Systèmes d’exploitation embarqués

Robotique et Systèmes Embarqués

Samuel Tardieu

samuel.tardieu@enst.fr

École Nationale Supérieure des Télécommunications

Institut de la Francophonie pour l’Informatique

Plan du cours

- Caractéristiques des systèmes d’exploitation
- Besoins de l’informatique embarquée
- Présentation de quelques systèmes

Système d’exploitation

Un système d’exploitation:

- fait le lien entre le logiciel (application) et le matériel
- abstrait certaines caractéristiques du matériel
- fournit des services communs (accès aux ressources, synchronisation, gestion de fichier)
- offre des possibilités de tests et de traces

Système embarqué

Généralement, un système embarqué:

- dispose de ressources limitées
- ne possède pas toujours de système de fichiers
- doit être le moins cher possible
- ne doit pas consommer d’énergie inutilement

Approche minimaliste

Dans une approche minimaliste, le système d'exploitation:

- initialise le matériel
- donne la main à l'application

Le système d'exploitation peut également être réduit à néant (carte nue).

Approche maximaliste

Dans une approche maximaliste, le système d'exploitation:

- fournit tous les services d'un ordinateur de bureau
- fournit des machines virtuelles
- fournit des bibliothèques graphiques

Exemple: le projet SPIF de l'ENST (Linux)

Approche intermédiaire

En général, le système d’exploitation fournit les services:

- de gestion de temps
- d’accès au réseau
- de gestion de la mémoire

Compromis

Des compromis doivent être faits:

- mono-programme / multi-programme
- mono-thread / multi-thread
- allocations statiques / allocation dynamique
- réservation des ressources / allocation des ressources à la demande

Exemple: PalmOS

- gestion de la mémoire simplifiée
- primitives de gestion de bases de données et de l'écran
- bibliothèques mathématiques
- applications minimalistes (philosophie du Palm)
- mono-application et mono-thread
- ne nécessite pas beaucoup de puissance (m68k 20MHz)

Exemple: Windows CE

- tous les services d’un Windows
- fiabilité d’un Windows (mauvaise)
- facilité de portage des applications
- pas de gestion du temps-réel
- nécessite un processeur très puissant (ARM 400MHz)

Exemple: Symbian OS

- orienté « téléphonie »
- gestion des contacts
- gestion de réseaux divers (SMS, BlueTooth, GSM, TCP/IP)
- gestion multimedia
- synchronisation sur réseau lent « over the air »
- supporte Java (JavaPhone)
- nécessite moins de ressources que Windows CE

Exemple: Linux

- tous les services de Linux
- fiabilité de Linux (bonne)
- multi-applications
- très gourmand en ressources
- logiciel libre
- nécessite un processeur puissant (PowerPC 50MHz)

Exemple: RTEMS et eCos

- exécutif configurable pour ne garder que ce qui est nécessaire
- services de synchronisation
- gestion du temps
- entrées/sorties
- gestion du réseau
- logiciel libre
- ne nécessite pas beaucoup de puissance (m68k à 33MHz)

Exemple: Forth

- langage de programmation et système à lui tout seul
- mono-application mais multi-threads en mode coopératif
- permet le test interactif
- ne nécessite pas beaucoup de ressources (PIC 16f876, 2k de programme, 80 octets de RAM)

Caractéristiques

Avant de choisir son système, il faut définir:

- le prix de revient maximum de chaque unité
- le temps de réponse aux actions de l'utilisateur
- la capacité totale de traitement
- la disponibilité et la possibilité de le modifier
- le support disponible et la fiabilité du fournisseur