

Systèmes d'exploitation

Introduction et historique

G. Arnould

IUT de Metz, dpt. Info

31 janvier 2009

Objectif : qu'est-ce qu'un système d'exploitation ?

Connaître les bases théoriques minimales des systèmes d'exploitation.

Contenu du cours

- Types et caractéristiques des systèmes d'exploitation
- Fichiers (types, droits, etc...)
- Commandes simples et paramétrées
- Programmes de commandes, scripts
- Programmes, processus, interaction avec le système d'exploitation

Présentation du cours

Introduction

Objectif : création d'un programme

Comprendre la chaîne de production d'un exécutable.

Contenu du cours

- Processus de compilation
- Bibliothèques logicielles
- Traduction, édition des liens, chargement
- Gestion et liaison des objets : portée, durée de vie

D'où viennent les systèmes d'exploitation ?

Machines à calculer

Au commencement, il y avait ...
... les machines à calculer mécaniques.

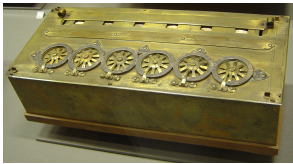


FIGURE: « Pascaline » –
Machine de Pascal (Musée des Arts et
Métiers – Photographie (c) 2005 David Monniaux)

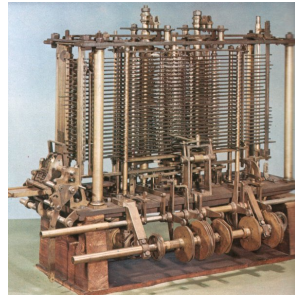


FIGURE: Machine de Babbage
(Whipple Museum of the History of Science –
University of Cambridge – Photographie (c) 2004
Andrew Dunn)

5/34



Machine différentielle : Charles Babbage (1833)

- Ordinateur numérique programmable
- Carte perforées
- Ada Lovelace : première série de cartes

Rôle et utilisation

- Tabulation de fonctions
- Opérations complexes par approximations
- Possibilité de répéter un même calcul
- Programmation et exécution séparées

Limitations

- Complexité des cartes perforées
- Machine à vapeur

7/34



«Pascaline» : Blaise Pascal (1642)

- calculateur mécanique
- 1673, améliorée par Leibnitz : mémoire, multiplications

Rôle et utilisation

- Calculs complexes
- Production *rapide* d'un résultat
- Traiter une grande quantité d'informations

Limitations

- *Mode opératoire* complexe
- En cas d'*erreur* : il faut tout recommencer
- *Un seul opérateur* à la fois

6/34

Bilan : machines à calculer mécaniques

Elles permettent :

- calculs complexes, rapide, bonne précision
- répéter des opérations
- programmation *puis* exécution

Limitations

D'un point de vue structurel :

- mise en œuvre complexe (vapeur, mécanique, etc...)
- réparations difficiles, évolutions presque impossibles

Principe de fonctionnement :

- une tâche après l'autre
- opérateur «humain» nécessaire
- programmation complexe : erreurs difficiles à repérer sur une carte perforée. Comment vérifier les résultats ?

8/34

Difficile évolution

... jusqu'en 1940

Machines à calculer → ordinateurs mécaniques

- rapidité de calcul
- puissance
- complexité des fonctions traitées
- programmes plus longs
- nombres à virgules flottantes, calcul différentiel, etc...

Progrès restreint

- toujours les mêmes limitations
- peu d'évolutions possibles
- limites technologiques atteintes

Évolution ← dépendante de nouvelles inventions

9/34

Électronique

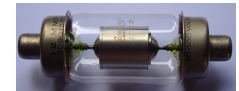
Avant 1945

- moteurs électriques
- machines analogiques
- éléments mécaniques



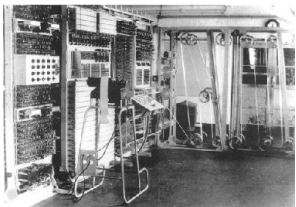
1945 – 1955

- relais
- condensateurs
- circuits électroniques
- tubes à vide



10/34

Premiers ordinateurs entièrement numériques



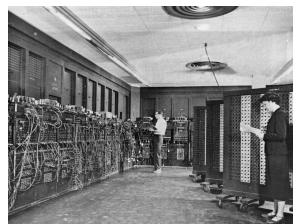
Colossus – 1943

- premier instrument de calcul complètement électronique
- déchiffrer les communications militaires allemandes

ENIAC – 1945



× 20000 =



11/34

ENIAC

Electronic Numerical Integrator and Computer

- **très fragile** : durée maximale de fonctionnement en continu de 116h !
- **très lent** :
 - fréquence : 100kHz
 - 5000 additions par seconde
 - 357 multiplications par seconde
 - 35 divisions par seconde
- pas de **langage de programmation**
- tableaux d'interrupteurs pour entrer les opérations
- un opérateur après l'autre

Stockage / entrée des données – 1950 - 1970

- cartes perforées avec enrouleurs automatiques
- bandes magnétiques

12/34

Miniaturisation...

Progrès de l'électronique

Première génération



FIGURE: l'IBM 650 prenait toute une salle. 900kg + 1350kg pour son alimentation.

Seconde génération

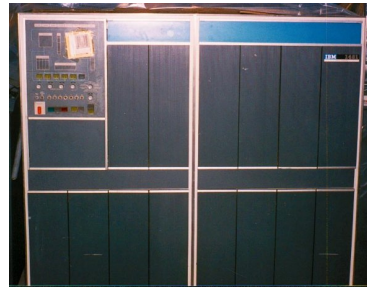


FIGURE: IBM 1401 : pas plus *gros* qu'une armoire



1958 : circuit intégré

- diminution de la taille des machines
- baisse des coûts
- hausse des performances

1968 : microprocesseur



Rationalisation de la production

- Production en nombre
- Gammes de machines
- puissance variable, adaptée à la demande

13/34

14/34

File d'attente ou fil à linge?

Toujours les mêmes limitations !

- un seul travail à la fois
- entrée des programmes : par un opérateur
- certains programmes plus **urgents** que d'autres

Université de Cambridge : **fil à linge** et **pincettes de couleur** pour trier les cartes perforées...

Du rôle de l'opérateur

- diminution de la durée d'exécution d'une tâche
- plus de tâches : moins de temps pour changer de tâche
- tâches plus complexes
- grand nombre de cartes perforées
- **risque d'erreur accru**

15/34

De nouveaux besoins

Gestion :

- des files d'attentes
 - des tâches à l'entrée de l'ordinateur
 - des résultats à la sortie
- des ressources de calcul
 - le temps de calcul coûte cher
 - partage de ressources
- de la compatibilité entre machines d'une même gamme

Hiérarchie des tâches

- plus efficace
- automatique
- plus rapide

Gestion du matériel – systèmes de :

- surveillance du(des) processeur(s)
- détection des pannes
- comptage des pages imprimées, des bandes magnétiques, etc...

16/34

Qu'est-ce qu'un système d'exploitation ?

17/34

Évolution des systèmes informatiques

- un ou plusieurs processeurs
- mémoire vive, mémoire ROM
- horloges
- terminaux
- stockage : disques durs, supports amovibles, clés USB, CD-ROM, DVD-ROM, ...
- interfaces réseau
- périphériques d'entrée sortie

Diversité

Fabriquants de périphériques :

- nombreux
- incompatibilité

Besoin de simplification $\Rightarrow$ logiciel de gestion du matériel

18/34

Problématique (2)

Deux types de programmes principaux :

Logiciels applicatifs

Résoudre les problèmes des utilisateurs :

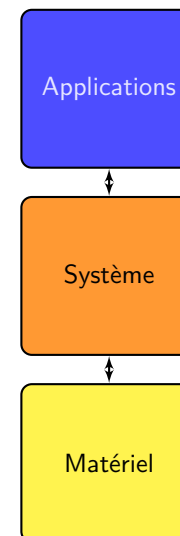
- calcul scientifique
- traitement de texte, tableur, ...
- application multimédia
- jeux
- ...

Logiciels système

Gestion des ressources, fonctionnement de la machine

19/34

Représentation par couches : point de vue utilisateur



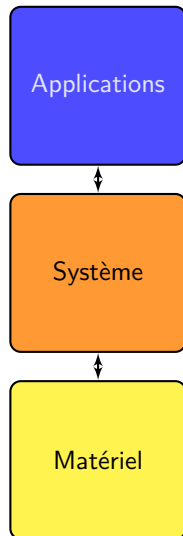
Point de vue de l'utilisateur

Applications :

- traitement de texte
- jeu vidéo
- navigateur www
- ...

20/34

Représentation par couches : programmes système



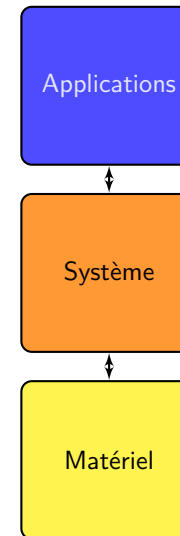
Programmes système

Deux sous couches :

- ① Applications système
 - Compilateurs C, C++, Delphi, ...
 - Machines virtuelles Python, Java, .NET, ...
 - Interprètes de commande : bash, csh, cmd.exe, ...
- ② Le système d'exploitation

21/34

Représentation par couches : partie matérielle

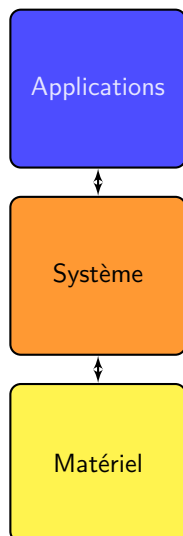


Couche matérielle

- ① Langage machine, assembleur, ...
- ② Microcode, firmware
- ③ Périphériques :
 - externes : écran, clavier, souris, imprimante, ...
 - internes : processeur(s), mémoire vive, carte graphique / son, disques durs, ...

22/34

Représentation par couches : partie matérielle (2)



Matériel

- composants électroniques
- carte mère, mémoire, stockage, périphériques d'E/S

Microprogramme

- logiciel de contrôle de périphérique
- interface au système physique
- BIOS, firmware de lecteur de DVD, de carte graphique, ...

Langage Machine

- instructions élémentaires
- connues par le processeur, le périphérique
- ≈ assembleur : ADD, MOVE, JUMP, instructions MMX, etc...

23/34

Abstraction et simplification

Problème

- Même fonction, grande variété de périphériques
 - carte graphique : Intel, NVidia, ATI/AMD, ...
 - interfaces réseau : Realtek, Netgear, Cisco, ...
- **mais** interfaces de programmation différentes...

Solution : système d'exploitation

- machine virtuelle gérant le matériel
- gestionnaire de ressources

24/34

Machine virtuelle

- instructions simples pour masquer la complexité réelle :
 - READ / WRITE : commandes contrôleur de disque dur
 - 13 paramètres sur 9 octets
 - gestion du déplacement des bras du lecteur
 - gestion des mémoires cache
 - ...
 - commandes système équivalentes :
 - read (descripteur, tampon, taille);
 - write (descripteur, tampon, taille);
- gestion des interruptions, des horloges, de la mémoire, interopération entre les périphériques, ...

25/34

Gestionnaire de ressources

- périphériques physiques
- plusieurs programmes utilisant ces périphériques
- plusieurs tâches simultanées : multitâches
- accès concurrent à tous ces périphériques ?

Problème

- fichier pdf, image, document texte
- envoyés par trois utilisateurs à une même imprimante
- comment gérer le conflit ?

26/34

Niveaux

Types de systèmes d'exploitation

Différents niveaux de fonctionnement

- ① Processeur : ordonnancement et arbitrage
- ② Système de fichiers : représentation des données sous forme *d'objets* (fichiers, répertoires, ...)
- ③ Opérations d'entrée / sortie : interface d'accès aux périphériques via des pilotes
- ④ Accès à la mémoire : partage, pagination, segmentation
- ⑤ Concurrence d'accès aux ressources
- ⑥ Sécurité et protection des accès : droits, segmentation, ...
- ⑦ Ressources distribuées : accès au réseau, échange de données, répertoires partagés, ...

27/34

Les grandes catégories de systèmes d'exploitation

28/34

Multitache ?

Systèmes monotâche

- un seul processus à la fois
- toutes les tâches sont exécutées en séquence
- MS-DOS, Windows 3.1, certains systèmes embarqués, Palm OS, ...

Systèmes multitâches

- plusieurs processus exécutés *simultanément*
- UNIX, IBM OS/2, Windows 95 et suivants, ...

29/34

Multiutilisateurs ?

Multi ou monutilisateur ?

- plusieurs utilisateurs se partagent les mêmes ressources matérielles
- accès partagé au système de fichier, au processeur
- besoin d'un système de priorités
- tâches urgentes ou non, ...

30/34

Type d'architecture ?

Architecture centralisée

- système complet (processeur, mémoire, périphériques, ...) groupé sur une seule machine physique
- SE : gestion des ressources locales uniquement
- UNIX, Windows, Mac OS, ...

Architecture décentralisée

- système réparti sur un ensemble de machines
- machine virtuelle : toutes les ressources sont présentées comme locales
- parallélisme, redondance, ...
- UNIX, Cluster, SETI, Google, ...

31/34

Licences et autres considérations

Systèmes fermés – systèmes propriétaires

- code source fermé
- fonctionnalités bloquées
- évolution limitée
- modèle de développement *pyramidal*
- UNIX, MS-DOS, Windows, Mac OS, Symbian, ...

Systèmes ouverts

- code source disponible
- architecture plus modulaire
- modèle de développement : souvent *bazar*
- {Net, Free, Open}BSD, GNU/Linux, Darwin, Rockbox, ...

32/34

Architecture mono-processeur

- plupart des systèmes grand public
- parallélisme virtuel : multiprogrammation
- multiplexage temporel entre les différentes tâches

Architecture multi-processeurs

- systèmes SMP, bi / quadri coeurs, ...
- exécution simultannée de plusieurs tâches, traitement séquentiel rapide

Systèmes temps-réel

- contraintes temporelles fortes
- validité des résultats d'un programme limitée dans le temps
- systèmes de pilotage :
 - avions, satellites, trains
 - centrales électriques
- systèmes embarqués / mobiles
 - téléphones portables
 - baladeurs numériques
 - PDA
 - ...
- cf. Linux RT