

Systemes d'Exploitation

Introduction / Historique

Didier Verna

didier@lrde.epita.fr

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

Version 2@1.10 – 30 novembre 2004



Table des matières

Introduction	3
Définition pragmatique d'un système d'exploitation	4
Définition(s) technique(s) d'un système d'exploitation	5
Préhistoire	6
Histoire	9
1945 – 1955 : lampes et jacks	10
1955 – 1965 : traitement par lots (batch).....	16
1965 : circuits intégrés	21
1965 – 1980 : multiprogrammation	23
1980 – ? : Ordinateurs personnels	29

Autres Systèmes contemporains	33
Systèmes répartis / distribués :	34
Systèmes temps réel	35
Rôles des systèmes d'exploitation	36
Organisation des systèmes d'exploitation	38
Machines virtuelles	39
Micro-noyaux	40

Introduction

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

Ça ?

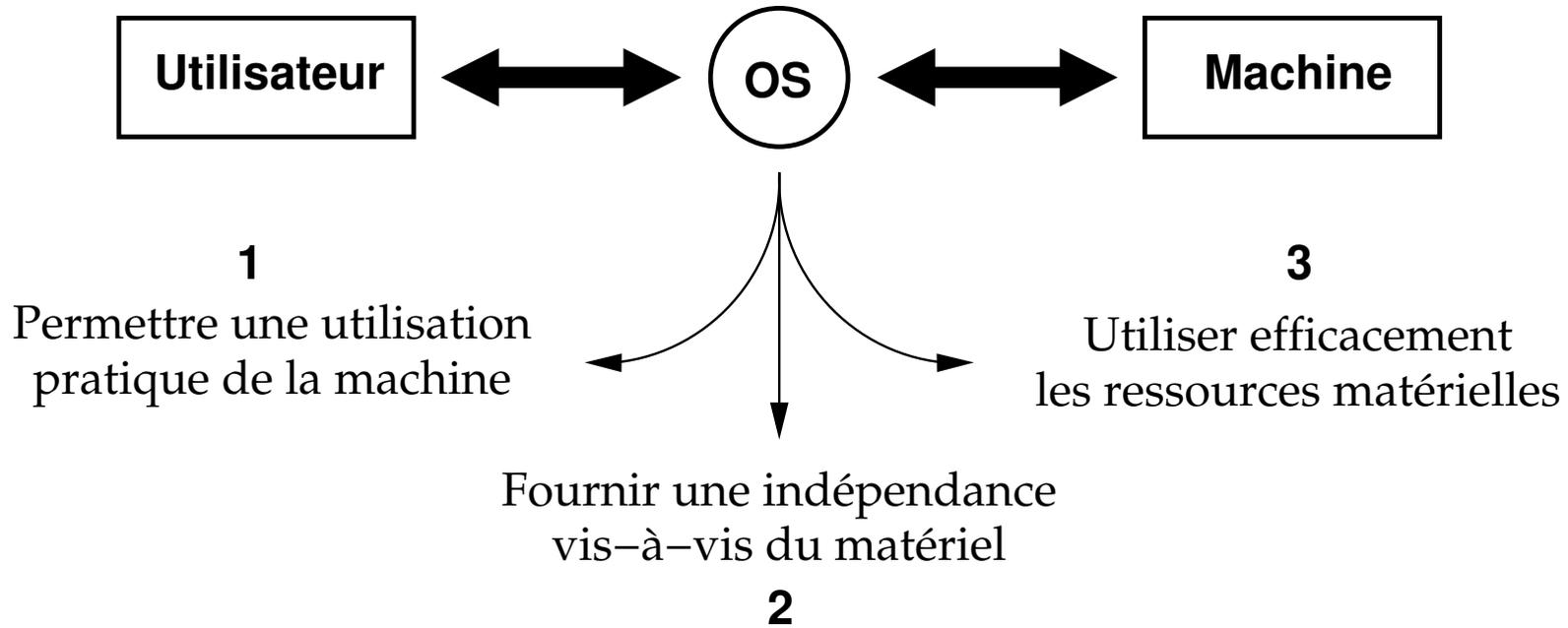


Ça ?



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

Définition pragmatique d'un système d'exploitation



Définition(s) technique(s) d'un système d'exploitation

- **Allocateur de ressources**
(temps CPU, mémoire, périphériques d'entrée/sorties. . .)
- **Contrôleur de ces ressources**
(vérification de l'exécution, des utilisations incorrectes. . .)
- Gestion et contrôle dans le temps et dans l'espace

Autres définitions courantes :

- Première couche logicielle au dessus du matériel
- Seul programme qui tourne constamment dans une machine
- Portion logicielle qui tourne en mode noyau (superviseur)

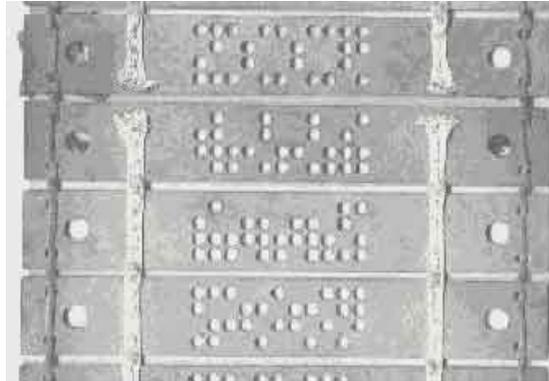
Préhistoire



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

- **1642 (Pascal) :**
Machines Arithmétiques : calcul dirigé manuellement.

Préhistoire (suite)



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

- **1725 (Jacquard) :**
Automates : métier à tisser doté d'un programme sur une carte en bois permettant de gérer le mouvement des engrenages.

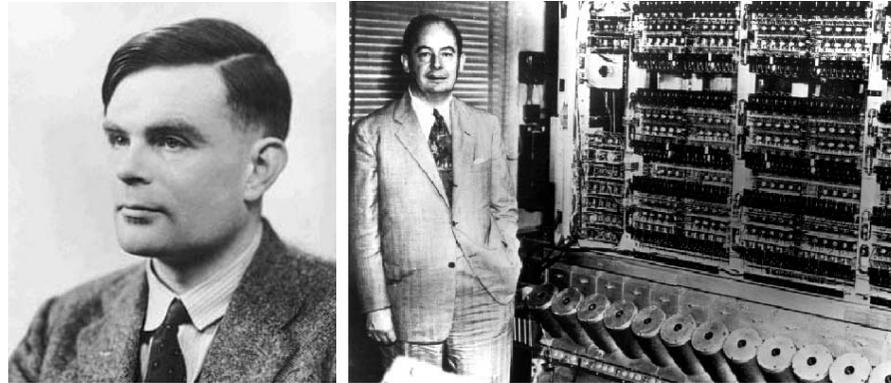
Préhistoire (suite)



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

- **1843 (Charles Babbage et la comtesse Ada Lovelace Byron) :**
Automate pour le calcul arithmétique (« difference engine ») : programme sur une carte en bois, « magasin » pour le stockage des résultats intermédiaires, automate pour l'exécution du programme.

Histoire



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

- **1936 (Alan Turing) :**
Définition mathématique précise de la notion d'algorithme. Modèle abstrait pour le calcul informatique. « Machine de Turing ».
- **1945 (Von Neumann) :**
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

- **Machines énormes**

Dizaines de milliers de relais mécaniques progressivement remplacés par des lampes.

- **Utilisation 100% manuelle**

Directement en langage machine, panneaux de contrôle électriques.

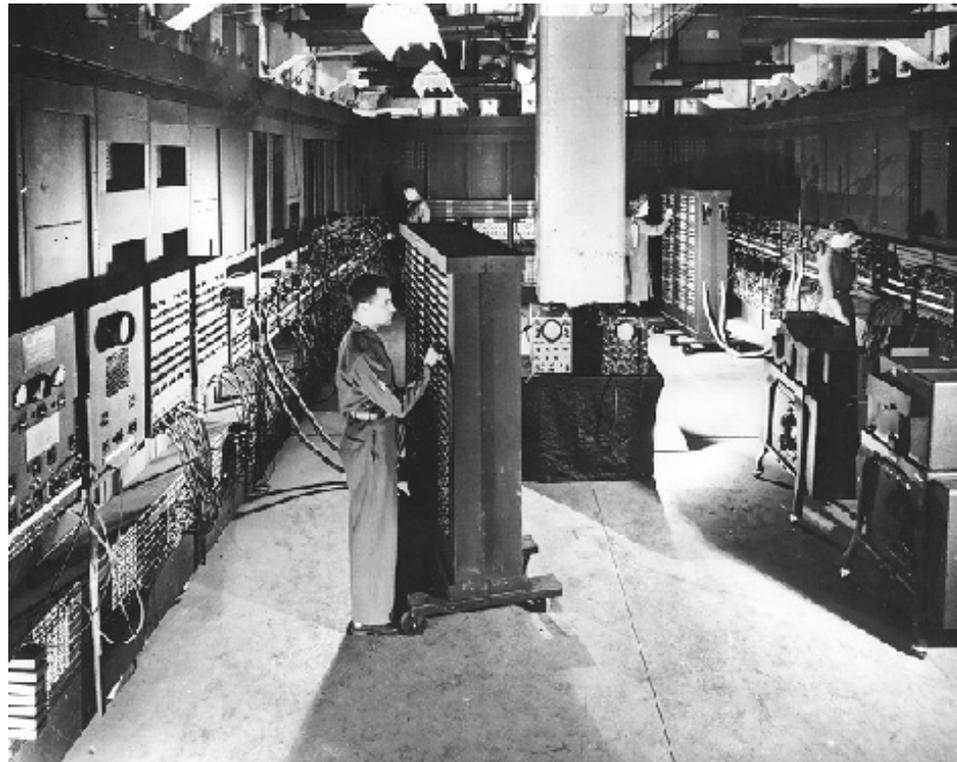
- Pas de distinction entre concepteurs, administrateurs et utilisateurs.

- **1950** Apparition des cartes perforées.

⇒ Temps de préparation considérable, unité centrale peu active.

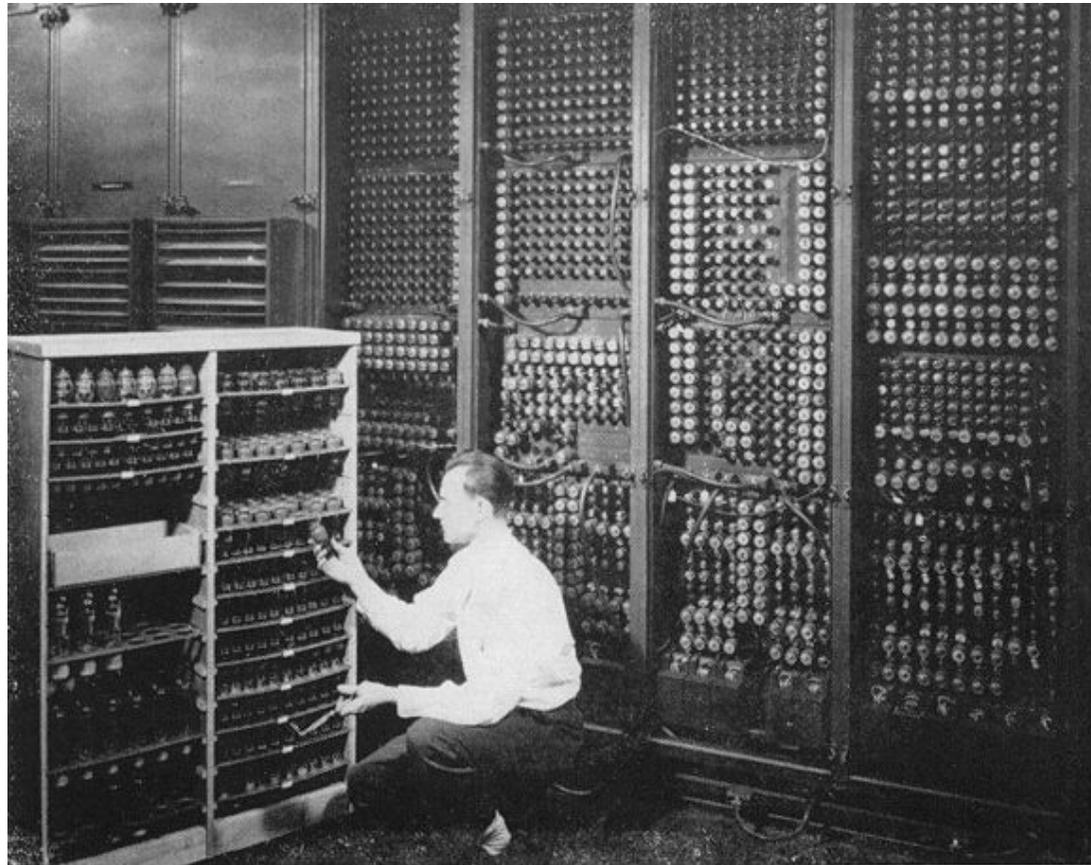
ENIAC (1)

Electronic Numerical Integrator and Computer



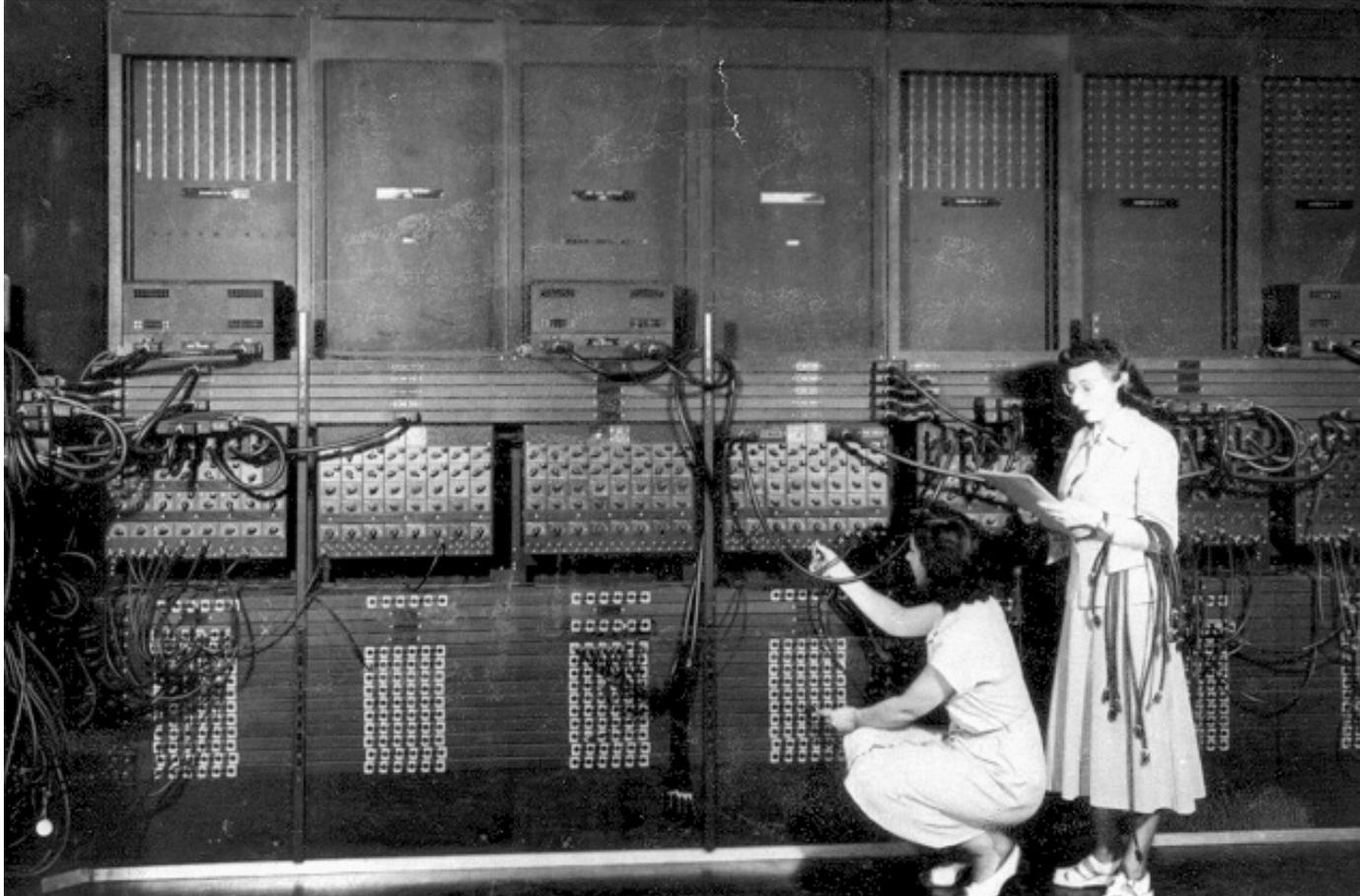
source : <http://www-gap.dcs.st-and.ac.uk/~history/>

ENIAC (2)

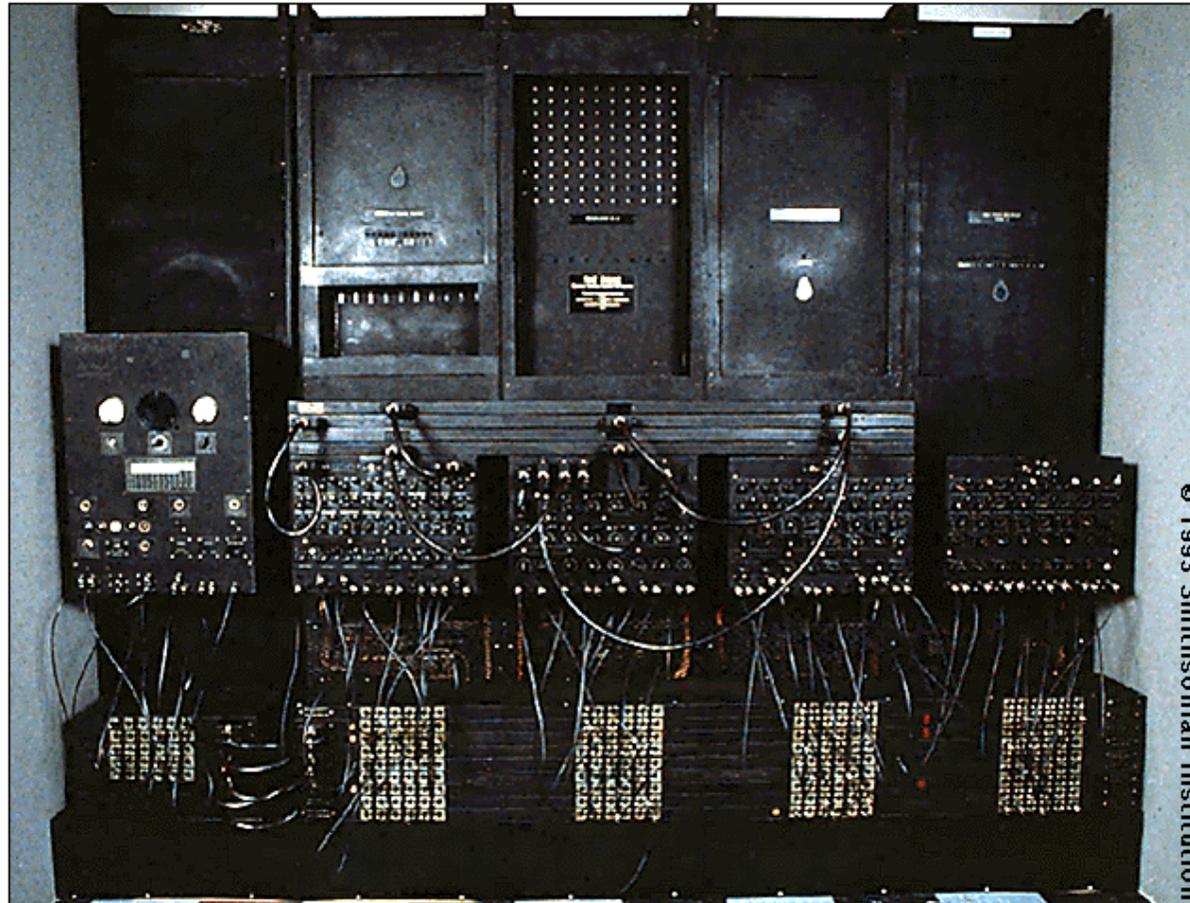


Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

ENIAC (3)



ENIAC (4)



ENIAC (5)



1955 – 1965 : traitement par lots (batch)

- **Utilisation de transistors** : machines plus fiables (mainframes).
- **Opérateur professionnel** engagé pour s'occuper du système.
Enchaînement des tâches, surveillance de leur déroulement. . .
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)

- **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).

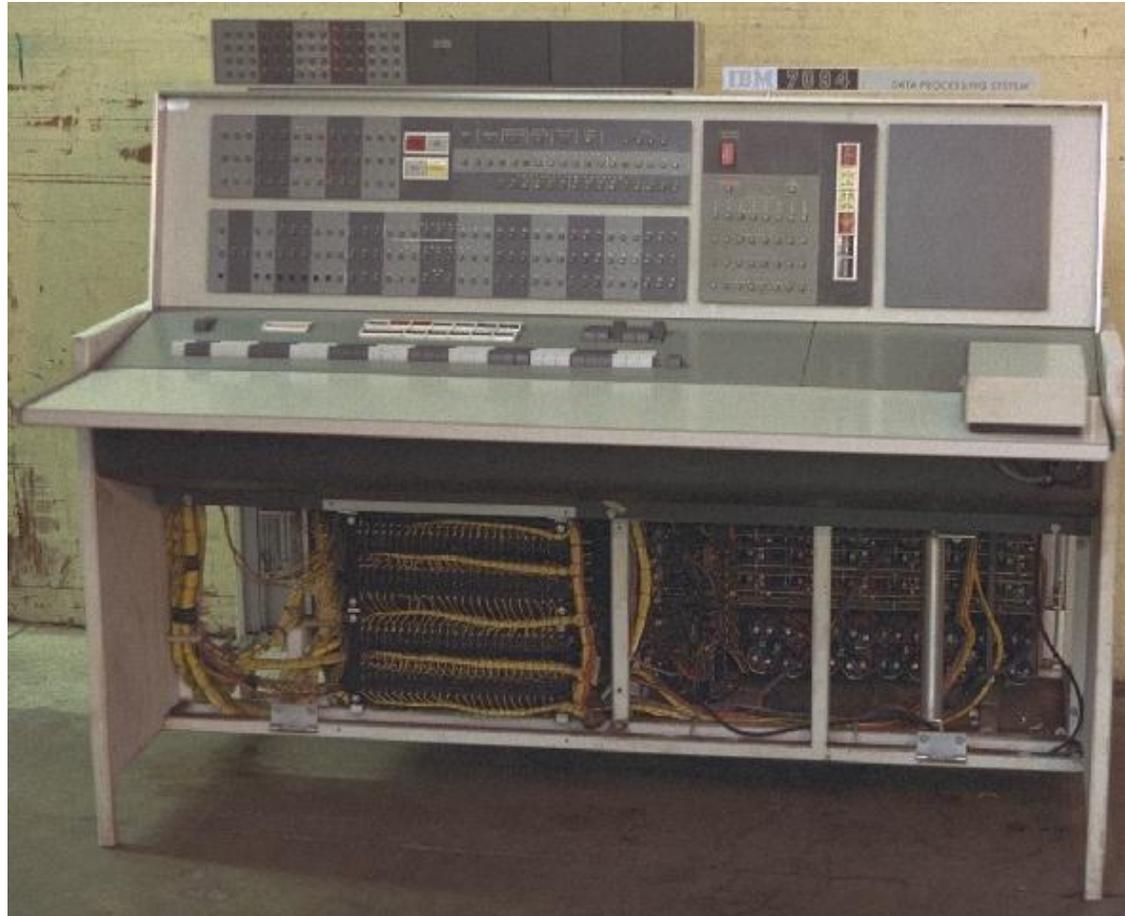
⇒ Aucune interactivité dans le traitement des tâches.

IBM-7094

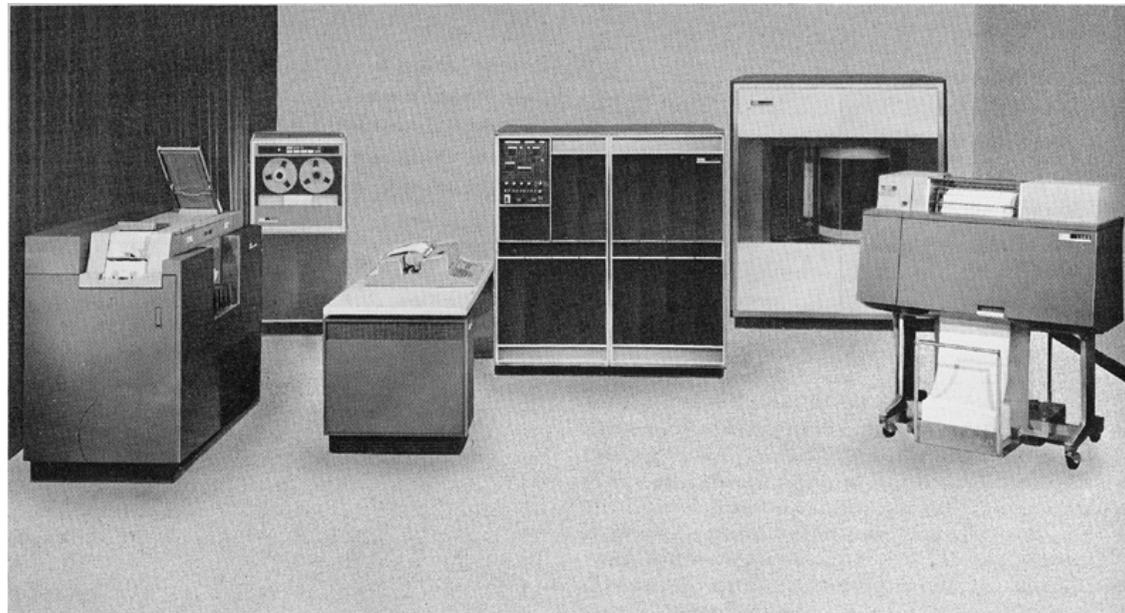


source : <http://www.columbia.edu/acis/history/>

Console 7094



IBM-1401



1965 : circuits intégrés

- **Volonté d'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...** : 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



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

1965 – 1980 : multiprogrammation

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. Quand un travail est en attente, en exécuter un autre.
- **Spooling** :
Simultaneous Peripheral Operation On-Line. Chevauchement lecture / écriture. Utilisation de disques (accès direct) plutôt que de bandes magnétiques (accès séquentiel). Plus besoin de conversion intermédiaire cartes ↔ bandes.

⇒ **Forme rudimentaire d'ordonnement (scheduling).**

- 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



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

Temps partagé (multitâche / timesharing)

- Extension logique de la multiprogrammation.
- 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...

- **Mini-ordinateurs** : DEC PDP-1 (5% du prix d'un 7094) sorti en 1961. La famille s'agrandi jusqu'au DEC 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).

Dec PDP-11/50



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

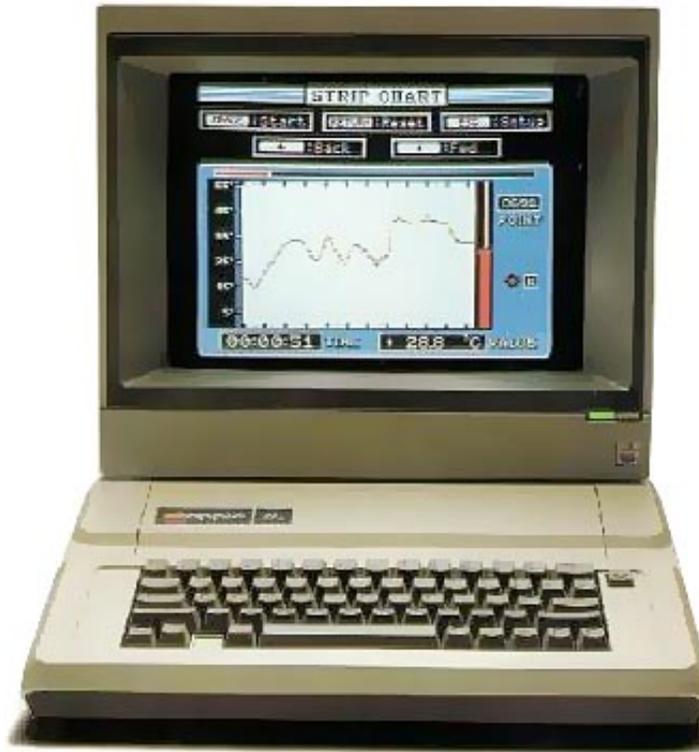
Console PDP-11/50



1980 – ? : Ordinateurs personnels

- **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 retourne vers Bill Gates... : – (
- **MS-DOS** (Microsoft Disk Operating System) : modification du DOS de Seattle Computer Products
- **1985 : Macintosh / Windows** : incorporation d'une interface graphique (Ebgelbart, 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...), Mac OS X.

Apple IIe



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

Commodore 64



www.old-computers.com

ATARI 800XL



www.old-computers.com

Autres Systèmes contemporains

Systèmes parallèles / multiprocesseurs :

- Plusieurs processeurs partagent les mêmes ressources matérielles (bus, mémoire, périphériques. . .).
- Plus grande capacité de traitement et meilleure fiabilité (« dégradation progressive »).
- **Multitraitement asymétrique :**
Tâches prédéfinies pour chaque processeur, où 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 :

Plusieurs processeurs *ne partagent pas* les mêmes ressources matérielles (bus, mémoire, périphériques. . .).

- Partage de ressources éloignées (imprimantes, fichiers. . .)
- Accélération du calcul (« répartition de charge »)
- Fiabilité (nécessite une certaine redondance)
- Communication (réseaux)

Systèmes temps réel

Les systèmes temps réel sont principalement utilisés pour des applications dédiées (ex. robotique industrielle). Ils doivent fournir les résultats attendus quelles que soient les contraintes de temps imposées.

- **Temps réel rigide :**

Système garantissant que les tâches critiques finissent à temps. Caractéristique incompatible avec les systèmes à temps partagé.

- **Temps réel souple :**

Système donnant la priorité aux tâches critiques jusqu'à ce qu'elles se terminent. Cette fonctionnalité se retrouve de plus en plus dans les systèmes modernes (UNIX en particulier).

Rôles des systèmes d'exploitation

Les rôles d'un système d'exploitation sont les suivants :

- **Gestion des processus** : création, suppression (a)normale, interruption et ordonnancement, communication, synchronisation
- **Gestion de la mémoire** : (dé)chargement de processus et de données, répartition et affectation, liaison d'adresses
- **Gestion de la mémoire auxiliaire** : organisation de l'espace libre, allocation, ordonnancement, swapping
- **Gestion des entrées / sorties** : interfaçage, pilotes, cache
- **Gestion des systèmes de fichiers** : création, suppression, hiérarchisation, manipulation, contrôle d'accès, transparence vis-à-vis du type de stockage, sauvegarde

Rôles des systèmes d'exploitation (suite)

- **Protection et sécurité** : contrôle d'accès aux ressources, détection d'erreurs et d'abus
- **Gestion des réseaux** : systèmes répartis, communication, partage de ressources, transparence
- **Appels systèmes** : interface élémentaire avec le programmeur (assembleur, C...)
- **Interprétation de commandes** : interface élémentaire avec l'utilisateur (shells, interfaces graphiques)

Organisation des systèmes d'exploitation

Structure monolithique : pas (ou peu) de structure.

- **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 : chaque couche du système ne peut utiliser que les fonctionnalités fournies par les couches inférieures. THE, anneaux MULTICS.

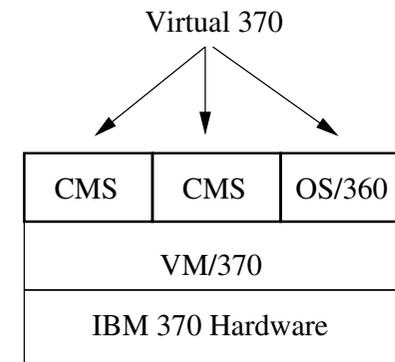
- **Avantages** : modularité, plus grande facilité de débogage.
- **Inconvénients** : difficulté de conception de chaque couche, perte d'efficacité des appels système.

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

Machines virtuelles

Simulation logicielle d'une architecture matérielle

- **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 voisin :

- Mode 8086 virtuel du Pentium
- Bytecode (ex. JVM)
- Exokernels : clônage de machines avec restriction de ressources

Micro-noyaux

- **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