

Cours de Systèmes d'Exploitation

RICM 4, Année 2011-2012

Fabienne Boyer
UFR IMA, Université Joseph Fourier

Fabienne.Boyer@imag.fr



Plan du cours de système d'exploitation

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. Présentation du système Nachos
8. Gestion de la mémoire secondaire

Organisation de l'enseignement

■ Equipe pédagogique

- ◆ Philippe Morat (Equipe ADELE, Laboratoire LIG)
- ◆ Fabienne Boyer (Equipe SARDES, Laboratoire LIG, INRIA Rhône-Alpes)
- ◆ Noël de Palma (Equipe SARDES, Laboratoire LIG, INRIA Rhône-Alpes)

■ Volume

- ◆ 12 Cours d'1H30
- ◆ 12 TD d'1H30
- ◆ 1 TP en Java
- ◆ 1 examen de 2 heures

Ce cours a été conçu à partir de...

■ E. Berthelot

◆ <http://www.iie.cnam.fr/~EBerthelot/>

■ A. Sylberschatz

◆ www.sciences.univ-nantes.fr/info/perso/permanents/attiogbe/SYSTEME/CoursSysteme.html

■ A. Griffaut

◆ <http://dept-info.labri.fr/~griffault/Enseignement/SE/Cours>

■ H. Bouzourfi, D. Donsez

◆ <http://www-adele.imag.fr/~donsez/cours/#se>

Bibliographie

■ Livres

- ◆ Silberschatz, Galvin and Gagne, Operating System Concepts, Addison-Wesley
- ◆ S. Krakowiak, Principes des systèmes d'exploitation des ordinateurs, Dunod
- ◆ A. Tanenbaum, Les Systèmes d'exploitation, Prentice-Hall, Inter-Editions
- ◆ Maurice Bach, La conception du système Unix, Masson
- ◆ J.M. Rifflet, La programmation sous Unix, McGraw-Hill

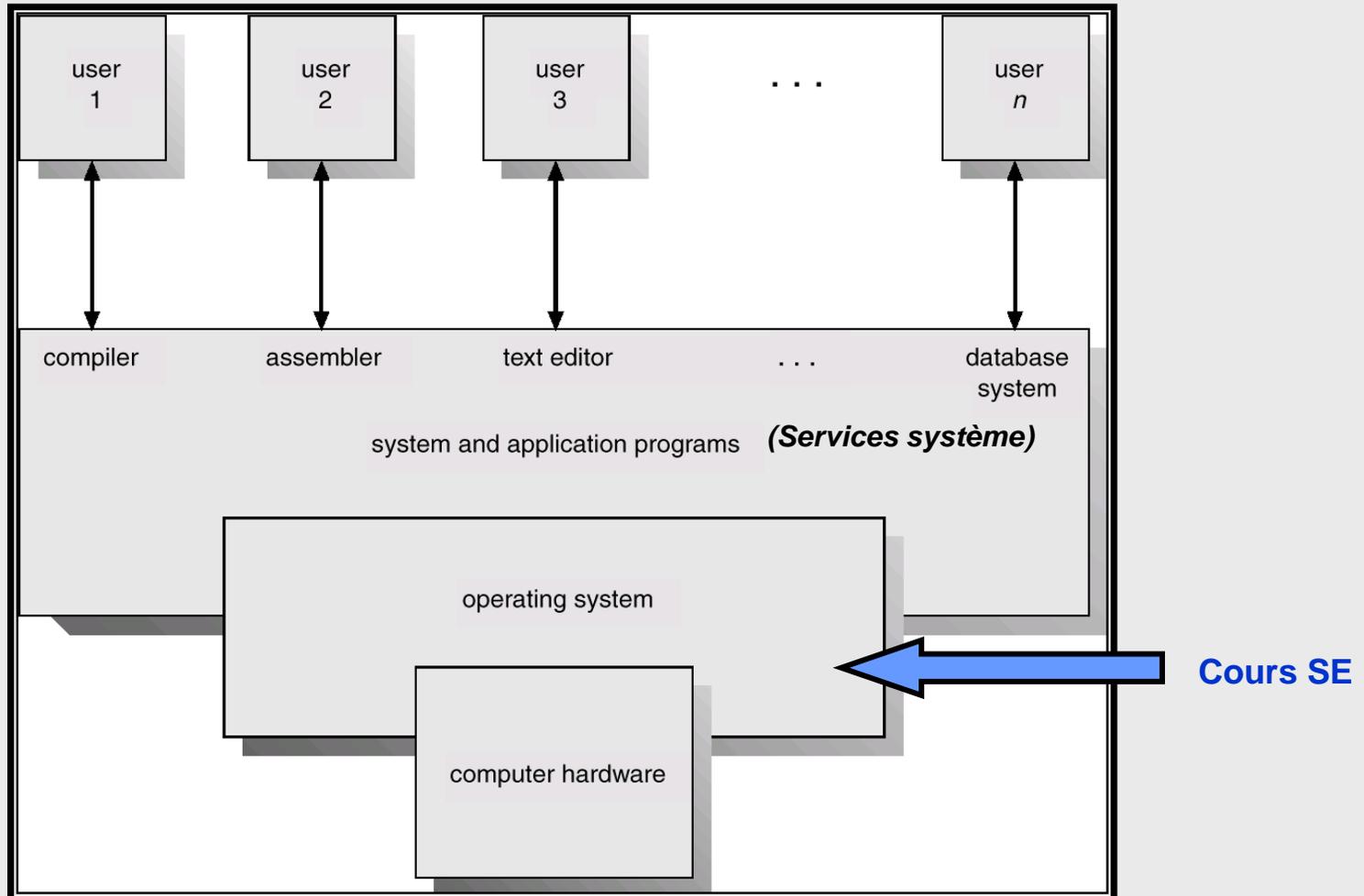
■ **Systèmes d'exploitation**

- ◆ **Rôles**
- ◆ **Organisations**
- ◆ **Historique**

■ Intermédiaire entre l'ordinateur et les applications/utilisateurs

- ◆ Ordinateur vu comme un ensemble de ressources (physiques + 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



A. Silberschatz, Calvin and Gagne, 2002

Structure d'un SE

■ Noyau

- ◆ Partie qui réside constamment en mémoire centrale
 - ❖ Routines qui requièrent un mode d'exécution privilégié
 - ❖ Données globales et partagées 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 / L4
- ◆ 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)

■ Fonctions système (API)

- ◆ Pour le programmeur d'applications, le SE apparaît comme un ensemble de fonctions appelables, permettant de manipuler les ressources physiques et logiques
- ◆ Une *fonction système* peut interagir avec un serveur système pour la réalisation d'un service particulier
- ◆ Au plus bas niveau, une fonction système peut engendrer un *appel système* (appel au noyau)

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

■ **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

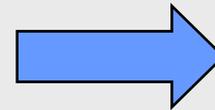
Ressources gérées par le SE

- Programme / application
- Processus
- Thread (processus léger)
- Pilotes



activités

- Mémoire centrale
- Fichiers
- Mémoires secondaires



données

Gestion des 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***

■ Variations 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”

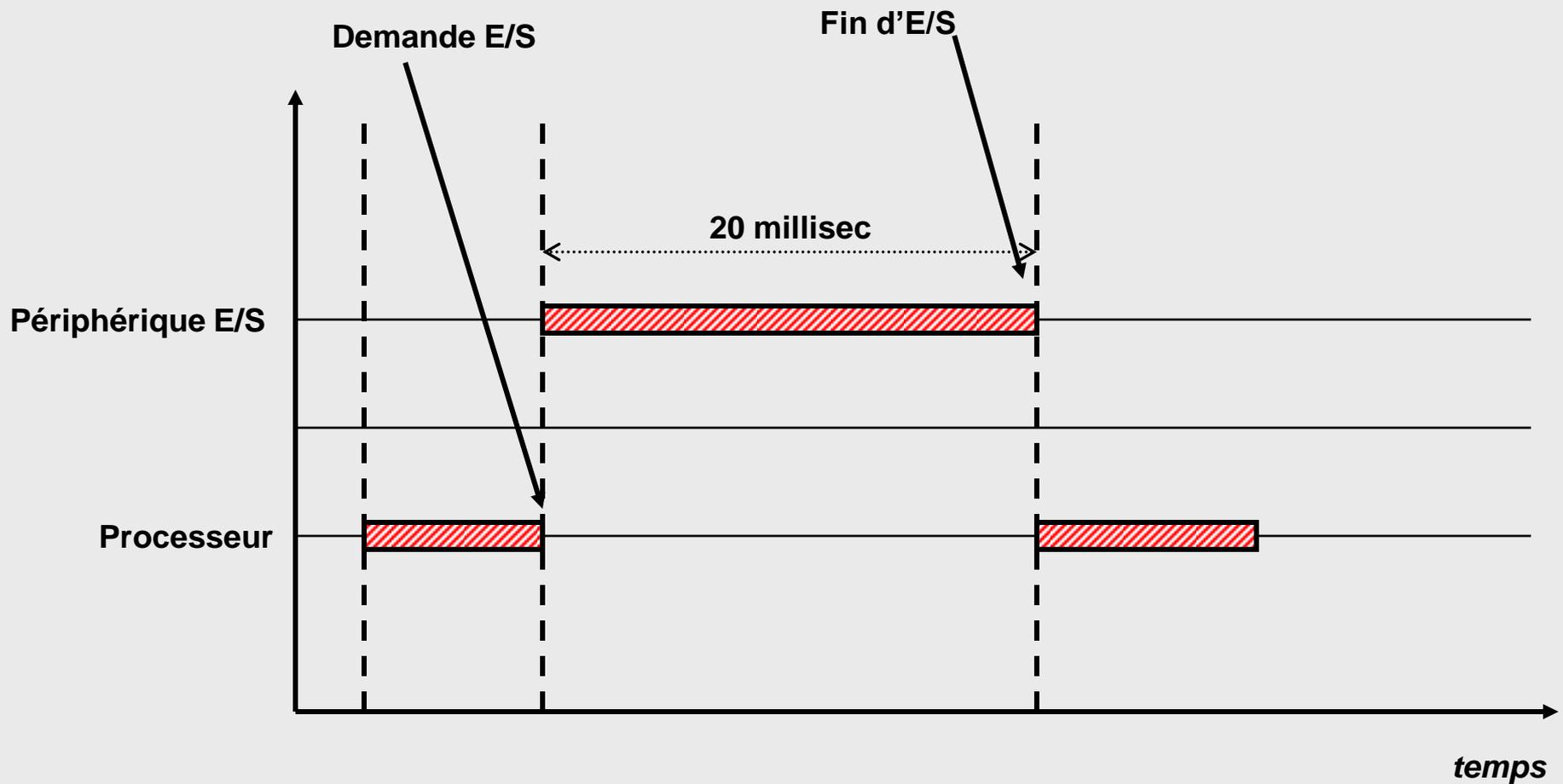
- ◆ Traitement par lots (cartes perforées)
- ◆ L’enchaînement des lots est décrit par une carte perforée spéciale (carte de controle)
- ◆ Le SE se limite à un moniteur résident qui enchaîne les lots

■ Inconvénients

- ◆ Lent
- ◆ Non interactif



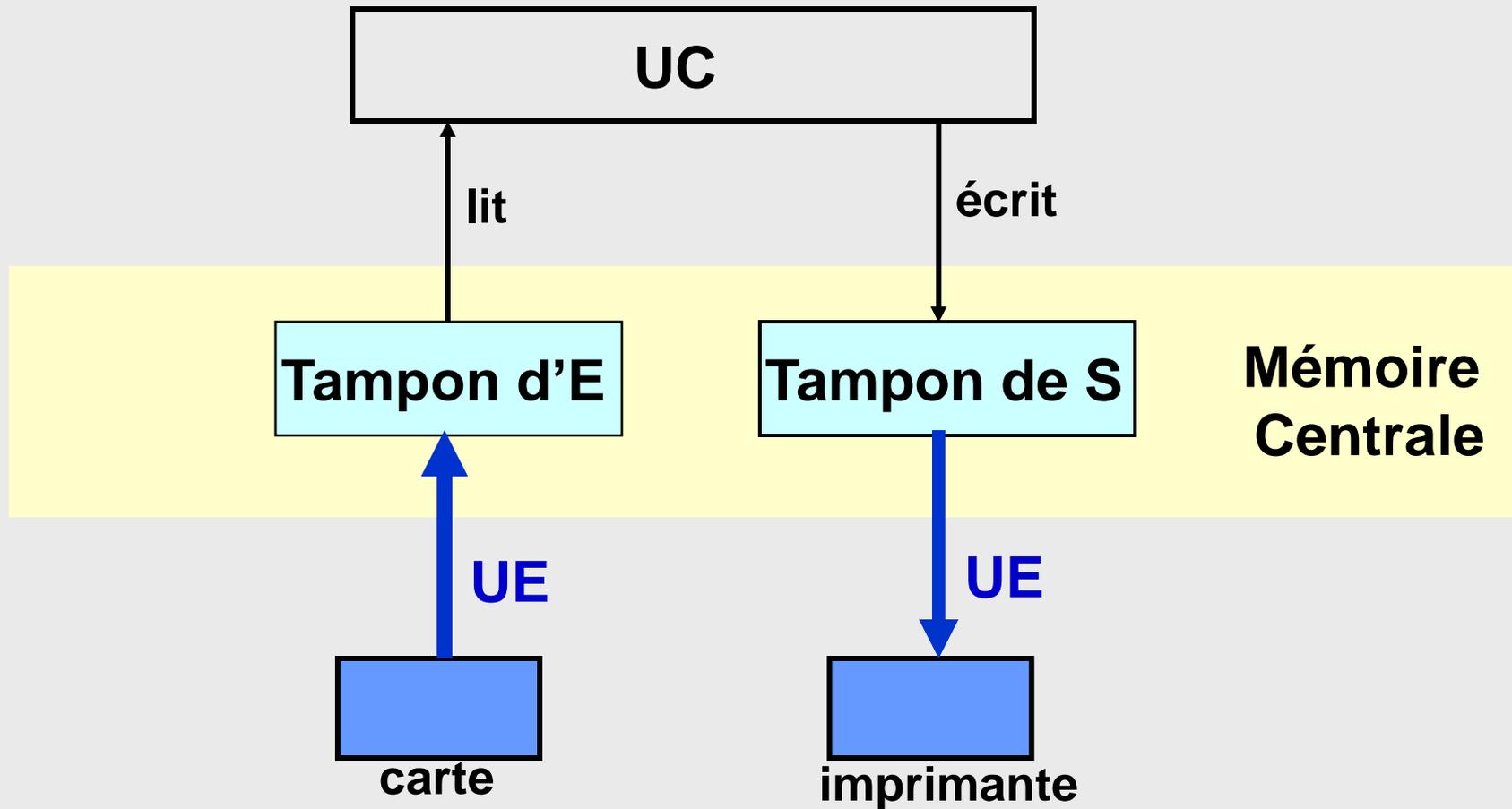
Mono-programmation



Evolution 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**

E/S Tamponnées



3ème génération (1960/1970)

■ Systèmes multi-programmés

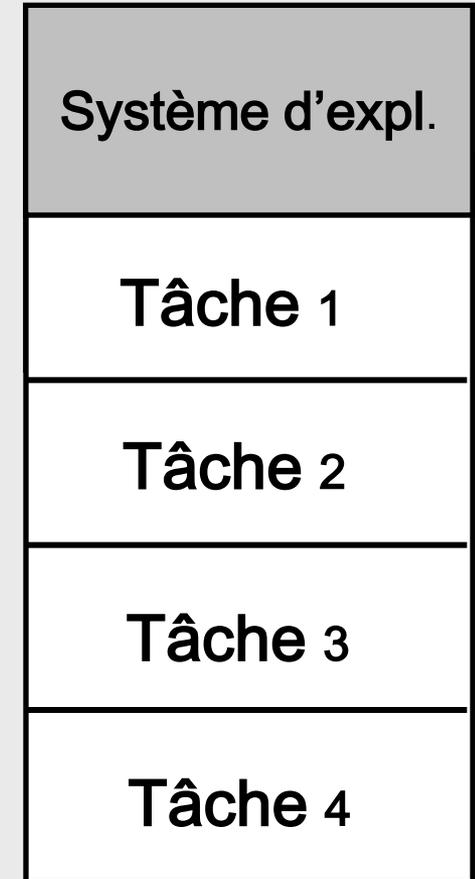
- ◆ Plusieurs tâches en mémoire simultanément
- ◆ Multiplexage du processeur entre les tâches
- ◆ Perte du processeur lors des E/S

■ Avantages

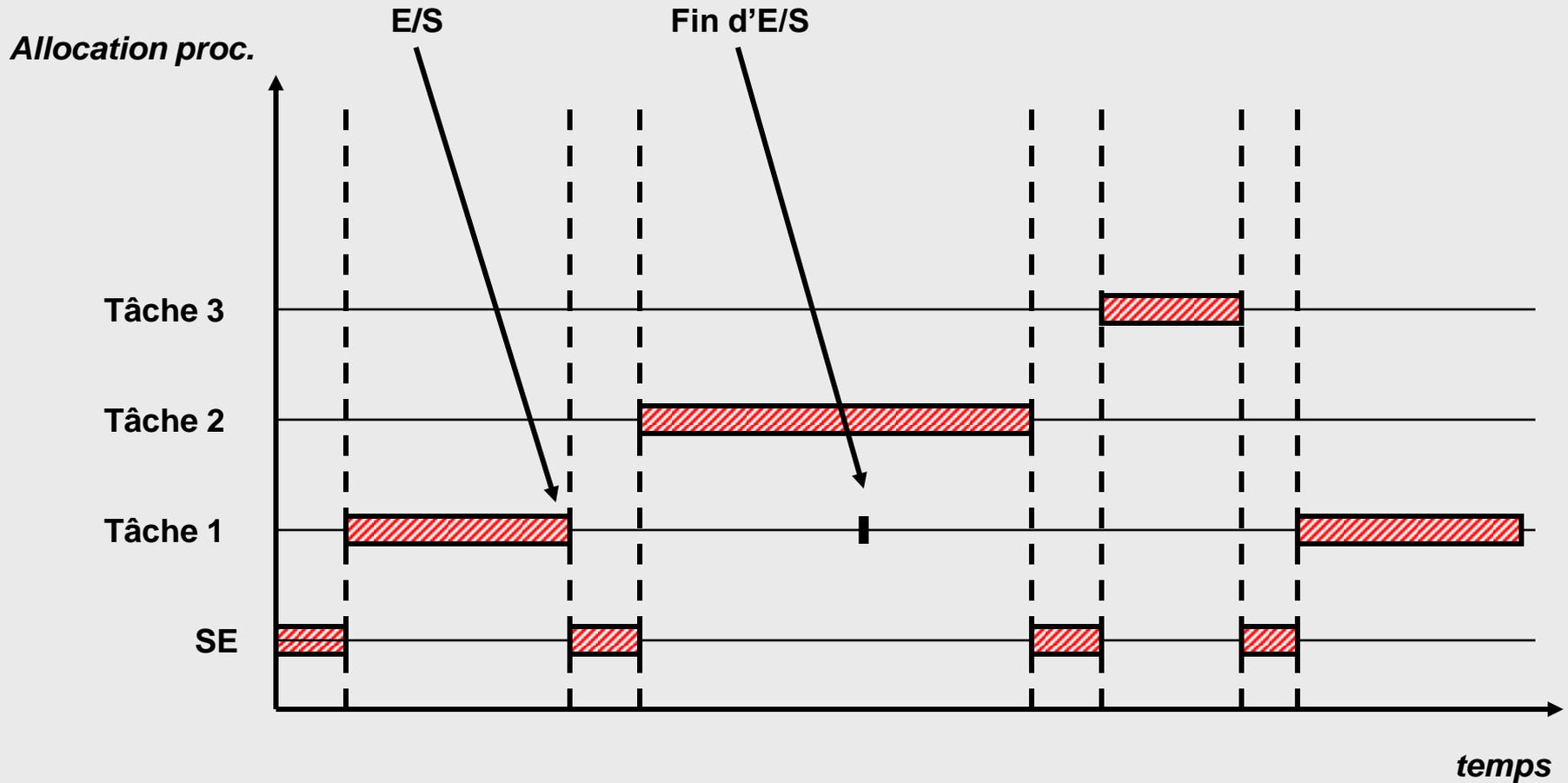
- ◆ Meilleure utilisation de l'UC

■ Inconvénients

- ◆ Complexité
- ◆ Taille de mémoire grande
- ◆ Partage et protection des ressources



Multi-programmation



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, ...)

3ème génération (suite)

■ **Systemes 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

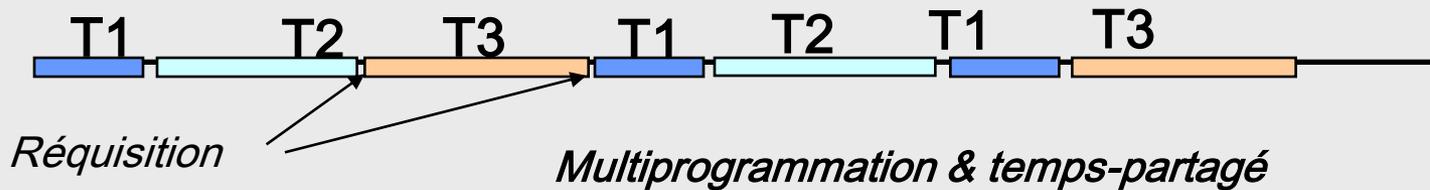
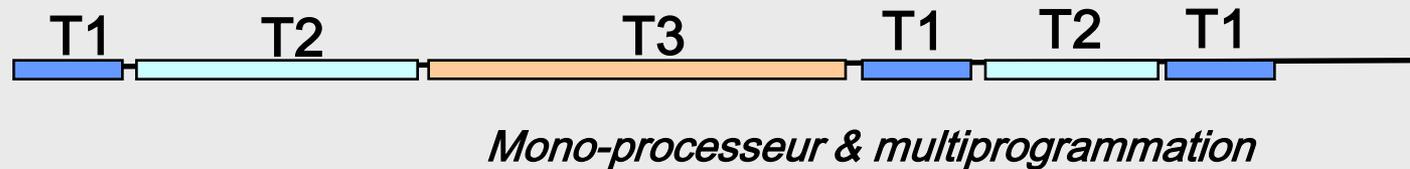
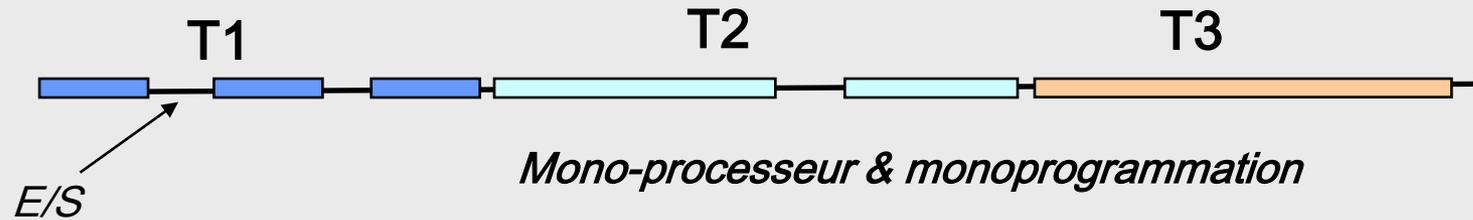
■ **Avantages**

- ◆ 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

Systemes mono/multiprogrammes et temps partage



4ème génération (après 1970)

- **Ordinateurs personnels**
- **Architectures spécialisées**
 - ◆ Parallèles, temps-réel, embarqué
- **Virtualisation**

- **Evolution vers les systèmes répartis**
 - ◆ Architectures cluster / 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
- ◆ Connections réseaux

Systemes temps-réel

■ Contraintes temporelles

- ◆ Temps d'exécution bornés

■ Utilisés dans des situations spécifiques

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

Systemes 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

Systemes mobiles

- **Personal Digital Assistants (PDAs)**
- **Téléphones cellulaires**
- **Large gamme d'applications**
- **Systemes d'exploitation spécifiques**
- **Contraintes**
 - ◆ **Mémoire limitée**
 - ◆ **Processeurs lents**
 - ◆ **Petits écrans**
 - ◆ **Faible consommation d'énergie**

Systemes parallèles multiprocesseurs (1/2)

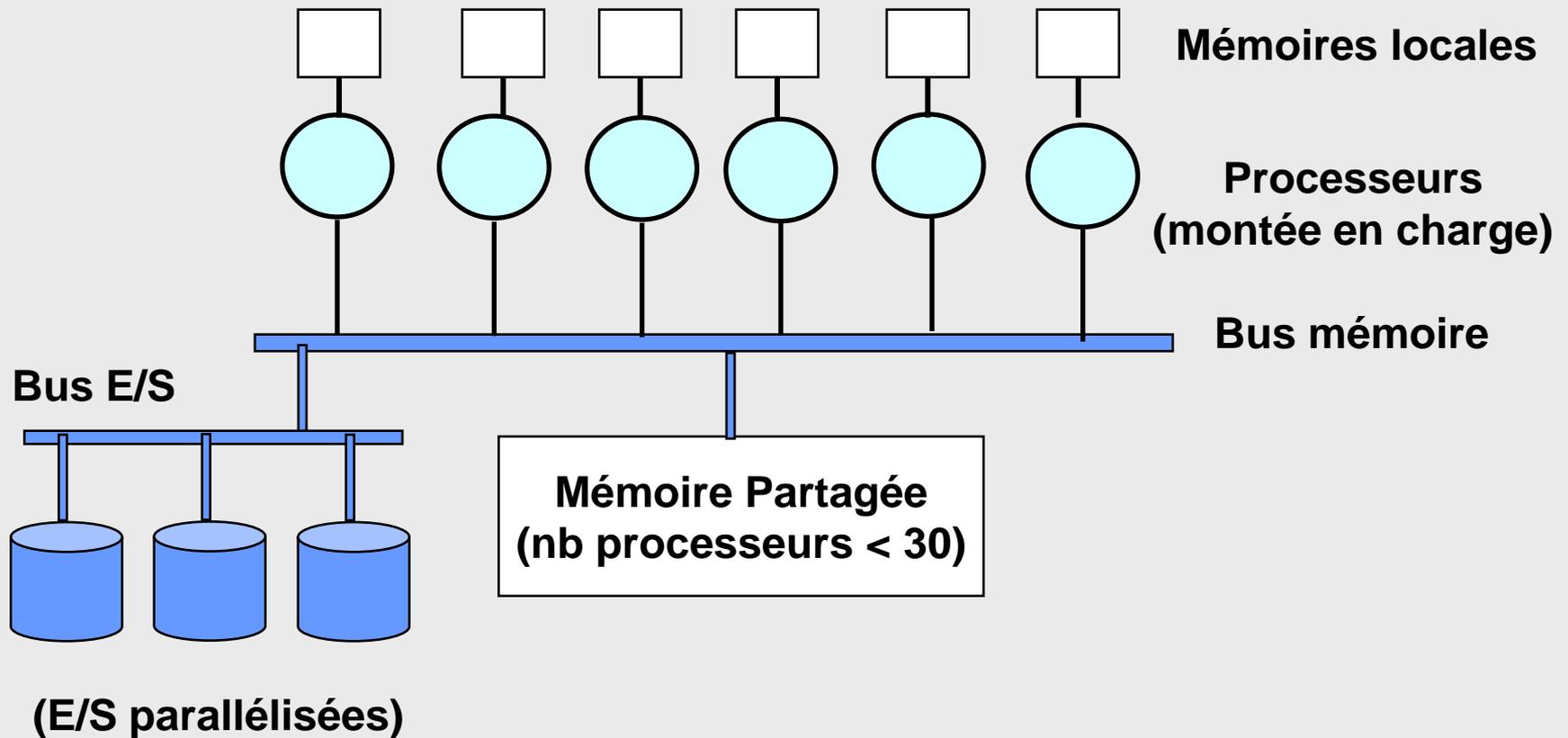
■ SMP (Symetric Multi Processeurs)

- ◆ Systemes 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)

Systemes paralleles multiprocesseurs (2/2)



Systemes clusterises (1/3)

■ Plusieurs noeuds

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

■ Connectés par un reseau specialise (SCI, Ethernet, ...)

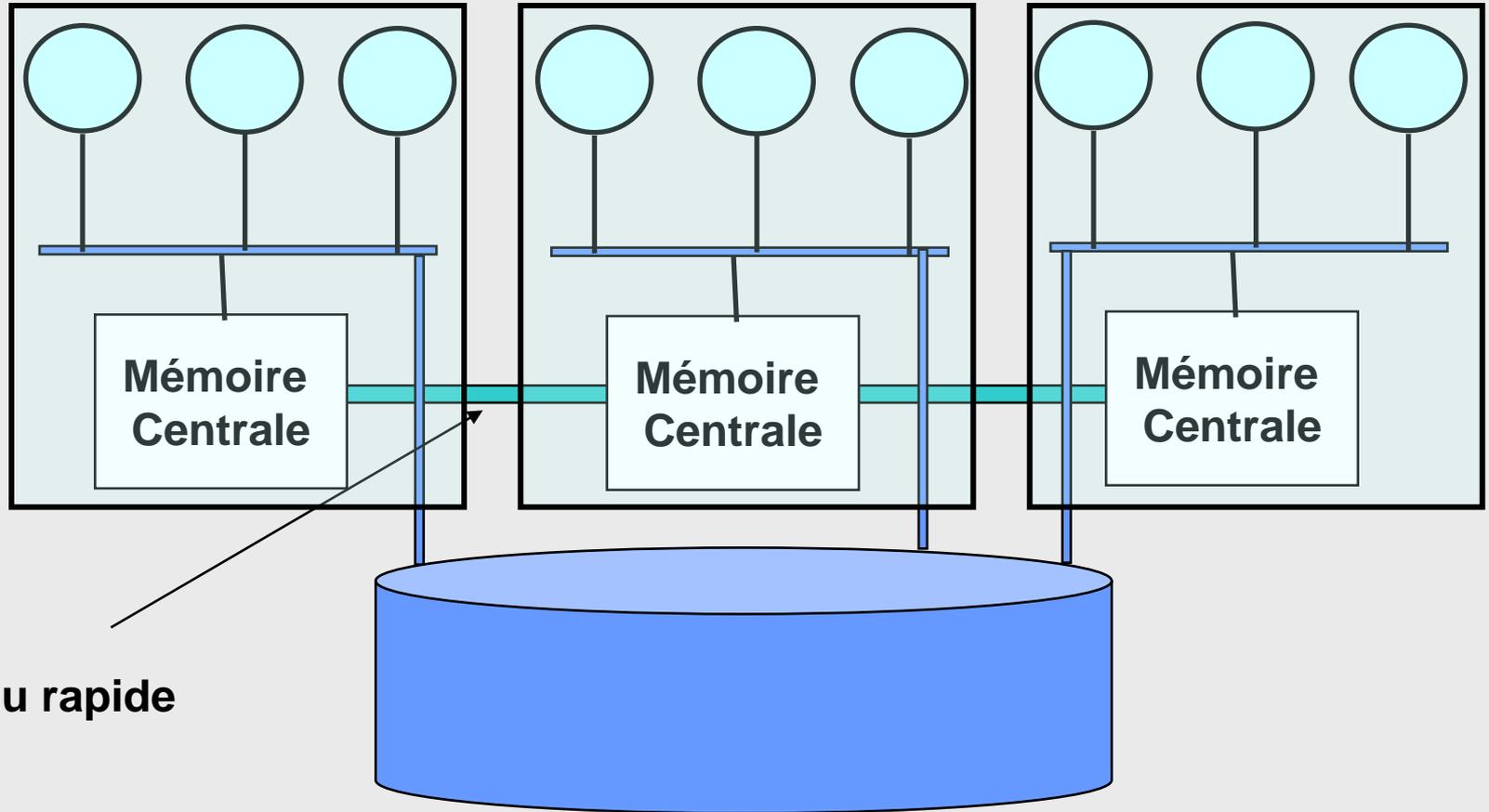
- ◆ Invention des reseau locaux à haut debit (Local Area Network)
- ◆ Echanges rapides de donnees de grandes taille

■ 3 caracteristiques :

- ◆ Montee en charge
- ◆ Passage à l'echelle (scalability)
- ◆ Haute disponibilite (serveurs back-up)

Systemes clusterises (2/3)

noeuds



Réseau rapide

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**

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**
- **Les services proposés par le Cloud peuvent s'exécuter au sein d'architectures cluster**
- **3 modèles**
 - ◆ **IaaS (Infrastructure as a Service)**
 - ◆ **PaaS (Platform as a Service)**
 - ◆ **SaaS (Software as a Service)**

Couverture du terme *systeme réparti*

■ Englobe maintenant

- ◆ Architectures distribuées fortement couplées :
 - ❖ Systèmes parallèles multi-processeurs
 - ❖ Architectures clusters

- ◆ Architectures distribuées faiblement couplées
 - ❖ Grilles de calcul
 - ❖ Cloud computing
 - ❖ Applications s'exécutant sur Internet

Motivations pour les systèmes répartis

- **Partage de ressources sur le réseau**
 - ◆ Processeur, mémoire, serveurs / services
 - ◆ Mobilité du code et des données

- **Potentiel**
 - ◆ Disponibilité
 - ◆ Montée en charge
 - ◆ Passage à l'échelle
 - ◆ Administration

- **Le SE réparti est mis en œuvre par une couche intergicielle fonctionnant au dessus des SE des serveurs du système réparti**