

- Comprendre les principes des OS temps réel
- Comprendre le code d'un OS temps réel :
  - concepts clefs :
    - interruptions, préemption, niveaux de priorité
    - commutation de tâches, sauvegarde de contexte
    - ordonnancement des tâches,
    - outils de communications entre tâches
  - maîtriser outils informatiques incontournables en langage C
    - pointeurs
    - allocation dynamique de mémoire
    - structures
    - listes chaînées, et listes doublement chaînées.
    - niveaux d'execution
    - compilation conditionnelle
- Être capable d'implémenter une solution temps réel sur cible :
  - Etre capable d'écrire son propre OS simplifié sur petite cible
  - Utiliser un OS temps réel multitâches (FreeRTOS)
  - Etre capable de porter FreeRTOS sur une cible inconnue
  - Améliorer ses capacités multitâches en Arduino et MBED

# Operating System Système d'Exploitation

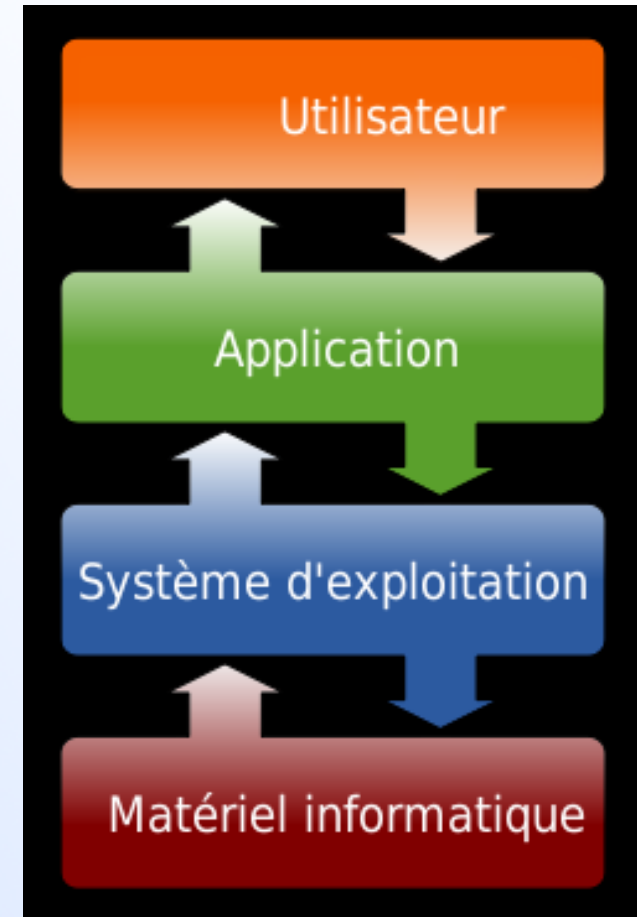
Définition (Wikipedia) : intermédiaire entre les logiciels applicatifs et le matériel.

Du cotés application : il propose une interface de programmation (API : Application Programming Interface).

Les applications sont donc dédiées à un OS ou tournent sur une Machine Virtuelle qui, elle, dispose d'une implémentation sur l'OS

Du cotés matériel : il coordonne et optimise les ressources matérielles.

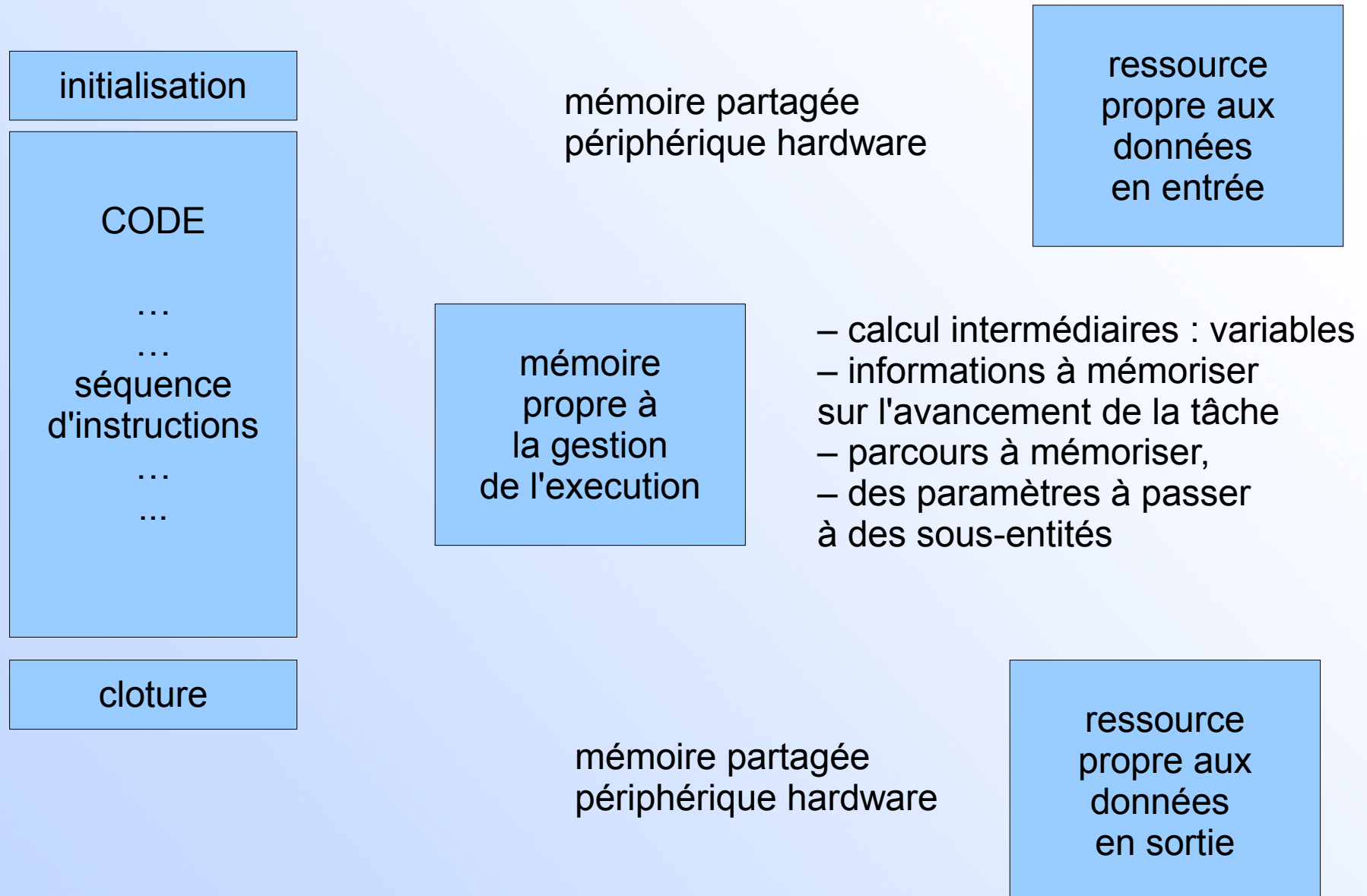
L'accès au matériel est contrôlé par l'OS dans certains cas.



# Mots clefs pour faire tourner un OS sur une cible

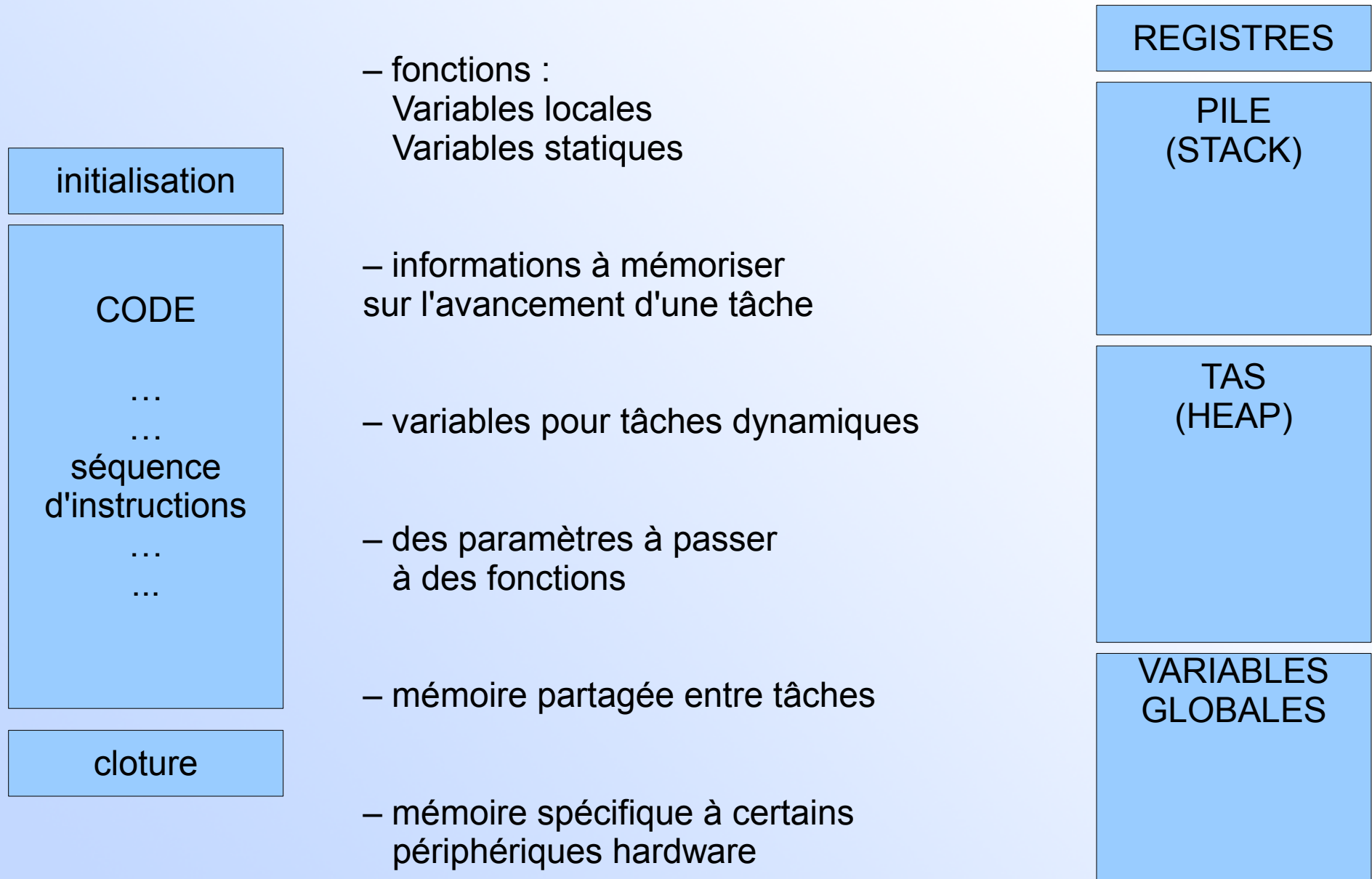
- Cible ?      Multi processeurs, Multi coeurs ?
- Monotâche / Multitâches?
- Système Collaboratif      ou      Système Préemptif ?
- Système Temps Réel ?

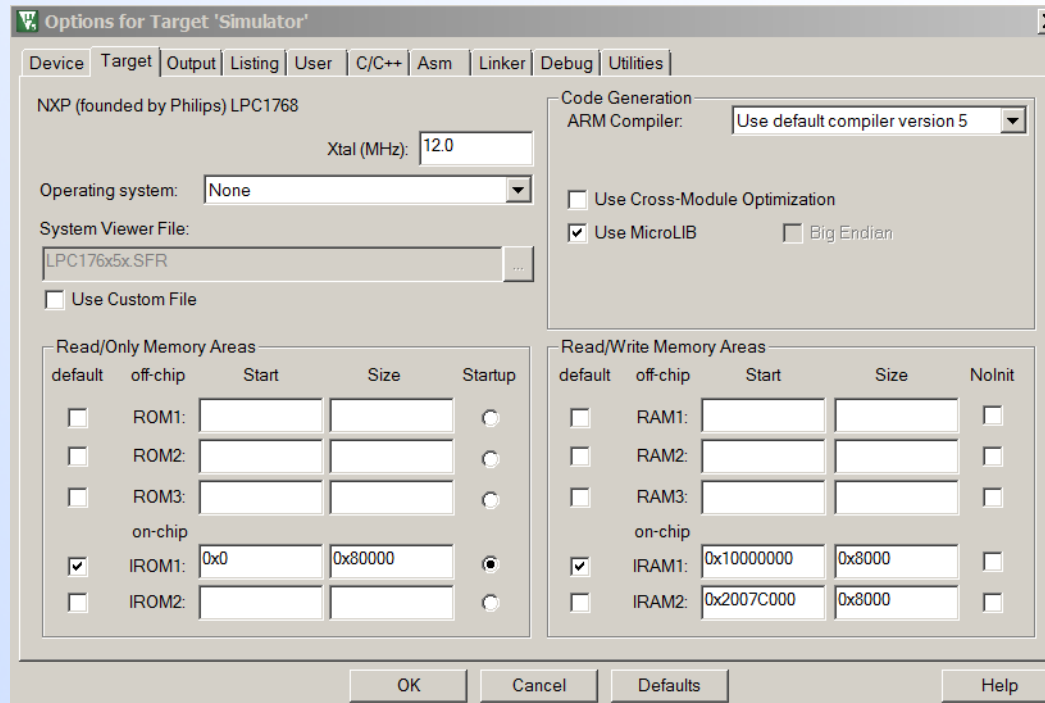
Unité d'exécution qui va réaliser une séquence d'instructions, en vue d'atteindre un résultat.



# Rappel d'organisation Mémoire

Différents éléments vont nécessiter de la mémoire





Dans un projet informatique, le fichier .MAP donne la cartographie mémoire utilisée mise en place par l'édition des liens :

- toutes les fonctions sont implémentées à une adresse et occupent de la mémoire

le nom de la fonction est en fait une étiquette à appeler

- toutes les variables occupent de la place,

leur nom est une étiquette pour les trouver

**ETIQUETTES ABSENTES DU MAP : MACRO, variables locales**



# Expression des besoins de temps réel

- Répondre à un événement en un temps donné (latence max)

Temps de réponse borné de façon absolue : Contrainte DURE ou CRITIQUE

- le non respect du temps donné engendre la disparition d'une information

Temps de réponse moyen : Contrainte DOUCE ou SOUPLE

- le non respect engendre un sentiment de non réactivité

- Effectuer certaines opérations à un instant donné

Contrainte d'agenda

- la non disponibilité au bon instant engendre un jitter de mesure ou d'action

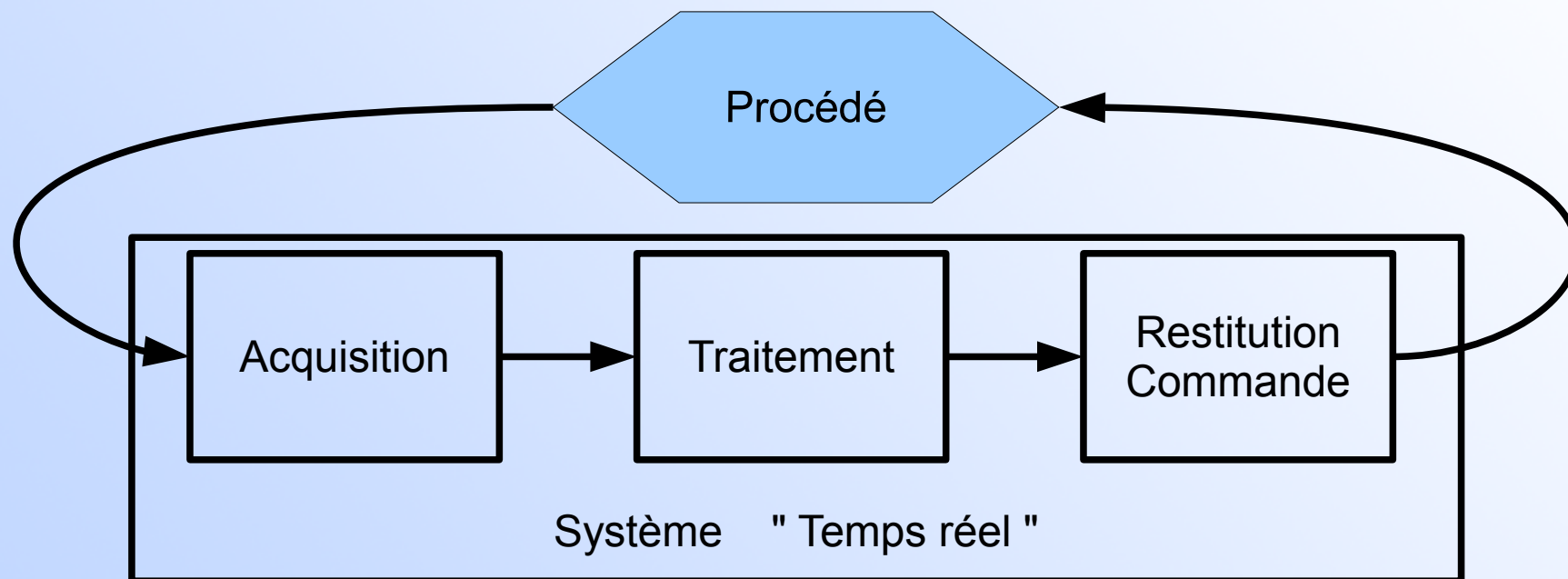
- Traiter un nombre donné d'événements par unité de temps

Contrainte de flux entre un flux entrant à transformer en un flux sortant

- l'incapacité de tenir un débit engendre un décalage grandissant entrée/sortie, se traduisant par un retard croissant jusqu'à explosion périodique du stockage

Système capable de contrôler (ou piloter) un procédé physique à une vitesse adaptée à l'évolution du procédé contrôlé :

- Réponse aux événements ou aux signaux dans un délai prévisible et garanti après leur occurrence. (déterminisme temporel)
- Comportement logique **déterministe** (Même cause, mêmes effets)
- Capacité à offrir le bon résultat dans le temps imparti de manière fiable.



"Un système est dit temps réel si l'information après acquisition et traitement reste encore pertinente"



C'est le délai qui s'écoule entre un évènement et la réaction perceptible.

Latences classiquement acceptables:

Mesure de la température <1000 ms

Bouton poussoir <100 ms

Simple contrôle moteur <10 ms

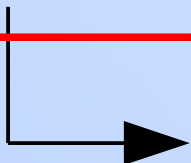
Application de contrôle multi-axes simple <1 ms

Asservissement de haute précision <10  $\mu$ s

Emulation d'un périphérique non existant <1  $\mu$ s

L'application définit les exigences temps réel souhaitées:

Temps réaction + Temps de traitement < Latence souhaitée



Retard à l'acquisition , Retard d'appel du traitement

# Temps de réaction techniques d'acquisition

Criticité du temps de réaction au niveau acquisition, sauvegarde?

- En cas de nécessité de datation précise d'un évènement, d'une durée.
- En cas de disponibilité éphémère de l'information

- Scrutation volontaire aléatoire ou périodique

- Génération d'une interruption liée à l'évènement ...prioritaire ?  
suivi d'une scrutation volontaire (capture manuelle)

- Action directe d'un périphérique (capture automatique)

# Temps de réaction appel du traitement

Criticité du temps d'appel du traitement :

- En cas de nécessité d'une réaction rapide après traitement
- En cas d'écrasement proche de l'information par l'information suivante si un buffer circulaire n'a pas été mis en oeuvre lors de l'acquisition

Comment on prévient qu'il faut agir?

Conséquences de la fréquence de la sollicitation ?

- Traitement lors du processus d'acquisition : regroupement
- Levée d'un Sémaphore lors de l'acquisition, lui même scruté plus tard
- Transfert automatique par un périphérique dédié (DMA) qui préviendra du lot à traiter : (interruption ou Flag hardware à scruter)

Une tâche est généralement caractérisée par :

- Un temps de calcul ( $C_i$ ),
- Une échéance ( $E_i$ ) qui est la date à laquelle la tâche doit avoir terminé son exécution,
- Une période ( $T_i$ ) pour les tâches périodiques, durée séparant ses instants d'activation.

Comme une tâche peut être interrompue par d'autres tâches,  
Pour respecter l'échéance  $E_i$ , il faut la démarrer en  
anticipation ...

en tenant compte du cas le plus défavorable ...

...interrompu ou retardé par les autres tâches ...

on peut calculer de combien il faut anticiper

## La Tâche est alors réalisable ou pas réalisable

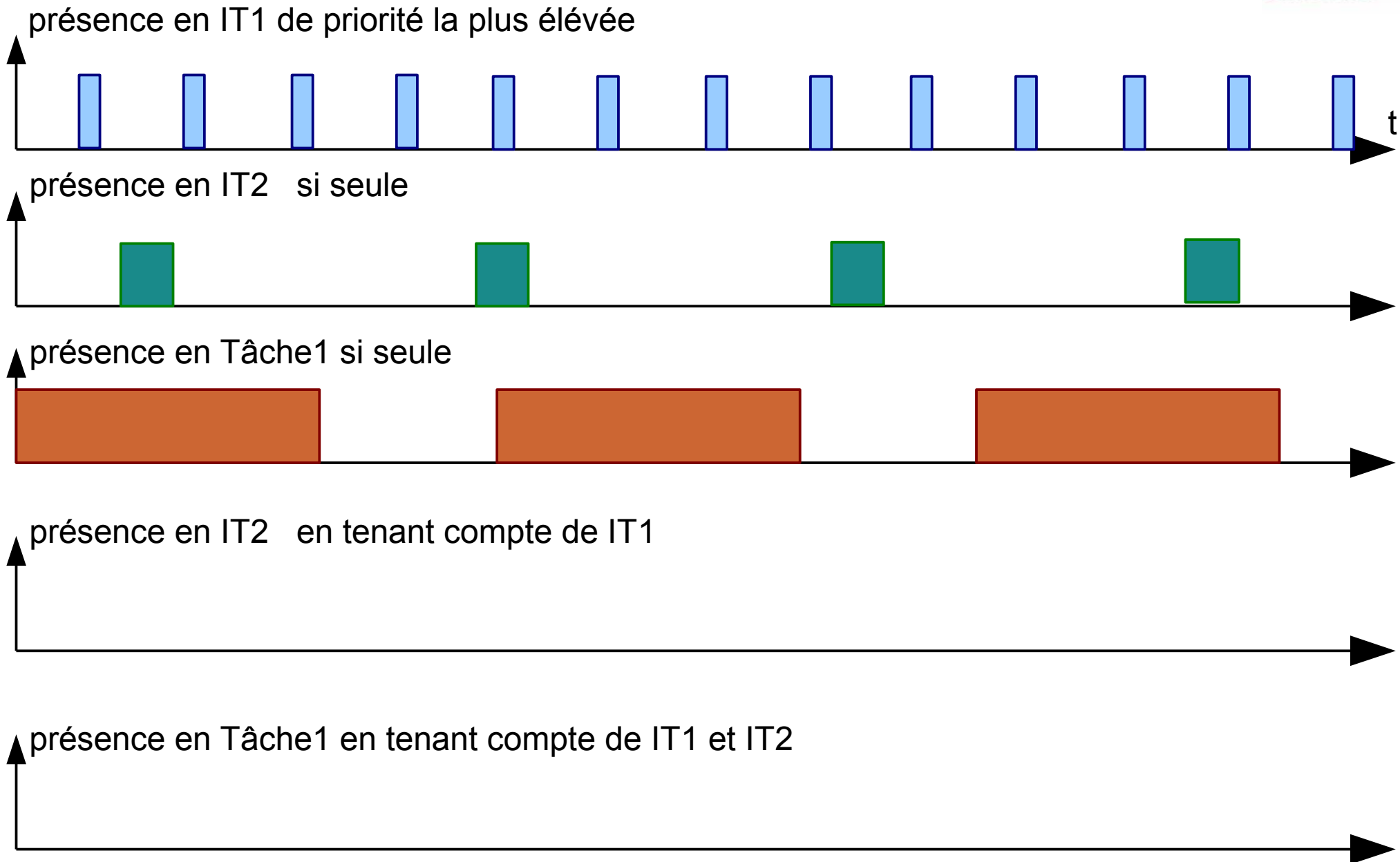
Le taux de charge est le taux d'occupation du processeur à réaliser les missions imparties: pourcentage du temps occupé à gérer un certain nombre de tâches sur une fenêtre temporelle.

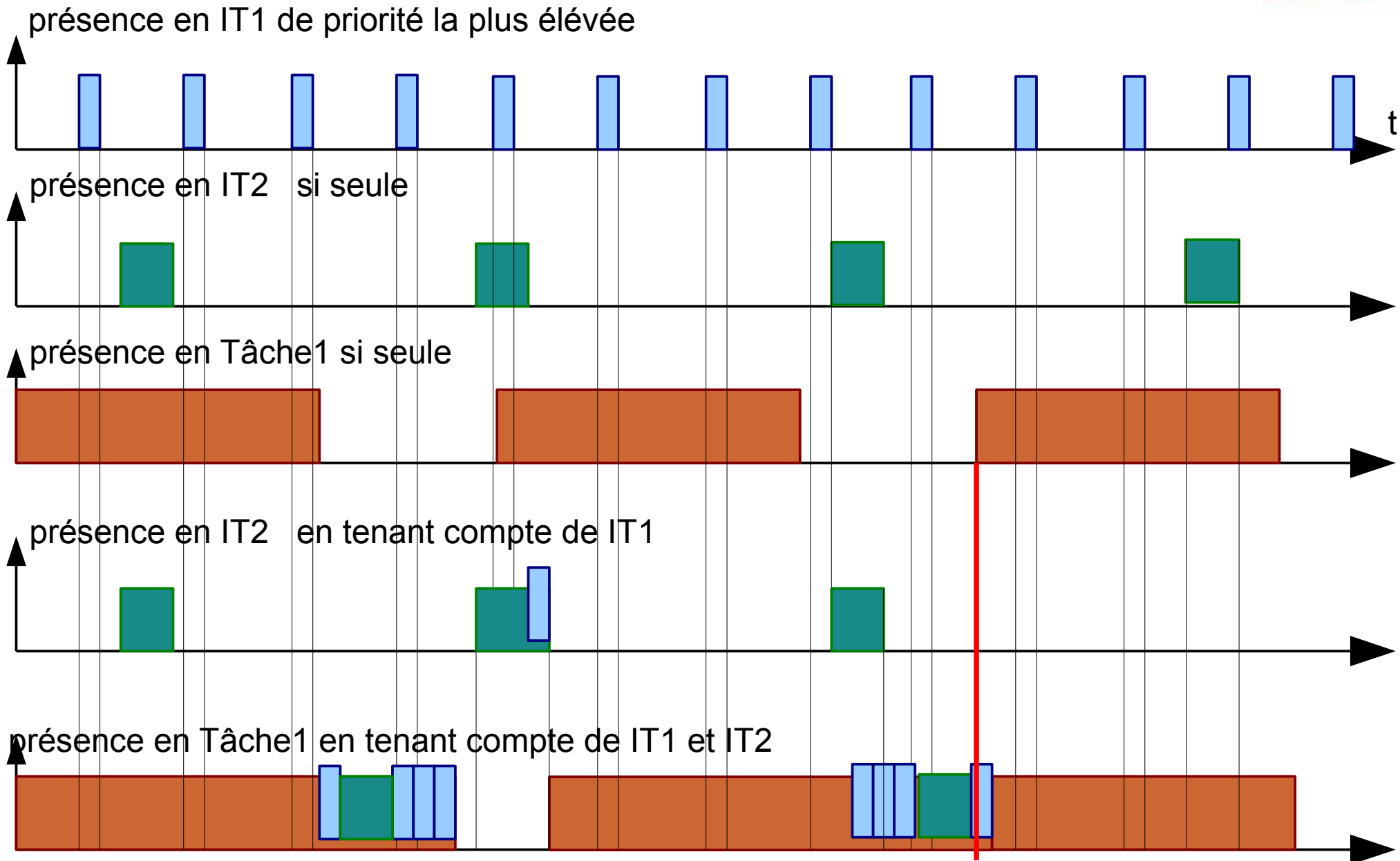
Ce taux de charge fluctue au cours du temps, d'une amplitude plus ou moins forte en fonction de la taille de la fenêtre choisie et des tâches qui peuvent apparaître une ou plusieurs fois dans cette fenêtre...

S'il atteint 100%, cela veut dire que le processeur n'arrive pas à traiter les tâches confiées sur la durée de la fenêtre considéré

Quand la fenêtre tend vers une "grande" durée, la fluctuation de ce taux de charge se réduit, et on parle de taux moyen.







Temps réel souple : condition de respect du temps moyen

- Adapté aux systèmes non critiques
- Débordement temporel acceptables au prix d'une perte de qualité
- Le système doit rattraper une échéance non respectée,  
(sinon il ne tient pas le temps réel, car il accumule les retards)
- Un retard ne crée pas d'erreur fatale, de mise en danger

Exemples : Multimédia sur Plate-forme généraliste. Déterminisme temporel faible à cause du matériel et du logiciel :

PC + OS + Débit Internet = Débits variables et difficiles à estimer

# Mots clef : Multitâches sur un processeur simple coeur

## Ordonnanceur :

- Coordonne l'utilisation du processeur.
- Attribue un ordre d'exécution aux tâches



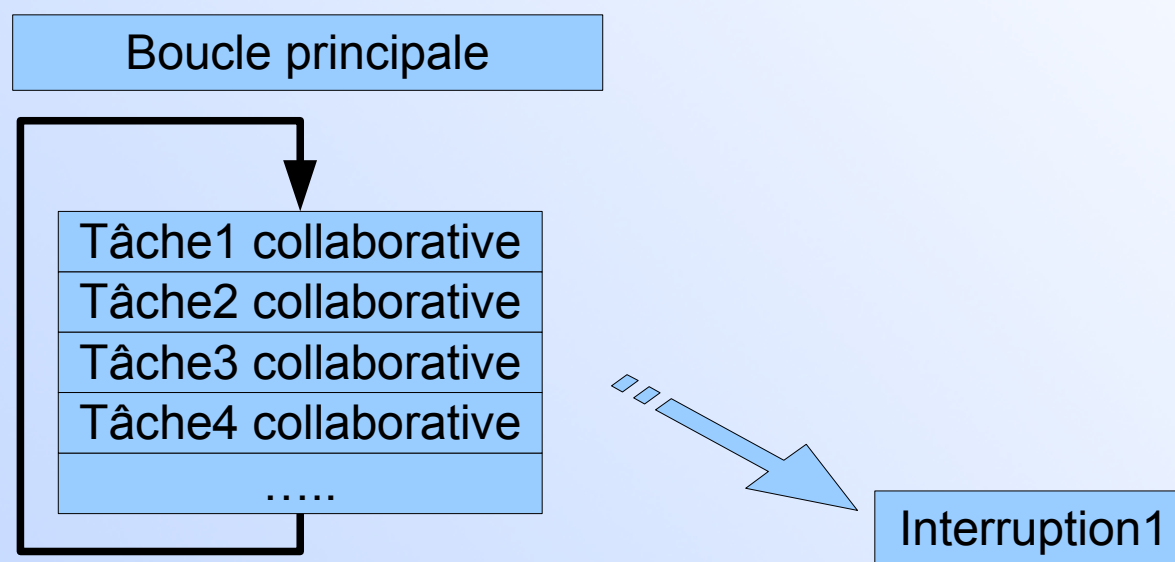
Mettre en place 3 clignoteurs indépendants :

1 Rouge 1Hz (0,3s ON 0,7s OFF)

1 Vert 2,5Hz (0,2s ON 0,2s OFF)

1 Bleu qui commute entre 1, 2, 3, 4 Hz

à chaque appui d'un bouton scruté à 10Hz



aucune tâche n'est bloquante