

La Carte à Microprocesseur

Un système embarqué en plein essor

Tegawendé F. Bissyandé

tegawende.bissyande@fasolabs.org

Cours préparé pour
L'Institut Supérieur de Technologie (IST Burkina)

April 1, 2015

La carte à puce dans nos vies...



Cartes spécifiques

- Réservées aux grands groupes industriels (*Cartes bancaires, cartes SIM, cartes de TV cryptées*)
- En général pour production de masse
- Lourde (*en terme de mise en oeuvre*) mais souple (*dans l'installation de l'OS, les fichiers, les instructions*)
- Besoin d'outils spécifiques fournis par le fabricant de la puce
- Coût élevé amorti par la production en masse \Rightarrow s'adresser à un encarteur tel Gemalto

Cartes personnalisables

- Développement possible d'une application
- OS déjà choisi par le fabricant de la carte / fonctionnalités sont limitées par le fabricant
- Le fabricant a aussi prévu des commandes propriétaires ciblant une certaine catégorie d'applications (*commandes de débit/crédit/calcul de solde pour paiement électronique*)

Cartes à OS ouverts

- Carte à puce "non terminée" – programmer son propre interpréteur de commandes
- Le microcontrôleur est programmé dans un langage évolué (tel Java) qui est précompilé

Cartes spécifiques

- Réservées aux grands groupes industriels (*Cartes bancaires, cartes SIM, cartes de TV cryptées*)
- En général pour production de masse
- Lourde (*en terme de mise en oeuvre*) mais souple (*dans l'installation de l'OS, les fichiers, les instructions*)
- Besoin d'outils spécifiques fournis par le fabricant de la puce
- Coût élevé amorti par la production en masse \Rightarrow s'adresser à un encarteur tel Gemalto

Cartes personnalisables

- Développement possible d'une application
- OS déjà choisi par le fabricant de la carte / fonctionnalités sont limitées par le fabricant
- Le fabricant a aussi prévu des commandes propriétaires ciblant une certaine catégorie d'applications (*commandes de débit/crédit/calcul.de.solde pour paiement électronique*)

Cartes à OS ouverts

- Carte à puce "non terminée" – programmer son propre interpréteur de commandes
- Le microcontrôleur est programmé dans un langage évolué (tel Java) qui est précompilé

Cartes spécifiques

- Réservées aux grands groupes industriels (*Cartes bancaires, cartes SIM, cartes de TV cryptées*)
- En général pour production de masse
- Lourde (*en terme de mise en oeuvre*) mais souple (*dans l'installation de l'OS, les fichiers, les instructions*)
- Besoin d'outils spécifiques fournis par le fabricant de la puce
- Coût élevé amorti par la production en masse \Rightarrow s'adresser à un encarteur tel Gemalto

Cartes personnalisables

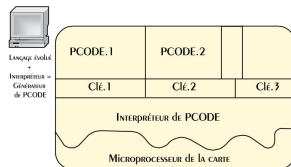
- Développement possible d'une application
- OS déjà choisi par le fabricant de la carte / fonctionnalités sont limitées par le fabricant
- Le fabricant a aussi prévu des commandes propriétaires ciblant une certaine catégorie d'applications (*commandes de débit/crédit/calcul.de.solde pour paiement électronique*)

Cartes à OS ouverts

- Carte à puce "non terminée" – programmer son propre interpréteur de commandes
- Le microcontrôleur est programmé dans un langage évolué (tel Java) qui est précompilé

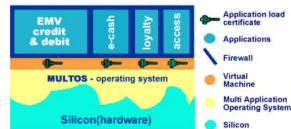
Cartes à OS ouvert (1)

- Précompilation vers un code intermédiaire : le P code
- Microcontrôleur est programmé avec un interpréteur de P code
- + Carte réellement programmable
- + Carte programmée en langage évolué (C, Basic, Java)
- + Pas de contrainte de production en masse
- + Application développée est portable



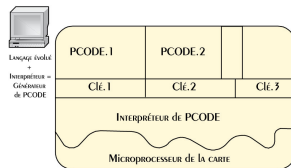
Systeme Multos

- L'ancêtre des cartes à OS ouverts
- OS multi-application proposé initialement par Mondex et Master Card
- Chaque application dispose d'un pare-feu
- Ajout et "Mise à jour" possible d'applications sans le fournisseur



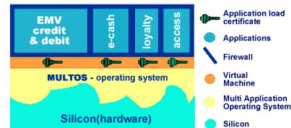
Cartes à OS ouvert (1)

- Précompilation vers un code intermédiaire : le P code
- Microcontrôleur est programmé avec un interpréteur de P code
- + Carte réellement programmable
- + Carte programmée en langage évolué (C, Basic, Java)
- + Pas de contrainte de production en masse
- + Application développée est portable



Systeme Multos

- L'ancêtre des cartes à OS ouverts
- OS multi-application proposé initialement par Mondex et Master Card
- Chaque application dispose d'un pare-feu
- Ajout et "Mise à jour" possible d'applications sans le fournisseur



La Basic Card

- Commercialisée par une petite société allemande (Zeit Control)
- Fonctionne comme les autres modèles à OS ouverts
- originalité: le P code provient de la compilation d'un programme en Basic
- Meme langage utilisé pour la carte et pour le terminal



La Java Card

- Se programme bien sûr en Java
- La carte contient un interpréteur de *bytecode*
- En developpement, la machine virtuelle Java est donc scindée en deux
- En fait, un sous-ensemble du langage java est supporté par la Java Card

Intérêt

- 1 La Java Card est vraiment normalisée
- 2 Sûreté des applications assurée par le langage
- 3 La Java Card autorise la coexistence de plusieurs applications
- 4 La Java Card supporte la modification dynamique des applications (cycle de vie)

La Basic Card

- Commercialisée par une petite société allemande (Zeit Control)
- Fonctionne comme les autres modèles à OS ouverts
- originalité: le P code provient de la compilation d'un programme en Basic
- Meme langage utilisé pour la carte et pour le terminal

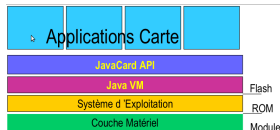


La Java Card

- Se programme bien sûr en Java
- La carte contient un interpréteur de *bytecode*
- En developpement, la machine virtuelle Java est donc scindée en deux
- En fait, un sous-ensemble du langage java est supporté par la Java Card

Intérêt

- 1 La Java Card est vraiment normalisée
- 2 Sûreté des applications assurée par le langage
- 3 La Java Card autorise la coexistence de plusieurs applications
- 4 La Java Card supporte la modification dynamique des applications (cycle de vie)



- Pas de chargement dynamique de classe
- Allocation dynamique d'objets supportée mais
 - pas de garbage collection
 - pas de désallocation explicite non plus :(
 - ...
- Quelques types de base (byte, int, boolean) – pas de char (ni de classe String), double, float, long
- Objets supportés / Mécanisme d'héritage supporté / Pas de threads
- Sécurité : Notion de paquetages & modifieurs public, private, protected

