

Cours de Systèmes d'Exploitation et Programmation Concurrente

RICM 4

Fabienne Boyer
UFR IMA, Université Joseph Fourier
Fabienne.Boyer@imag.fr



Objectifs du cours

→ **Savoir utiliser les fonctions et ressources fournies par le système**

- **Processus & threads**
- **Mémoire**
- **Fichiers**

→ **Comprendre leur implantation**

- **Techniques de virtualisation**

Plan du cours

1. Introduction générale
2. Processus et threads
3. Synchronisation
 - ◆ Solutions de bas niveau
 - ◆ Solutions de haut niveau
4. Gestion des interblocages
5. Gestion de la mémoire virtuelle
6. Le système de gestion de fichiers
7. Gestion de la mémoire secondaire
8. Présentation du système Nachos

Organisation de l'enseignement

■ Equipe pédagogique

- ◆ Philippe Morat, Philippe.Morat@imag.fr, Equipe ADELE / LIG)
- ◆ Fabienne Boyer (Fabienne.Boyer@imag.fr, Equipe ERODS / LIG)
- ◆ Noël de Palma (Noel.dePalma@imag.fr, Equipe ERODS / LIG)

■ Volume

- ◆ 12 Cours d'1H30
- ◆ 12 TD d'1H30
- ◆ 4 TPs en C (processus, shell, gestion mémoire)
- ◆ 1 TP en java (programmation concurrente)
- ◆ 1 examen de 2 heures

Bibliographie

- ◆ Silberschatz, Galvin and Gagne, Operating System Concepts, Addison-Wesley, 2nd edition, 2014
- ◆ S. Krakowiak, Principes des systèmes d'exploitation des ordinateurs, Dunod, 1993
- ◆ A. Tanenbaum, Modern Operating Systems, dec 2015

Introduction générale

■ Systèmes d'exploitation

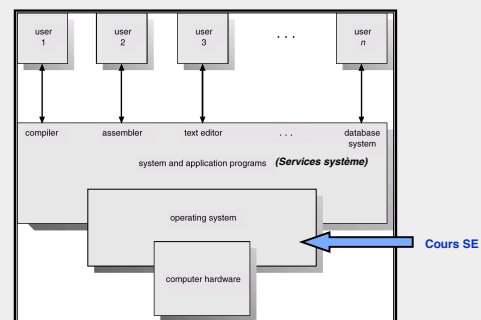
- ◆ Rôles
- ◆ Organisations
- ◆ Historique

Rôle d'un Système d'Exploitation (SE)

■ Intermédiaire entre l'ordinateur et les applications/ utilisateurs

- ◆ Ordinateur vu comme un ensemble de ressources physiques
 - ◆ Fournir des ressources **logiques**
 - ◆ Fournir une interface **d'accès simplifiée** aux ressources
 - ◆ Gérer le **partage efficace et équitable** des ressources
 - ◆ Assurer le **contrôle d'accès** aux ressources
- Le SE est un gestionnaire de ressources qui met en oeuvre une machine virtuelle

Architecture générale



Structure d'un SE

■ Noyau

- ◆ Partie qui réside constamment en mémoire centrale
 - ❖ Routines qui requièrent un mode d'exécution privilégié
 - ❖ Données globales et partagées ou pas par les serveurs système

■ Serveurs système

- ◆ Partie qui peut être commutée / déchargée temporairement

■ Pilotes (ou contrôleurs)

- ◆ Modules spécialisés qui dialoguent avec les périphériques réels

Conception d'un SE

■ Noyau minimal (micro-noyau / client-serveur)

- ◆ Mach / Chorus (1ere génération) / L4 (2^{ème} génération)
- ◆ Maximise les fonctions gérées par des serveurs systèmes
- ◆ Meilleure extensibilité et adaptabilité
- ◆ Meilleure isolation des fautes

■ Noyau maximal (monolithique)

- ◆ Unix, Linux, Windows XP
- ◆ Meilleures performances
- ◆ Le SE est un ensemble de procédures, ce qui diminue les IPC (Inter Processus Call)

API du SE

■ Fonctions système (API)

- ◆ Pour le programmeur d'applications, le SE apparaît comme un ensemble de fonctions appelables (*fonctions système*), permettant de manipuler les ressources physiques et logiques
- ◆ Au plus bas niveau, une fonction système peut engendrer un *appel système* (appel au noyau)
- ◆ Un appel système peut faire appel à un serveur système

■ Langages de haut niveau : appels systèmes cachés par compilateurs / interpréteurs / bibliothèques

Utilisation d'un SE

■ Interpréteur de commandes

- ◆ Permet à un utilisateur d'exécuter de manière interactive des requêtes de haut niveau, qui mettent en jeu des appels systèmes
 - ❖ Manipulation de processus
 - ❖ Manipulation de fichiers
 - ❖ Manipulation de mémoires secondaires
 - ❖ ...
- ◆ Est lui-même un programme applicatif pour le SE

Ressources gérées par le SE

- Programme / application
 - Processus
 - Thread (processus léger)
 - Pilotes
- activités
-
- Mémoire centrale
 - Fichiers
 - Mémoires secondaires
- données

Gestion des activités

- ◆ Lancement d'un programme → création d'un **processus**
 - ◆ Un processus = un programme en cours d'exécution
 - ◆ Durée de vie d'un processus = celle du programme exécuté
-
- ◆ Le SE
 - ❖ crée et détruit les processus
 - ❖ gère l'allocation des ressources aux processus
 - ❖ fournit des moyens de communication entre processus locaux ou distants

Gestion des données

- ◆ Mémoire centrale = un grand tableau d'octets (contenu volatile)
 - ◆ Les fichiers contiennent des données persistentes (programmes, données)
 - ◆ Mémoires secondaires = supports de stockage permanents
-
- ◆ Le SE
 - ❖ gère l'allocation de la mémoire centrale aux processus
 - ❖ gère la création, la destruction, l'accès aux fichiers
 - ❖ gère l'allocation de la mémoire secondaire aux fichiers

Historique rapide

- Evolutions liées
 - ◆ Aux évolutions technologiques
 - ❖ Miniaturisation des composants
 - ❖ Puissances croissantes

 - ◆ Au domaine d'utilisation
 - ❖ Informatique embarquée
 - ❖ Informatique domestique
 - ❖ Informatique d'entreprise
 - ❖ ...

1ère génération (~1950)

■ Ordinateur

- ◆ Ensemble de tubes (~20 000)
- ◆ Chargement manuel par le programmeur à l'aide d'interrupteurs
- ◆ L'utilisateur exécute son programme immédiatement après le chargement
- ◆ Pas de SE

■ Inconvénients

- ◆ Très encombrant
- ◆ Très lent
- ◆ Très peu fiable

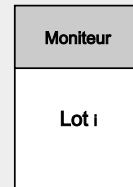
2ème génération

■ Ordinateurs "mainframes"

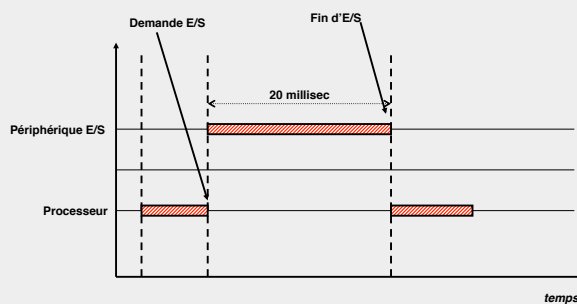
- ◆ Traitement par lots (cartes perforées)
- ◆ L'enchaînement des lots est décrit par une carte perforée spéciale (carte de contrôle)
- ◆ Le SE se limite à un moniteur résident qui enchaîne les lots

■ Inconvénients

- ◆ Lent
- ◆ Non interactif



Mono-programmation



E/S tamponnées

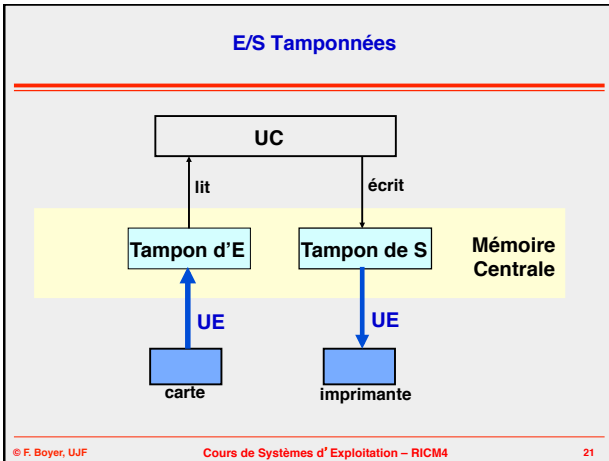
■ Utilisation d'unités d'échange (UE) capables :

- ◆ de fonctionner simultanément avec l'UC (asynchronisme)
- ◆ de transférer des tampons = blocs d'octets

■ Exécution pgm_i // chargement pgm_{i+1}

■ Les cartes sont lues par l'UE et stockées dans des tampons (buffers) d'entrée

■ L'UC lit les données dans le tampon d'entrée, place ces données dans la mémoire, et produit le résultat dans un tampon de sortie

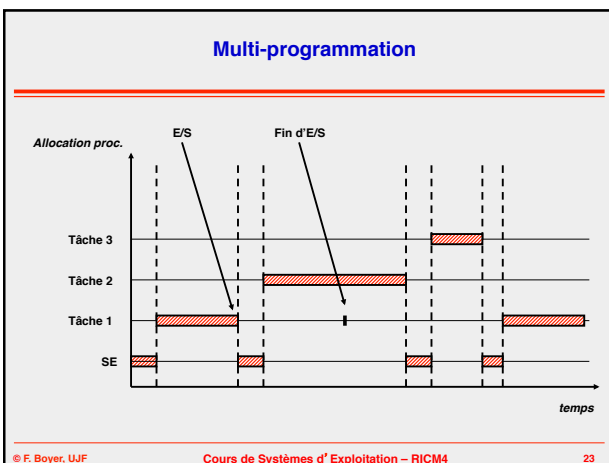


3ème génération (1960/1970)

- **Systèmes multi-programmés**
 - ◆ Plusieurs tâches en mémoire simultanément
 - ◆ Multiplexage du processeur entre les tâches
 - ◆ Perte du processeur lors des E/S
- **Avantages**
 - ◆ Meilleure utilisation de l'UC
- **Inconvénients**
 - ◆ Complexité
 - ◆ Taille de mémoire grande
 - ◆ Partage et protection des ressources

Système d'expl.
Tâche 1
Tâche 2
Tâche 3
Tâche 4

© F. Boyer, UJF Cours de Systèmes d'Exploitation – RICM4 22



Multi-programmation et protection

- Eviter qu'une tâche puisse lire / écrire dans la zone mémoire affectée à une autre tâche
- Eviter qu'une tâche puisse manipuler la zone réservée au système autrement que par les appels système
- Eviter qu'une tâche puisse lire / écrire des données d'E/S d'une autre tâche

→ Complexifie la gestion des ressources (mémoire, E/S, ...)

© F. Boyer, UJF Cours de Systèmes d'Exploitation – RICM4 24

3ème génération (suite)

■ Systèmes à temps partagé (1970)

- ◆ Partage du temps processeur entre les tâches
 - ❖ Quantum de temps
- ◆ Tâches en mémoire ou "swappée" sur disque
 - ❖ Plus grand nombre de tâches
 - ❖ Une mémoire plus grande allouée à chaque tâche

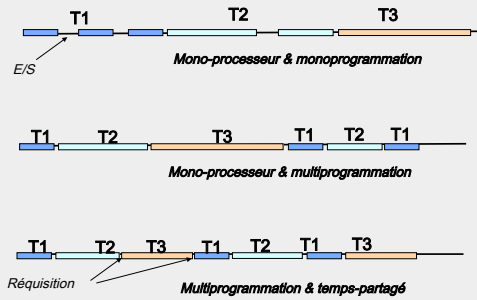
■ Avantages

- ◆ Temps de réponse corrects pour tâches courtes, même en présence de tâches longues et non interactives

■ Inconvénients

- ◆ Complexité
- ◆ L'utilisation du processeur peut être moins bonne

Systèmes mono/multiprogrammés et temps partagé



4ème génération (après 1970)

■ Ordinateurs personnels

■ Architectures spécialisées

- ◆ Parallèles, temps-réel, embarqué

■ Systèmes répartis

- ◆ Architectures cluster /grille/ cloud

■ Des critères de qualité souvent communs

- ◆ Fiabilité, disponibilité, efficacité, extensibilité, faible consommation de ressources, adaptabilité

Ordinateurs personnels

■ Initialement

- ◆ Mono-tâches (MS/DOS, Windows3/11)
- ◆ Dédiés à un utilisateur
- ◆ Pas de protection / sécurité
- ◆ Très interactifs et conviviaux

■ Evolution

- ◆ Multi-tâches
- ◆ Multi-utilisateurs
- ◆ Connexions réseaux

Systèmes temps-réel

■ Contraintes temporelles

- ◆ Temps d'exécution bornés

■ Utilisés dans des situations spécifiques

- ◆ Systèmes d'imagerie médicale
- ◆ Systèmes de contrôle industriel
- ◆ Robotique
- ◆ ...

Systèmes temps-réel

■ *Hard* real-time systems

- ◆ Peu ou pas de mémoire secondaire
- ◆ Pas de temps-partagé
- ◆ Temps de commutation très court
- ◆ SE complètement spécifique

■ *Soft* real-time systems

- ◆ Utilisés dans les applications multimedia, réalité virtuelle
- ◆ Contraintes de temps plus souples
- ◆ Gestion de priorités de tâches
- ◆ Gestion spécifique de la mémoire

Systèmes mobiles

■ Personal Digital Assistants (PDAs)

■ Téléphones cellulaires

■ Large gamme d'applications

■ Systèmes d'exploitation spécifiques

■ Contraintes

- ◆ Mémoire limitée
- ◆ Processeurs lents
- ◆ Petits écrans
- ◆ Faible consommation d'énergie

Systèmes parallèles multiprocesseurs

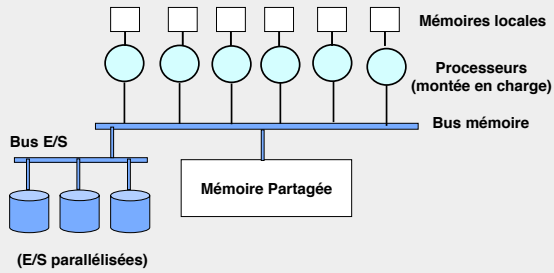
■ SMP (Symetric Multi Processeurs)

- ◆ Systèmes généralistes (serveurs BD, Web, SI, NFS, ...)
- ◆ Processeurs standards
- ◆ Partage total de la mémoire

■ Machines de calcul parallèles

- ◆ Architectures spécialisées
 - ❖ Processeurs spécifiques pour le calcul vectoriel
 - ❖ Réseaux spécialisés
 - ❖ Partage total ou partiel de la mémoire (bancs)

Systèmes parallèles multiprocesseurs



© F. Boyer, UJF

Cours de Systèmes d'Exploitation – RICM4

33

Systèmes clusterisés

■ Plusieurs noeuds

- ◆ Homogènes
- ◆ Partageant un disque (share disk) ou bien ne partageant rien (share nothing)

■ Connectés par un réseau spécialisé (SCI, Ethernet, ...)

- ◆ Réseaux locaux à haut débit (Local Area Network)
- ◆ Echanges rapides de données de grandes taille

■ 3 caractéristiques :

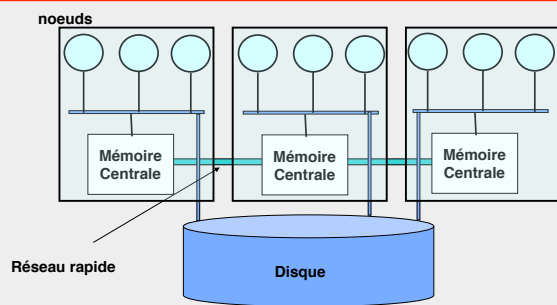
- ◆ Montée en charge
- ◆ Passage à l'échelle (scalability)
- ◆ Haute disponibilité (serveurs back-up)

© F. Boyer, UJF

Cours de Systèmes d'Exploitation – RICM4

34

Systèmes clusterisés



© F. Boyer, UJF

Cours de Systèmes d'Exploitation – RICM4

35

Grilles de calcul

■ Infrastructures réparties composées d'un ensemble de machines hétérogènes et potentiellement distantes

■ Milliers de machines

- ◆ Grid5000

■ Calcul intensif

■ Garanties de disponibilité

■ Système de réservation des ressources matérielles

© F. Boyer, UJF

Cours de Systèmes d'Exploitation – RICM4

36

Cloud computing

■ **Système réparti dans lequel les clients utilisent des services déportés sur un ou plusieurs serveurs distants, sans avoir à administrer ces services ni les machines qui les supportent**

■ **3 modèles**

- ◆ IaaS (Infrastructure as a Service)
- ◆ PaaS (Platform as a Service)
- ◆ SaaS (Software as a Service)

■ **Amazon EC2, Azure, ...**