

Présentation des Systèmes d'Exploitation Embarqués

I. Définitions (SE et SEE)

Un **système embarqué (SE)** est un ordinateur qui prend en charge une machine. Les exemples incluent les ordinateurs dans les voitures, les feux de circulation, les téléviseurs numériques, les guichets automatiques, les commandes d'avion, les terminaux de point de vente (POS), les appareils photo numériques, les systèmes de navigation GPS, les ascenseurs, les récepteurs multimédias numériques et les compteurs intelligents, parmi de nombreuses autres possibilités [1].

Un **système d'exploitation embarqué (SEE)** est un système d'exploitation spécialisé conçu pour effectuer une tâche spécifique pour un périphérique qui n'est pas un ordinateur. La tâche principale d'un système d'exploitation embarqué consiste à exécuter le code qui permet à l'appareil de faire son travail. Le système d'exploitation intégré rend également le matériel de l'appareil accessible au logiciel qui s'exécute sur le système d'exploitation [1].

II. Pourquoi un Système d'exploitation embarqué

L'utilisation d'un Système d'exploitation embarqué permet les avantages suivants [2] :

- Libérer le développeur de logiciel embarqué de l'obligation de bien connaître le matériel, le système d'exploitation embarqué contrôle tous le matériel par des pilotes (drivers).
- L'accès des applications aux services offerts par le système d'exploitation embarqué se font en utilisant des API (Application Programming Interface), ce qui permet de rendre les codes des programmes réutilisable, portable et interopérable.
- Offrir la possibilité de bénéficier des mêmes avancées technologiques que les applications classiques (TCP/IP, HTTP, etc.).
- Permettre au développeur d'application embarqué d'utiliser des environnements de développement performant.
- Minimiser le temps de commercialisation du système embarqué (Time To Market).

III. Contraintes des Systèmes embarqués à tenir en compte

Les systèmes embarqués exécutent des tâches prédéfinies et ont un cahier des charges contraignant à remplir, qui peut être d'ordre [3] :

- **D'espace compté** : l'espace mémoire peut être limité de l'ordre de quelques Go maximum (bien que la taille vienne à être de moins en moins limitée grâce à la miniaturisation des éléments). Il convient de concevoir des systèmes embarqués qui répondent aux besoins au plus juste pour éviter un surcoût.
- **De puissance de calcul** : il convient d'avoir la puissance de calcul juste nécessaire pour répondre aux besoins et aux contraintes temporelles de la tâche prédéfinie. Ceci en vue d'éviter un surcoût de l'appareil et une consommation excédentaire d'énergie (courant électrique).



- **D'autonomie** : la consommation énergétique doit être la plus faible possible, due à l'utilisation de batteries et/ou, de panneaux solaires voire de pile à combustible pour certains prototypes.
- **Temporel** : les temps d'exécution et l'échéance temporelle d'une tâche sont déterminés (les délais sont connus ou bornés a priori). Cette dernière contrainte fait que généralement de tels systèmes ont des propriétés temps réel.
- **De sûreté de fonctionnement** : car s'il arrive que certains de ces systèmes embarqués subissent une défaillance, ils mettent des vies humaines en danger ou mettent en péril des investissements importants. Ils sont alors dits « critiques » et ne doivent jamais faillir. Par « jamais faillir », il faut comprendre toujours donner des résultats justes, pertinents et ce dans les délais attendus par les utilisateurs (machines et/ou humains) des dits résultats.

IV. Caractéristiques des SEE

Les systèmes d'exploitation embarqués ont un ensemble de caractéristiques qui les diffèrent des systèmes d'exploitation de bureau. Parmi ces caractéristiques :

- **Ciblé** : limité aux fonctions pour lesquelles il est conçu.
- **Fiable** : car destiné à un fonctionnement autonome et/ou critique.
- **Maintenable pour toute la durée de vie du produit** (ex. 10 ans pour une automobile, 40 ans pour un avion).
- **Doté d'une IHM spécifique** : l'IHM se réduit à un afficheur de petite taille de type LCD, les périphériques d'entrée sont de petits boutons (évolution de l'IHM vers l'écran tactile comme dans le téléphone, mais nécessite une bibliothèque graphique adaptée).
- **Optimisé.**
- **De petite taille** pour le tester, le vérifier et/ou le certifier.
- **Embarqué sur des équipements produits à grande échelle.**
- **Performant** : du point de vue temps de réponse.
- **Configurabilité nécessaire [4]** : pas de SEE pour tous, la configurabilité est nécessaire pour éviter les surcharges/gaspillages dus à du code non utilisé. Parmi les techniques pour implémenter la configurabilité :
 - Ne pas inclure les fonctions non utilisées (fait par l'éditeur de liens)
 - Compilation/assemblage conditionnel (directives #if et #ifdef)
 - Évaluation et optimisation au moment compilation/assemblage
- **Les pilotes de périphériques sont typiquement laissés au développeur [4]** : Dans le cas des SE du bureau, des dispositifs standards sont supposés toujours présents (disques, Ethernet, audio, etc.), par contre dans un système d'exploitation embarqué c'est au développeur d'intégrer les pilotes des dispositifs qui sont présent dans son système embarqué.

V. Architecture d'un SEE

V.1. Architecture en couche

Un système d'exploitation est typiquement organisé en couches distinctes [7]:

- La couche supérieure est l'interface de programmation avec les logiciels applicatifs (dont font partie les logiciels utilitaires fournis avec le système d'exploitation).
- Au centre, on trouve une ou plusieurs couches qui contiennent les composants principaux du système d'exploitation tels que : les programmes pour les systèmes de fichiers et le réseau, la gestion de mémoire (), les pilotes, l'ordonnanceur, le gestionnaire d'interruption.
- La couche inférieure, appelée couche d'abstraction matérielle (anglais hardware abstraction layer abrégé HAL), est chargée de masquer les particularités matérielles.

V.2. Le noyau

Le noyau (anglais kernel) est un espace mémoire isolé, dans lequel est placé tout ou partie du système d'exploitation. Dans le langage courant le terme kernel désigne l'emplacement ainsi que l'ensemble des programmes qu'il contient et qui forment le cœur rigide du système d'exploitation. Le contenu du noyau ne peut pas être modifié par inadvertance (inattention) par les logiciels applicatifs – une modification provoquerait un crash de l'ordinateur [7].

Il existe plusieurs types de noyaux, parmi lesquels :

- **Noyau monolithique** : l'architecture est dite à noyau monolithique lorsque la totalité des programmes du système d'exploitation (en particulier les pilotes, les programmes qui traitent le réseau et le système de fichier) résident dans l'espace du noyau. Dans cette architecture chaque utilisation d'une fonction de l'interface de programmation provoque une commutation de contexte et le passage en mode kernel. Linux, FreeBSD, OpenVMS ou Solaris sont des systèmes d'exploitation à noyau monolithique [7].
- **Micro-noyau** : l'architecture est dite micro-noyau (anglais microkernel) lorsque le noyau contient le strict minimum, c'est-à-dire l'ordonnanceur et le programme qui simule la mémoire virtuelle, et que la grande majorité des programmes se trouvent en dehors : les pilotes, les programmes qui traitent les systèmes de fichiers ou l'interface graphique, ainsi que les logiciels applicatifs. Minix, BeOS, Mac OS X ou QNX sont des systèmes d'exploitation à micro-noyau [7].
- **Noyau hybride** : divers systèmes d'exploitation ont une architecture qui a certaines caractéristiques des micro-noyaux et en même temps des noyaux monolithiques. Cette architecture est appelée noyau hybride macro-noyau ou micro-noyau modifié. Dans cette architecture, la totalité des programmes du système d'exploitation résident dans le noyau, à la façon d'un noyau monolithique. Cependant de nombreux programmes du système d'exploitation sont exécutés en concurrence comme des logiciels applicatifs, à la manière de l'architecture micro-noyau (processus noyau). Windows NT ou NetWare sont des systèmes d'exploitation à noyau hybride [7].
- **Exo-noyau** : dans l'architecture exo-noyau (du grec exos = hors de), il n'y a pas d'emplacement isolé, et le système d'exploitation est composé de programmes et de bibliothèques logicielles de

construction identique à celles des logiciels applicatifs. Mac OS, AmigaOS et Oberon sont des systèmes d'exploitation en architecture exo-noyau [7].

VI. Exemples d'usage des SEE

VI.1. Parcomètre

Pour gérer les entrées des utilisateurs et suivre le temps et les frais, les parcomètres urbains intelligents nécessitent un système d'exploitation intégré. Ces appareils ont une gamme de fonctionnalités intégrées, selon la conception. Par exemple, certains utilisent des capteurs pour détecter l'arrivée et le départ des véhicules, tandis que d'autres demandent au conducteur de saisir l'espace de stationnement ou le permis du véhicule. Une interface utilisateur fournit au conducteur des sélections, telles que la possibilité de définir l'heure de retour prévue et de payer en conséquence [6].

VI.2. Système de navigation d'un avion

Le système de navigation d'un avion est un excellent exemple de système d'exploitation en temps réel. L'ordinateur principal d'un avion qui est interconnecté avec la plupart des systèmes de contrôle tels que les commandes de moteur, d'aile, de sécurité et de pression. Ainsi, il est spécifiquement conçu pour fonctionner à l'intérieur d'un avion et aider à effectuer des opérations telles que le décollage, l'atterrissage et les urgences [6].

VI.3. Système de navigation automobile

Le système de navigation automobile est un petit ordinateur qui comprend un écran tactile qui permet au conducteur de parcourir divers menus tels que le GPS et la cartographie d'itinéraire, la lecture audio, la radio, les appels mains libres et les systèmes de surveillance du niveau de carburant et de la pression des pneus. Dans les véhicules connectés, ces systèmes sont encore plus complexes, incorporant des systèmes de communication véhicule-à-intersection et véhicule-à-n'importe quoi ou "V2X". L'ordinateur est conçu pour effectuer toutes ces tâches et améliorer l'expérience de conduite [6].

VI.4. Équipement médical

L'équipement médical intégré aux systèmes d'exploitation embarqués surveille automatiquement les constantes biologiques, administre les médicaments et envoie des alertes au personnel au cas où les constantes biologiques dépasseraient ou seraient inférieures à la valeur seuil. Ainsi, l'équipement médical peut aider les médecins à traiter leurs patients, à surveiller leur santé et à leur sauver la vie [6].

VII. QNX, VxWorks et LynxOS

VII.1. QNX

QNX est un système d'exploitation en temps réel commercial initialement développé pour les systèmes embarqués et développé à l'origine par Quantum Software Systems au début des années 1980. La société basée au Canada a ensuite été renommée QNX Software Systems et finalement acquise par Research in Motion (RIM) pour servir de base à leur système d'exploitation BlackBerry 10 et à leur populaire BlackBerry Playbook. La version du système d'exploitation utilisée dans le Playbook était QNX Neutrino, qui reposait sur une véritable architecture de micronoyau. QNX a été le premier système d'exploitation basé sur un micro-noyau à succès commercial. QNX est essentiellement un système d'exploitation de type Unix basé sur une véritable conception de micro-noyau et une architecture modulaire où les éléments du système d'exploitation s'exécutent en tant que tâches, appelées gestionnaires de ressources, ce qui permettrait aux développeurs de désactiver



toute fonctionnalité qu'ils pensent ne pas 'être requis dans le système particulier. Une particularité importante de QNX est qu'il utilise sur une communication interprocessus basée sur des messages [8].

VII.2.VxWorks

VxWorks est un système d'exploitation en temps réel spécialement conçu pour l'informatique distribuée pour les applications en temps réel avec des systèmes embarqués (première version en 1987). Il s'agit d'un système d'exploitation propriétaire en temps réel développé par Wind River Systems, une société californienne spécialisée dans le développement de logiciels embarqués pour les systèmes intelligents connectés. VxWorks est le RTOS leader du secteur qui construit des systèmes et des appareils embarqués depuis plus de trois décennies. C'est un noyau monolithique avec des communications inter-processus étendues et des fonctions de synchronisation. Chaque version de VxWorks est unique en raison de son architecture où l'ensemble du système d'exploitation fonctionne dans l'espace du noyau, ce qui signifie que tous les services du noyau et les services utilisateur existent dans le même espace d'adressage, ce qui accélère l'exécution du système d'exploitation. Il fournit des fonctions de débogage, de surveillance des performances, de gestion de la mémoire, de planification du processeur et d'autres fonctions du système d'exploitation via des appels système [8].

VII.3.LynxOS

LynxOS est un système d'exploitation compatible UNIX, conforme POSIX, multiprocessus et multithread conçu pour les applications temps réel complexes qui nécessitent une réponse rapide et déterministe. Le noyau LynxOS a été spécialement conçu pour les applications temps réel durs. Depuis sa première version en 1988, LynxOS est entièrement préemptif, ré-entrant (la **réentrance** est la propriété pour une fonction d'être utilisable *simultanément* par plusieurs tâches utilisatrices) et compact. La modularité inhérente à l'architecture LynxOS rend le système d'exploitation hautement évolutif et configurable. Au plus petit, LynxOS peut être configuré avec uniquement le noyau et lié à une application pour former une image ROMable pour les applications embarquées spécialisées. Au maximum, LynxOS est un environnement de développement auto-hébergé composé d'un large éventail d'outils de développement logiciel, d'utilitaires compatibles UNIX, d'un réseau standard de l'industrie, d'une interface utilisateur graphique et d'un système de fichiers hiérarchique de type UNIX [9].

Références

- [1] <https://internetofthingsagenda.techtarget.com/definition/embedded-operating-system>
- [2] <http://syst.univ-brest.fr/boukhobza/index.php/systemes-dexploitation-pour-lembarque>
- [3] https://fr.wikipedia.org/wiki/Syst%C3%A8me_embarqu%C3%A9
- [4] <https://docplayer.fr/73367884-Systemes-d-exploitation-embarques-traduit-et-ou-adapte-de-materiel-recueilli-sur-internet-dont-les-notes-de-chung-ta-king.html>
- [5] Systèmes Embarqués et Temps-Réel par Tristan Crolard
- [6] <https://www.digi.com/blog/post/what-is-an-embedded-operating-system>
- [7] <https://www.techno-science.net/glossaire-definition/Systeme-d-exploitation-page-4.html>
- [8] <http://www.differencebetween.net/technology/difference-between-qnx-and-vxworks/>
- [9] Exploring LynxOS, How-Shen Chang

