



Cours n° 1

Introduction au Noyau

Mécanismes internes du noyau (UNIX)

Processus,
Ordonnancement
Fichiers,
Mémoire virtuelle

Organisation

Sur 10 semaines

Début TD/TME : La semaine du 24 septembre

Note finale = 40 % CC + 60 % Examen

Equipe enseignante

Cours : Pierre Sens

Pierre.Sens@lip6.fr

TD/TME : Emmanuelle Encrenaz, Sebastien Monnet, Gael Thomas
{Nom.Prenom}@lip6.fr

Bibliographie

Programmation système

J.M. Rifflet, «La Programmation sous UNIX»,
J.M. Rifflet, «Les communications sous UNIX»,
C. Blaess, «Programmation système en C sous Linux»

Mécanismes internes du noyau UNIX

M.J. Bach, «Conception du système UNIX»,
S.J Leffler & al. , «Conception et implémentation du système 4.4 BSD»,
U. Vahalia, «Unix internals --the new frontiers»

Noyau Linux :

D. P. Bovet, M. Cesari, « Le noyau Linux »
R. Love, « Linux Kernel Development »
R. Card & al., «Programmation Linux 2.0»
Linux Magazine Sept./Octobre 2003 - « Voyage au centre du noyau

Mécanismes internes Windows

Inside Windows 2000, 3rd Edition

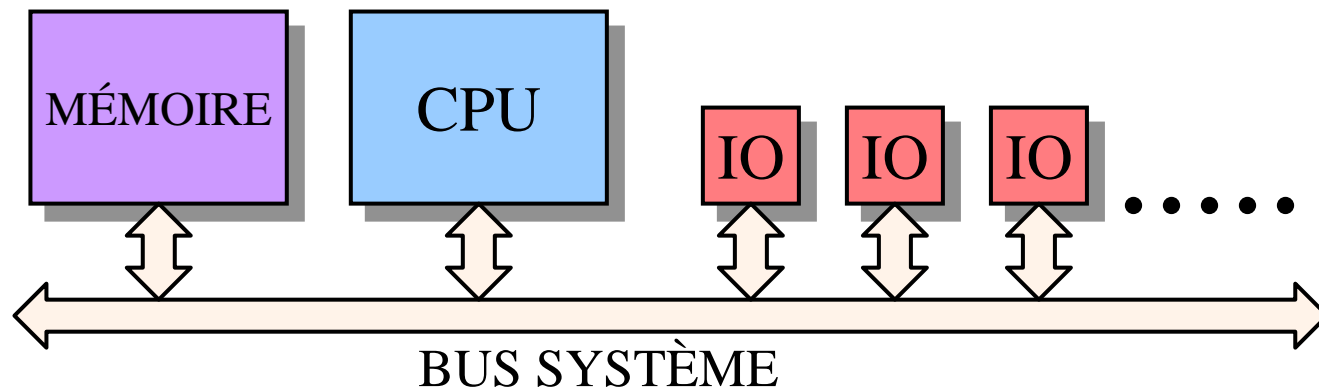
Liens web : voir sur la page du module

<http://www.infop6.jussieu.fr/lmd/2004/master/ue/noyau-2004oct>

Vue d'une machine

Ensembles de composants organisées autour de bus

Composants de base: mémoire, CPU, I/O, bus système



I/Os “standard”: cartes SCSI et/ou IDE, clavier, souris, haut-parleurs, etc.

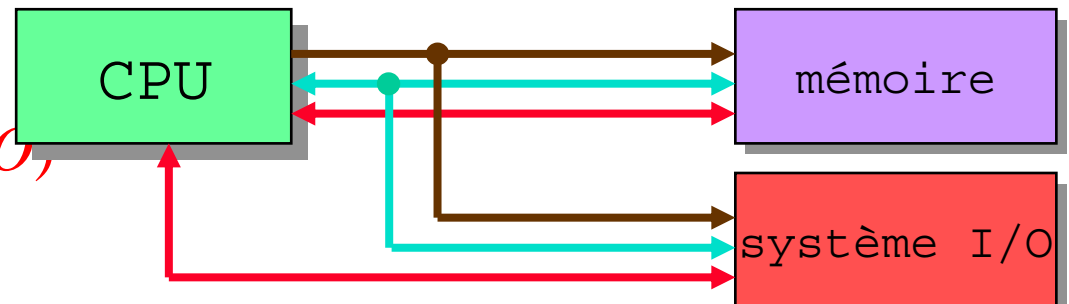
La vitesse du bus système devient le facteur prédominant pour la performance d'un ordinateur.

Vue d'une machine : Accès aux périphériques

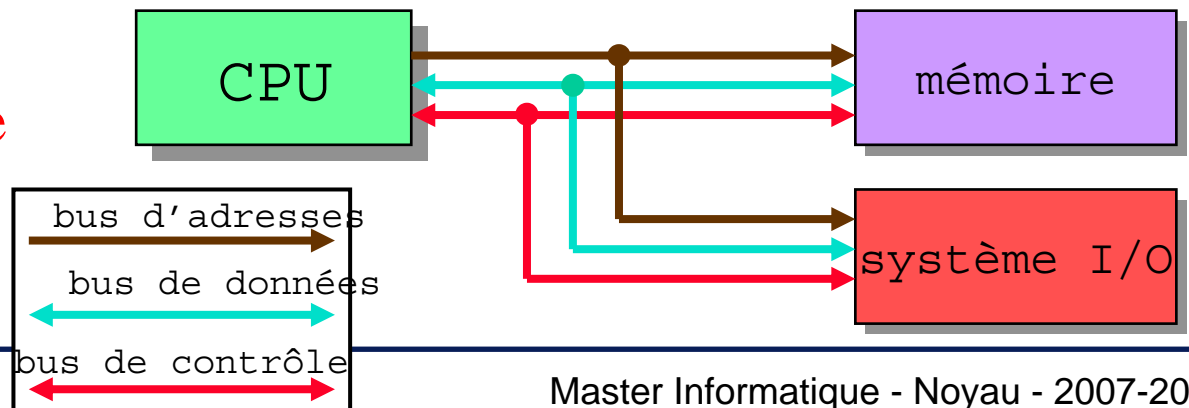
Bus dédié (*isolated I/O*)



Adresses partagés (*shared I/O*)



E/S mappée en mémoire
(*memory-mapped I/O*)

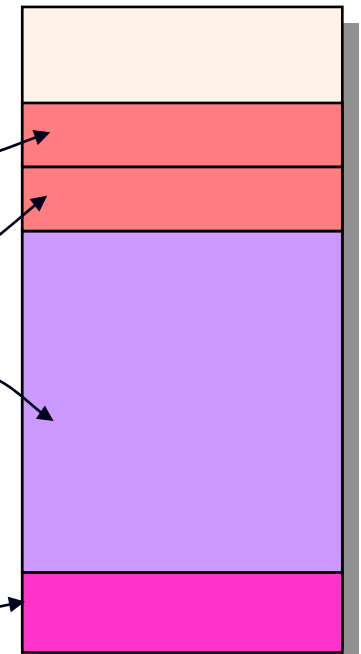
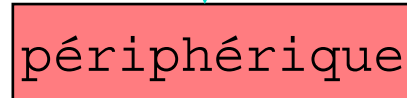
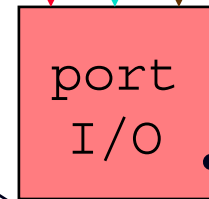
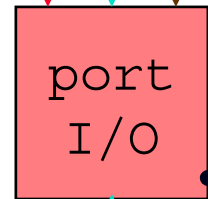
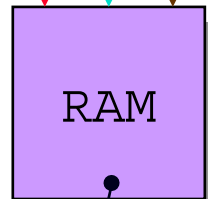
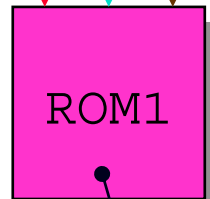
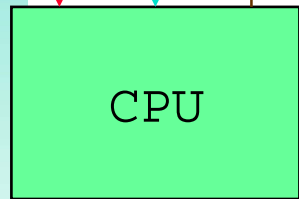


Vue d'une machine : E/S mappées en mémoire

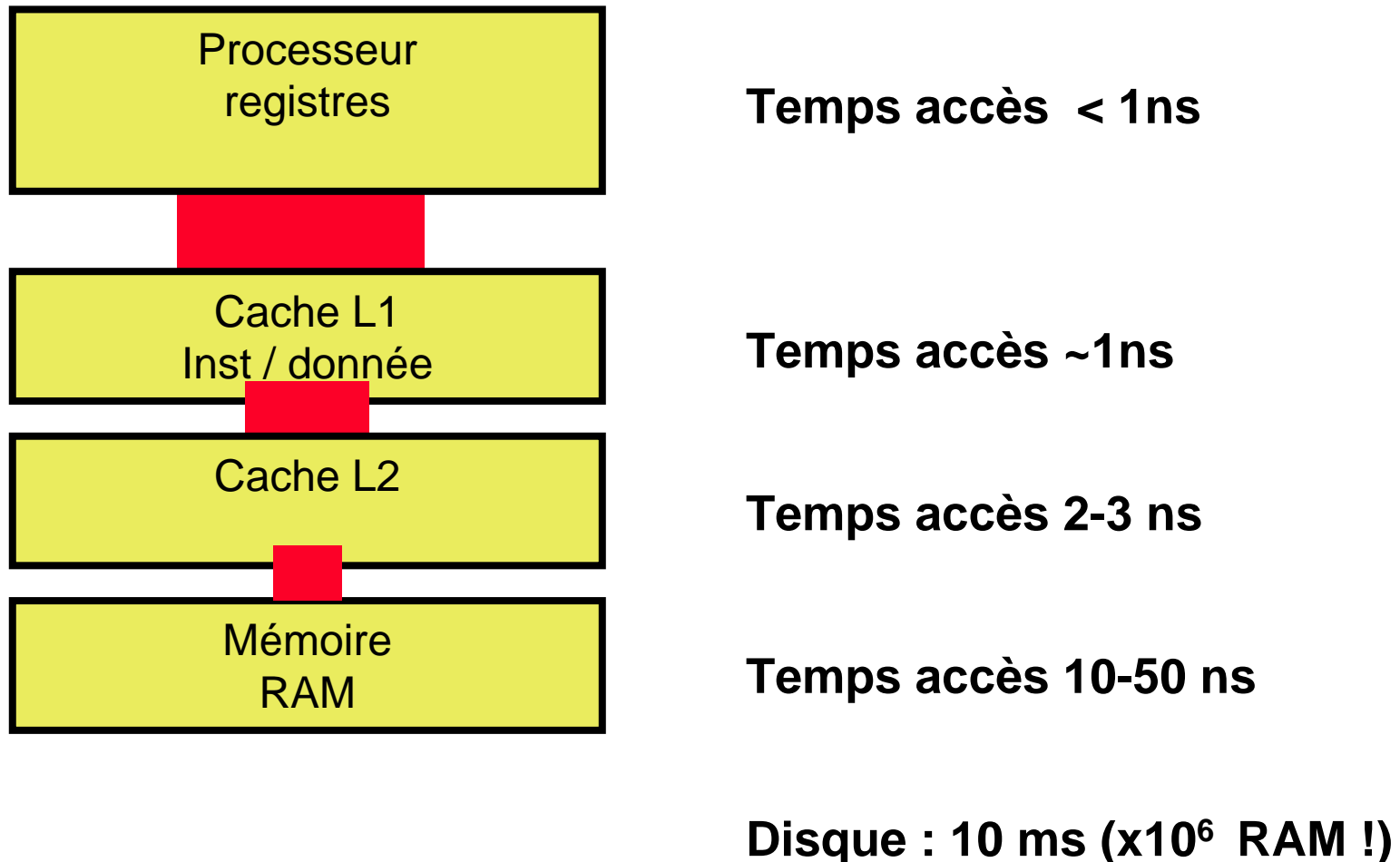
bus d'adresses

bus de données

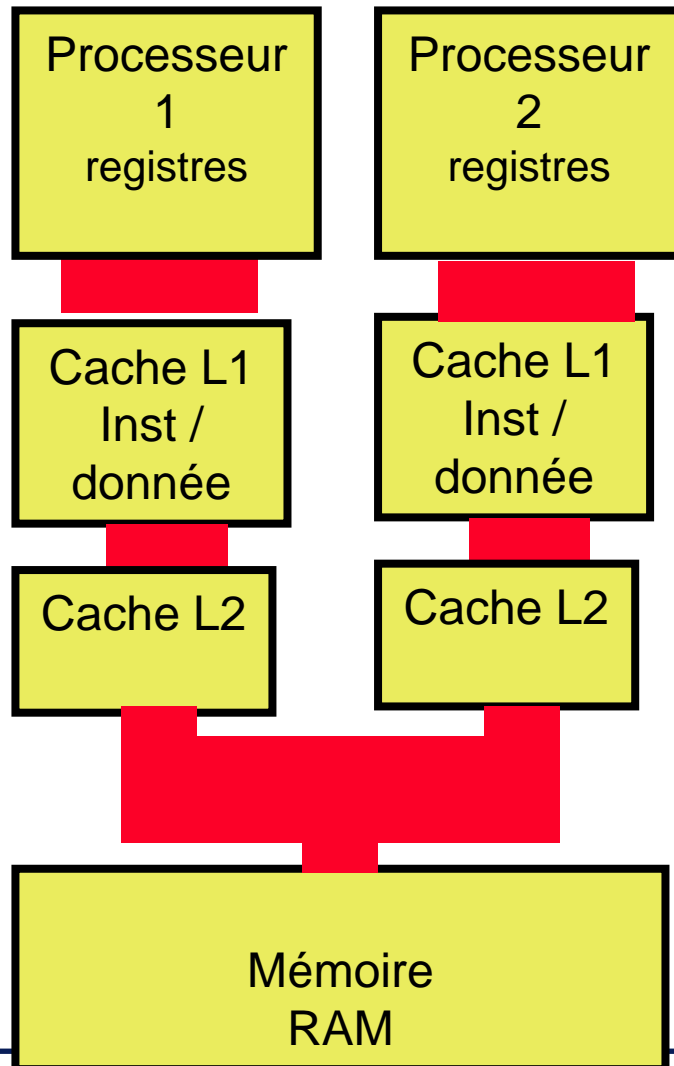
bus de contrôle

espace des
adresses
physiques

Vue d'une machine : Les niveaux de caches



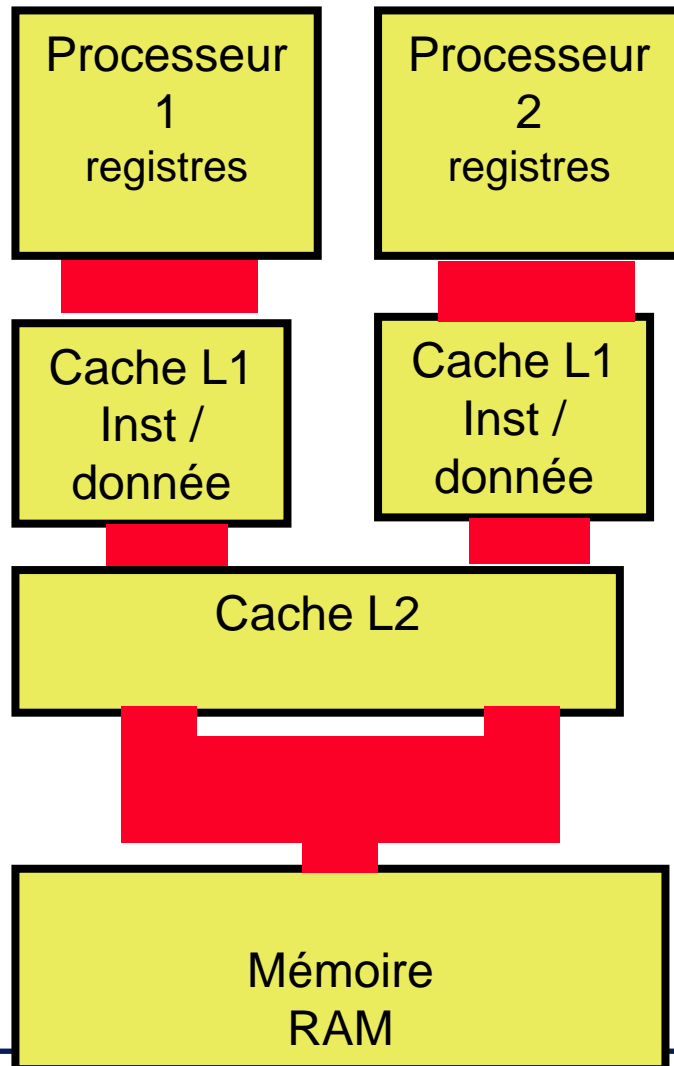
Vue d'une machine : Multi-processeur SMP



Architecture *Symmetric Multiprocessor (SMP)*

- Processeurs distincts, pas de partage de caches
- Partage de la RAM
- + Gestion de flux indépendants
- Maintien de la cohérence des caches
- Conflit d'accès au bus mémoire (nb de processeurs limités)

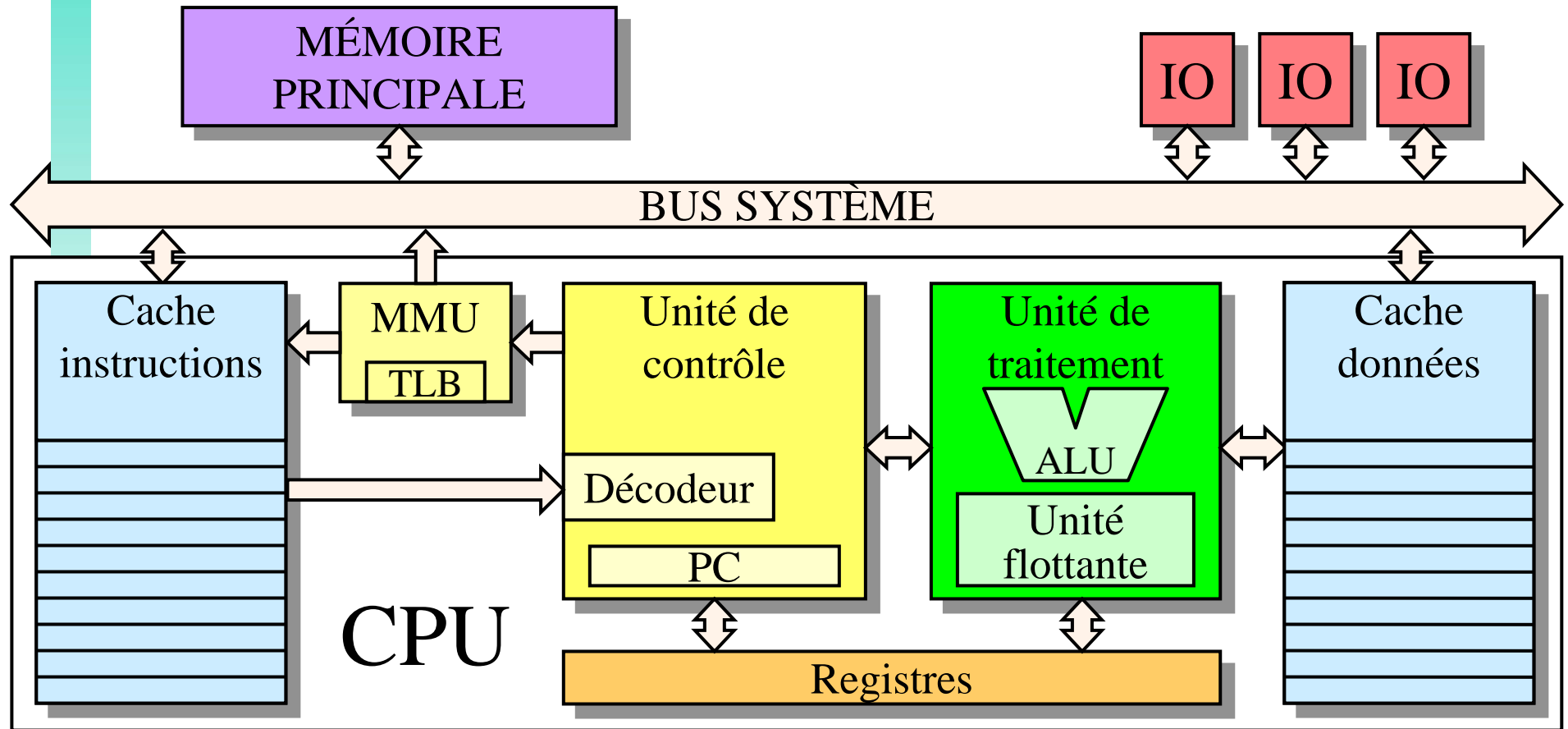
Vue d'une machine : Multi-core



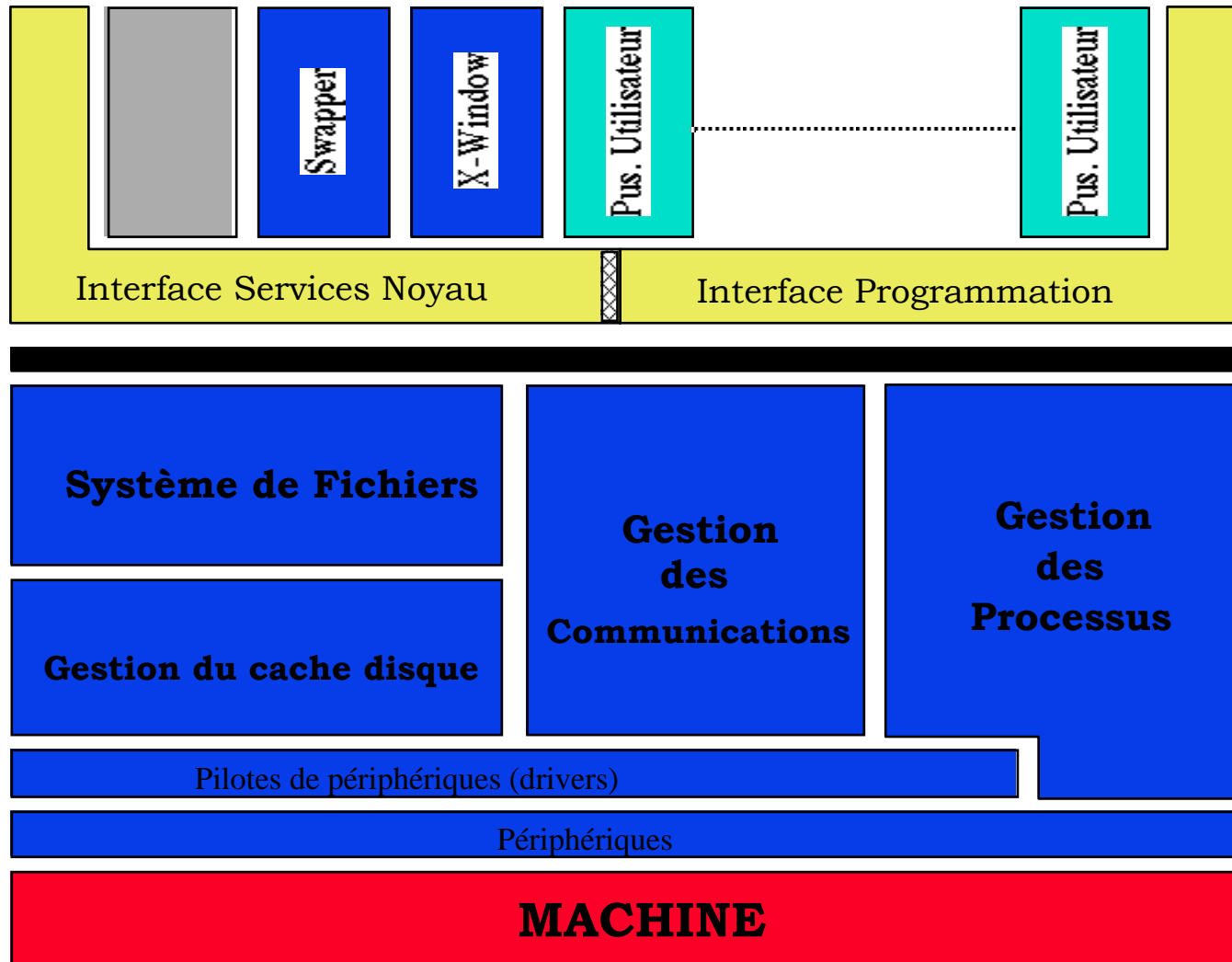
Architecture *Multi-core*
(ex. Intel Dual/Quad core, Cell)

- Deux cœurs de processeurs distincts sur un même support, partage de cache possible (L2 ou L3)
- Partage de la RAM
- + Gestion de flux indépendants
- + Moins coûteux que SMP
- + Moins de pb de cohérence
- Moins de cache disponible

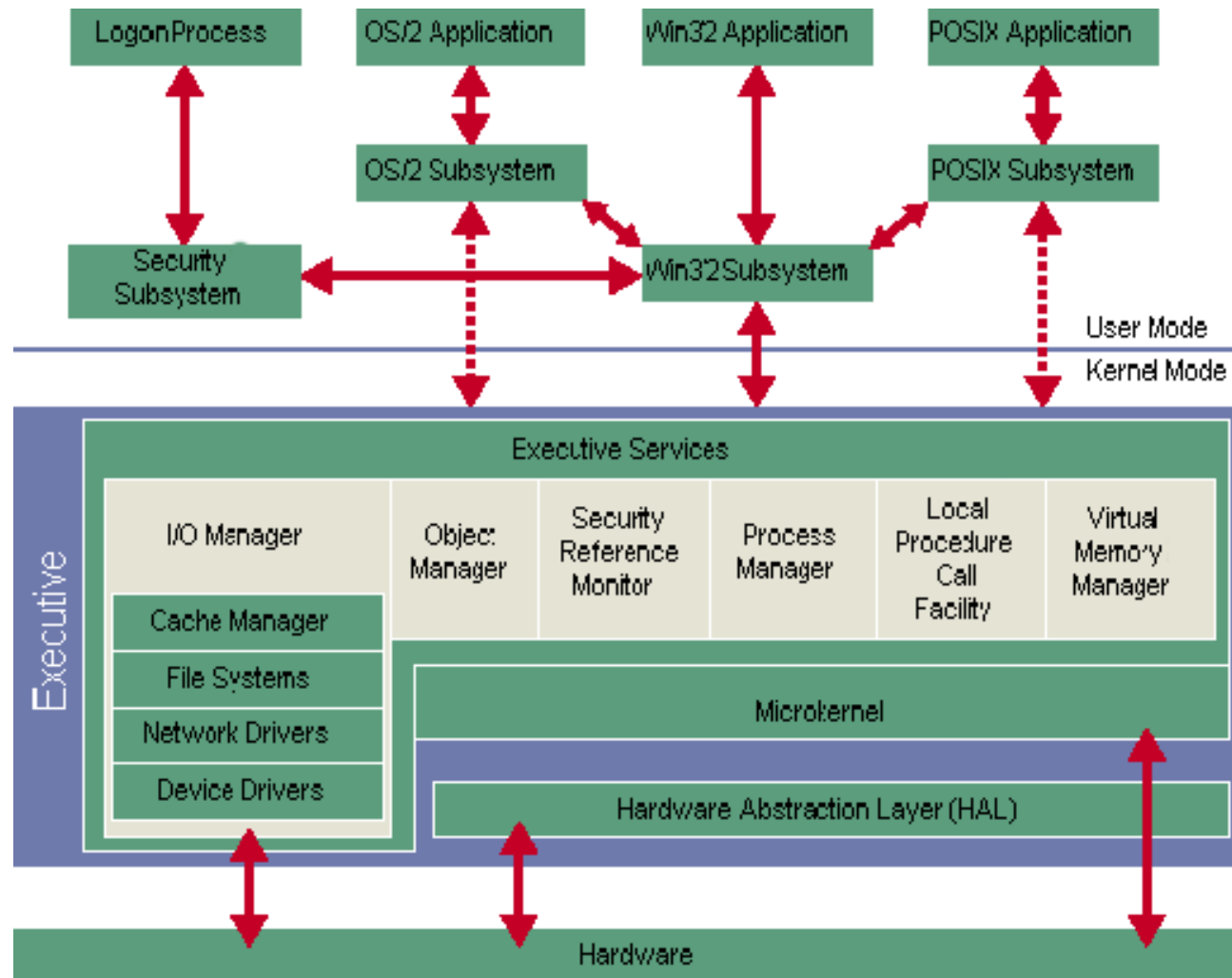
Vue d'une machine



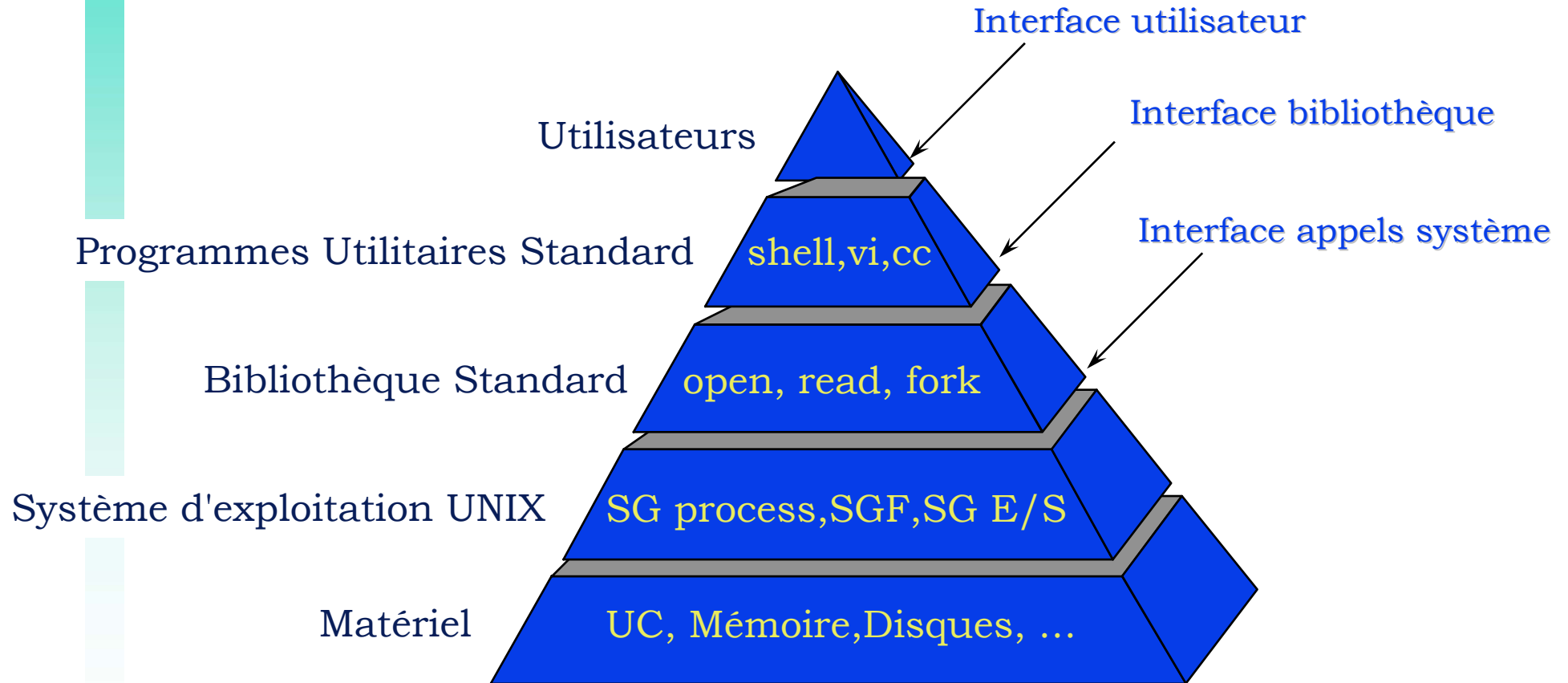
Vue du système : architectures classiques



Vue du système : architectures modulaire (Windows NT)



Généralités (1)



Généralités (2)

Système interactif en temps partagé, Multi-Utilisateurs et Multi-Tâches

Principes

- Système de gestion de fichiers hiérarchisé
- Entrées/Sorties
- Création dynamique de processus (Père / Fils)
- Communication inter-processus (Pipes, Sockets)

Langage de commande extensible (Shell)

Noyau monolithique portable

- Le noyau est écrit en C à 95%.
- UNIX existe sur de nombreuses machines (PC, Stations RISC, CRAY-YMP, Hypercubes, ...)

Les objectifs

Simplicité et efficacité (par opposition aux gros systèmes MULTICS ...):

Efficacité dans la gestion des ressources

Fournir des services d'exécution de programmes

Charger, Exécuter, Gérer les erreurs, Terminaison

Entrées / Sorties à partir de périphériques (Créer, Lire. Ecrire, ...).

Détecter les erreurs (CPU, mémoire, E/S, ...).

Fournir des services d'administration

Allocation des ressources système.

Gestion des utilisateurs.

Comptabilité et statistiques (accounting)

Configuration.

Protection des ressources.

Ajout et retrait de gestionnaires de périphériques (drivers).

Tentatives de Normalisation

POSIX : Compromis entre BSD et Systeme V

Proche de V7 de Bell Labs + signaux + gestion des terminaux

O.S.F (Open Software Foundation) : IBM, DEC, HP, ...

Conforme aux normes IEEE + outils

X11 : système de fenêtrage,

MOTIF : interface utilisateur

DCE : calcul réparti

DME : gestion répartie

...

U.I (UNIX International) : AT&T, Sun, ...

Système V release 4.0

mais aussi ...

**AIX (IBM), Spix (Bull), Ultrix (Digital), HP-UX (HP), SCO-UNIX (SCO),
SunOS & Solaris(Sun Microsystems), ... LINUX, FreeBSD, ...**

Historique (1)

A l'origine UNICS (UNIplicated information and Computing System)

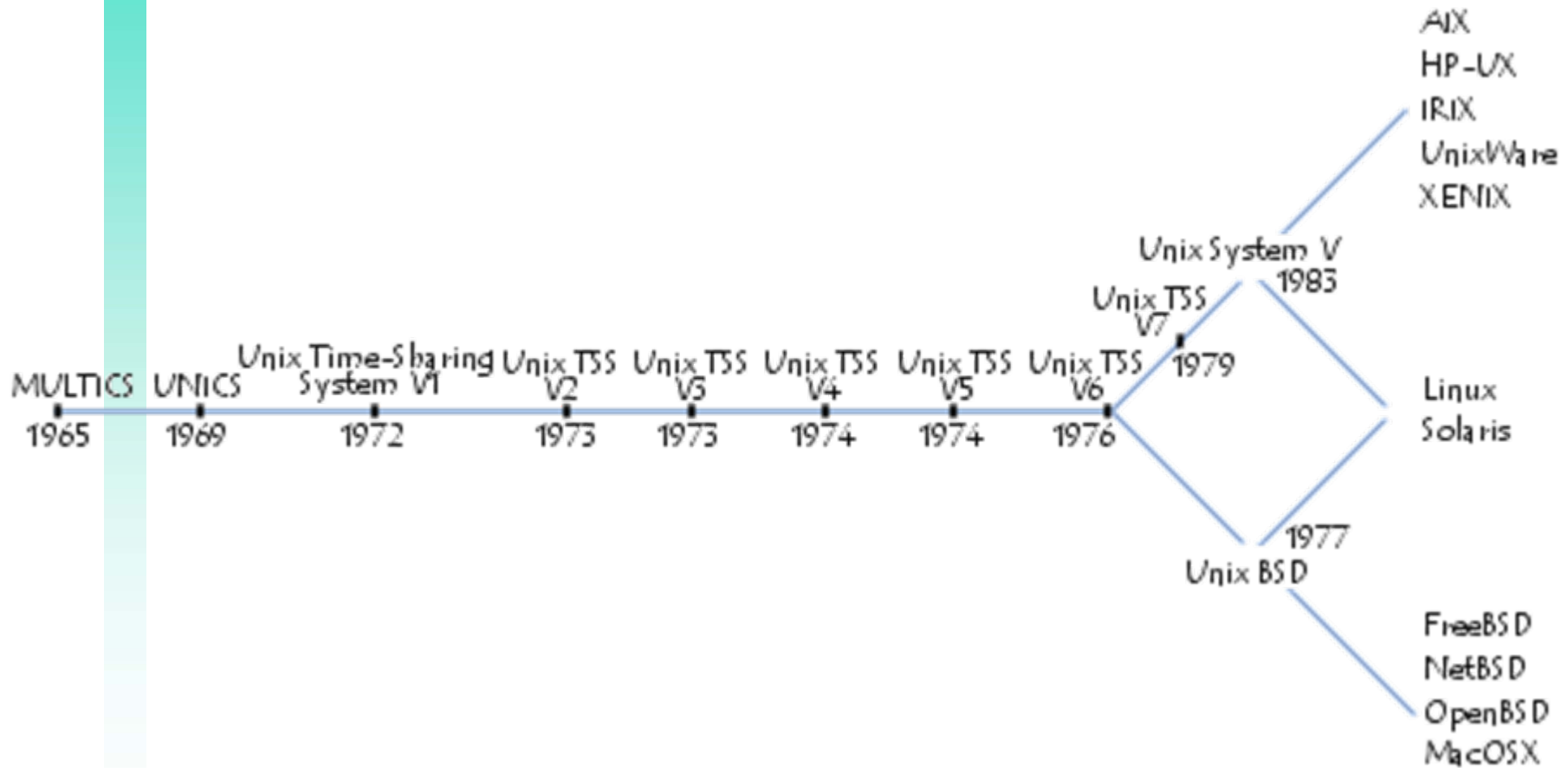
1969 UNIX/PDP-7 Ecrit en assembleur sur un PDP-7 pour développer un traitement de texte aux Bell Labs

1973 UNIX V5 Langage C (90%)

1976 UNIX V 6 (coopération de Bell Labs/universités américaines)

	Berkeley	Bell Labs	AT&T
1977	1.0 BSD	-	-
1978	2.0 BSD	V7	-
1979	3.0 BSD	-	-
1980	4.0 BSD	-	-
1981	4.1 BSD	-	-
1982	-	-	System III
1984	4.2 BSD	V8	System VR2
1986	4.3 BSD	-	-
1989	-	V 10	-
1993	4.4 BSD	-	System VR4
1991		Linux	

Historique (2)



Historique (3)

POSIX (Portable Open System Interface eXchange)

1993 - Compromis entre BSD et Systeme V

1003.0 Guide et présentation

1003.1 Appels Systèmes

1003.2 Shell et Utilitaires

1003.3 Méthodes de tests et conformité

1003.4 Extensions temps réel

1003.6 Sécurité

1003.7 Administration système

1003.8 Accès commun aux fichiers

1003.9 Appels Fortran-77

1003.10 Interface pour les supercalculateurs

1003.11 Extensions pour le transactionnel

1003.12 Extensions communications inter processus

...

Types d'API

Contrôler l'exécution d'un processus,
Faire des accès et de la gestion sur le système de fichiers,
Faire et contrôler des accès réseau,
Créer et gérer de l'espace mémoire,
Envoyer et recevoir des messages entre les processus,
Gérer ou s'informer sur l'état du système,
Contrôler des permissions d'accès,
Allouer et gérer des ressources diverses

Limites d'Unix

Complexité de certaines versions => problèmes de robustesse

Interface utilisateur

Prolifération des versions => situation chaotique

Approche monolithique => difficilement extensible