

L3 Informatique - Systèmes d'exploitation

Introduction/Historique

D. Béchet

`Denis.Bechet@univ-nantes.fr`

Université de Nantes

Faculté des Sciences et Techniques

2, rue de la Houssinière BP 92208

44322 Nantes cedex 3, France

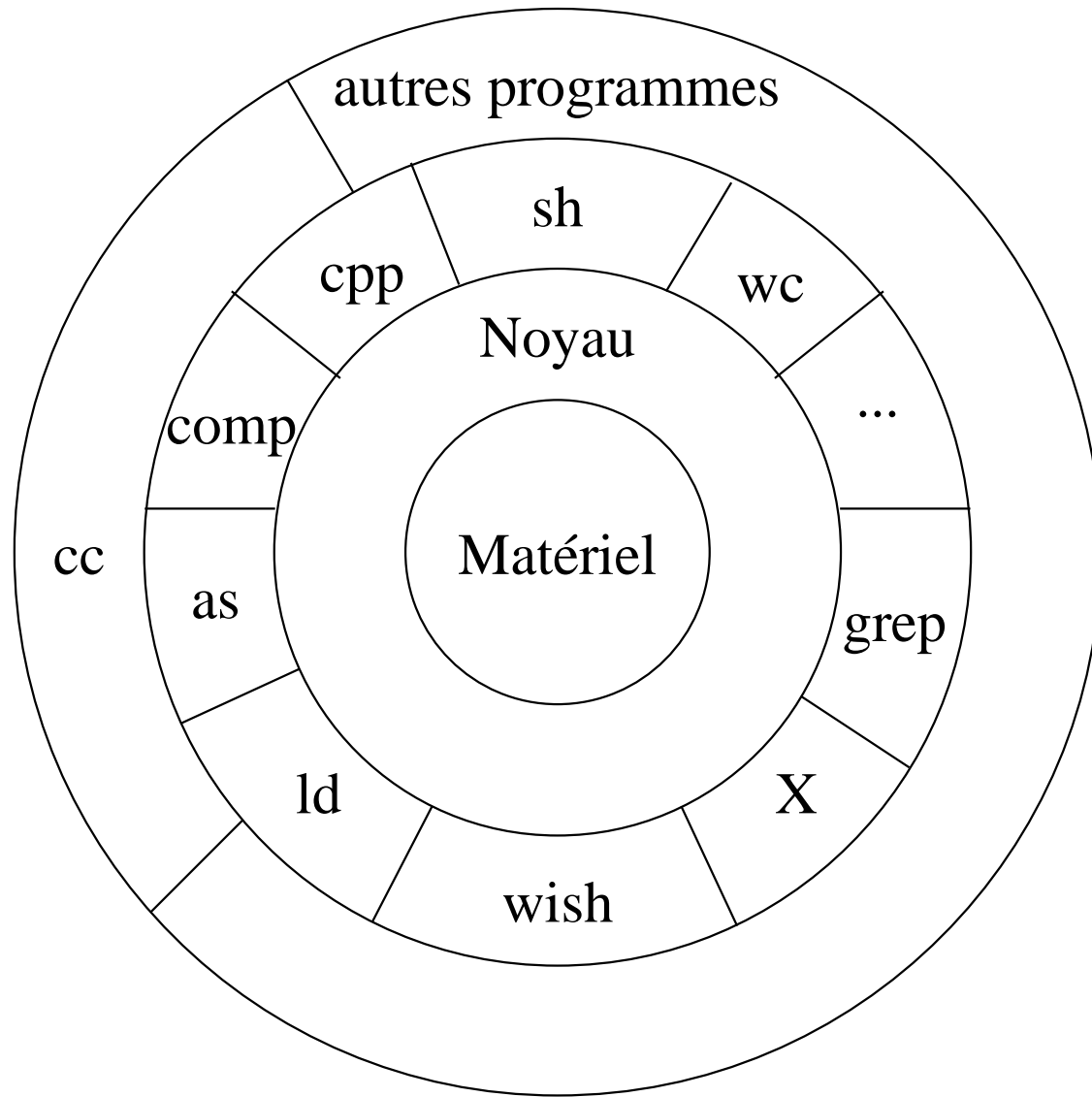
<http://www.sciences.univ-nantes.fr/info/perso/permanents/bechet>



Système d'exploitation

- Qu'est ce qu'un système d'exploitation ?
 - Aspects externes (fonctionnalités)
 - Aspects internes (architecture)
 - Aspects historiques
- Structure des systèmes d'exploitation

Systeme d'exploitation



SE = machine étendue

Exemple: gestion des entrées/sorties sur disquettes

- Matériel = contrôleur NEC PD765
 - 16 commandes de 1 à 9 octets: déplacement de la tête, lecture, écriture, formatage d'une piste...
 - `read et write`: 13 paramètres (9 octets). Fin de l'opération par interruption du processeur. Retour: 23 champs de status et d'erreurs (7 octets).
 - Gestion de l'arrêt du moteur
- ⇒ Trop frustré pour le programmeur d'une application
- ⇒ Pas de souplesse d'adaptation (disques durs)
- ⇒ E/S par blocs (unité = 512 octets)
- ⇒ Pas de structuration de l'espace (fichiers)

SE = machine étendue

Exemple: gestion des entrées/sorties sur disquettes

● Solution:

- Une interface simple (API) à base de fichiers = une machine **étendue** ou **virtuelle**
- Gestion des périphériques par le SE: E/S vers le matériel, gestion des interruptions = **pilotes des périphériques**
- Gestion de systèmes de fichiers par le SE: gestion de l'espace des supports secondaires (disquettes, disques durs) et des requêtes des programmes via l'API

SE = machine étendue

Domaines des API systèmes

- Accès aux périphériques: clavier, écran, imprimantes, lignes séries, cartes réseaux, modems, lecteurs de disquettes, disques durs, horloge interne, temporisateur...
- Accès aux systèmes de fichiers
- Création/suppression des processus et communication interprocessus
- Accès aux réseaux et aux services
- Accès à l'interface graphique (gestion des événements)
- ...

SE = gestionnaire de ressources

Exemple: gestion d'une imprimante par des processus

- Une seule impression à la fois
- Solutions possibles:
 - Attribution de l'imprimante à un processus
 - Attribution de l'imprimante à un processus avec file d'attente
 - Création de files d'attente de fichiers = **spoulage**
 - Gestion par le SE ou un service associé au SE

SE = gestionnaire de ressources

Domaines des ressources

- Allocation de la mémoire secondaire aux fichiers, gestion de quotas par utilisateur,...
- Allocation de la mémoire primaire aux processus
- Allocation du temps de calcul aux processus
- Gestion de la concurrence

Autres fonctions

- Surveillance du système
- Mise en route ou arrêt du système
- Mise en veille ou en hibernation (d'une partie) du système

Historique

- Première génération 1945–1955: relais, lampe (tube à vide) et tableaux d'interrupteurs ▷
- Deuxième génération 1955–1965: transistors, cartes perforées et traitements par lots ▷
- Troisième génération 1965–1980: circuit intégrés, multi-programmation, multi-utilisateurs (temps partagé et terminaux en ligne), CTSS, MULTICS, UNIX ▷
- Quatrième génération 1980-aujourd'hui: circuits (V)LSI, micro-processeurs, micro-ordinateurs, CP/M, MS-DOS, IHM graphique (LISA, Macintosh, Windows) ▷

Première génération 1945–1955

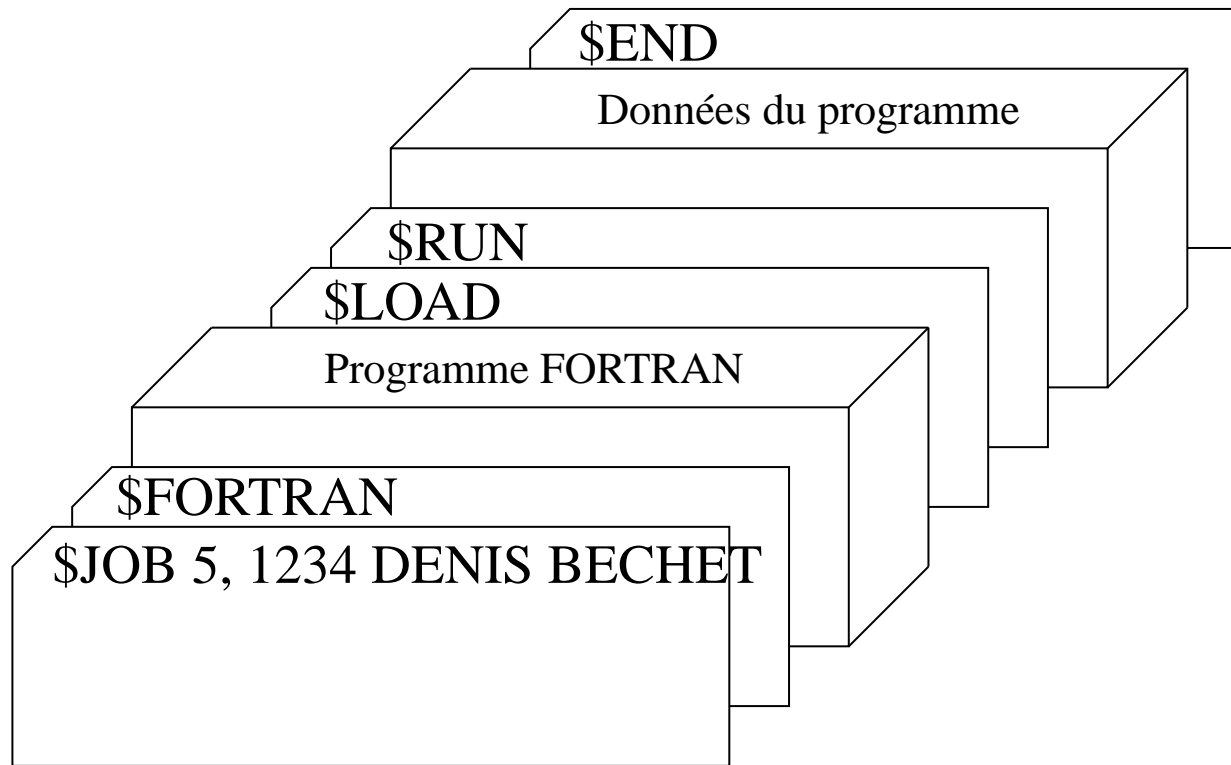
- Premiers moteurs de calcul avec des relais mécaniques \Rightarrow très lents (1 à 10 secondes par cycle)
- Remplacement progressif des relais par des lampes (tubes à vide) \Rightarrow ordinateurs énormes (une pièce entière remplie de lampes)
- Programmation par des tableaux d'interrupteurs
- Pas de langage de programmation, pas de SE
- L'utilisateur devait réserver la salle sur un planning, positionner les interrupteurs en fonction de son programme et attendre le résultat en espérant qu'aucune des lampes ne tomberait en panne !
- Après 1950, amélioration du protocole avec l'utilisation des cartes perforées à la place des interrupteurs

Deuxième génération 1955–1965

- Remplacement des lampes par des transistors
⇒ **fiabilité**
- Apparition des **mainframes** (ordinateurs centraux) vendus par des sociétés spécialisées (IBM par exemple) et placés dans des pièces climatisées
- Programme source (en FORTRAN, par exemple) et données codés sur cartes perforées
- L'utilisateur portait ses cartes à la salle de soumissions des *jobs*. Quand l'ordinateur terminait le job courant, un opérateur récupérait la trace du travail sur l'**imprimante** et la stockait dans la salle des résultats. Puis, il prenait l'un des paquets de cartes de la salle des soumission plus, éventuellement, celles du compilateur (FORTRAN) pour les mettre dans la **machine**

Retour au plan de l'historique ◀

Cartes perforées et jobs



- **SE** : FMS (Fortran Monitor System), IBSYS (IBM)
- **Languages** : FORTRAN et assembleur
- **Périphériques** : lecteur de carte, de bandes magnétiques, imprimantes

[Retour au plan de l'historique](#) ◀

Troisième génération 1965–1980

- Remplacement des transistors par des circuits intégrés
⇒ **prix faible** par rapport aux transistors
- Apparition des séries de machines de performances différentes mais **compatibles** (série system/360 d'IBM)
⇒ SE unique pour toute la série (OS/360 = usine à gaz)
- Apparition de la **multi-programmation** : plusieurs jobs sont chargés sur différentes partitions de la mémoire pour optimiser le temps CPU (si attentes sur les E/S)
- Puis méthode du **spoulage** (de spool, Simultaneous Peripheral Operation On Line) chargeant automatiquement le prochain job lorsque l'un d'entre eux se termine

Troisième génération 1965–1980

- Introduction du **temps partagé** et de la notion de **multi-utilisateur** permettant de travailler directement sur l'ordinateur sans passer par un service de soumission des jobs: CTSS (Compatible Time Sharing System) du MIT
- Projet **MULTICS** (MULTIplexed Information and Computing Service) regroupant le MIT, Bell Labs et General Electric : une et une seule machine partagée par tous !
- Création aux Bell Labs du premier **UNIX** sur PDP-7 par Ken Thompson (Brian Kernighan l'a surnommé UNICS (Uniplexed Information and Computing Service) par plaisanterie) bientôt rejoint par Dennis Ritchie: portage sur des machines plus puissantes (PDP-11) et réécriture en C.

Retour au plan de l'historique ◀

Quatrième génération 1980–aujourd'hui

- Intégration de plus en plus poussée des circuits intégrés (VL)LSI \Rightarrow **prix et volume très faibles**
- Apparition des premiers micro-processeurs (sortie du 8080 en 1974 par Intel)
- Apparition des premiers micro-ordinateurs avec lecteur de disquette et son SE: **CP/M** (Control Program for Microcomputers) écrit par Gary Kildall (un consultant de Intel devenu le fondateur de Digital Research)
- Sortie de l'IBM PC (1980) et écriture par Bill Gates de la première version de **MS-DOS** (Microsoft Disk Operation System) depuis le system **DOS** (Disk Operation System) de Seattle Computer Products.

Quatrième génération 1980–aujourd'hui

- Récupération par le Xerox PARC puis par Steve Jobs (Apple) du concept d'IHM graphique (Interface Homme-Machine) inventé par Doug Engelbart à l'institut de recherche de Stanford (années 1960) \Rightarrow lancement de LISA puis du Macintosh (1984)
- Intégration (en une surcouche) d'une IHM à MS-DOS : le premier Windows évoluant puis sur deux SE (Windows 95 et Windows NT) ancêtres de l'actuel Windows XP
- Ajout au dessus de UNIX de X Window, une IHM développée depuis 1984 dans le projet Athena du MIT

Quatrième génération 1980–aujourd'hui

- Côté UNIX, évolution vers deux versions principales **System V** d'AT&T et **BSD** (Berkeley Software Distribution) de l'Université de Berkeley.
- Création de MINIX (1987), un mini-UNIX à vocation éducative, par Andrew Tanenbaum, système réécrit par un étudiant finlandais, Linus Torvalds, et appelé **LINUX**
- Unification d'une partie significative des systèmes UNIX avec le standard **POSIX** de l'IEEE

Types de SE

- Mainframes ▷
- Serveurs ▷
- Systèmes multiprocesseurs ▷
- Systèmes personnels ▷
- Systèmes temps réel ▷
- Systèmes embarqués ▷
- Smart cards ou cartes de crédit ▷

Mainframes

- Très gros système : un millier de disques — plusieurs téraoctets de mémoire au total
- Gestion optimisée des ressources
- Exemple : OS/390 d'IBM descendant de l'OS/360

Serveurs

- Système avec d'importantes ressources utilisées en parallèle par de nombreux utilisateurs
- SE spécialisés, en fait, optimisés pour cet usage: UNIX, Linux ou Windows XP Server
- Exemples : Serveur de fichiers, serveur Web, serveur de courriels

Systèmes multiprocesseurs

- Plusieurs CPU fonctionnent sur une même plateforme: ordinateurs parallèles, multi-ordinateurs ou multiprocesseurs
- SE spécialisés ou variantes plus ou moins optimisés pour la communication entre processus et les accès concurrents aux ressources
- Exemple : la Connection Machine, le Cray, les grappes de PC ou les PC bi-processeurs

Systèmes personnels

- Un seul utilisateur
- Interface conviviale
- Orienté vers traitement de texte, le multimédia ou l'accès à internet
- Exemple : Mac OS, Windows XP et Linux

Systemes temps réel

- **Contraintes temporelles** : contrôle du temps de réponse
- **Exemple** : micro-contrôleur dans une voiture (ABS, Air-bag)
- **SE spécialisé** : VxWorks, QNX, LynxOS, etc

Systèmes embarqués

- Palmtops ou PDA (Personal Digital Assistant), téléphones portables, ordinateurs de machine à laver, etc
- **Contrainte de taille** (programmes et/ou données)
- **SE spécialisé**: PalmOS, VRTX, Windows-CE (Consumer Electronics), etc

Smart cards ou cartes de crédit

- SE souvent rudimentaire
- **Parfois** : mise-en-oeuvre du SE à l'aide d'un **langage de haut niveau** comme Java ou Basic
- Applications souvent spécialisées : cartes de crédit, cartes téléphoniques, badge d'accès, etc

Concepts de base des SE

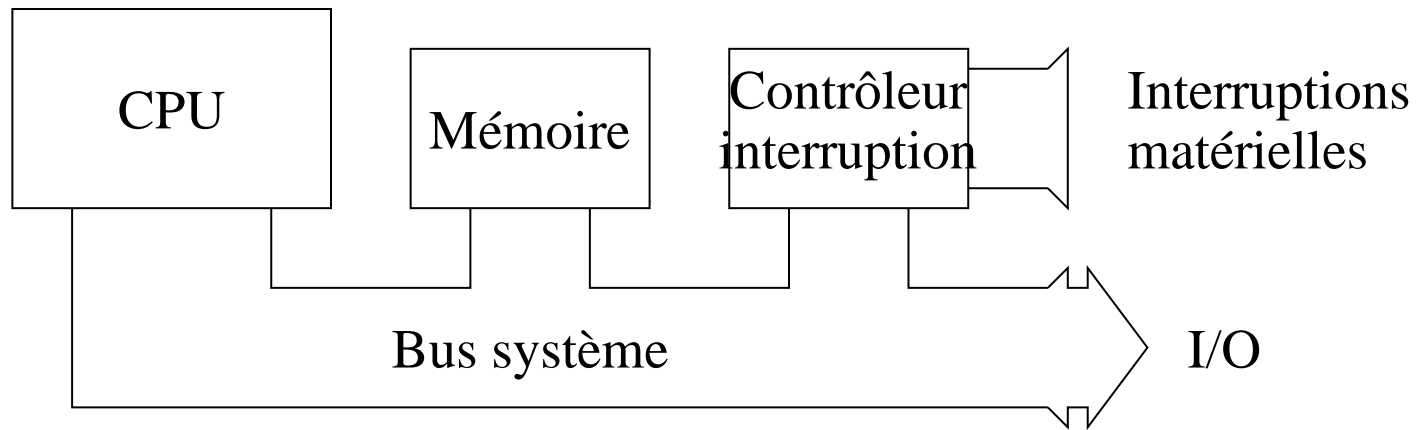
- CPU - Mémoire vive - Interruptions ▷
- Espace d'adressage ▷
- E/S ▷
- Mode noyau/mode utilisateur ▷
- Processus - tâche - thread ▷
- Système de fichiers ▷

CPU - Mémoire vive - Interruptions

- CPU : le “cerveau” de l’ordinateur
 - Registre du **compteur ordinal** pointe sur l’instruction en langage machine qu’il va exécuter
 - Registre du **pointeur de pile** qui donne le haut de la pile courante
 - **Registre d’état**
 - **Registres généraux** pour le calcul ou l’adressage
 - **Registres spécialisés** : registre des vecteurs d’interruption, segmentation ou pagination de la mémoire, etc
 - Unité(s) de calcul, d’adressage et de contrôle
 - Eventuellement, mémoire cache

CPU - Mémoire vive - Interruptions

- Mémoire vive externe
- Canaux d'interruption et leur contrôleur



Les interruptions (matérielles ou logicielles) interrompent le travail du CPU pour exécuter un code spécifique appelé **procédure de service de l'interruption**

Provenance : horloge, contrôleurs de disque, du clavier, de l'écran, division par zéro, instruction inconnue, etc

Espace d'adressage

- Le CPU accède à la mémoire vive en utilisant un mécanisme plus ou moins directe
 - Adresse physique : accès directe à l'emplacement
 - Segment : accès à l'aide d'un identificateur de segment se référant à une zone de la mémoire physique et d'un déplacement dans ce segment
 - Pagination : accès à l'aide d'une (ou plusieurs) table d'indirection donnant la correspondance entre les pages virtuelles et les pages physiques
- Avec la possibilité de placer temporairement des pages sur disque, ceci crée une notion de mémoire virtuelle qui, en générale, est spécifique à chaque processus qu'on appelle l'espace d'adressage du processus

E/S

- Les entrées/sorties sont très **variées**
- Leur accès se fait, en général, à l'aide d'**instructions spéciales** du processeur avec un espace d'adressage particulier \neq mémoire vive
- De nombreux contrôleurs sont utilisés, certains pouvant directement adresser la mémoire vive comme les contrôleurs de DMA (Direct Memory Access)

Mode noyau/mode utilisateur

- Pour des raisons de sécurité, un CPU fonctionne suivant plusieurs modes ou niveaux, deux en général :
 - **Mode noyau ou superviseur** : pas de restriction : accès aux registres spéciaux sensibles, E/S vers les périphériques, accès directe à la mémoire physique et par extension, à l'espace d'adressage de n'importe quel processus, etc
 - **Mode utilisateur** : ceci est interdit
- Les SE exploitent cette propriété pour exécuter les programmes normaux dans le mode utilisateur et les routines du SE dans le mode noyau.

Mode noyau/mode utilisateur

- Passage du mode utilisateur au mode noyau :
 - Violation d'un privilège : création d'une interruption "violation de privilège"
 - Erreur lors de l'exécution d'une instruction : création d'une interruption (division par zéro, défaut de page, instruction inconnue, etc)
 - instruction demandant la création d'une interruption logicielle utilisée pour débiter un appel système (les paramètres sont, en général, placés dans les registres généraux du CPU)
 - Prise en compte par le CPU d'une interruption matérielle

Mode noyau/mode utilisateur

- L'exécution continue en mode noyau dans la procédure de service de l'interruption
- Passage du mode noyau au mode utilisateur : au "retour" de l'interruption
- Les processus utilisateurs s'exécutent parfois en mode utilisateur, parfois en mode noyau
- Les processus systèmes s'exécutent en mode noyau
- La commutation entre processus à lieu (magiquement) en mode noyau

Processus - tâche - thread

- Un processus est une notion du SE. Il regroupe :
 - L'état des **registres** du CPU associé au processus
 - L'espace d'adressage en mode utilisateur et en mode noyau
 - Les descripteurs des ressources (fichier, verrou) associés au processus
 - Des informations diverses : nom ou numéro du propriétaire, priorité d'exécution, temps d'exécution, etc
- **Thread** (processus léger) : est un processus qui partage tout sauf l'état des registres du CPU avec les autres threads de son groupe
- **Tache** : un groupe de threads

Systeme de fichier

