



Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel

Systèmes d'Exploitation

Introduction

Didier Verna

didier@lrde.epita.fr
<http://www.lrde.epita.fr/~didier>



Table des matières

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel

- 1 Introduction
- 2 Préhistoire
- 3 Histoire
- 4 Autres Systèmes contemporains
- 5 Organisation des systèmes d'exploitation



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

Ça ?



Ça ?



Que dire de `command.com`, des shellutils, de X11... ?

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



Définition pragmatique

Systèmes
d'Exploitation

Didier Verna
EPITA

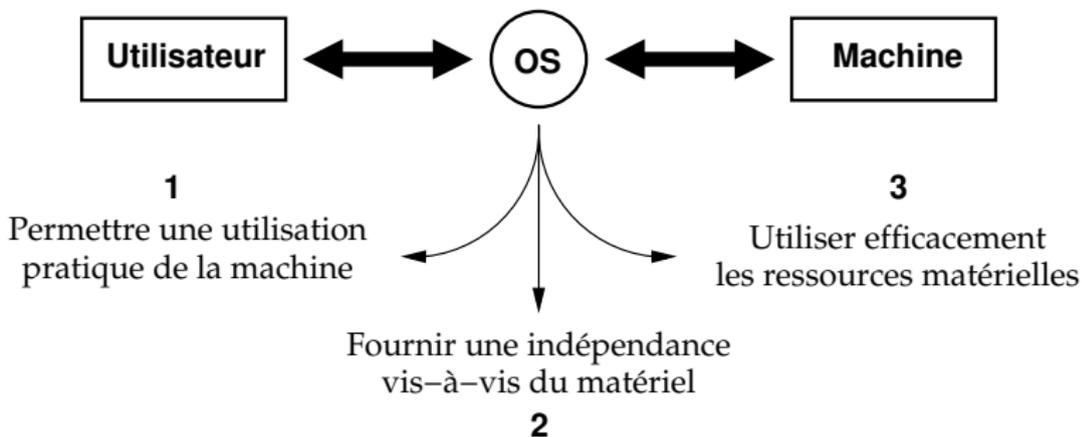
Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel





Définition(s) technique(s)

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel

■ La plus répandue :

- ▶ Allocateur de ressources
(CPU, mémoire, périphériques *etc.*)
- ▶ Contrôleur de ressources
(gestion et contrôle dans le temps et dans l'espace)

■ Autres :

- ▶ Première couche logicielle au dessus du matériel
- ▶ Seul programme permanent dans une machine
- ▶ Portion logicielle qui tourne en mode noyau



1642 : Pascal

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://davidguy.brinkster.net/computer/>

Machines Arithmétiques : calcul dirigé manuellement.



1801 : Jacquard

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



http://lyc-jacquard.scola.ac-paris.fr/historique_1p.html

Automates : métier à tisser doté d'un programme sur une carte en bois permettant de gérer le mouvement des engrenages.



1843 : Charles Babbage / Ada Byron

Systèmes
d'Exploitation

Didier Verna
EPITA

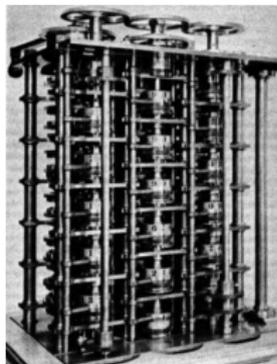
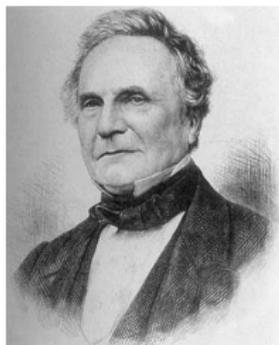
Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://www-gap.dcs.st-and.ac.uk/~history/>

Automate pour le calcul **arithmétique** : programme sur une carte en bois, « magasin » pour le stockage des résultats intermédiaires, automate pour l'exécution du programme.



1936 : Alan Turing

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://www-gap.dcs.st-and.ac.uk/~history/>

Définition mathématique précise de la notion d'algorithme.
Modèle abstrait pour le calcul informatique. « Machine de Turing ».



1945 : Von Neumann

Systèmes
d'Exploitation

Didier Verna
EPITA

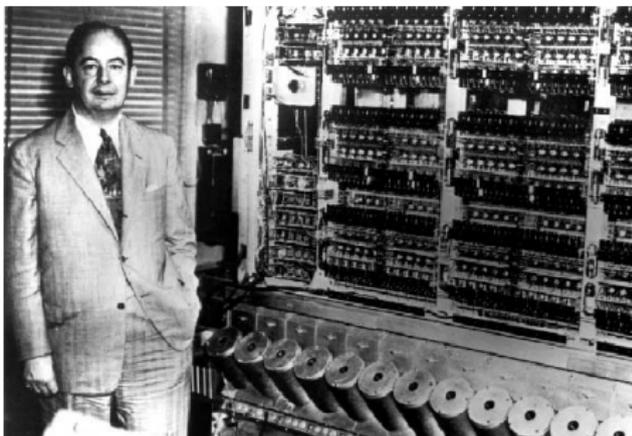
Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://www-gap.dcs.st-and.ac.uk/~history/>

Stockage du programme dans le « magasin » (la mémoire), codage de l'information en binaire. Seul le « logiciel » sait distinguer les instructions et les données.



1945 – 1955 : lampes et jacks

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel

- **Machines énormes**

Dizaines de milliers de relais mécaniques progressivement remplacés par des lampes

- **Utilisation 100% manuelle**

Directement en langage machine, panneaux de contrôle électriques.

- Pas de distinction entre concepteurs, administrateurs et utilisateurs

- **1950** : apparition des cartes perforées

⇒ Temps de préparation considérable, unité centrale peu active.



ENIAC (1)

Electronic Numerical Integrator and Computer

Systèmes
d'Exploitation

Didier Verna
EPITA

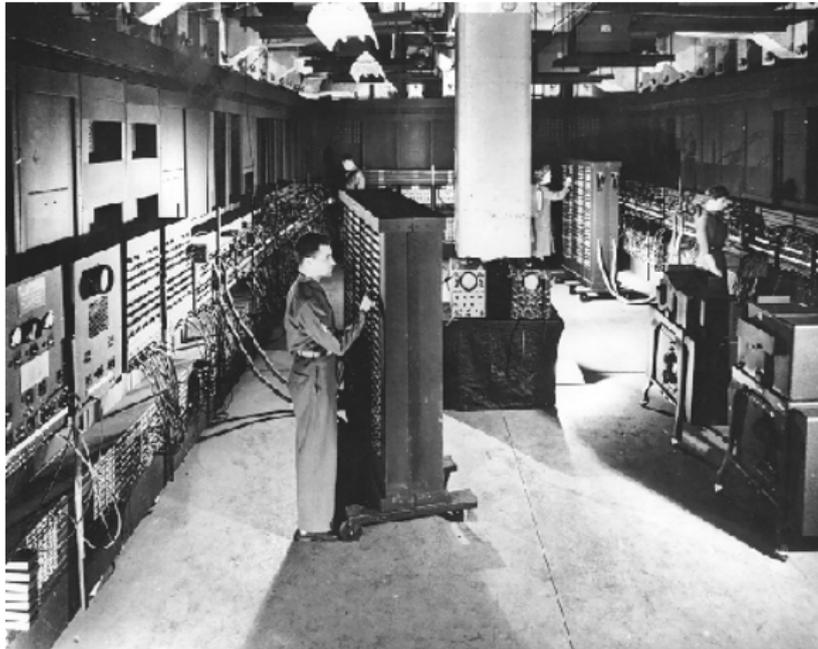
Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://www-gap.dcs.st-and.ac.uk/~history/>



ENIAC (2)

Systèmes
d'Exploitation

Didier Verna
EPITA

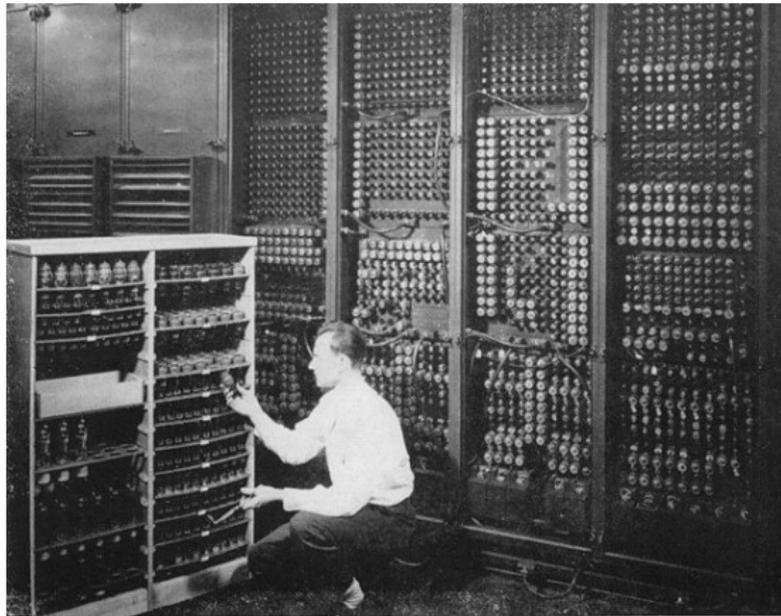
Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

<http://www-gap.dcs.st-and.ac.uk/~history/>



ENIAC (3)

Systèmes
d'Exploitation

Didier Verna
EPITA

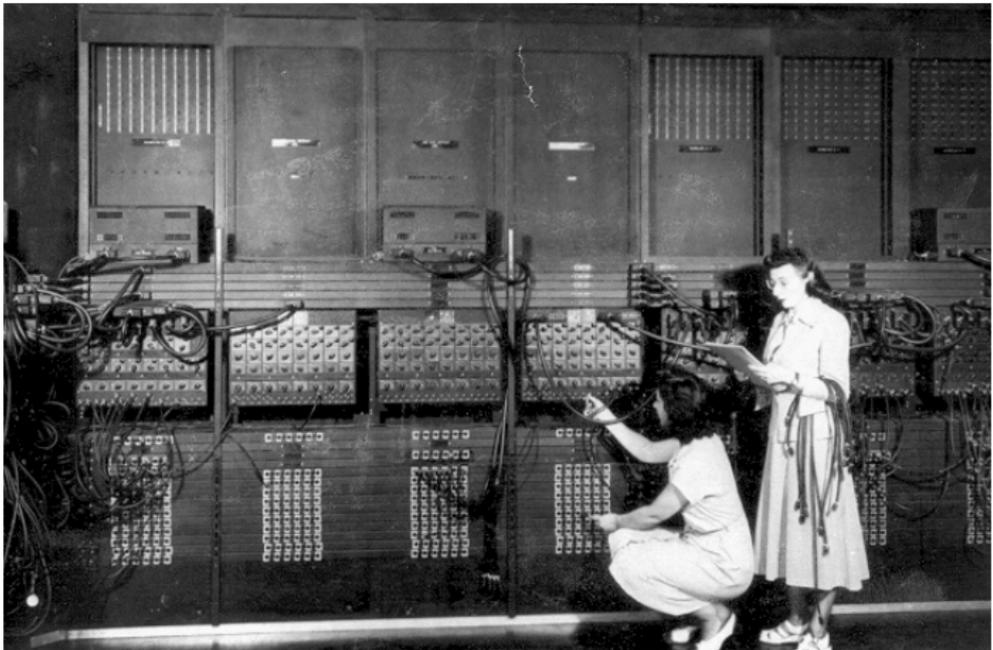
Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://www-gap.dcs.st-and.ac.uk/~history/>



ENIAC (4)

Systèmes
d'Exploitation

Didier Verna
EPITA

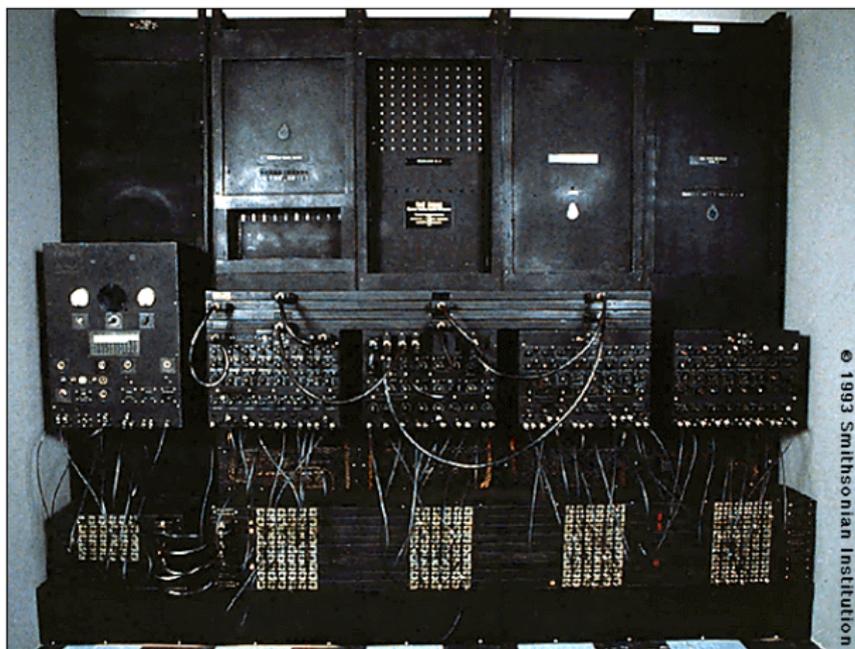
Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://www-gap.dcs.st-and.ac.uk/~history/>



ENIAC (5)

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://www-gap.dcs.st-and.ac.uk/~history/>



1955 – 1965 : traitement par lots (batch)

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel

■ Utilisation de transistors

Machines plus fiables (mainframes)

■ Opérateur professionnel

- ▶ Engagé pour s'occuper du système
- ▶ Enchaînement des tâches, surveillance de leur déroulement *etc.*
- ▶ Le programmeur n'a plus accès à la console

■ Mise en commun des tâches similaires

Exemple : deux programmes FORTRAN utilisent le même compilateur

⇒ Toujours de longues périodes d'inactivité de l'unité centrale.



Traitement par lots (suite)

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel

■ **Traitement hors ligne (offline)**

Utilisation de bandes magnétiques comme buffers E/S de l'unité centrale (IBM-7094). Conversion cartes ↔ bandes sur des machines distinctes (IBM-1401).

■ **Moniteur résident**

Premier système d'exploitation rudimentaire : programme d'enchaînement automatique des tâches contrôlé par des cartes spéciales (\$LOAD, \$RUN *etc.*).

⇒ Aucune interactivité dans le traitement des tâches.



IBM-7094

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://www.columbia.edu/acis/history/>



Console 7094

Systèmes
d'Exploitation

Didier Verna
EPITA

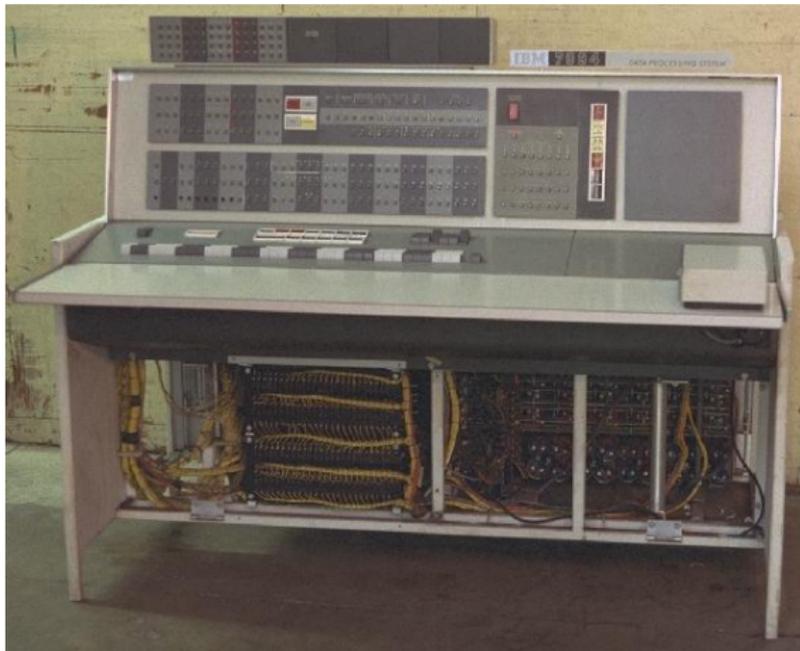
Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://www.columbia.edu/acis/history/>



IBM-1401

Systèmes
d'Exploitation

Didier Verna
EPITA

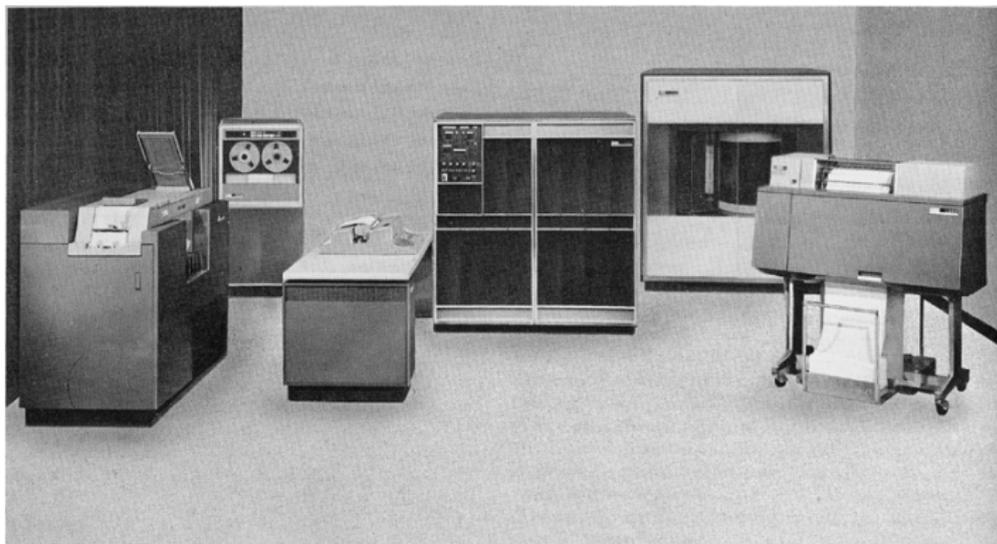
Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://www.columbia.edu/acis/history/>



1965 : circuits intégrés

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel

- **Offrir une gamme de machines compatibles**
Éviter les machines dédiées (IBM-1401 vs. IBM-7094).
Permettre la montée en puissance progressive.
- **IBM System/360, 370, 4300 etc.**
Première famille d'ordinateurs dotés de circuits intégrés. Rapport performance / coût cassé.
- **OS/360** : des millions de lignes d'assembleur. Des milliers de programmeurs. Autant de bugs.



IBM-360/91

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://www.columbia.edu/acis/history/>



1965 – 1980 : multiprogrammation

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel

■ Traitement par lots multiprogrammé

- ▶ Idée : 1960, mise en œuvre : 1970.
- ▶ **Problème** : inactivité du CPU pendant les E/S.
- ▶ **Solution** : faire coexister plusieurs travaux dans la mémoire, et faire travailler celui qui peut.

■ Spooling (Simultaneous Peripheral Operation Online)

- ▶ Utilisation de disques (accès direct) plutôt que de bandes (accès séquentiel)
- ▶ Plus besoin de conversion cartes ↔ bandes

■ Forme rudimentaire d'ordonnancement

- ▶ Le système choisi un candidat parmi les travaux disponibles (modification de l'ordre initial des travaux)
- ▶ Premier cas de prise de décision à la place des utilisateurs



Disques magnétiques

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://davidguy.brinkster.net/computer/>



Temps partagé (multitâche / timesharing)

Extension logique de la multiprogrammation

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel

- Plusieurs travaux sont lancés simultanément (multiprogrammation)
- Commutation très fréquente entre les différents programmes en cours (illusion de parallélisme)
- Possibilité de dialogue avec le programmeur (celui-ci redevient le contrôleur de ses travaux)
- CTSS (Compatible Time Sharing System), MULTICS (MULTIplexed Information and Computing Service).

⇒ Un accès interactif aux données nécessite un système de fichiers, et pose donc les premiers problèmes de sécurité.



Pour la p'tite histoire...

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel

- **Mini-ordinateurs** : DEC PDP-1 (5% du prix d'un 7094) sorti en 1961. La famille s'agrandi jusqu'au PDP-11.
- Ken Thompson (Bell Labs) développa une version simplifiée de MULTICS sur un PDP-7. Ce travail est à l'origine d'UNIX.
- UNIX s'est développé en **System V** (AT&T) et **Bsd** (Berkeley Software Distribution). POSIX est un effort de normalisation IEEE.
- **1987** : Andrew Tanenbaum livre MINIX, un clone d'UNIX à vocation pédagogique.
- MINIX est à la base de LINUX (Linus Torvalds).



PDP-11/50

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://www.columbia.edu/acis/history/>



Console PDP-11/50

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://www.columbia.edu/acis/history/>



1980 – ? : Ordinateurs personnels

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel

- **LSI** (Large Scale Integration)
mini-ordinateurs → micro-ordinateurs
- **1974 : CP/M** (Control Program for Micro-computers).
Premier système d'exploitation pour Intel (8080) par Digital Research. Basé sur floppy (8-pouces).
- **1980 : IBM PC**. Bill Gates fourni son interpréteur BASIC (Beginners All-purpose Symbolic Instructions Language). Digital Research refuse de fournir son système d'exploitation. IBM se tourne vers Bill Gates : – (
- **Ms-Dos** (Microsoft Disk Operating System)
Modification du DOS de Seattle Computer Products



1980 – ? : Ordinateurs personnels (suite)

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel

- **1985 : Macintosh / Windows** : incorporation d'une interface graphique (Engelbart, Xerox PARC). Windows n'est qu'une interface graphique au dessus de MS-DOS.
- **Windows 95 / 98 / Me** : ne retiennent MS-DOS que pour le boot et la compatibilité des applications. Encore beaucoup de code assembleur Intel 16 bits.
- **Window NT / 2000** : vrai 32 bits.
- **Et bien sûr** : LINUX (X11, KDE *etc.*), Mac OS X.



Apple IIe

Systèmes
d'Exploitation

Didier Verna
EPITA

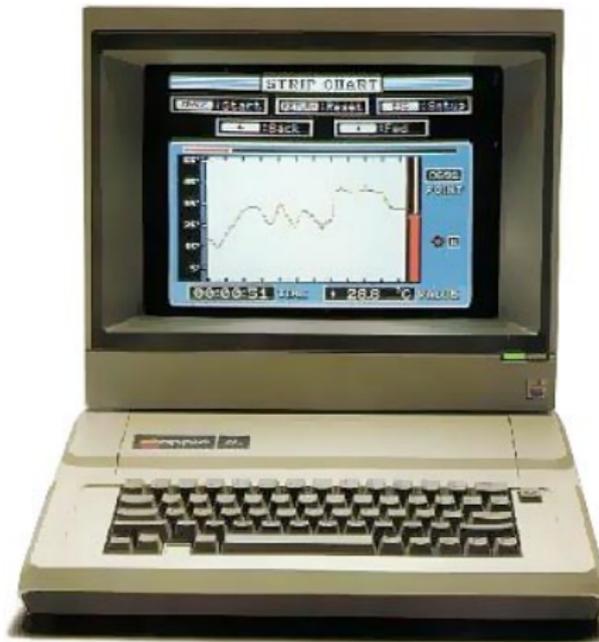
Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://www.old-computers.com/museum/>



Commodore 64

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://www.old-computers.com/museum/>



ATARI 800 XL

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel



<http://www.old-computers.com/museum/>



Systèmes parallèles / multiprocesseurs :

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel

- Plusieurs processeurs partagent les mêmes ressources matérielles (bus, mémoire, périphériques *etc.*)
 - ▶ Plus grande capacité de traitement et meilleure fiabilité (« dégradation progressive »)
- **Multitraitement asymétrique**
Tâches prédéfinies pour chaque processeur, philosophie maître - esclave (ex. SunOS 4)
- **Multitraitement symétrique**
Chaque processeur exécute une copie du système d'exploitation et communique avec les autres pour l'allocation des tâches (ex. SunOS 5 (Solaris 2) et LINUX)



Systèmes répartis / distribués

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel

- Plusieurs processeurs *ne partagent pas* les mêmes ressources matérielles (bus, mémoire, périphériques *etc.*)
 - ▶ Partage de ressources éloignées (imprimantes, fichiers *etc.*)
 - ▶ Accélération du calcul (« répartition de charge »)
 - ▶ Fiabilité (nécessite une certaine redondance)
 - ▶ Communication (réseaux)



Systemes temps reel

Systemes
d'Exploitation

Didier Verna
EPITA

Introduction

Prehistoire

Histoire

Aujourd'hui

Genie Logiciel

Les systemes temps reel sont principalement utilises pour des applications dediees (ex. robotique industrielle). Ils doivent fournir les resultats attendus quelles que soient les contraintes de temps imposees.

- **Temps reel rigide**

Systeme garantissant que les taches critiques finissent a temps. Caracteristique incompatible avec les systemes a temps partage.

- **Temps reel souple**

Systeme donnant la priorite aux taches critiques jusqu'a ce qu'elles se terminent. Cette fonctionnalite se retrouve de plus en plus dans les systemes modernes (UNIX en particulier).



Structure monolithique

Pas (ou peu) de structure

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel

- **MS-DOS** : conçu initialement pour Intel 8088. Pas de protection matérielle disponible, pas de contrôle d'accès, accessibilité de toutes les fonctionnalités aux utilisateurs.
- **UNIX** : séparation du système en noyau et programmes systèmes, séparation du noyau en pilotes et interface.



Approche en couches

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel

Chaque couche du système ne peut utiliser que les fonctionnalités fournies par les couches inférieures (ex. anneaux MULTICS).

- **Avantages** : modularité, plus grande facilité de débogage.
- **Inconvénients** : difficulté de conception de chaque couche, perte d'efficacité des appels système.

⇒ Limiter le nombre de couches, étendre leurs fonctionnalités, favoriser la modularité.



Machines virtuelles

Simulation logicielle d'une architecture matérielle

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

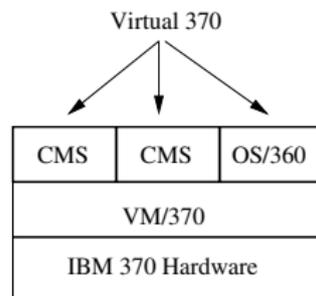
Génie Logiciel

- **Idée d'origine** : séparer la multiprogrammation de la notion de « machine étendue ».

- ▶ **Avantages** : protection des ressources, facilité de développement de nouveaux systèmes, portabilité.
- ▶ **Inconvénient** : perte d'efficacité à cause de la couche logicielle supplémentaire.

- **Concepts voisins**

- ▶ Mode 8086 virtuel du Pentium
- ▶ Bytecode (ex. JVM)
- ▶ Exo-kernels : clonage de machines avec restriction de ressources





Micro-noyaux

Systèmes
d'Exploitation

Didier Verna
EPITA

Introduction

Préhistoire

Histoire

Aujourd'hui

Génie Logiciel

- **Le moins possible en mode noyau**

Fonctionnalités système déplacées en espace utilisateur (userland).

- **Modèle client / serveur**

Les appels système deviennent des requêtes de clients (processus utilisateurs) vers des serveurs (fonctionnalités système en userland).

- **Avantages :**

- ▶ Tolérance aux pannes
- ▶ Systèmes distribués