

Cours Systèmes Temps Réel

CH3: **Cheddar**: Outil de Simulation des Ordonnancements Temps Réel

Par:

Hassan EL FADIL

elfadil.h@gmail.com

1.1. Description:

- Cheddar est un outil de simulation et d'analyse d'ordonnancement pour les systèmes temps réel embarqués.
- Il permet de vérifier la faisabilité des algorithmes d'ordonnancement et de s'assurer que les contraintes temporelles sont respectées.
- Outil open Source.

1.2. Caractéristiques:

- Support de divers algorithmes d'ordonnancement (fixes et dynamiques).
- Analyse de faisabilité d'ordonnancement.
- Interface graphique pour configurer et visualiser les simulations.
- Simulation des tâches et des ressources partagées.

Tableau comparatif des **5 logiciels** pour les simulations de systèmes temps réel et tests d'ordonnançabilité :

Logiciel	Description	Points forts	Langue/Interface	Usage principal	Site officiel
Cheddar	Outil open-source pour l'analyse et la simulation de systèmes temps réel.	Support de nombreux algorithmes d'ordonnancement.	Interface graphique	Académique et recherche	Cheddar
SimSo	Simulateur Python pour l'ordonnancement soft temps réel.	Facilité de personnalisation et interface intuitive.	Python (script/GUI)	Simulation académique	SimSo
RTSim	Bibliothèque C++ pour simuler des tâches et stratégies d'ordonnancement.	Flexibilité et extensibilité.	C++ (code uniquement)	Recherche et simulation avancée	RTSim
UPPAAL	Modélisation et vérification de systèmes temps réel via automates temporisés.	Vérification formelle puissante.	Interface graphique	Analyse formelle et académique	UPPAAL
Tracealyzer	Analyseur pour systèmes embarqués temps réel, basé sur FreeRTOS.	Visualisation graphique des tâches et des performances.	Interface graphique	Systèmes embarqués industriels	Tracealyzer

2. Téléchargement de l'Outil Cheddar

- Créateurs de l'outil: équipe de chercheurs de l'Université de Bretagne Occidentale (UBO) , Brest, France
- Le Nom est inspiré du fameux fromage originaire du village anglais de Cheddar.



Description:

<https://beru.univ-brest.fr/cheddar/>



Télécharger et installer Cheddar:

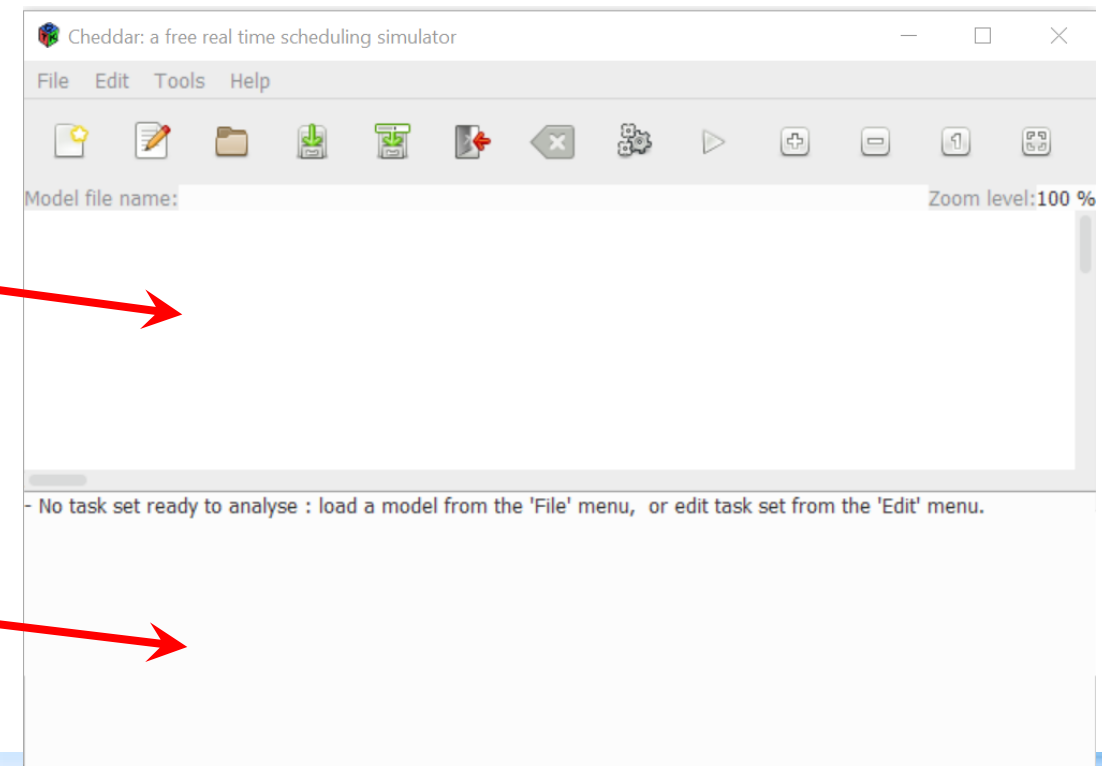
https://beru.univ-brest.fr/cheddar/ug/ug_v3_3/pages/download_install/

3.1. Fenêtre Cheddar:

- 1 Décompresser le dossier téléchargé
- 2 Exécuter Cheddar. Double-cliquer sur le fichier .exe



- La partie haute affiche des chronogrammes représentant la simulation du système temps réel étudié.
- La partie basse affiche divers résultats d'analyse.



Remarque:

Si les fenêtres **s'affichent trop grandes ou trop petites**, vous pouvez ajuster **l'échelle (zoom) d'affichage** de Windows :

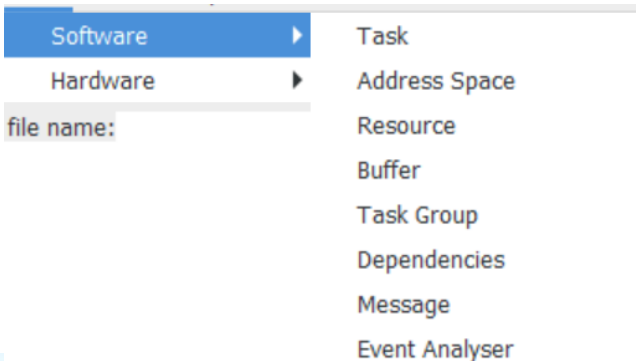
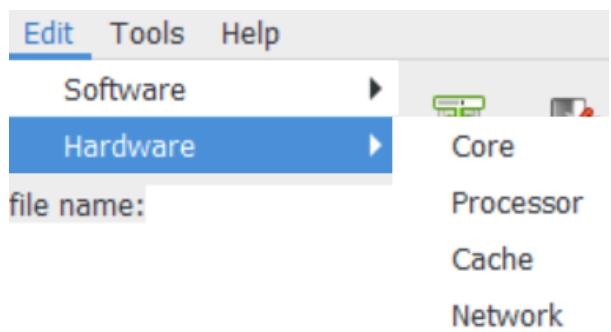
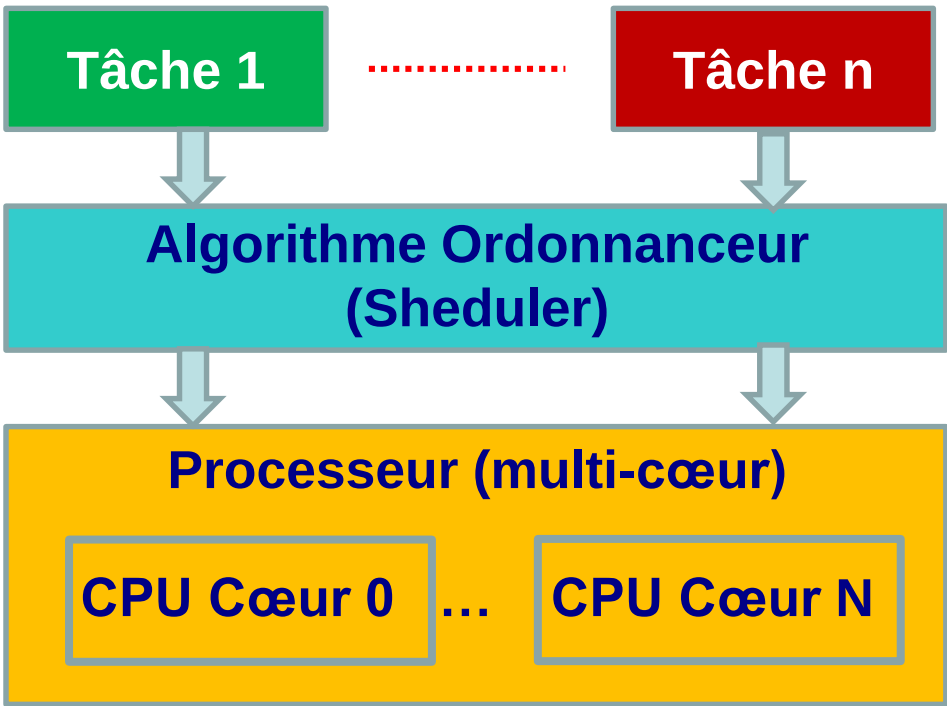
1. Faites un **clic droit sur le Bureau**, puis sélectionnez **Paramètres d'affichage**.
2. Rendez-vous à la section **Mise à l'échelle**.
3. Choisissez la valeur qui vous convient :
 - **100 %** (taille normale)
 - **125 %**
 - **150 %**, etc.

Choisir la mise à l'échelle adéquate



3.2. Partie matérielle/logicielle:

- Cheddar comporte une description de la partie matérielle et de la partie logicielle du système à analyser
- Partie matérielle: Coeur, Processeur, Cache, Réseaux.
- Partie logicielle: Tâches, espace d'adresses, ressources, etc



4. Exemple d'étude

4.1. Système de tâches:

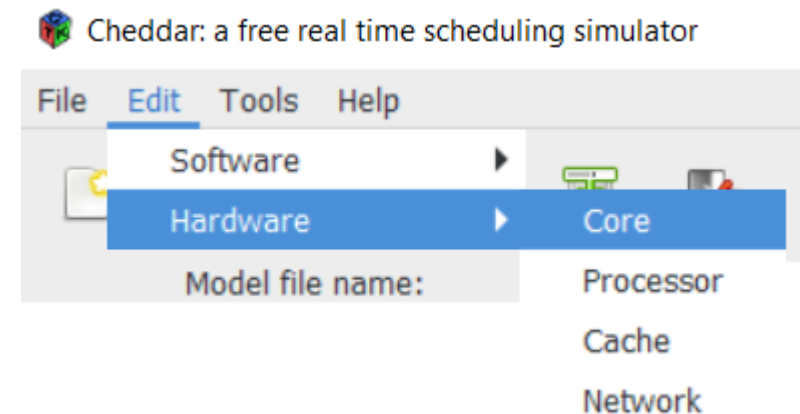
Soit un système constitué par 3 tâches, périodiques, synchrones et à échéances sur requêtes, avec les paramètres ci-contre:

Tâche	Période P_i	Durée d'exécution C_i	Echéance D_i	Priorité
T1	6	2	6	100
T2	9	2	9	90
T3	12	3	12	120

On souhaite analyser ce système d'ordonnancement selon l'algorithme Rate Monotonic (RM) en utilisant l'outil Cheddar

4.2. Elaboration du modèle à analyser:

- Dans un premier temps, la **plate-forme d'exécution** matérielle est décrite.
- Pour Cheddar, cette plate-forme peut être constituée d'un ou plusieurs composants « **processor** » incluant un ou plusieurs composants « **core** » et éventuellement des composants **cache**.
- Pour décrire un processeur comportant un core, utilisez le menu **Edit/Hardware/Core**



1 Commencer par créer un « Core »: Edit/Hardware/Core

Donner un nom unique: ici Core1

Définir la méthode d'ordonnancement: Ici RM

Spécifier si l'ordonnanceur doit être préemptif ou non: Ici Preemptive

Garder les autres paramètres par défaut !

Pas d'exécution: 1 unité de temps

Terminer par cliquer sur Add puis Close

Name	Scheduler	Quantum	Preemptive	Automaton
Core1	Rate Monotonic Protocol	0	Preemptive	

2 Créer ensuite un « Processor »: Edit/Hardware/Processor

Une fois Core1 défini, il est possible de décrire le processor qui utilisera Core1 avec le menu Edit/Hardware/Processor

Processor

Name

CPU1

Network

Processor Type

Monocore Type

Migration Type

No Migration Type

Cores Table

Core Name

Core1

Delete

Add

Core Name

Core1

Close

Cancel

Delete

Modify

Add

Name	Network	Processor Type	Migration Type
CPU1		Monocore Type	No Migration Type

Donner un nom au Processeur: ici CPU1

Définir le type de processor
Monocore/Multicore: Ici Monocore

Ajouter un Core: Ici Core1

Terminer par cliquer sur Add puis Close

- Par la suite, il est nécessaire de décrire la partie logicielle du système à analyser :
 - *Il s'agit ici des tâches (**composant task**) mais aussi des ressources partagées entre les tâches (**composant resource**).*
 - *Cheddar requiert par ailleurs la modélisation des entités mémoires regroupant tâches et ressources, c'est le rôle des composants **Address Space**.*

3 Définir l'espace d'adresse : Edit/Software/Address space

Seuls les champs Name et Cpu Name doivent être renseignés ici

Donner un nom à l'espace d'adresse mémoire: ici Adr1

Address Space

Name

Adr1

Processor Name

CPU1

Name	Processor Name	Text Size	Stack Size	Data Size	Heap Size	Scheduler	Quantum
Adr1	CPU1	0	0	0	0	No Scheduling Protocol	0

Sélectionner le Processeur: Ici CPU1

Garder les autres paramètres par défaut !

Close

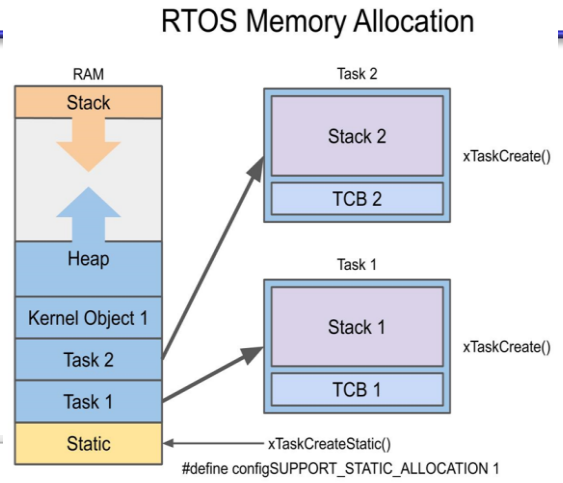
Cancel

Delete

Modify

Add

Terminer par cliquer sur Add puis Close



Tâche	Période	Durée d'exécution	Echéance	Priorité
T1	6	2	6	100
T2	9	2	9	90
T3	12	3	12	120

... Exemple d'étude / Elaboration du modèle

4 Terminer la configuration par décrire les tâches du système : Edit/Software/Task

Name

T1

Task Type

Periodic

Processor Name

CPU1

Core Name

Address Space Name

Adr1

Capacity

2

Deadline

6

Start Time

0

Priority

100

Blocking Time

0

Policy

Sched Fifo

Text Memory Size

0

Stack Memory Size

0

Criticality

0

Jitter

0

Period

6

Activation Rule

Name

Task Type

Processor

Address Space

Core

Capacity

Deadline

Start time

T1

Periodic

CPU1

Adr1

2

6

0

T2

Periodic

CPU1

Adr1

2

9

0

T3

Periodic

CPU1

Adr1

3

12

0

Nom de la tâche: ici T1 pour la première tâche

Type: Ici Périodique

Nom Processeur: Ici CPU1

Espace d'adresse: Ici Adr1

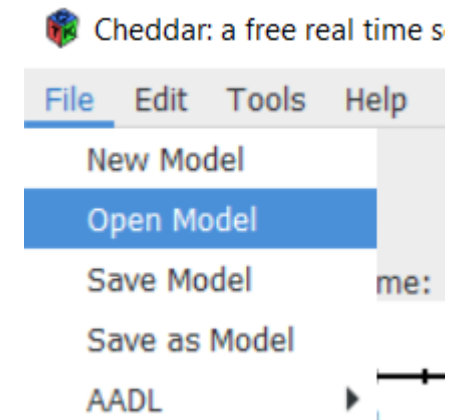
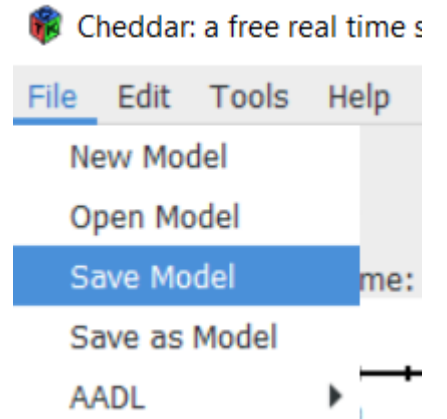
Paramètres de la tâche T1:
C=2; D=6; Prio=100; temps de 1^{er} réveil r1=0

Période de la tâche T1: P=6

Terminer par cliquer sur Add puis Close

5 Sauvegarder, enfin, votre Projet Cheddar : **File/Save Model**

Remarque: un modèle peut être chargé dans Cheddar pour être analysé:
File/Open Model



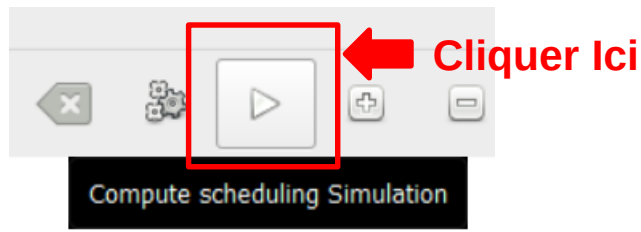
4.3. Analyse du modèle:

Dans cette partie, nous allons analyser le modèle ultérieurement élaboré.

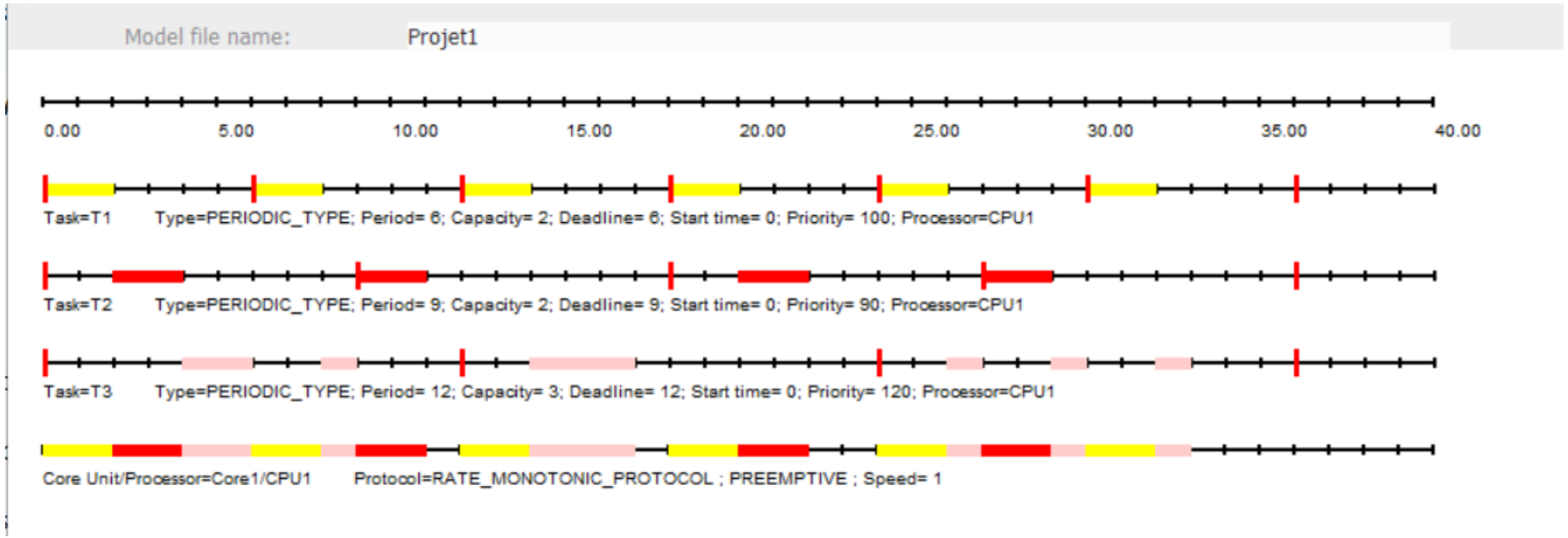
A partir d'un tel modèle, nous pouvons appliquer trois types d'analyse, toutes disponibles à partir du menu **Tools** :

1. *Des outils de simulation permettant de calculer l'ordonnancement du jeu de tâches et de déterminer, à partir de cette simulation, divers critères de performance comme le respect des échéances, les temps de réponse (pire ou moyen), les temps de blocage sur ressources partagées par exemple.*
2. *Divers tests de faisabilité permettant de démontrer l'ordonnançabilité du jeu de tâches, en calculant les pires temps de réponse, le taux d'occupation des processeurs, ...*
3. *Des outils permettant de modifier le modèle d'architecture (dans un objectif d'exploration d'architectures) si ce dernier ne conduit pas au respect des échéances des tâches : algorithmes pour affecter les priorités des tâches, pour placer les tâches sur les processeurs ou modifier leurs paramètres selon leurs dépendances par exemple.*

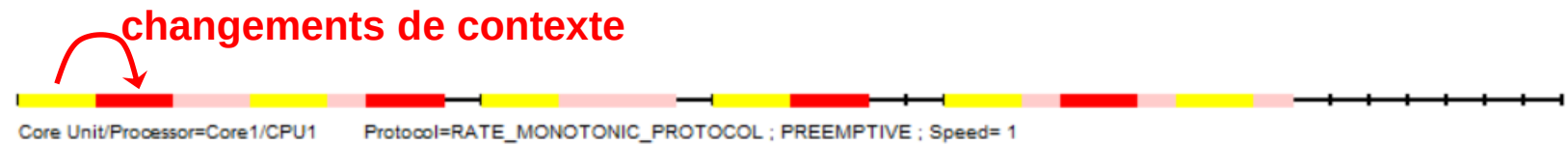
6 Exécuter la simulation d'ordonnancement



Résultats:
Diagramme de Gantt



Résultats:



Scheduling simulation, Processor CPU1 :

- Number of context switches : 15
- Number of preemptions : 3
- Task response time computed from simulation :
 - T1 => 2/worst
 - T2 => 4/worst
 - T3 => 9/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.

Nombre de changements de contexte (nombre de changements d'exécutions de tâches)

Temps de rotation ou temps de séjour (Turnaround time)

Scheduling feasibility, Processor CPU1 :

1) Feasibility test based on the processor utilization factor :

- The feasibility interval is 36.0, see Leung and Merrill (1980) from [21].
- 7 units of time are unused in the feasibility interval.
- Number of cores hosted by this processor : 1.
- Processor utilization factor with deadline is 0.80556 (see [1], page 6).
- Processor utilization factor with period is 0.80556 (see [1], page 6).
- In the preemptive case, with RM, we cannot prove that the task set is schedulable because the processor utilization factor 0.80556 is more than 0.77976 (see [1], page 16, theorem 8).

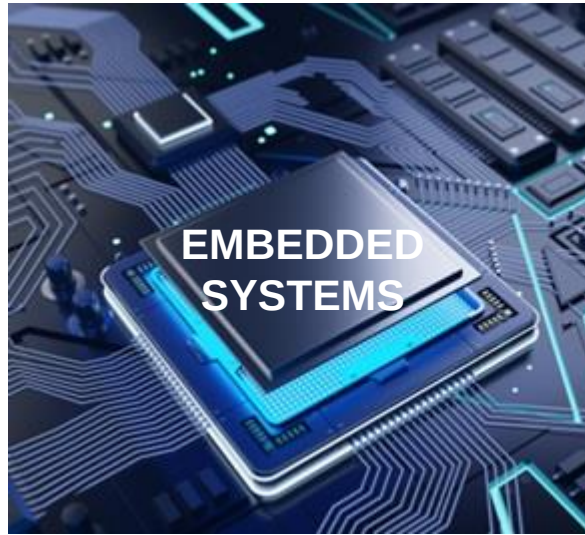
Taux d'utilisation CPU:
$$U = \sum_{i=1}^n \frac{C_i}{P_i}$$

5. Exercice d'application

Simuler le système de tâches précédent avec les algorithmes:

- **DM** : *Deadline Monotonic*
- **EDF** : *Earliest Deadline First*
- **LLF** : *Least Laxity First*

Tâche	Période P_i	Durée d'exécution C_i	Echéance D_i	Priorité
T1	6	2	6	100
T2	9	2	9	90
T3	12	3	12	120



FIN !
Questions ?