques (1)

- exactitude **logique** (comme tout système) : les sorties sont déterminées en fonction des entrées et de l'état interne,
- exactitude **temporelle** (propre au STR) : les sorties (résultats) sont fournies au moment approprié (temps nécessaire), en d'autres termes, la réaction du système aux événements extérieurs doit se produire pendant son évolution,

# Caractéristiques (2)

- Conception dans le contexte de son environnement (système dédié et système embarqué),
- C. de temps: pas de **minimisation** de la moyenne des temps de réponse, mais de **satisfaire** les échéances de toutes les tâches individuelles,
- Prédicibilité est plus importante que la rapidité de la réponse,

# Types des STR

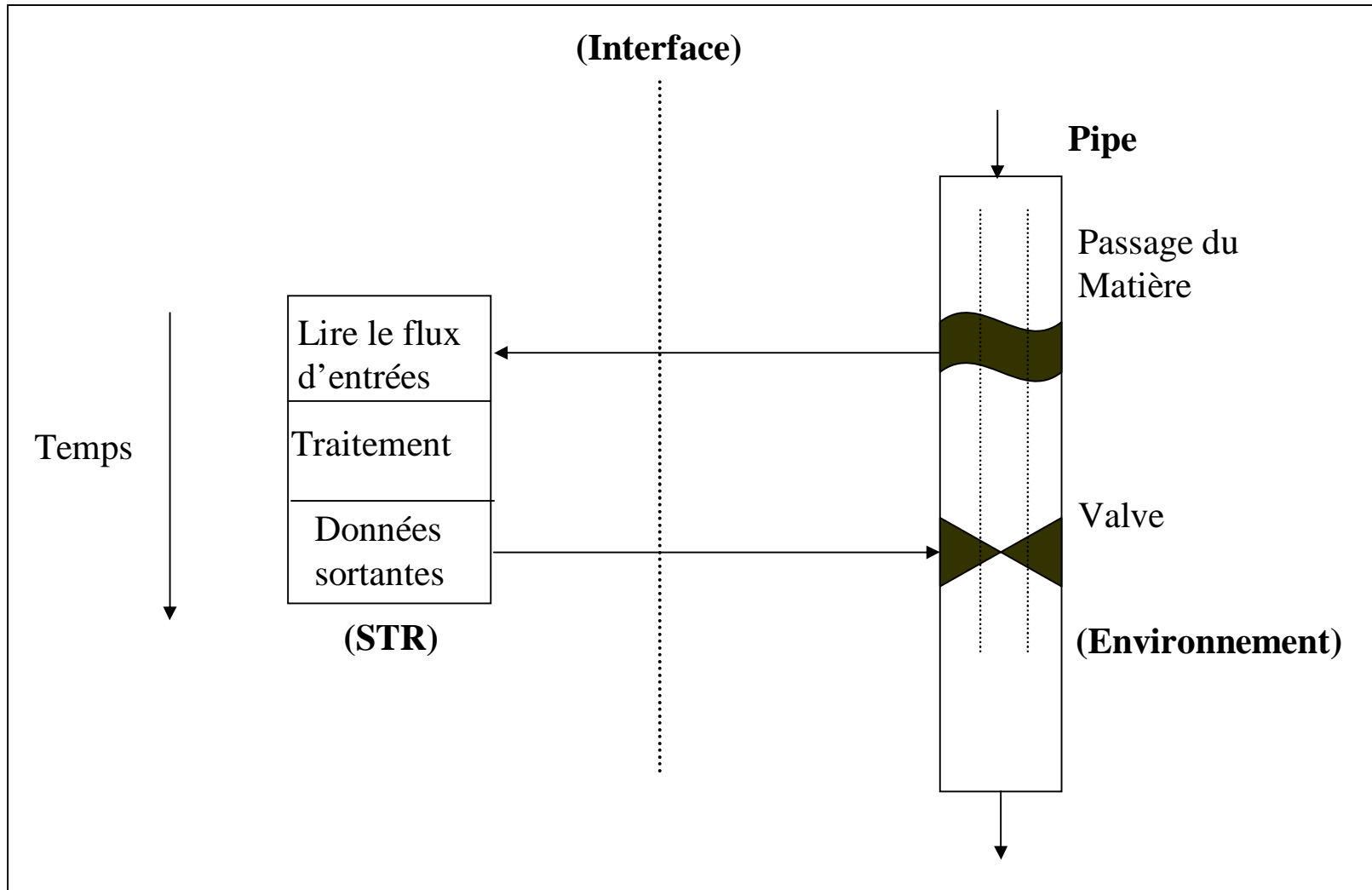
- **Critère**: gravité causée par le retard de la réponse
  1. STR **mou**: un retard dans l'obtention du résultat n'est pas dramatique (distributeur de billets)
  2. STR **ferme**: un retard, s'il arrive très peu souvent, peut être toléré (téléphonie)
  3. STR **Dur**: un retard dans l'obtention du résultat le rend inutile (détection de missile)

# Domaines d'applications des STR

## Domaines divers:

- contrôle d'usines chimiques, de centrales nucléaires
- contrôle de procédures de production complexes
- applications dans l'automobile
- systèmes de contrôle de vol, de circulation des trains
- télécommunications
- robotique
- médecine
- réalité virtuelle
- etc...

# Exemple: Contrôle de pipe



# Problèmes traités pour les STR

- **A\** Fonctionnement: (Allocation de ressources)  
partage des ressources, ordonnancement des tâches, tolérance aux fautes,...
- **B\** Matériel et OS appropriés: (Architecture)  
caractéristiques du processeur, du système de communication, des Entrées/Sorties
- **C\** Conception et développement:  
Cahier des charges, spécification de besoins et des objectifs, modélisation, implémentation, et validation.

# Fonctionnement: Allocation de ressources

- **Problème principal** : l'ordonnancement
  - partage de temps de processeur entre les tâches  
Livraison (*dispatch*) / suspension (*preempt*)
- **Objectif** de l'ordonnancement : assurer le respect des échéances *de manière prouvable*  
(*gestion de tâches et de temps*)
- Choix d'une **méthode d'ordonnancement** est caractérisée par:
  - Ordonnançabilité (Scheduling): (online/offline)



# Matériel et OS appropriés (1)

- Généralement,  
Un choix d'une **architecture** (matériel + OS)  
pour les applications temps réel offre:
  1. temps d'exécution des instructions
  2. opérations avec la mémoire
  3. traitement des interruptions
  4. adaptabilité
  5. ...

# Matériel et OS appropriés (2)

## Extensibilités et particularités:

Tolérance aux fautes

Support pour des communications rapides et prédictibles

Offrir des outils appropriés: (ex. parallélisme, pas de processeurs superscalaires etc.)

# Matériel et OS appropriés (3)

- Concevoir et usage des OS pour les algorithmes d'ordonnancement (préemption rapide, contextes multiples, priorité etc...):
  1. gestion des interruptions,
  2. Synchronisation d'horloge,
  3. ...etc.

# Conception et développement (1)

## A. Spécification des besoins

- Aspect logique : résultats ordinaires (calcul)
- Aspect temporel : contraintes temporelles
- Précision des deux aspects précédents (avant l'implémentation)

# Conception et développement (2)

## **B.** Choix et extension de langage:

1. pour gérer le temps et l'allocation de ressources,
2. permettant la communication et la concurrence,
3. support de l'ordonnançabilité à la compilation

# Conception et développement (3)

## C. Environnement d'exécution:

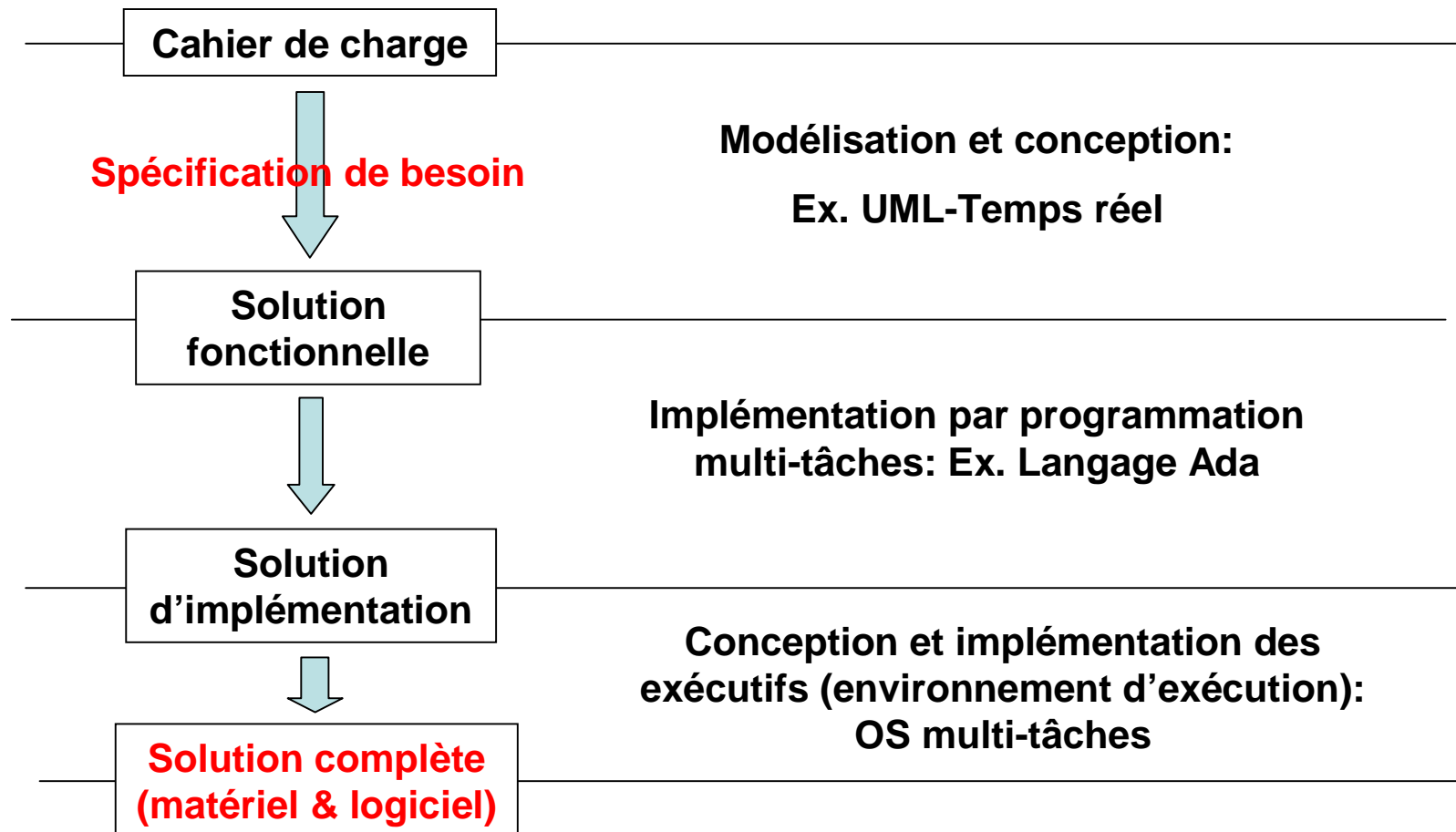
Concevoir et implémenter les exécutifs: système d'exploitation temps réel

### Objectif:

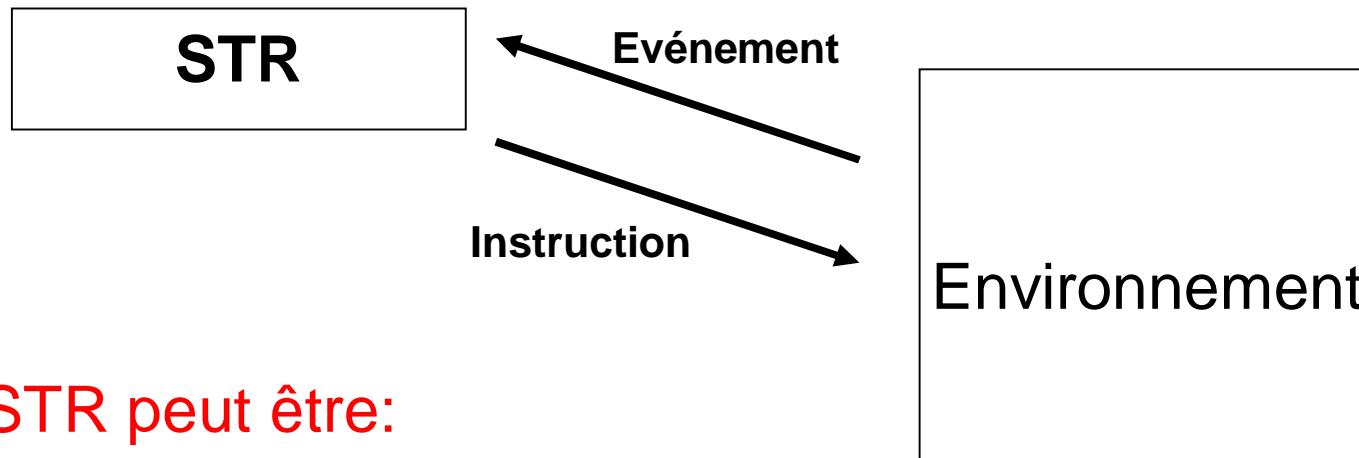
- Gestion des tâches (création, ordonnancement)
- Gestion des synchronisations (précédences + partage de ressources)
- Gestion des communications (ex : messages)
- Gestion de la mémoire
- Gestion du temps: délai, activation périodique
- Gestion des périphériques

# Conception et développement (4)

## Synthèse



# STR vs Environnement



- **STR peut être:**
- Ordinateur (monoprocasseur)
- Système multiprocasseur à mémoire commune
- Système réparti
- Automate programmable
- Circuit spécifique (Boite noire)



# Fonctionnement d'un STR

- **Fonctionnement général** : boucle infinie

**Tant que** TOUJOURS **faire**

Acquisition des entrées (données, mesures...)

Calcul des ordres à envoyer au processeur

Émission des ordres

**Fin tant que**

- Il existe deux modes de fonctionnement :

A. fonctionnement **périodique** (cyclique)

B. fonctionnement **apériodique** (événementiel)

N.B. On peut avoir un fonctionnement mixte : traitements périodiques et apériodiques (en dépendance des événements)

# Fonctionnement périodique

## Principe:

Consultation périodique de la mémoire des entrées:  
üéchantillonnage des entrées sur l'horloge du système  
üactivation du système à chaque top d'horloge

**A chaque** top d'horloge **faire**

Lecture de la mémoire des entrées

Calcul des ordres à envoyer au processeur

Émission des ordres

**Fin**

# Fonctionnement apériodique

**Principe:** activation du système à chaque événement (Interruption)

**A chaque interruption faire**

Lecture de l'information arrivée

activation du traitement correspondant

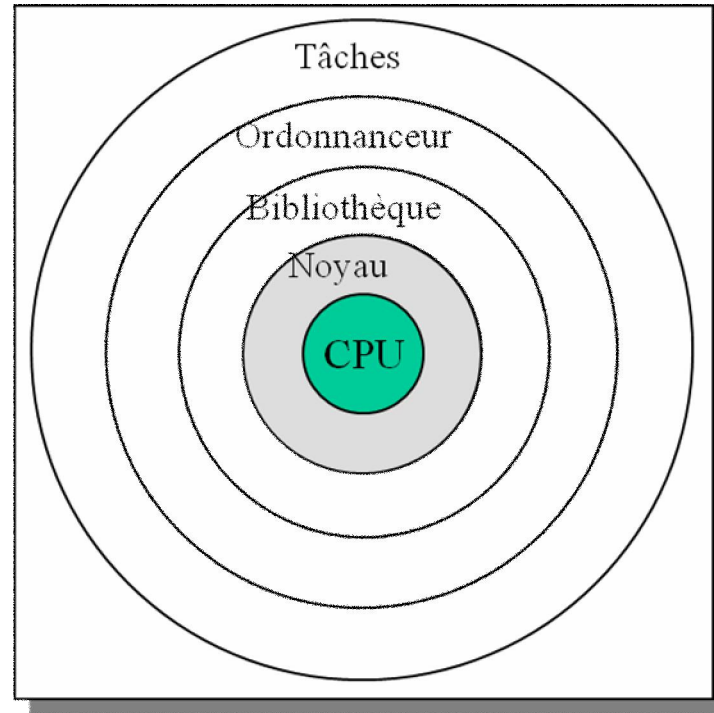
Émission des ordres issus de ce traitement

**Fin**

**Problème :** que faire si une interruption survient alors que le système est en train de traiter une interruption précédente ?

- => notion de priorité des interruptions
- => notion de "tâche" associée à une ou plusieurs interruptions
- => mécanisme de préemption et de reprise de tâche
- => gestion de l'exécution concurrente des tâches (ordonnancement)

# Organisation générale



**Conclusion:** Un système temps réel est souvent un système multitâche incluant un gestionnaire de tâches (Ordonnanceur)

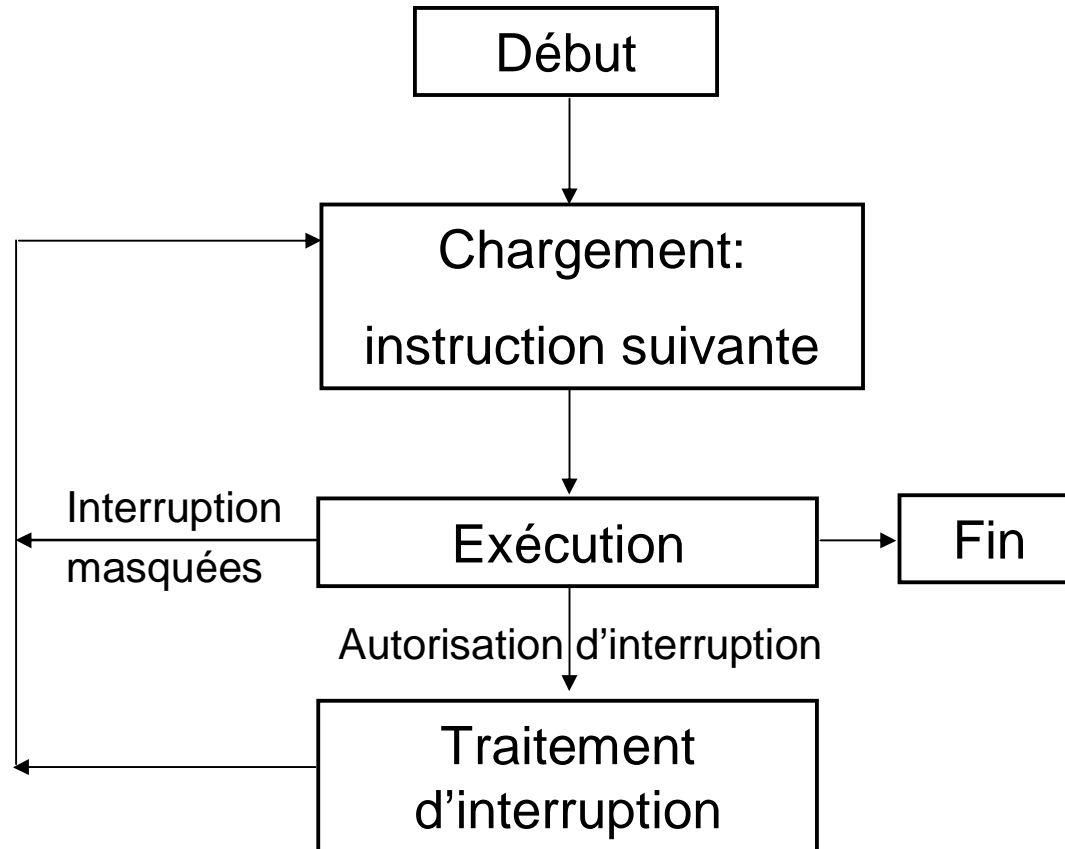
# Sys. Exp. Temps Réel (1)

- SE (**définition**): ensemble processus (activités concurrentes et/ou coopérantes).
- exécuter plusieurs processus de façon quasi-simultanée, (ex. en temps partagé)
- **Objectifs**:
  - Gestion de ressources
  - Allocation du processeur
  - Gestion de fichiers
  - Support des entrées-sorties
  - ...

# Sys. Exp. Temps Réel (2)

- Sollicitation périphériques-processeur (3 manières):
  1. Polling: test périodique du périphérique,
  2. Par DMA (Direct Memory Access): communication directe entre deux périphériques (Charger un contrôleur de DMA et libérer le processeur),
  3. **Interruptions**: interrompre le processeur lors d'une sollicitation d'un périphérique.

# Traitement d'interruption



# Tâche (processus) (1)

## Systeme d'exploitation:

1. alloue les ressources aux processus (tâches) (mémoires, temps processeur, entrées/sorties),
2. assure un fonctionnement par isolation.

## Programme:

Ensemble d'instructions.

## Processus (tâche):

1. Entité dynamique composée de:
  - a. Programme (composante statique)
  - b. Données, contexte d'exécution (composante dynamique)
2. Descripteur : structure de données regroupant les informations sur un processus (nom, contexte d'exécution, ...)



# Tâche (processus) (2)

notion dynamique (temps: début et fin).

Deux types de processus :

1. Immédiat: exécution lors de l'arrivée d'une interruption,
2. Différé: ordonnancement par le hard ou le soft (pas d'interruption).

# Cycle de vie d'un processus (Etats) ?

Remarque:

- 1 processus immédiat peut interrompre 1 processus différé ou 1 processus immédiat (notion de priorité):  
—> notion de **répartition** de processus (changement d'état) et **ordonnancement**:

