


Cours Systèmes d'Exploitation embarqués et Temps réel




Chapitre I : Introduction au Systèmes d'Exploitation Embarqués et Temps réel

2023/2024
Cours RTOS
1

1

**Introduction au Systèmes d'Exploitation
Embarqués et Temps réel**




Définition

- C'est un système électronique et informatique autonome qui est dédié à une tâche particulière et contenu dans un système englobant. Il n'est « généralement » pas programmable.
- Pas d'E/S standards.
- Il est constitué de parties matérielles et logicielles qui sont conçues spécifiquement pour réaliser une fonction dédiée.
- Il n'est pas visible mais il est intégré dans un équipement doté d'une fonction, on dit que alors c'est un système enfoui.

2023/2024
Cours RTOS
2

2




Introduction au Systèmes d'Exploitation Embarqués et Temps réel

Caractéristiques principales d'un système embarqué

- Système principalement numérique
- Met généralement en œuvre un processeur
- Exécute une application logicielle dédiée précise (non pas une application grand public)
- Il n'a pas réellement de clavier standard et l'affichage est limité.
- Ce ne sont pas des PC, mais des architectures similaires (X86, ARM, PowerPC, RISC-V, ARM) à basse consommation (l'une des contraintes ...)
- Autonomie élevée: Une fois en marche sur l'application, il n'est plus accessible
- Temps réel: Le temps de réponse de ces systèmes est aussi important que l'exactitude des résultats

2023/2024
Cours RTOS
3

3



Introduction au Systèmes d'Exploitation Embarqués et Temps réel

Caractéristiques principales d'un système embarqué

- Ressources limitées (mémoire, vitesse processeur, énergie).
- Interaction avec son environnement (capteurs)
- Garantit la robustesse (stabilité de sa performance malgré des conditions externes)
- Sécurisé
- Maintenable dans le temps: certains systèmes sont censés durer jusqu'à 20 ans et plus
- Optimisé: généralement il utilise des logiciels de petite taille car plus c'est grand, plus il y a de chance d'avoir des bugs.
- Tolérant aux fautes (continuer à fonctionner, éventuellement de manière réduite au lieu de tomber complètement en panne, lorsque l'un de ses composants ne fonctionne plus correctement).

2023/2024
Cours RTOS
4

4

Introduction au Systèmes d'Exploitation Embarqués et Temps réel

Champs d'application

- Il existe 4 grandes classes:
 1. Calcul généraliste:
 - ✧ Similaire aux applications bureau mais embarqué (assistant personnel, téléphone portable, etc.)
 - ✧ Consoles de jeux vidéo, décodeur, appareils photographiques et caméras, ...
 2. Contrôle de systèmes: Moteur, voiture, avion, processus chimique, nucléaire, navigation, ...
 3. Traitement du signal: Compression vidéo, radar, flux de données ,etc.
 4. Réseaux et communications: Transmission de données, routage, etc.

2023/2024
Cours RTOS
5

5


Introduction au Systèmes d'Exploitation Embarqués et Temps réel

Architecture d'un système embarqué

- Un système embarqué peut être structuré autour d'un:
 - ✧ Capteurs couplés à des convertisseurs analogique/numérique.
 - ✧ Actionneurs (LED, etc) couplés à des convertisseurs numérique/analogique.
 - ✧ Calculateur (processeur embarqué et ses E/S).
 - ✧ Mémoire ROM, pour les programmes à exécuter, et RAM, pour le stockage temporaire des données.
- Possibilité d'avoir un/des FPGAs et/ou ASICs pour jouer le rôle de co-processeurs (accélération matérielle)

2023/2024
Cours RTOS

6




Introduction au Systèmes d'Exploitation Embarqués et Temps réel

Métriques des systèmes embarqués

- Puissance de calcul: sélection du processeur d'après la charge de travail à effectuer et d'après la largeur des registres.
- Débit: le système peut avoir besoin de prendre en charge plusieurs données en même temps.
- Temps de réponse: le système doit réagir vite.
- Mémoire: l'estimation la plus précise possible de l'utilisation mémoire.
- Consommation d'énergie: la conception matérielle et logicielle doit prendre en compte la consommation de l'énergie.
- Nombre d'unités produites
- Durée de vie est un paramètre important lors de la décision des composants à inclure.
- Testabilité et débogage

2023/2024
Cours RTOS
7

7



Introduction au Systèmes d'Exploitation Embarqués et Temps réel


Logiciel/matériel embarqué

- Logiciel embarqué: programme/application utilisé dans un équipement et complètement intégré dans ce dernier.
- Système embarqué: Matériel(s) + logiciel(s) **(+OS)**
- Deux types de systèmes embarqués:
 - Systèmes embarqués destinés à l'utilisateur (high-end): utilisation d'un OS (ex: Linux). Ex: routeurs, Assistant numérique personnel, etc.
 - Systèmes embarqués profondément enfouis: peu de fonctions, très petite empreinte mémoire, généralement construit *from scratch*. Ex: Appareil photo numérique, téléphones portables, etc.
- Dans les systèmes embarqués, les tâches sont des fonctionnalités limitées et elles ne changent pas avec le temps
 - Dimensionner au mieux et optimiser le système en fonction de ses besoins

2023/2024
Cours RTOS
8

8

Introduction au Systèmes d'Exploitation Embarqués et Temps réel


 **Logiciel/matériel embarqué**

- Les ressources matérielles doivent être gérées en assurant leurs partages entre les différents utilisateurs.
- L'OS va offrir une interface homogène et générique mieux adaptée aux utilisateurs
- L'OS va donner la possibilité de bénéficier des mêmes avancées technologiques que les applications classiques (TCP/IP, HTTP, etc.)
- L'OS va permettre aux développeurs de logiciel embarqué de bien connaître le matériel
 - Les applications doivent avoir un accès aux services de l'OS via des APIs (réutilisabilité du code, interopérabilité, portabilité, maintenance aisée)
 - Le temps de développement va devenir raisonnable, afin de limiter le temps de mise sur le marché (time-to-market), et ainsi d'assurer le succès du produit.

2023/2024
Cours RTOS
9

9

Introduction au Systèmes d'Exploitation Embarqués et Temps réel

 **Systèmes d'Exploitation Embarqués**

- L'OS garantit un environnement de développement plus performant
- L'OS offre un contrôle de processus sans (ou à faible) contrainte temporelle pour les systèmes à temps partagé
 - Garantir le partage équitable du temps et des ressources
- L'OS offre un contrôle de processus avec contrainte temps réel pour les systèmes temps réel
 - Garantir les temps de réponse
 - Systèmes à contraintes souples/molles : systèmes acceptants des variations minimales de temps de réponse (systèmes multimédias)
 - Systèmes à contraintes dure : gestion stricte de temps pour conserver l'intégrité du système (déterminisme logique et temporelle et fiabilité)

2023/2024
Cours RTOS
10

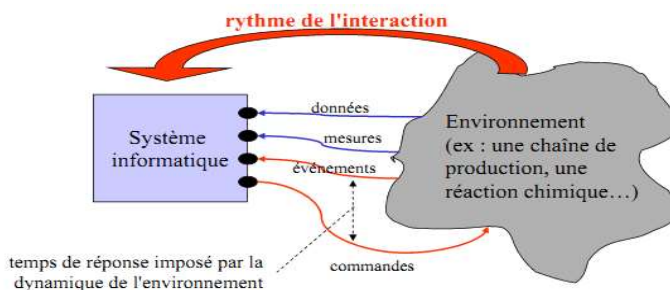
10

Introduction aux Systèmes d'Exploitation Embarqués et Temps réel



Définition d'un système Temps réel

Est appelé *temps réel* le comportement d'un système informatique dont le fonctionnement est assujéti à l'évolution dynamique d'un procédé industriel connecté à lui. On dit que le processus est contrôlé, piloté ou supervisé par le système qui *réagit* aux changements d'état du processus.



Système temps réel (STR): système dont le comportement dépend non seulement de l'exactitude des traitements effectués, mais également du temps où les résultats de ces traitements sont fournis, c.-à-d. qu'un retard dans la production d'un résultat est considéré comme une erreur.

Cours RTOS

11

11

Introduction aux Systèmes d'Exploitation Embarqués et Temps réel



Conséquence de la Définition

➤ Conséquences de la définition :

↳ Temps Réel ≠ aller vite

- temps de réaction court (1 ms) pour le contrôle d'un avion de combat
- temps de réaction moins court (10 ms) pour le contrôle d'un avion de transport civil
- temps de réaction moins court (1s) pour une IHM
- temps de réaction moins court (1mn) pour le contrôle d'un chaîne de production (lente)
- temps de réaction moins court (1h) pour le contrôle d'une réaction chimique (lente)
- temps de réaction de quelques heures pour l'établissement d'une prévision météorologique
- temps de réaction de quelques jours pour le calcul de la paie...

⇒ **On parlera de temps réel à chaque fois qu'il sera question de contraintes de temps et que ces dernières seront respectées :**

⇒ **un résultat juste mais hors délai est un résultat faux**

⇒ et parfois réagir trop tôt est aussi une faute (cf. contrôle d'un avion souple)

↳ **Temps Réel = Ponctualité**


2023/2024

Cours RTOS

12

12

Introduction au Systèmes d'Exploitation Embarqués et Temps réel

 ISSAT


- Un peu de classification... ou un autre point de vue du temps réel.
 - ↳ Système Transformationnel :
 - activité de calcul, qui lit ses données et ses entrées lors de son démarrage, qui fournit ses sorties, puis meurt.
 - ↳ Système Interactif :
 - système en interaction quasi permanente avec son environnement, y compris après l'initialisation du système ; la réaction du système est déterminée par les événements reçus et par l'état courant (fonction des événements et des réactions passés) ; le rythme de l'interaction est déterminé par le système et non par l'environnement.
 - ↳ Système **Réactif** ou **Temps Réel** :
 - système en interaction **permanente** avec son environnement, y compris après l'initialisation du système ; la réaction du système est déterminée par les événements reçus et par l'état courant (fonction des événements et des réactions passées) ;
 - **mais le rythme de l'interaction est déterminé par l'environnement et non par le système.**

=> capacité à pouvoir appréhender un flux d'évènements asynchrones issus d'un processus, sans perdre un seul de ces évènements et de traiter chacun d'eux en un temps déterminé.

Cours RTOS

13

Introduction au Systèmes d'Exploitation Embarqués et Temps réel

 ISSAT

Notions de criticité

Deux notions de criticité face au manquement d'une échéance

- ↳ Contraintes temps réel **strictes** : le dépassement d'une échéance est catastrophique
 - **Contraintes de temps Fortes**
- ↳ Contrainte temps réel **relatives** : le dépassement d'une échéance peut être toléré (dans une certaine mesure)
 - **Contraintes de temps Faible**

↳ => **Temps réel Dur / Temps réel Lâche !**

- ↳ => dans le cas des systèmes Temps Réel Dur, on cherchera à obtenir un comportement **prévisible, déterministe et fiable**
 - => utilisation de techniques mathématiques (ordonnancement, évaluation des pires cas...)
- ↳ => dans le cas des systèmes Temps Réel Lâche, on cherchera à minimiser la probabilité de rater une échéance plusieurs fois de suite...

Cours RTOS

14

Introduction au Systèmes d'Exploitation Embarqués et Temps réel

Quelques exemples de systèmes embarqués temps réels

- **Système RADAR de détection des objets.**

➔ Les contraintes temporelles sont critiques où le non respect de ces contraintes pourrait conduire à des défaillances du système avec des conséquences pouvant être graves.
- **Système de téléconférence**

➔ Les images et le son doivent être synchronisées, mais des dérives de synchronisation (dus à des messages tardifs) sont souvent tolérées. De même pour les systèmes de réservation des compagnies aériennes, car si une requête de réservation dure un peu plus longtemps que prévu, la seule conséquence regrettable pourra être la perte d'un client qui quitte l'agence.

2023/2024 Cours RTOS 15

15

Introduction au Systèmes d'Exploitation Embarqués et Temps réel

Utilisation du temps

Au sein d'une application ou d'un système temps-réel, il faut pouvoir gérer le **temps (horloge)**. Le temps sera utilisé de plusieurs façons:

- Soit en définissant la date à laquelle une action doit être **commencée**
- Soit en définissant la date à laquelle une action doit être **finie**

• Il peut être nécessaire de pouvoir modifier ces paramètres en cours d'exécution et de pouvoir préciser les actions à prendre en cas de faute temporelle.

DECOUPAGE EN TACHES OU EN PROCESSUS (concurrents)

Dans le monde réel les périphériques et l'environnement du système évoluent simultanément (en parallèle et parfois en concurrence). Pour calquer fidèlement la réalité il faut s'appuyer sur de la programmation concurrente :

- utiliser un **modèle de tâches**,
- utiliser des moyens de **communication et de synchronisation inter-tâches** ou inter-processus (mémoire partagée, boîtes aux lettres, files de messages, sémaphores ...)

2023/2024 Cours RTOS 16

16

Introduction au Systèmes d'Exploitation Embarqués et Temps réel

Conception d'un système temps réel

- La conception d'un Système Temps Réel passe par plusieurs phases :
 1. Identification de la tâche à réaliser et les contraintes temporelles qui doivent être satisfaites
 2. Ecriture du code
 3. Mesure du temps d'exécution de chaque tâche et test de rendez-vous horaire pour vérifier qu'aucune tâche ne dépassera son temps limite pendant le fonctionnement du système.
 - ✧ Avec les tests, il est possible de garantir qu'aucun temps limite ne sera dépassé.
 - ✧ Si les tests ne sont pas concluants, il est nécessaire de reprendre la conception depuis le début, en choisissant un processeur plus rapide ou en utilisant d'autres algorithmes pour implémenter les tâches.

2023/2024 Cours RTOS 17

17

Introduction au Systèmes d'Exploitation Embarqués et Temps réel

➤ • Un système temps réel permet :

- De définir des tâches avec des priorités entre-elles
- De définir des tâches ayant des dates d'exécution imposées qui deviendront prioritaires quand la date s'approchera
- De définir des tâches déclenchées par des interruptions :
 - Avec un niveau de préemption entre elles
 - Avec un niveau de préemption vis-à-vis du SE lui-même (ordonnanceur)
- De bloquer/activer les interruptions du CPU (dangereux mais efficace en cas d'extrême urgence)
- De créer et gérer des sémaphores et des Mutex
- De communiquer entre tâches par :
 - Files (pipes)
 - Boîtes à lettres
 - Mémoire partagée
- De communiquer par réseau industriel (type CAN)

2023/2024 Cours RTOS 18

18

Introduction au Systèmes d'Exploitation Embarqués et Temps réel

Systèmes d'Exploitation Embarqués et Temps réel

- Le système informatique se décompose lui-même en deux parties :
 - ✧ l'application temps réel proprement dite construite par un ensemble de tâches, qui sont activées de manière:
 - ✓ périodique pour réaliser les prises de mesures
 - ✓ aperiodique pour prendre en compte les événements du procédé.
 - ✧ Les services d'un noyau temps réel appelé généralement exécutif temps réel pour respecter les contraintes temps réel des tâches et utiliser les ressources matérielles ou périphériques.

2023/2024 Cours RTOS 19

19


Introduction au Systèmes d'Exploitation Embarqués et Temps réel

Pourquoi utiliser un RTOS ?

- Concentration sur l'application, et non sur l'exécution de tâches
 - Gestion automatique du temps de CPU entre unités d'exécution (threads), suivant un critère de sélection prédéfini (priorité ou autre)
 - Synchronisation de l'accès aux ressources
- Ajout/retrait de modules fonctionnels sans modifier l'application existante
- Services aux périphériques quand disponibles, incluant des protocoles de communication:
 - TCP/IP, USB, serveurs Web, WI-FI, communication CAN, etc.

2023/2024 Cours RTOS 20

20




Introduction au Systèmes d'Exploitation Embarqués et Temps réel

RTOS

- C'est Système d'exploitation temps-réel (RTOS - Real Time Operating Systems), qui se caractérise par la présence de contrainte temps réel
- Un RTOS est construit autour d'un noyau multitâche qui contrôle l'allocation de temps des tâches.
- Un laps de temps est la durée d'exécution d'une tâche donnée avant qu'elle sera arrêtée et remplacée par une autre tâche.
- Ce processus est connu par le changement du contexte, qui se répète de façon continue.
- Un RTOS fournit aussi les messages qui passent entre les tâches, synchronisation des tâches et allocation de ressources partagées aux tâches

2023/2024
Cours RTOS
21

21



Introduction au Systèmes d'Exploitation Embarqués et Temps réel


Les services de bases d'un RTOS

1. **Tâche:**
 - ✧ Processus ou thread: elle est définie par son identificateur, comporte un contexte permettant à tout instant de connaître son état et ainsi de pouvoir arrêter/redémarrer son exécution sur le processeur.
 - ✧ Création de processus et de threads:
 - ✓ Statique: toutes les tâches sont connues à l'avance il n'est pas possible d'en créer pendant que le système tourne.
 - ✓ Dynamique: appels système permettant de créer et détruire des tâches à la volée (au fur et à mesure):
 - ☐ Système plus flexible (qui peut s'adapter aux circonstances).
 - ☐ Plus de complexité (allocation dynamique, gestion d'erreurs,...)

2023/2024
Cours RTOS
22

22

Introduction au Systèmes d'Exploitation Embarqués et Temps réel

 iisat


2. Ordonnancement:

- ✧ Entité qui décide quelle tâche doit être exécutée par le processeur
- ✧ Faire un compromis entre la complexité d'implémentation, et délai d'exécution
- ✧ RTOS supporte plusieurs politiques d'ordonnancement, le choix revient au programmeur:
 - ✓ FIFO avec priorité (statique)
 - ✓ Date limite la plus proche (priorité dynamique)
 - ✓ Serveur sporadique (perte de priorité en fonction du temps processeur consommé).
 - ✓ ...

2023/2024 Cours RTOS 23

23

Introduction au Systèmes d'Exploitation Embarqués et Temps réel

 iisat


3. Interruptions:

- ✧ Les interruptions permettent de prendre en compte des événements au processeur. La prise en compte d'une interruption provoque l'arrêt du programme en cours et l'exécution d'un programme associé à cette interruption.
- ✧ Les interruptions permettent de signaler des requêtes asynchrones
- ✧ Gestion de l'aspect temps réel

2023/2024 Cours RTOS 24

24

Introduction au Systèmes d'Exploitation Embarqués et Temps réel




4. **IPC (Boîtes aux lettres):** Service permettant d'envoyer (send) et de recevoir (receive) des messages. La tâche en attente d'une lettre est suspendue jusqu'à l'arrivée de cette dernière.
5. **Événements :** Le principe des événements est similaire à celui des boîtes aux lettres, à la différence près qu'au lieu de recevoir ou d'envoyer des données, un signal est émis ou attendu.
6. **Gestion de la mémoire**
 - ✧ Allocation: Allouer à chaque tâche la mémoire dont elle a besoin
 - ✧ Mapping: Faire la correspondance entre la mémoire physique et l'adressage utilisé par les tâches.
 - ✧ Protection: Etablir un ensemble de comportements à adopter lorsqu'une tâche utilise de la mémoire non allouée.

2023/2024 Cours RTOS 25

25

Introduction au Systèmes d'Exploitation Embarqués et Temps réel



7. **Sémaphores:** Les sémaphores utilisés avec les systèmes temps réel servent à partager une ressource entre plusieurs tâches (création d'une section critique). Les sémaphores servent également à synchroniser plusieurs tâches entre elles.
8. **Timers ou chronomètre:** Ces services permettent de réaliser des traitements cycliques ou de détecter des dépassements dans les réceptions de données par exemple.
9. **Support réseau**
 - ✧ Le standard POSIX (Portable Operating System Interface for Computing Environments) :
 - Accès à n'importe quel mode/protocole de communication en réseau

2023/2024 Cours RTOS 26

26

Introduction au Systèmes d'Exploitation Embarqués et Temps réel

Classification des systèmes temps réel

1. **Temps réel dur:** Un système d'exploitation est dit Temps Réel dur s'il est capable de répondre à des sollicitations ou événements (internes ou externes) dans un temps maximum connu.
 - Si ces événement sont traités très tardivement, ils provoquent des conséquences catastrophiques.
2. **Temps réel mou:** Ces systèmes se rapprochent fortement des systèmes d'exploitation classiques à temps partagé. Ils garantissent un temps moyen d'exécution pour chaque tâche.
 - Si ces événement sont traités très tardivement, ils ne provoquent pas des conséquences catastrophiques.

2023/2024 Cours RTOS 27

27


Introduction au Systèmes d'Exploitation Embarqués et Temps réel

Classification des RTOS

- RTOS à noyau RT
 - Généralement conçu pour la rapidité de réponse.
 - Peut être inadapté aux systèmes complexes, car priorisation de la rapidité sur la prédictibilité
 - En général propriétaire : QNX, PDOS, VxWORKS, VTRX32, VxWORKS
- SE standard à noyau non-RT modifié
 - Extension RT gère l'exécution des tâches et considère le noyau standard comme l'une d'elles pour les réponses en temps réel
- Interface de programmation d'applications (API) standard versus dédiés
 - p. ex. POSIX extension-RT d'Unix, ITRON, OSEK

2023/2024 Cours RTOS 28

28


Introduction au Systèmes d'Exploitation
Embarqués et Temps réel


QUIZ

Un OS est-il obligatoire?

- **Absolument pas!**
 - Certains langages sont un OS à eux tout seul (Forth).
 - Certains langages incluent des options de concurrence plus ou moins avancées (Ada, Java, Rust).
 - Certains projets sont très simples

2023/2024
Cours RTOS
29

29


Introduction au Systèmes d'Exploitation
Embarqués et Temps réel

QUIZ

Un OS est-il utile?

- **Absolument!**
 - La plupart des programmes embarqués ont des besoins comparables (concurrence, temps-réel, USB, TCP/IP, fichiers, Bluetooth, WiFi).
 - Il est plus facile d'utiliser une API portable plutôt que de recoder des fonctionnalités de base.
 - **Pourquoi réinventer la roue systématiquement?**

2023/2024
Cours RTOS
30

30

Introduction au Systèmes d'Exploitation Embarqués et Temps réel



QUIZ

Un OS résout-il tous les problèmes?

- **Non!**
 - La gestion de la mémoire reste à la charge du développeur (d'où un conseil : ne gérez pas la mémoire et privilégiez les allocations statiques).
 - Les tâches ne s'ordonnent pas toutes seules : il faut en choisir les priorités statiques et dynamiques.
 - Un OS n'empêche pas les bugs dans le code, mais bénéficie généralement de plus d'utilisateurs que votre code : profitez de l'effet de masse.

2023/2024

Cours RTOS

31

31

Introduction au Systèmes d'Exploitation Embarqués et Temps réel



Quelques célèbres RTOS

- **VxWorks et pSOS:** (WindRiver)
 - Noyau temps réel le plus utilisé dans l'industrie
 - Inclut en natif un support TCP/IP
 - Coût important de la licence
 - Utilisation d'un environnement de *compilation croisée*
- **QNX:** (QNX)
 - Noyau temps réel de type UNIX (conforme à POSIX)
 - Gratuit pour des applications non commerciales
 - Développement direct sur la plateforme cible
 - Très faible *empreinte mémoire*
- **μC/OS** (Micrium): (uCosII ou III)
 - Environnement de très petite taille (microcontrôleur 68HC111)
 - Utilisable gratuitement pour l'enseignement/recherche.
 - Peut intégrer des protocoles standards



Mars Reconnaissance Orbiter



Tablet BlackBerry



2023/2024

Cours RTOS


32

32

Introduction au Systèmes d'Exploitation Embarqués et Temps réel

Quelques célèbres RTOS

- **Windows CE** : (Microsoft)
 - Cantonné à l'équipement de nombreux assistants personnels et téléphone (actuellement Nokia en remplacement de Symbian OS)
- **Nucleus** (Mentor Graphics)
 - Noyau temps réel; couche TCP/IP; interface graphique, serveur http, etc. (open source)
- **eCos**: embeddable Configurable OS (Cygnus puis RedHat)
 - Temps réel, bien adapté aux très faibles empreintes mémoire.
 - Disponibilité des protocoles standards (TCP/IP, etc.)
 - Basé sur Linux et la chaîne de compilation GNU
 - Conforme à la norme POSIX



2023/2024 Cours RTOS 33

33

Introduction au Systèmes d'Exploitation Embarqués et Temps réel

La norme POSIX

- POSIX sont les initiales de "Portable Operating System Interface" (Interface Portable pour Systèmes d'Exploitation).
- C'est un standard qui tend à obtenir la portabilité des logiciels aux niveau du code source.
 - un programme destiné à un système d'exploitation et respectant le standard POSIX devrait se compiler et s'exécuter sous n'importe quel système POSIX, même s'il provient d'un fabricant différent.
- POSIX est développé par l'IEEE (Institute of Electrical and Electronic Engineering) et standardisé par ANSI (American National Standards Institute) et ISO (International Standards Organisation).
- POSIX est basée sur UNIX. La majorité des systèmes d'exploitation tendent à être compatibles POSIX aux travers de leurs différentes versions.

2023/2024 Cours RTOS 34

34

Introduction au Systèmes d'Exploitation Embarqués et Temps réel

Exemples:

Exemples	Sig. extensions
POSIX 1003.1	Services de base (ex : fork, exec, ect)
POSIX 1003.2	Commandes shell (ex : sh)
POSIX 1003.1b [GAL 95]	Temps réel
POSIX 1003.1c [RIF 95]	Threads
POSIX 1003.5	POSIX et Ada
etc	

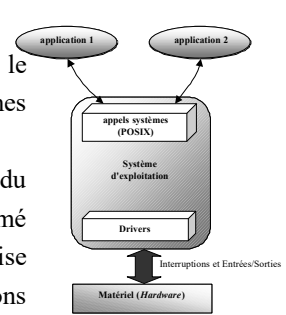
2023/2024 Cours RTOS 35

35

Introduction au Systèmes d'Exploitation Embarqués et Temps réel

Les extensions POSIX Temps Réel

- Les extensions POSIX sont conçues pour améliorer le contrôle de la gestion des ressources de systèmes d'exploitation
- Chaque groupe se charge de concevoir un aspect du système d'exploitation. Par exemple, le groupe nommé POSIX.4 (renommées 1003.1b depuis 1993) autorise l'usage d'un système d'exploitation dans des situations temps réel.
- Exemple de systèmes temps réel avec ou non la norme IEEE POSIX 1003.1



Nom	Fournisseur	Posix	Architectures
iRMX/INtime	Intel / TenAsys http://www.tenasys.com/	-	ix86
QNX	QNX Software Systems / BlackBerry http://www.qnx.com/	oui	ix86, arm
VxWorks	WindRiver Systems http://www.windriver.com/	-	arm, powerpc, ix86, mips,...
LynxOS	Lynuxworks http://www.lynuxworks.com/	oui	ix86, powerpc
RTLinux	FSMLabs http://www.fsmlabs.com/	oui	ix86, powerpc,...

2023/2024 Cours RTOS 36

36

Introduction au Systèmes d'Exploitation Embarqués et Temps réel



Autres standards

- **UNIX98**: normalisation de l'OS UNIX. Cette norme incorpore plusieurs des normes de POSIX
- **EL/IX**: API pour les systèmes embarqués. Se veut un sous ensemble des normes POSIX et ANSI.
- **ITRON**: norme japonaise pour les systèmes embarqués
- **OSEK**: norme allemande pour une architecture ouverte reliant les divers contrôleurs électroniques d'un véhicule.
- **RT Spec pour Java**: spécification pour un *runtime* qui édicte des prescriptions (ramasse miettes, certaines politiques d'ordonnancement, etc.)
- **Ada95**: ex: MarteOS, OpenRavenscar.
- **RT Corba**: un ensemble de spécification temps réel

2023/2024

Cours RTOS

37

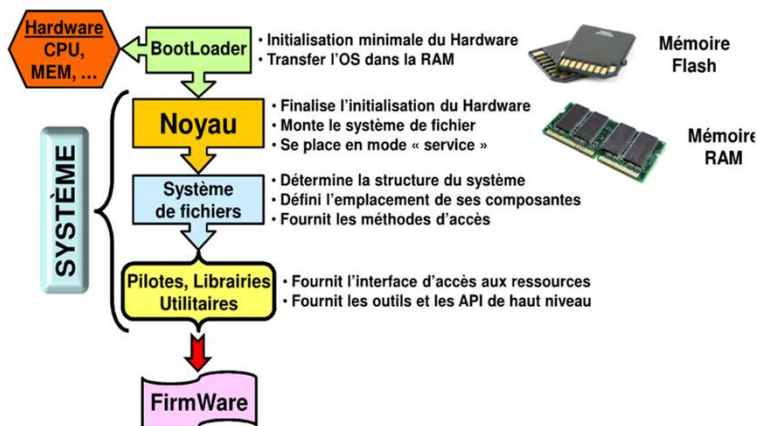
37

Introduction au Systèmes d'Exploitation Embarqués et Temps réel



Démarrage sur un système embarqué

Système embarqué : Partie logicielle



2023/2024

Cours RTOS

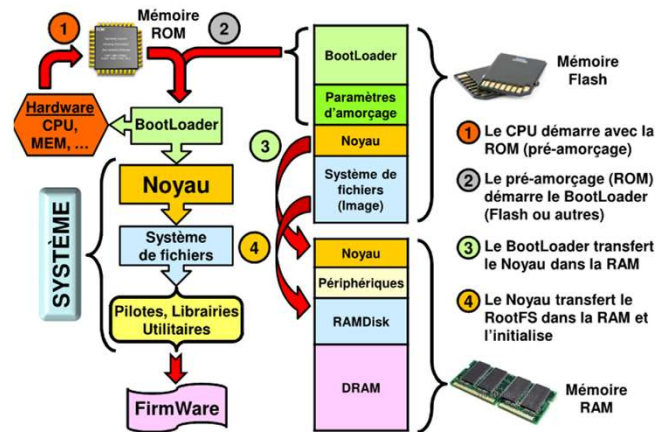
38

38

Introduction au Systèmes d'Exploitation Embarqués et Temps réel



Démarrage sur un système embarqué



2023/2024

Cours RTOS

39