

Système d'exploitation pour IoT

Un système d'exploitation est le programme de base de Projets IoT. Le système d'exploitation IoT moderne utilise la technologie de *cloud computing* pour contrôler les appareils IoT partout dans le monde. Avec une faible empreinte mémoire et une efficacité supérieure, chaque système d'exploitation représenté ci-dessous peut répondre aux besoins de l'utilisateur.

1. Contiki/Contiki-NG

Inventé en 2002, Contiki est un système d'exploitation IoT open source particulièrement populaire pour les microcontrôleurs basse consommation et autres appareils IoT. Il fonctionne efficacement à l'aide des protocoles Internet IPv4 et IPv6. Ce système d'exploitation prend en charge la norme sans fil CoAP, 6LowPAN, RPL.

Ce système d'exploitation IoT est très approprié pour une connectivité Internet de faible puissance.

Aperçu de Contiki

- Contiki contient une suite de protocoles Internet intégrée.
- Seuls 10Ko de RAM et 30Ko de ROM sont nécessaires pour exécuter ce système d'exploitation.
- Le langage de base de ce système d'exploitation est le langage C. Un simulateur, appelé Cooja, permet de tester firmware développé.
- Le modèle de programmation Contiki utilise un type de programmation pour gérer efficacement la mémoire (Protothread.).
- Portée sur les cibles (par ex.) : TI MSP430x, Atmel AVR, Atmel Atmega128rfa1.

2. Android Things

Android Things (anciennement Brillo) est un système d'exploitation IoT inventé par Google. Il peut fonctionner en mode « faible consommation » et prend en charge les technologies Bluetooth et WiFi. Android Things vise à supprimer tous les obstacles et à simplifier le développement de l'IoT. Si Android Things fonctionne bien sur le marché, on peut s'attendre à ce que Google lance une boutique d'applications IoT.

Aperçu de Android Things

- Android Things n'utilise que 32-64Ko de RAM.
- Avec Android Things, Google annonce qu'il fournira un protocole de réseau de communication appelé Weave.
- Comme Android Things et Weave sont connectés, il est possible de détecter chaque appareil IoT par smartphone Android.
- Le kit de développement permet de tester, construire et déboguer chaque solution IoT.
- Android Things est une technologie open source et est régulièrement mise à jour (toutes les 6 semaines).

3. RIOT OS

Riot est un système d'exploitation IoT open source gratuit. Riot a une énorme communauté de développement (publié sous une Licence publique générale limitée GNU). Des universitaires, des amateurs et différentes entreprises ont mis leurs contributions en commun pour développer le système d'exploitation Riot.

Aperçu de RIOT OS

- Avec une approche « low power », Riot est construit sur une architecture de micronoyau avec le langage C, C++.
- Ce système d'exploitation IoT open source prend en charge le multithreading complet et les bibliothèques SSL/TSL, par exemple, wolfSSL.
- Utilisable sur des processeurs de 8bits, 16bits ou 32bits.
- Un portage de ce système d'exploitation permet de fonctionner comme Linux.
- Fournit des protocoles de mise en réseau et de réseau centrés sur le contenu tels que TCP, UDP et CoAp.

4. Apache Mynewt

Semblable à Riot, ce système d'exploitation IoT est conçu pour les petits appareils IoT intégrés. Il s'agit d'un système d'exploitation en temps réel sous licence Apache 2.0 qui fournit un environnement complet pour le développement, la gestion et l'exploitation des appareils IoT. Avec des bibliothèques riches, les systèmes d'exploitation modulaires comme Apache Mynewt peuvent fonctionner longtemps.

Aperçu d'Apache Mynewt

- Avec un noyau de 6KB, Mynewt est très utile pour construire des systèmes embarqués (équipements IoT industriels, dispositifs médicaux).
- Il intègre une pile Bluetooth 4.2 avec une approche à faible consommation d'énergie.
- Apache Mynewt gère la planification des priorités, le multithreading préemptif, le *watchdog*, l'allocation mémoire dynamique, etc.

5. Huawei LiteOS

En 2015, le géant chinois Huawei a lancé un système d'exploitation IoT nommé LiteOS. Celui-ci est un système d'exploitation sécurisé, interopérable et à faible consommation d'énergie. LiteOS utilise un *middleware* pour faciliter le développement. La taille du noyau est de 6KB.

Aperçu de Huawei LiteOS

- Intègre les protocoles suivants : NB-IoT, Ethernet, Bluetooth, Wifi, Zigbee, etc.
- Pour des raisons de sécurité, LiteOS fournit des mises à niveau à distance pour les terminaux, une authentification à deux facteurs et une transmission cryptée.
- Convient aux composants du système d'exploitation tels que la file d'attente, la mémoire, la gestion du temps et des tâches, etc.

- Selon un rapport, Huawei exporte 50 millions d'appareils IoT, chacun contenant LiteOS.
- L'utilisation des fonctions statiques, l'approche « faible consommation d'énergie » et son orientation « temps réel » sont les principales caractéristiques du noyau LiteOS.

6. Zephyr

Zephyr est un système d'exploitation temps réel (RTOS) conçu pour des applications IoT (soutenu par la Linux Foundation). L'intégration facile de diverses architectures IoT la rend populaire parmi les spécialistes de l'IoT. La technologie d'interconnectivité (exemple : Bluetooth LE, Wifi, 6Lowpan, NFC) est la caractéristique la plus importante de ce système d'exploitation IoT.

Aperçu de Zephyr

- Zephyr est un système d'exploitation IoT open source modulaire hautement configurable qui utilise la prise en charge de l'arborescence des appareils (DTS).
- Offre une suite complète de services de noyau, une prise en charge du stockage non volatile, une prise en charge des fichiers virtuels, etc.
- 8KB de RAM et 512KB de ROM sont nécessaires pour faire fonctionner ce système d'exploitation.

7. TinyOS

TinyOS est un système d'exploitation open source. Le langage de base de TinyOS est nesC qui est un dialecte du langage C. TinyOS est populaire parmi les développeurs pour ses caractéristiques d'optimisation de la mémoire. Le groupe de développeurs de ce système d'exploitation IoT est TinyOS Alliance.

Aperçu de TinyOS

- ESTCube-1 est un programme spatial qui utilise ce système d'exploitation.
- Les protocoles réseau, les pilotes de capteurs, les outils d'acquisition de données font partie des bibliothèques de composants.
- Utiliser principalement des réseaux de capteurs sans fil car son architecture est conçue de cette manière.

8. Fuchsia

Les exigences des divers systèmes d'exploitation IoT augmentent de jour en jour. Fuchsia est un système d'exploitation basé sur un micro-noyau composé de solutions de connectivité efficaces. Fuchsia fonctionne bien dans les appareils de faible puissance. Les informaticiens pensent qu'il est possible que Fuchsia remplace le système d'exploitation Android dans un avenir proche.

Aperçu du Fuchsia

- L'utilisation de Node.js sur le système d'exploitation garantit que l'application puisse s'exécuter sur un téléphone, une tablette et des appareils IoT.
- Les langages de développement de ce système d'exploitation est Dart, Go, Rust, C/C++.
- Utilise le kit de développement logiciel Flutter pour l'interface utilisateur et les applications.
- Le nom du micro-kernel de Fuchsia est Zircon.

9. Windows IoT

Windows 10 IoT fait partie de la famille de systèmes d'exploitation Windows 10. Windows IoT est divisé en deux parties. La première est le noyau Windows 10 IoT pour prendre en charge les petits appareils intégrés. La deuxième est Windows 10 IoT Enterprise pour une perspective industrielle.

Aperçu de Windows IoT

- Le système d'exploitation s'exécute sur le processeur ARM.
- Il tire parti de la connectivité IoT, de l'expérience du cloud et propose à diverses organisations de se connecter aux appareils IoT.
- Le noyau Windows IoT offre une *gérabilité* comme le système d'exploitation Windows 10, bien qu'il agisse comme une application.
- Avec le noyau hybride, ce n'est pas un système d'exploitation open source.

10. TizenRT

Il s'agit d'un système d'exploitation basé sur Linux inventé en 2011 pour les applications mobiles et les petits systèmes embarqués. Une version améliorée de TizenRT peut prendre en charge les téléviseurs intelligents, les véhicules, les appareils ménagers, etc. Samsung Electronics utilise ce système d'exploitation principalement pour son développement IoT. Le développement de TizenRT s'est déroulé de manière à offrir une flexibilité aux développeurs d'applications, aux fabricants d'appareils et aux opérateurs de téléphonie mobile.

Aperçu de TizenRT

- Il utilise une infrastructure partagée appelée « Tizen Common » pour soutenir l'objectif principal du développement de l'IoT.
- Cet IoT OS a été développé pour que les opérateurs mobiles puissent personnaliser leurs produits en fonction des besoins démographiques d'une zone.
- Les langages de programmation C, C++ et html5.
- Le type de noyau de TizenRT est monolithique et comparable à AndroidOS, et celui-ci est beaucoup plus léger.

11. Raspbian

Raspbian est l'un des OS les plus utilisés pour le développement de l'IoT. Raspbian est très flexible pour les processeurs des cartes Raspberry Pi. Raspbian fournit un grand nombre de logiciels préinstallés, à des fins expérimentales, éducatives, etc. Il s'agit d'un système d'exploitation IoT basé sur Debian pour tous les modèles de Raspberry Pi.

Aperçu de Raspbian

- Le développement actif de Raspbian se poursuit alors que la demande pour ce système d'exploitation augmente.
- Raspbian Buster et Raspbian Stretch sont deux versions du système d'exploitation Raspbian.
- L'environnement de bureau principal est PIXEL, qui est un environnement x-window amélioré par Raspberry PI.
- Le noyau est similaire au noyau Unix.

12. Amazon FreeRTOS

Amazon FreeRTOS est un système d'exploitation open source. Les bibliothèques logicielles enrichies facilitent la connexion avec les petits appareils IoT. Ce système d'exploitation IoT utilise le service cloud d'Amazon Web Service appelé AWS IoT Core pour exécuter les applications IoT. L'empreinte mémoire n'est que de 6 à 15KB.

Aperçu d'Amazon FreeRTOS

- La modularité du code, les fonctionnalités de hiérarchisation des tâches aident à respecter les délais de traitement avec une optimisation de la puissance.
- L'utilisation du profil d'accès générique standard et du profil d'attribut générique (GAP) via Bluetooth low energy le rend plus efficace.
- Amazon a investi beaucoup d'argent dans le développement de Sécurité des données IoT.
- Le testeur d'appareils IoT garantit la possibilité d'intégration des appareils IoT avec le service cloud.
- Il est devenu un standard des systèmes d'exploitation basés sur des microcontrôleurs au cours des dernières années.

13. Linux embarqué

Linux embarqué est un système d'exploitation conçu pour les appareils embarqués, bien qu'il utilise le noyau Linux. La taille et la puissance réduites de Linux embarqué permettent d'intégrer toutes les exigences des appareils IoT. Le système d'exploitation Android utilisait Linux embarqué avec des interfaces personnalisées pour le smartphone. De même, ce système d'exploitation IoT s'applique également aux téléviseurs intelligents, aux appareils de navigation, aux tablettes, aux téléviseurs intelligents, aux routeurs sans fil, etc.

Aperçu de Linux embarqué

- Comme Linux embarqué est gratuit et open-source avec une large communauté, de nombreux contributeurs mettent à jour régulièrement ce système d'exploitation.
- Les plateformes logicielles qui utilisent des noyaux Linux intégrés sont Busybox, Mobilinux et Maemo.
- Il utilise 100KB d'espace mémoire, ce qui le rend plus rapide et fiable.
- Applications embarquées (exemple : SQL Lite, Boa, thttpd, PEG, NANO) prises en charge.

14. Système d'exploitation Mbed

Pour le développement de produits embarqués IoT, le système d'exploitation Mbed utilise un processeur ARM. Il s'agit d'un système d'exploitation gratuit et open source axée sur les projets IoT. Un nombre important d'options de connectivité incluent Wifi, Bluetooth, 6LowPan, Ethernet, Cellular, RFID, NFC, Thread, etc. La sécurité multicouche de ce système d'exploitation IoT offre une grande fiabilité.

Aperçu de Mbed OS

- Le développeur peut créer un prototype d'applications IoT à l'aide d'appareils ARM cortex M.
- À partir de la riche bibliothèque, les mises à jour s'ajoutent automatiquement aux applications IoT.
- L'API Mbed OS peut garder votre code propre et portable.
- Utilise les protocoles de sécurité SSL et TLS pour la sécurité des applications en ligne.
- Il fournit un grand nombre d'exemples de code pour montrer comment intégrer l'API sur chaque cible.