UNIVERSITÉ
CÔTE D'AZUR

Introduction aux Systèmes et Logiciels Embarqués

Présentation: **Stéphane Lavirotte**
Auteurs: ... et al*

(*) Cours réalisé grâce aux documents de :
Christophe Blaess, Loïc Cuvillon, Pierre Ficheux,
Stéphane Lavirotte, Iulian Oiber, Thomas Petazzoni

Mail: Stephane.Lavirotte@univ-cotedazur.fr
Web: http://stephane.lavirotte.com/
Université Côte d'Azur

UNIVERSITÉ
CÔTE D'AZUR

Introduction aux Aspects Temps Réel dans les OS

Microprocesseurs

UNIVERSITÉ CÔTE D'AZUR

Problématique du Temps Réel

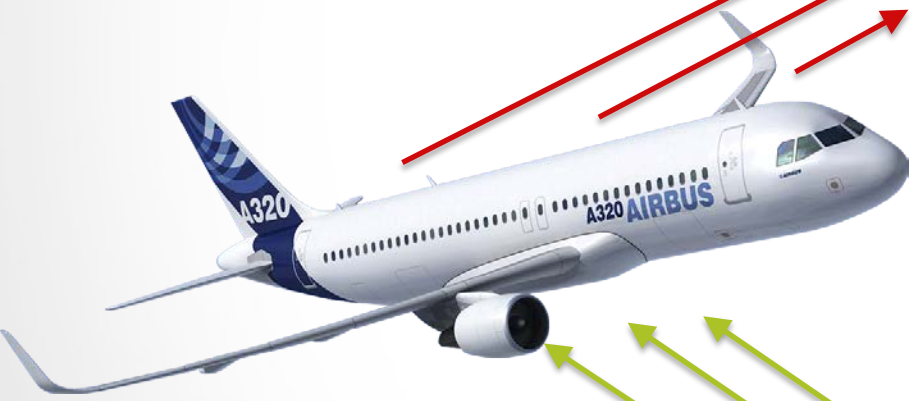
- ✓ Simple: 1 tâche périodique
  - Pas besoin de système d'exploitation
  - La tâche est exécutée à chaque interruption d'une horloge
  - **Temps Réel si: Temps d'exécution < Période de la tâche**
- ✓ Complexe: multitude de tâches en parallèle, de périodes différentes
  - Tâche de supervision, d'archivage, de l'IHM, du réseau, d'accès aux supports de stockage, d'asservissement, ...

⇨ Problème de concurrence pour l'accès au CPU, la mémoire, aux bus de données, ...

⇨ Politique d'ordonnancement pour assurer les échéances des tâches

UNIVERSITÉ CÔTE D'AZUR

Exemple de Système Temps Réel



Acquisition des données capteurs

Calcul de la réaction par un contrôleur temps réel (matériel + logiciel)

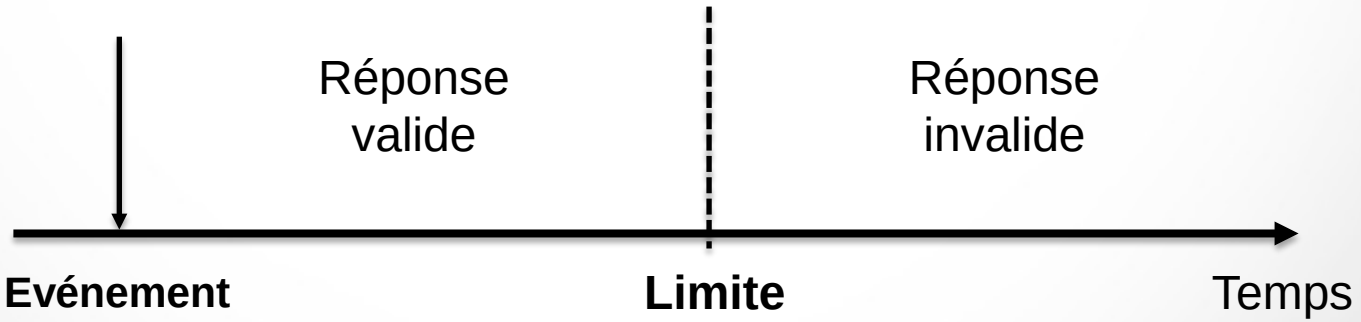
Application des commandes sur les actionneurs

Environnement

UNIVERSITÉ CÔTE D'AZUR

Définition Système Temps Réel

- ✓ Définition générale
  - Système (matériel et logiciel) qui satisfait aux contraintes fonctionnelles et temporelles qui lui sont imposées
- ✓ Répondre à des stimuli extérieurs
  - En prenant en compte l'écoulement du temps
  - Indépendamment du flux d'instructions traité par le processeur
- ✓ Notion de contrainte temps réel:

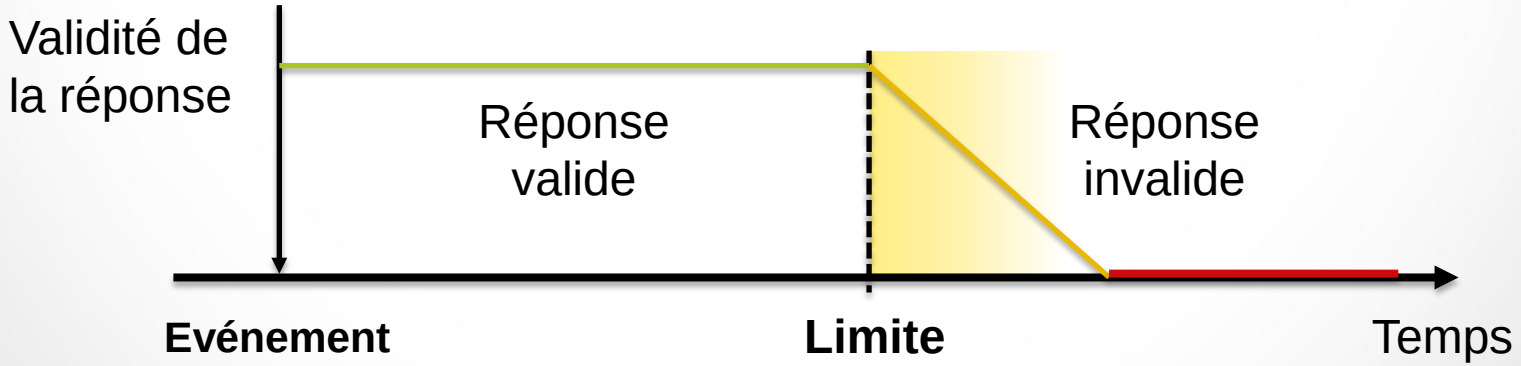


- ✓ Différents classes de systèmes temps réel

UNIVERSITÉ CÔTE D'AZUR

Temps Réel mou ou souple (Soft Realtime)

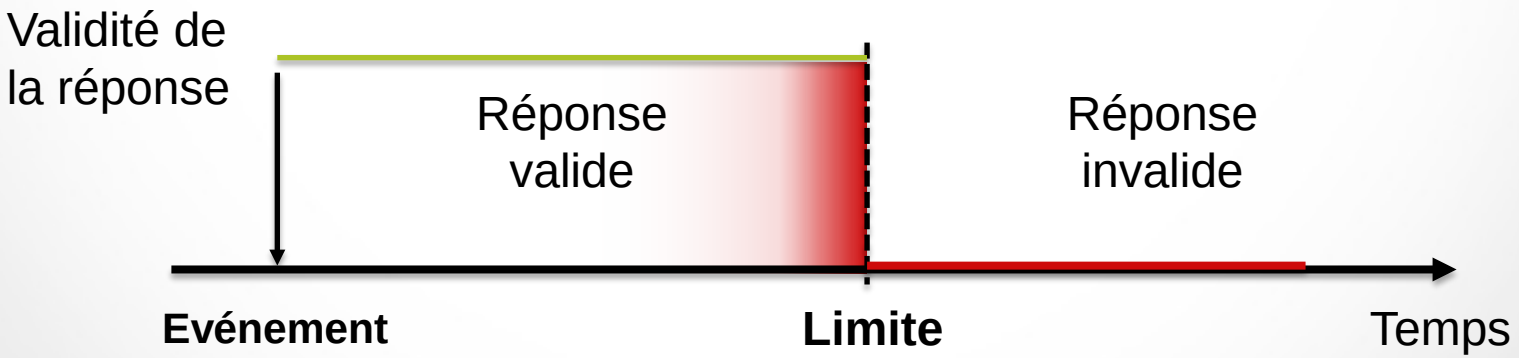
- ✓ **Définition**
  - Système devant satisfaire des échéances temporelles mais avec un degré de flexibilité, de tolérance sur l'échéance
- ✓ **La majorité des systèmes informatiques sont de ce type**
  - Un lecteur mp3
  - Un lecteur vidéo (variation possible du framerate: fps)



UNIVERSITÉ CÔTE D'AZUR

Temps réel dur ou critique (Hard Realtime)

- ✓ **Définition**
  - Système devant respecter des échéances avec une tolérance nulle
  - Utilité nulle des résultats obtenus après l'échéance
- ✓ **Pénalité élevée: panne du système + danger pour l'intégrité physique du système ou des êtres humains**



Le critère entre temps réel dur et mou est la pénalité en cas de non-respect de l'échéance

UNIVERSITÉ CÔTE D'AZUR

Système Temps Réel dur (Hard Realtime)

- ✓ Doit respecter des limites temporelles données même dans la pire des situations d'exécution possibles

- ✓ Majoritairement des tâche périodiques
  - Tâche de régulation et d'avertissement d'un système physique
    - Tout retard déstabilise l'asservissement
  - Tâche de supervision

- ✓ Application du temps réel dur
  - Aéronautique/spatial (pilote automatique, ...)
  - Automobile (ABS, ESP, Airbag, ...)
  - Robotique (perception environnement, contrôle des moteurs)
  - Processus industriel (centrales nucléaires, ...)

UNIVERSITÉ CÔTE D'AZUR

Autre Classification de Systèmes Temps Réel

- ✓ Temps Réel mou ou souple (*soft realtime*)
  - Organisation sous forme de priorités entre les tâches applicatives
  - L'ordonnanceur choisit une tâche applicative dont la priorité est la plus élevée parmi les tâches prêtes
- ✓ Temps réel strict non certifiable
  - Mécanisme d'ordonnancement visant à approcher le temps réel strict certifiable mais sans garantie
- ✓ Temps Réel dur ou strict certifiable (*hard realtime*)
  - De légères fluctuations (centaines de nano secondes) pourront intervenir dans les temps de réponses aux événements extérieurs mais toujours **prédictible** et **borné**
- ✓ Temps Réel absolu:
  - Temps parfaitement connu, identique stable et donc prédictible (système à base de FPGA par exemple)

UNIVERSITÉ CÔTE D'AZUR

Prédictibilité

- ✓ « LA » caractéristique requise avant tout d'un système temps réel:
  - La prédictibilité de la réponse du système
- ✓ « Définition »:
  - sous un ensemble d'hypothèses concernant la charge (*e.g.*, fréquence des entrées) et les erreurs non-contrôlables (*e.g.*, fréquence des erreurs de bit sur le bus), arriver à prouver que:
    - toutes les contraintes (notamment les délais) sont respectées
    - au moins pour les tâches critiques du système
- ✓ Exemple: Que préférez-vous pour votre sécurité ?
  - un ordinateur de bord qui déclenche l'ABS à 1ms du blocage des roues dans 99% de cas ?
  - ou un qui le déclenche à 10ms, mais dans 100% de cas ?

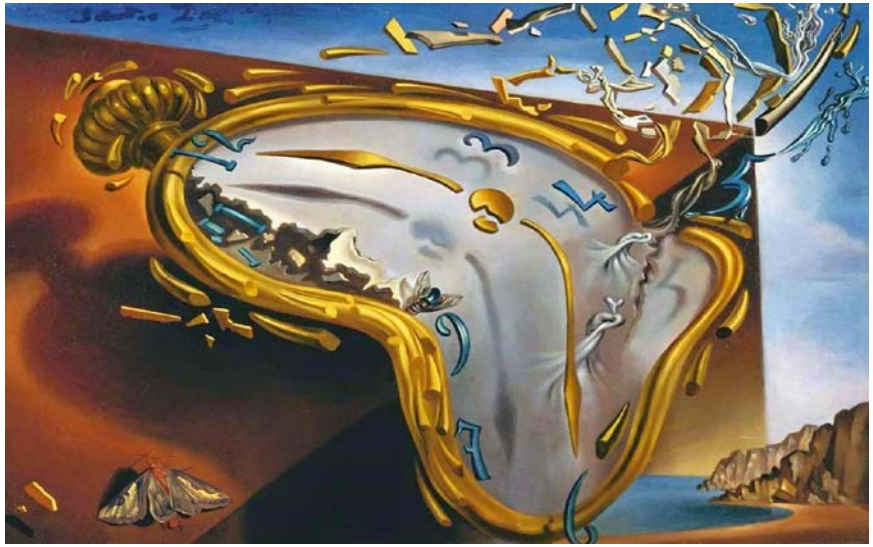
UNIVERSITÉ CÔTE D'AZUR

Récapitulatif

- ✓ **Donc un système temps réel**
  - n'est pas forcément un système « qui va vite »
  - mais un système qui satisfait à des contraintes temporelles

Type de Temps réel	Temps de réponse	Exemple sur Raspberry Pi
Souple	Non garanti Indépendant des activités moins prioritaires en espace utilisateur	Linux Vanilla
Strict non-certifiable	Non garanti Indépendant des activités moins prioritaires du système	Linux PREEMPT_RT Xenomai
Strict certifiable	Fluctuant mais borné Prédictible	RTEMS
Absolu	Fixe donc Prédictible	Non

UNIVERSITÉ
CÔTE D'AZUR



Temps Réel et OS

Problématique de l'Ordonnancement

UNIVERSITÉ CÔTE D'AZUR

Des malentendus fréquents sur les Systèmes Temps Réel

- ✓ **Tous ces préjugés sont faux !**
  - Système temps réel = système rapide et performant
  - Programmation temps réel veut dire assembleur, interruptions et pilotes
  - Il n'y a aucune science derrière le développement des systèmes temps réel, tout est une question de bidouillage
  - Tous les problèmes du temps réel ont déjà été résolus dans d'autres domaines de l'informatique
  - L'augmentation de la vitesse des processeurs va résoudre ce qui reste

- ✓ Par contre, un des problèmes principaux à résoudre est
  - L'ordonnancement (allocation des ressources processeur)

UNIVERSITÉ CÔTE D'AZUR

OS Temps Réel

✓ Caractéristiques

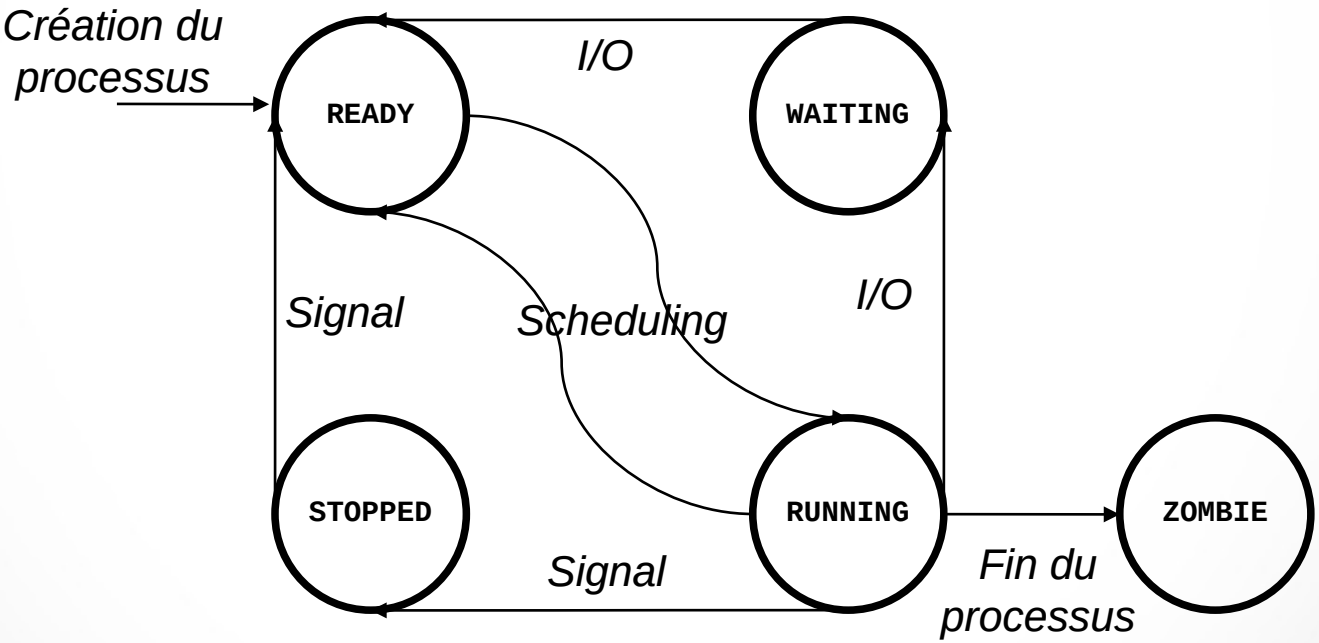
- Politique d'ordonnancement « Temps Réel »
  - Résoudre la concurrence des tâches pour le CPU
- Temps de réponse (latence) assez court pour l'application visée

✓ Usage

- N'assure pas le respect des contraintes temps réel d'un programme quelconque!
  - Exemple: inversion de priorité de PathFinder (NASA)
- Mais donne les primitives pour y parvenir si la conception et les timings ont été validés par expérience ou simulation

Ordonancement

- ✓ Ordonnanceur (*scheduler*)
  - Choix d'un processus à exécuter parmi les processus prêts



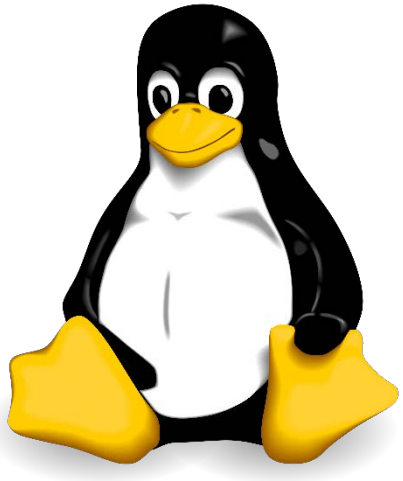
- ✓ Objectifs de l'ordonnanceur temps réel
  - Assure l'ordonnancement des tâches et leur échéance

UNIVERSITÉ CÔTE D'AZUR

Ordonnancement préemptif à priorité fixe

- ✓ Ordonnanceur préemptif à priorité fixe
  - Le plus utilisé
  - Préemption: capacité à interrompre une tâche pour une autre
  - Priorité fixe: chaque tâche a une priorité d'exécution invariante
  - Règle: exécution de la tâche de plus haute priorité non-bloquée à chaque appel à l'ordonnanceur
- ✓ Ordonnancement RM (rate-monotonic)
  - Etudié dans les années 70 (Liu & Leyland) et étendu par la suite
  - Hypothèses:
    - Tâches périodiques et indépendantes uniquement
    - Priorité P de la tâche inverse à sa période T
- ✓ Pour tout système de n tâches: $\sum_{i=1}^{n} \frac{C_i}{T_i} \leq N(2^{\frac{1}{N}} - 1) \leq 1$

UNIVERSITÉ
CÔTE D'AZUR



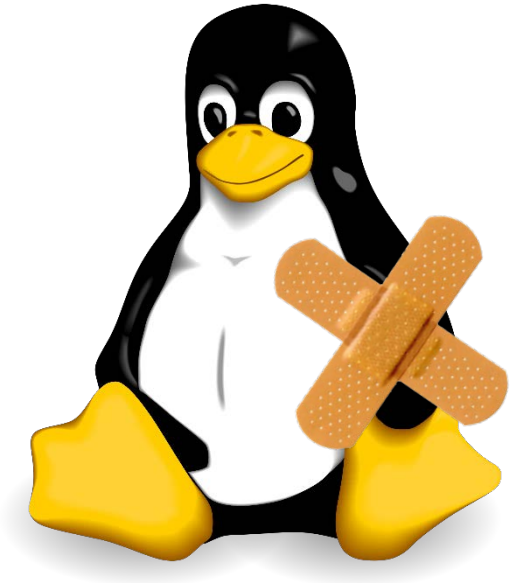
Linux et le Temps Réel

Linux un OS Temps Réel ???

UNIVERSITÉ CÔTE D'AZUR

Linux Embarqué et Temps Réel

- ✓ Linux est un système d'exploitation de la famille Unix
- ✓ Il a été conçu comme un système en temps partagé
  - Dans le but de fournir la meilleur cadence/flux de production en fonction du matériel disponible
  - En faisant le meilleur usage possible des ressources
  - Le déterminisme temporel n'a pas été pris en compte
- ✓ Mais les systèmes temps réel impliquent la prédictibilité (ou déterminisme) quelques soient les conditions
- ✓ Deux approches pour apporter les besoins temps réel
  1. En modifiant le noyau pour gérer les aspects temps réel (latence bornée et des API temps réels): **PREEMPT_RT**
  2. En ajoutant une couche « sous » le noyau pour lui apporter les contraintes temps réel: RTLinux, **RTAI** ou **Xenomai**



PREEMPT_RT

Approche 1

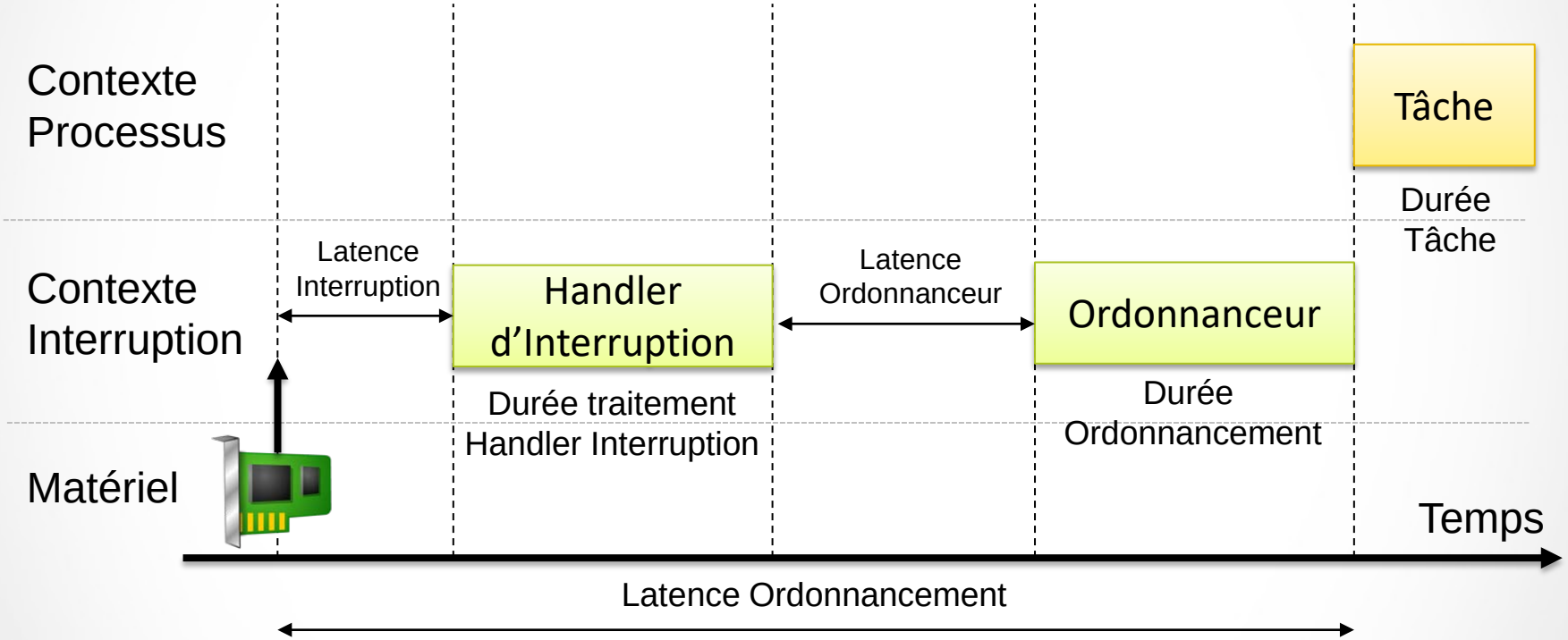
UNIVERSITÉ CÔTE D'AZUR

Comprendre la Problématique de Latence

- ✓ Scénario typique pour des applications temps réel avec un système comme Linux
  - Un événement provient du monde physique
  - Le CPU en est notifié par une interruption
  - Le handler d'interruption reconnaît et prend en charge l'événement
  - Il réveille alors la tâche utilisateur qui réagit à cet événement
  - Au bout d'un certain temps, la tâche utilisateur s'exécute et traite la réaction à l'événement du monde physique
- ✓ Le temps réel est: être capable de garantir la latence dans le pire cas pour réagir à cet événement

UNIVERSITÉ CÔTE D'AZUR

Exemple de Latence

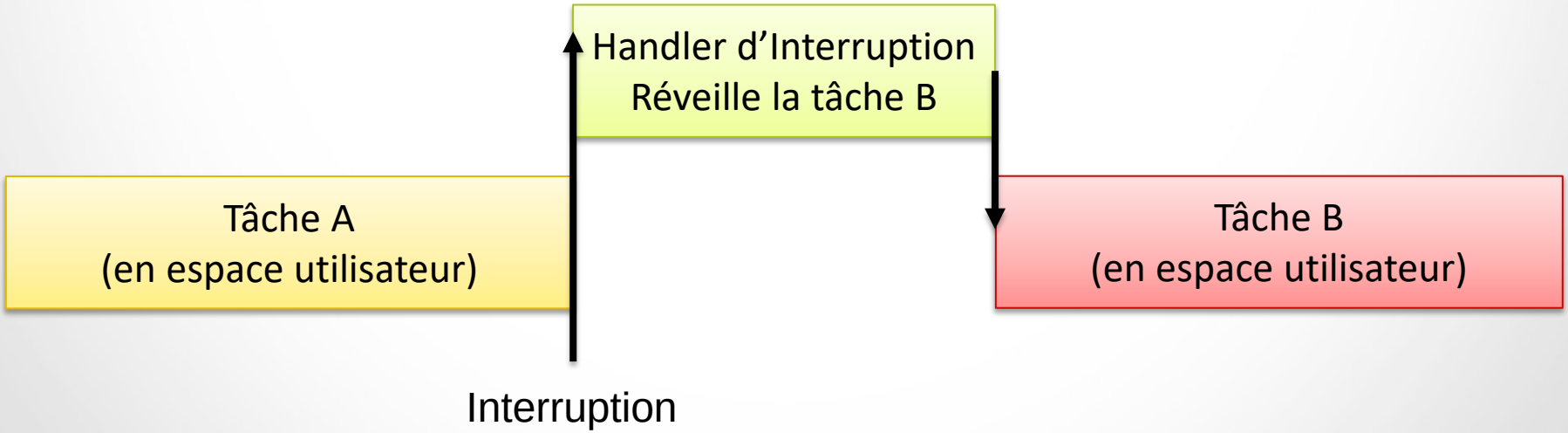


- ✓ *Latence interruption: temps écoulé avant le traitement de l'interruption*
- ✓ *Latence ordonnanceur: temps écoule avant d'exécuter l'ordonnanceur*

UNIVERSITÉ CÔTE D'AZUR

Comprendre la Préemption 1/2

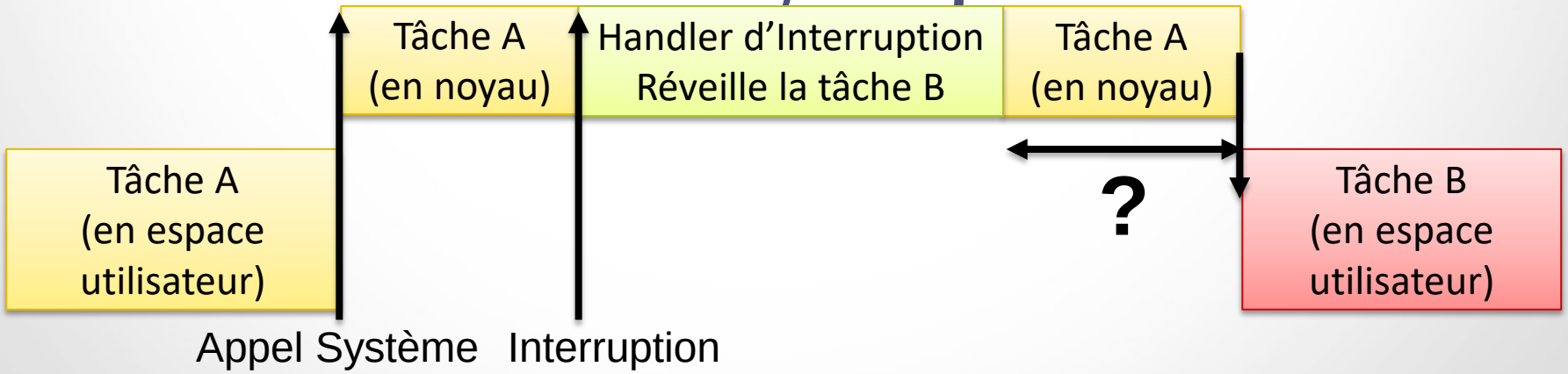
- ✓ Le noyau Linux est un système préemptif
- ✓ Quand une tâche s'exécute dans l'espace utilisateur
  - Si elle est interrompue par une interruption
  - Et si l'interruption réveille une autre tâche
  - Cette tâche peut être exécutée dès le retour du traitement de l'interruption



UNIVERSITÉ CÔTE D'AZUR

Comprendre la Préemption 2/2

- ✓ Quand une interruption arrive quand une tâche exécute un appel système
  - Cet appel système doit finir avant qu'une autre tâche soit lancée
- ✓ Par défaut, le noyau Linux n'est pas préemptif sur le noyau lui-même (seulement sur les tâches utilisateur)
- ✓ Donc le temps avant l'appel de l'ordonnanceur (pour lancer une nouvelle tâche) **n'est pas borné !**



UNIVERSITÉ CÔTE D'AZUR

Autre Mécanismes non Déterministes dans Linux

- ✓ D'autres mécanismes non déterministes dans Linux peuvent affecter les tâches temps réel

- ✓ Linux est basé sur l'utilisation de mémoire virtuelle (comme fournie par un MMU)
  - L'allocation de mémoire est faite à la demande
  - Quand une application accède pour la première fois à un code ou une donnée
  - Il y a un chargement à la demande qui peut créer des délais importants
- ✓ De nombreux services de la librairie C ou du noyau ne sont pas conçus pour des préoccupations temps réel

UNIVERSITÉ CÔTE D'AZUR

Le projet PREEMPT_RT

- ✓ But du projet:
  - Améliorer le noyau Linux pour les aspects temps réels
  - Ajouter ces améliorations au noyau principal (patch)
    - Le développement de PREEMPT_RT travaille au plus proche du développement de la branche principale du noyau
  - De nombreuses améliorations conçues, développées et débuguées dans PREEMPT_RT ont été incluses dans la branche principale du noyau
    - Le projet est un développement au long terme d'une branche du noyau qui devrait disparaître quand tout sera inclus dans la branche principale du noyau

- ✓ https://wiki.linuxfoundation.org/realtime/start

UNIVERSITÉ CÔTE D'AZUR

3 Modèles de Préemption principaux

- ✓ CONFIG_PREEMPT_NONE
  - Code noyau jamais préempté (interruption, exceptions, appels systèmes)
  - Comportement par défaut dans le noyau standard
- ✓ CONFIG_PREEMPT_VOLUNTARY
  - Code noyau peut se préempter lui-même
  - Pour des applications desktop plus réactives aux entrées utilisateur
- ✓ CONFIG_PREEMPT_RT_FULL
  - La plupart du code noyau peut être préempté à n'importe quel moment
  - Pour des systèmes embarqués avec une latence de l'ordre de quelques millisecondes

UNIVERSITÉ CÔTE D'AZUR

High Resolution Timers

- ✓ Résolution des timers liés aux « tick » réguliers horloge
  - Habituellement 100 Hz ou 250 Hz, dépendant de l'architecture et de la configuration
  - Donc une résolution de 10 ou 4ms
  - Augmenter la fréquence des tick n'est pas une bonne option car cela consommera plus de ressources pour traiter ces interruption
- ✓ Infrastructure des timers haute résolution introduite en 2.6.21
  - Permet d'utiliser les timers matériels disponibles pour programmer des interruptions au bon moment
  - Les timers matériels sont multiplexés, de telle sorte qu'un seul timer matériel est suffisant pour gérer un grand nombre de timers logiciels
  - Utilisable depuis l'espace utilisateur en utilisant l'API

UNIVERSITÉ CÔTE D'AZUR

Utilisation de PREEMPT_RT

- ✓ Distribué sous la forme d'un patch à la branche principale du noyau
  - Toutes les version de noyau ne sont pas supportées
  - Simple patch à appliquer sur les sources
- ✓ Activer les options suivantes dans le noyau
  - `CONFIG_PREEMPT_RT_FULL`
  - `CONFIG_HIGH_RES_TIMERS`
- ✓ Vous disposez alors d'un noyau Linux avec support temps réel
  - Besoin de configurer le système comme la priorité des threads d'interruption qui dépend de votre application

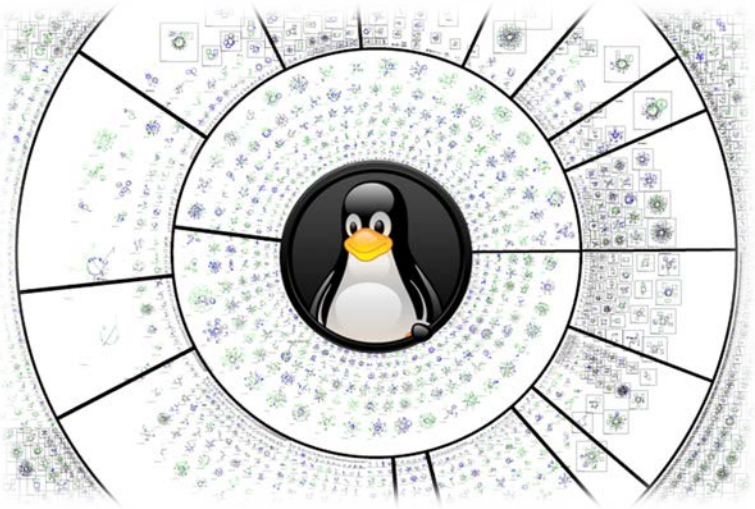
UNIVERSITÉ CÔTE D'AZUR

Développement d'Application Temps Réel

- ✓ Pas de librairie spécifique nécessaire
  - L'API temps réel POSIX fait partie de la librairie C standard
  - Les librairies C glibc et eglibc sont recommandées
    - Support temps réel dans la uClibc pas encore disponible

- ✓ Compilation d'une application
  - `gcc –o myprog myprog.c -lrt`

UNIVERSITÉ
CÔTE D'AZUR



Extension Temps Réel au Noyau Linux

Approche 2

UNIVERSITÉ CÔTE D'AZUR

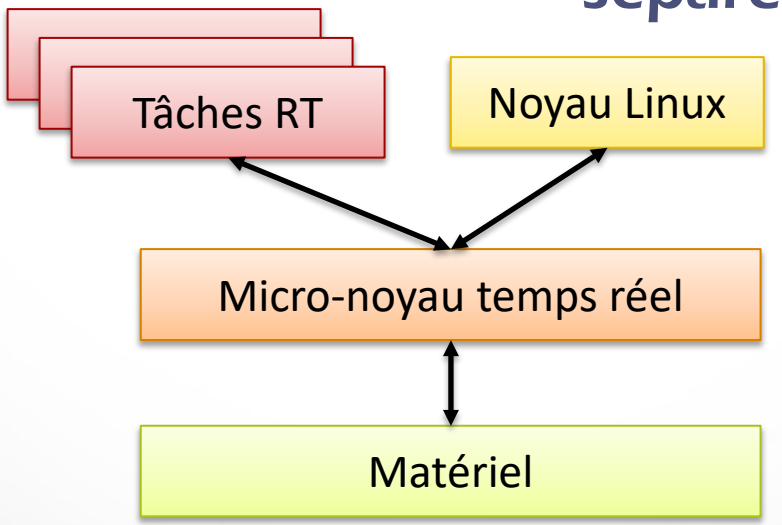
Extensions Temps Réel au Noyau Linux

Trois Générations

- ✓ RTLinux
- ✓ RTAI
- ✓ Xenomai

Un principe Commun

- ✓ Ajouter une couche entre le matériel et le noyau Linux pour gérer les tâche temps réel séparément



UNIVERSITÉ CÔTE D'AZUR

Quoi Utiliser ?

- ✓ RTLinux
  - En 2007 revente des droits de RTLinux à Wind River
  - Plus supporté par la communauté
  - Fournit sous le nom « Real-Time Core for Wind River Linux »
  - https://www.windriver.com/solutions/learning/rtos
- ✓ RTAI
  - Real-Time Application Interface for Linux, créé en 1999
  - Activement maintenu sur x86, pas vraiment maintenu sur ARM
  - http://www.rtai.org/
- ✓ Xenomai
  - Démarré en 2001
  - Pas pour vocation d'être intégré au noyau Linux
  - http://www.xenomai.org/

UNIVERSITÉ CÔTE D'AZUR

Bibliographie

- ✓ Solutions temps réel sous Linux: Cas pratique : le Raspberry Pi 3 (3ème édition)
  - Christophe Blaess - Collection Eyrolles – 01/2019
- ✓ Real-Time Linux Wiki
  - http://rt.wiki.kernel.org/index.php/Main_Page
- ✓ Le temps réel sous Linux:
  - https://www.linuxembedded.fr/2019/09/le-temps-reel-sous-linux/