

SYSTÈMES D'EXPLOITATION

SIF-1015

Contenu du cours 1

- Informations utiles
 - Source des notes de cours, capsules vidéos et exemples
 - <ftp://dmiftp.uqtr.ca/FMeunier/sif1015/>
 - Local: 3081R
 - Email: Francois.Meunier@uqtr.ca
 - No. de téléphone: 819 376 5011 ext. 3833
 - Laboratoires: 3038R, 3043R, 3045R (accès: 1-5-32-4)

Contenu du cours 1

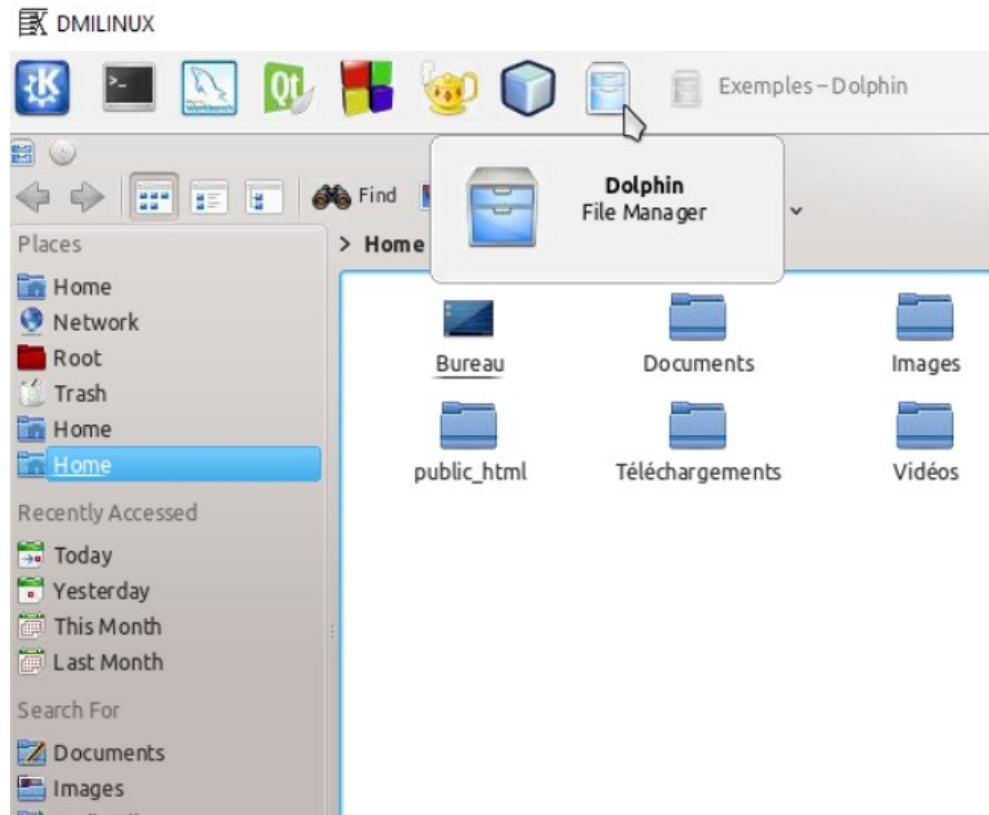
- Introduction aux systèmes d'exploitation (SE)
 - Organisation d'un système informatique
 - Fonctionnalités de LINUX
 - Rôles des SE
 - Structures générales
 - Développement sous LINUX/UNIX
- Environnement de travail LINUX et langage C
 - Langage C sur LINUX (exemples de programmes)
 - Utilitaire ***make***
- LECTURES: Chapitres 1, 8 et 9 (Prog. LINUX)
 - Chapitre 1 (OSC, Prog. système en C)
- Notes de cours (sur **dmiftp**, répertoire Documentation)

Contenu du cours 1

- Exemples en langage C
 - Sur dmiftp.uqtr.ca (site des notes de cours, répertoire **Exemples**)
 - Accès au répertoire **Exemples_Meunier** sur DMILinux

Contenu du cours 1

- Exemples en langage C
 - Accès au répertoire **Exemples_Meunier** sur DMILinux
 - Pour accéder aux exemples, d'abord ouvrir le répertoire **/Home/Exemples_Meunier**
 - Ouvrir ensuite le gestionnaire de fichiers **Dolphin**



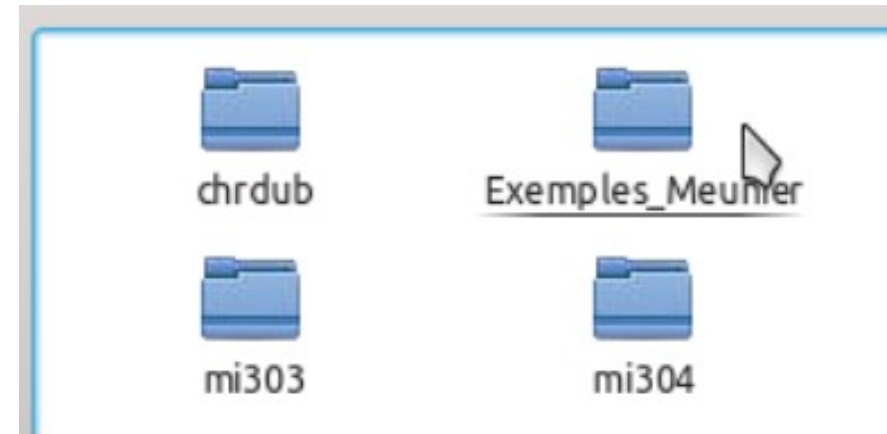
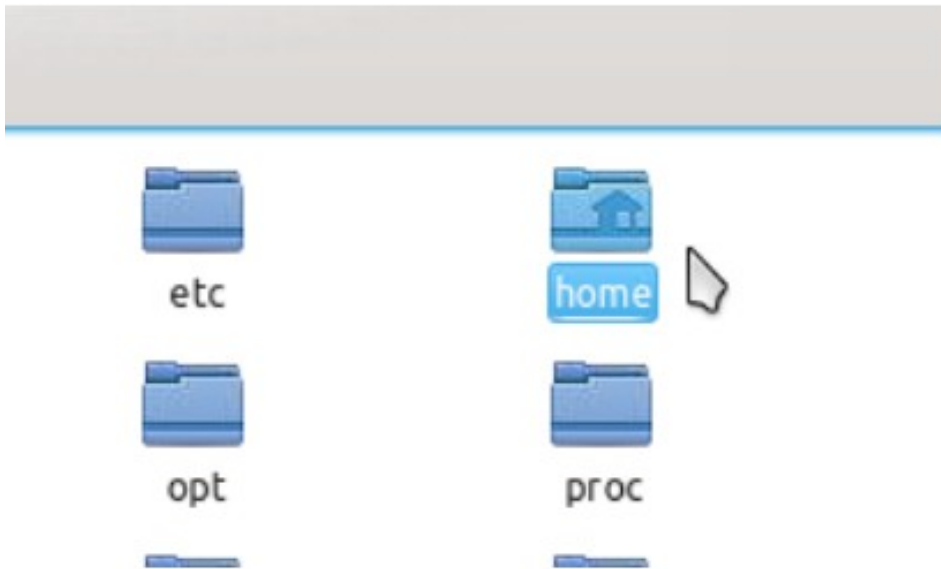
Contenu du cours 1

- Exemples en langage C
 - Accès au répertoire **Exemples_Meunier** sur DMILinux
 - Pour accéder aux exemples, d'abord ouvrir le répertoire **/Home/Exemples_Meunier** (suite)
 - Sélectionner la source du répertoire **Exemples_Meunier**, sélectionner **Hard Drive**



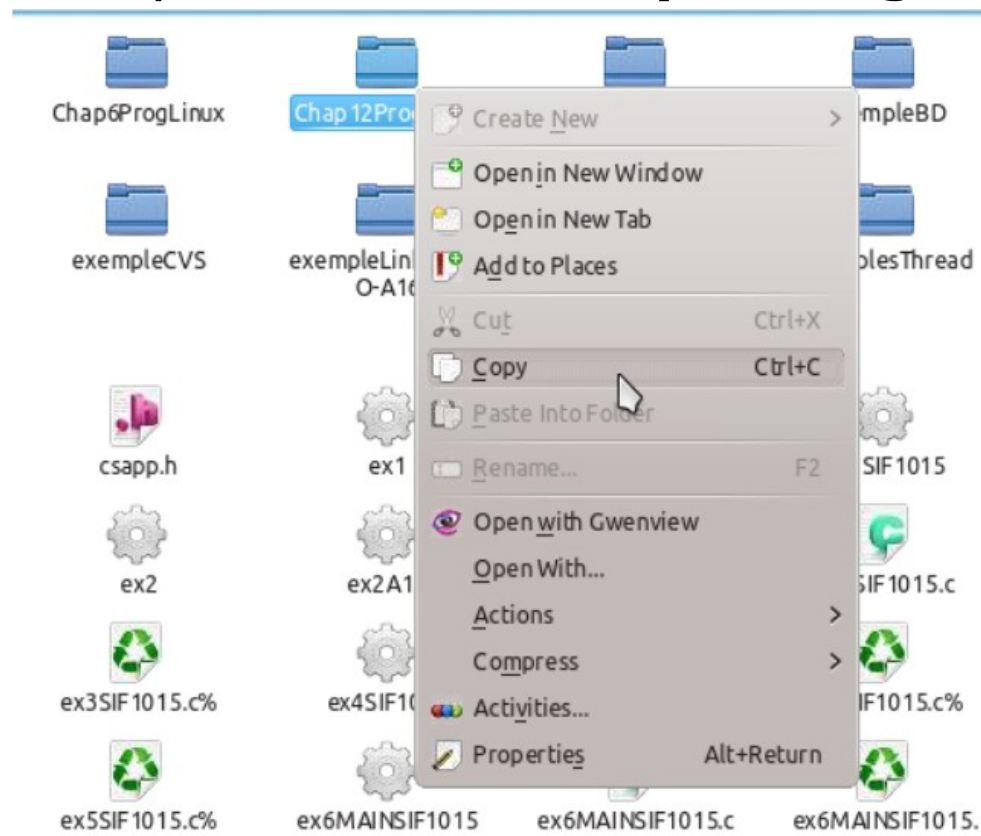
Contenu du cours 1

- Exemples en langage C
 - Accès au répertoire **Exemples_Meunier** sur DMILinux
 - Pour accéder aux exemples, d'abord ouvrir le répertoire **/Home/Exemples_Meunier** (suite)
 - Sélectionner le répertoire **/Home** et ensuite **/Exemples_Meunier**



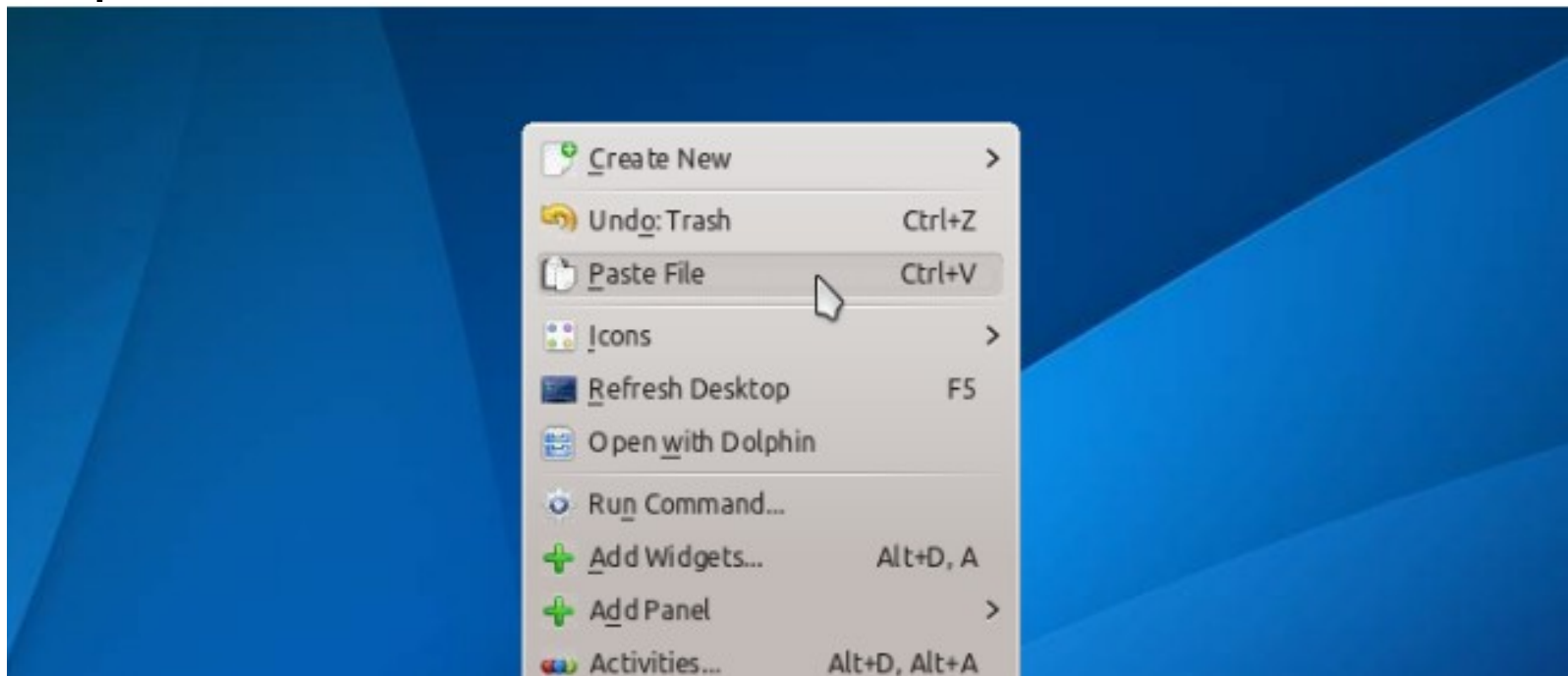
Contenu du cours 1

- Exemples en langage C
 - Accès au répertoire **Exemples_Meunier** sur DMILinux
 - Pour accéder aux exemples, d'abord ouvrir le répertoire **/Home/Exemples_Meunier** (suite)
 - Copier un répertoire (ex: **Chap12ProgLinux**)



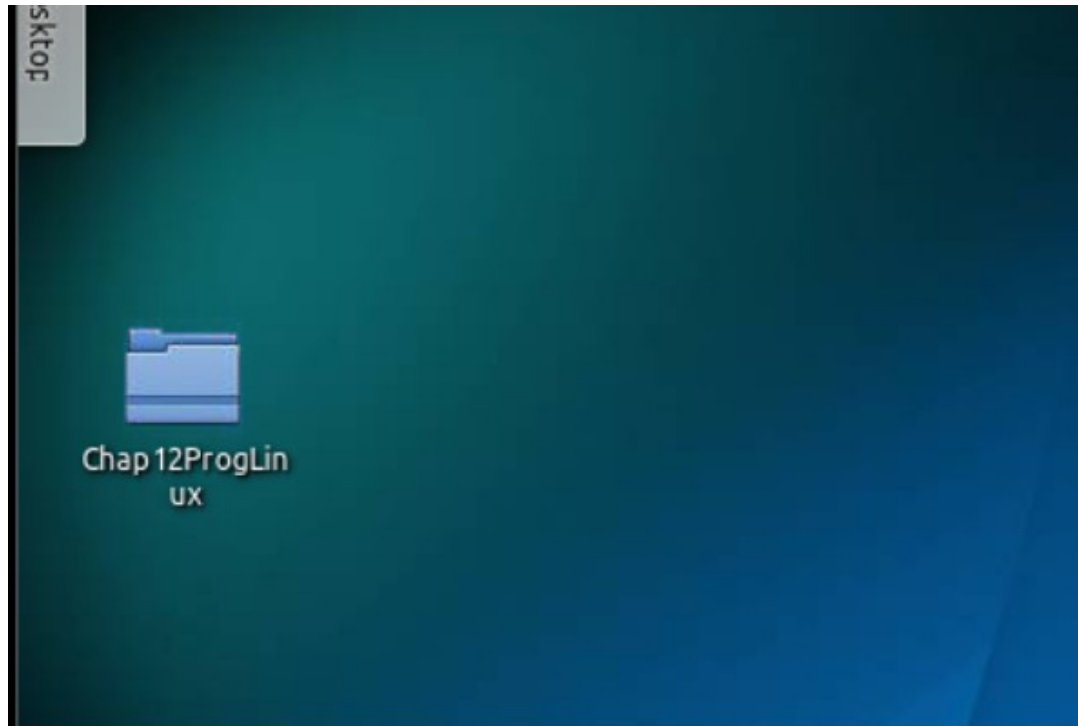
Contenu du cours 1

- Exemples en langage C
 - Accès au répertoire **Exemples_Meunier** sur DMILinux
 - Pour accéder aux exemples, d'abord ouvrir le répertoire **/Home/Exemples_Meunier** (suite)
 - Coller ce répertoire (ex: **Chap12ProgLinux**) sur votre



Contenu du cours 1

- Exemples en langage C
 - Accès au répertoire **Exemples_Meunier** sur DMILinux
 - Pour accéder aux exemples, d'abord ouvrir le répertoire **/Home/Exemples_Meunier** (suite)
 - Voir le répertoire **Chap12ProgLinux** sur votre bureau



Contenu du cours 1

- Voir guide de l'utilisateur des principales commandes UNIX ***quick_guide.txt*** sur le site ftp UQTR des notes de cours (répertoire **Documentation**)

Contenu du cours 1

- Petit survol rapide de LINUX avec des commandes de l'environnement Linux
 - <http://linuxreviews.org/beginner>
- Autre petit survol de linux
 - http://www.linuxhotbox.com/adminguide/lnag_basics.html
- Introduction à la programmation en langage C
 - http://docs.linux.cz/programming/c/c_marshall/CE.html
- Révision des structures de données importantes (ex: **liste simplement chaînée**) en langage C
 - dmiftp.uqtr.ca/FMeunier/sif1015/Documentation/Structures de données Élémentaires TP1-A19.ppt

Contenu du cours 1

- Livre de référence en français (traduction du livre de Mitchell)
 - <http://mtodorovic.developpez.com/linux/programmation-avancee/>
 - <http://gl.developpez.com/tutoriel/outil/makefile/>
- Livre de référence en anglais (Mitchell)
 - <http://www.advancedlinuxprogramming.com/alp-folder>
- Livre de référence en anglais (**Linux System Programming: Love**)
- Livre de référence en anglais (**Beginning Linux Programming: Matthew & Stones**)
- Programmation C/C++ d'interfaces avec Qt
 - <http://www.bogotobogo.com/cplusplus/files/c-gui-programming-with-qt-4-2ndedition.pdf>

Contenu du cours 1

- **Installation d'un client X2go pour pouvoir travailler sur DMILinux dans le même environnement *kde***
 - Télécharger le client X2go en suivant les instructions contenues dans le fichier *.pdf* situé sur le site ftp des notes de cours (UQTR):
dmiftp.uqtr.ca/FMeunier/sif1015/Documentation/DMILINUX-20160912.pdf
 - Vous pouvez consulter ce document pour obtenir d'autres informations sur d'autres services offerts sur le serveur DMILINUX

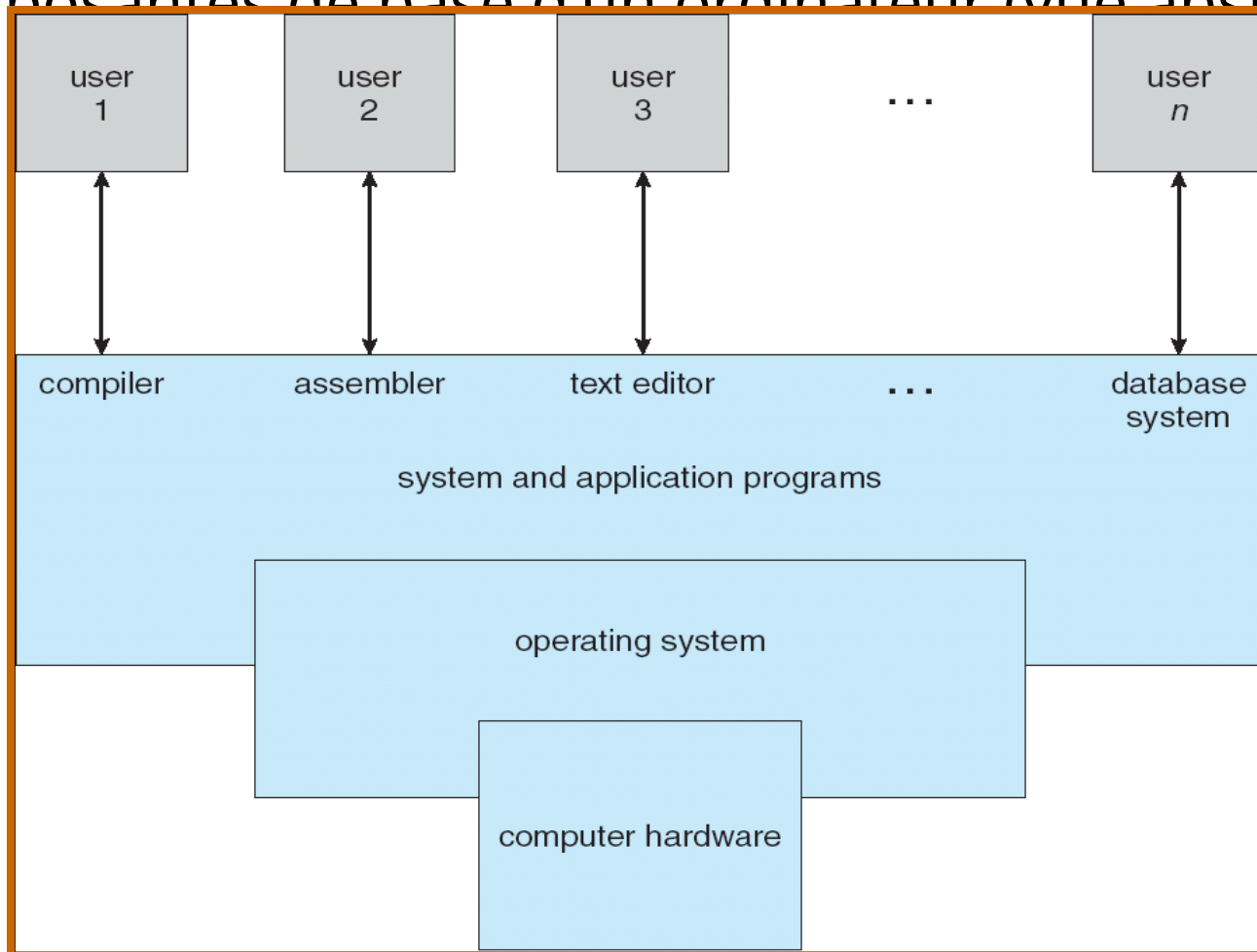
Contenu du cours 1

- **LINUX MINT (17)** est un distribution de Ubuntu installé dans les laboratoires (3038R, 3043R et 3045R) du D



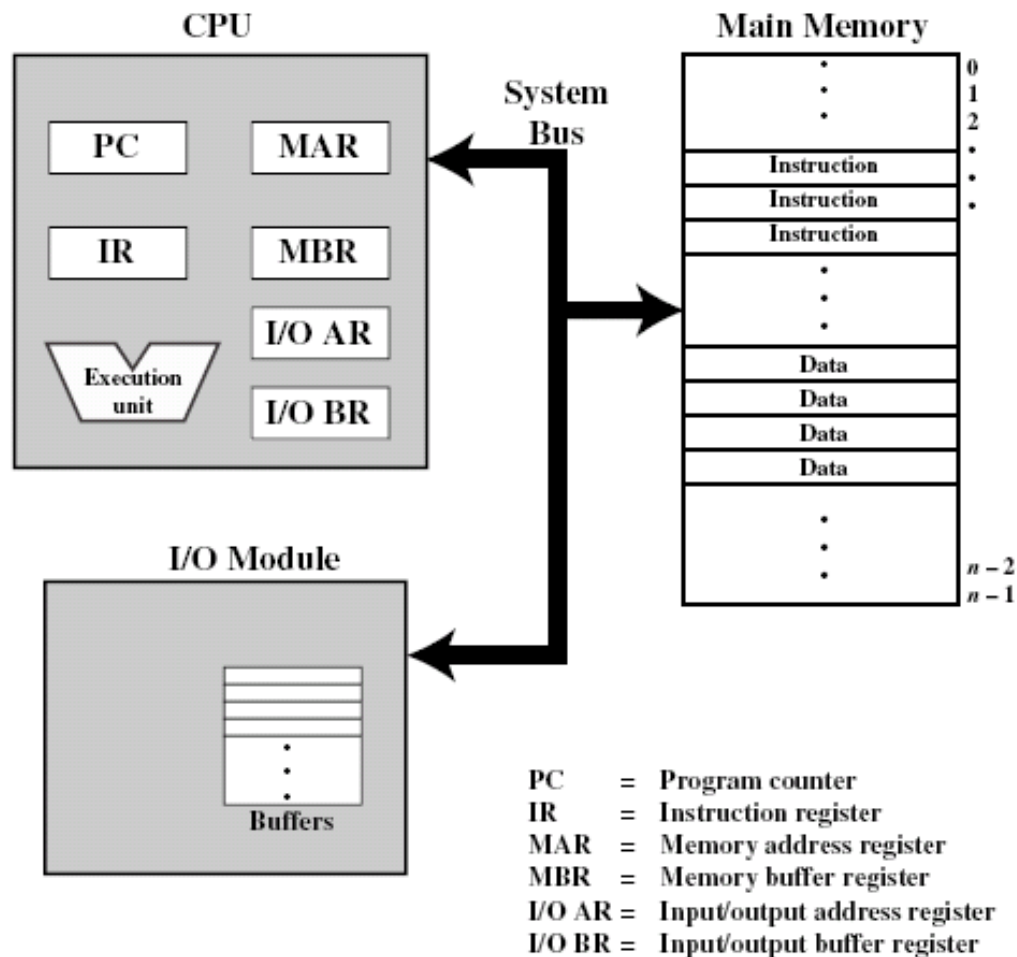
Introduction aux systèmes d'exploitation (SE)

- Organisation d'un système informatique (Aspect matériel)
 - Composantes de base d'un ordinateur (vue abstraite)



Introduction aux systèmes d'exploitation (SE)

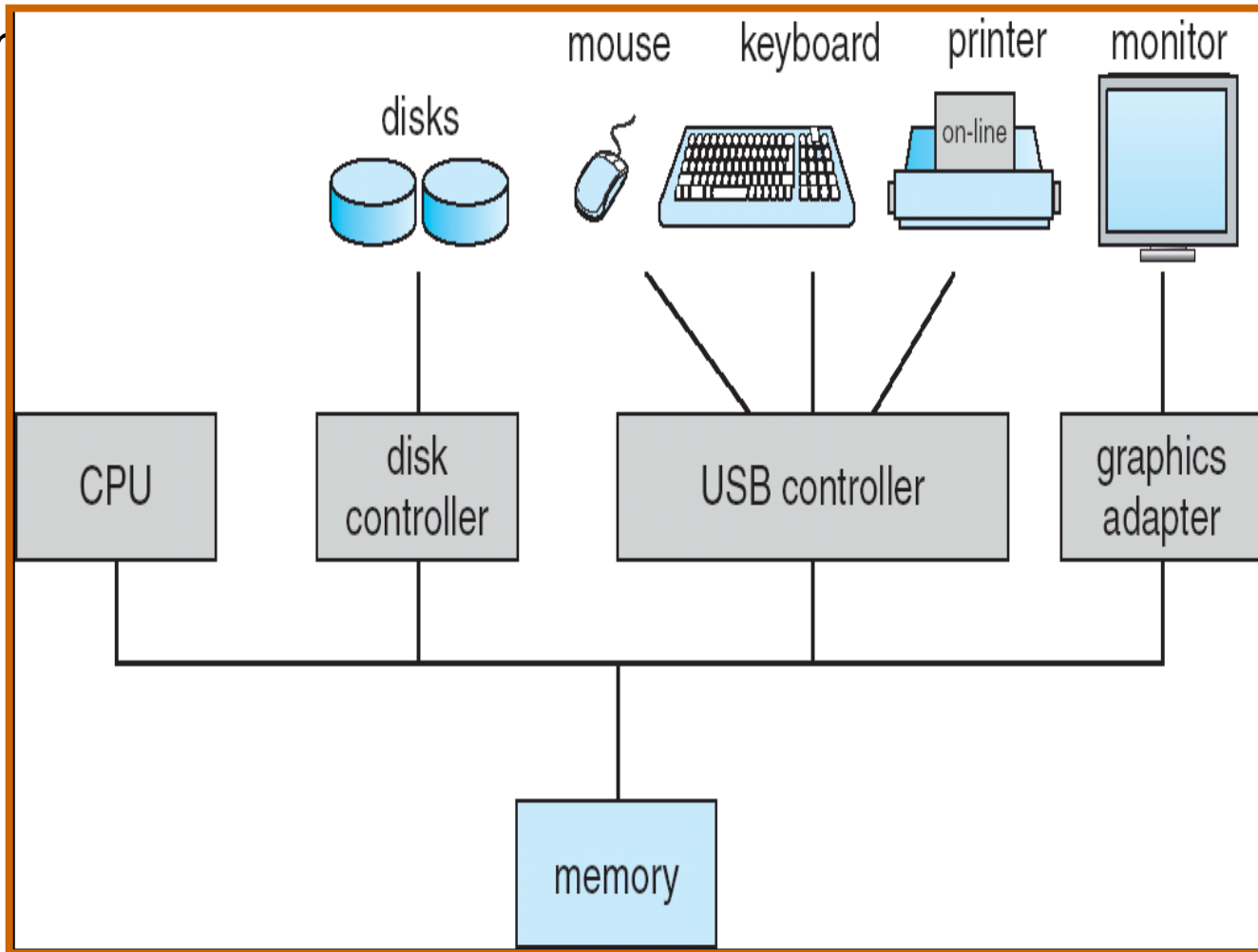
- Organisation d'un système informatique (Aspect matériel)
 - Composants matériels



Introduction aux systèmes d'exploitation (SE)

- Organisation d'un système informatique (Aspect matériel)

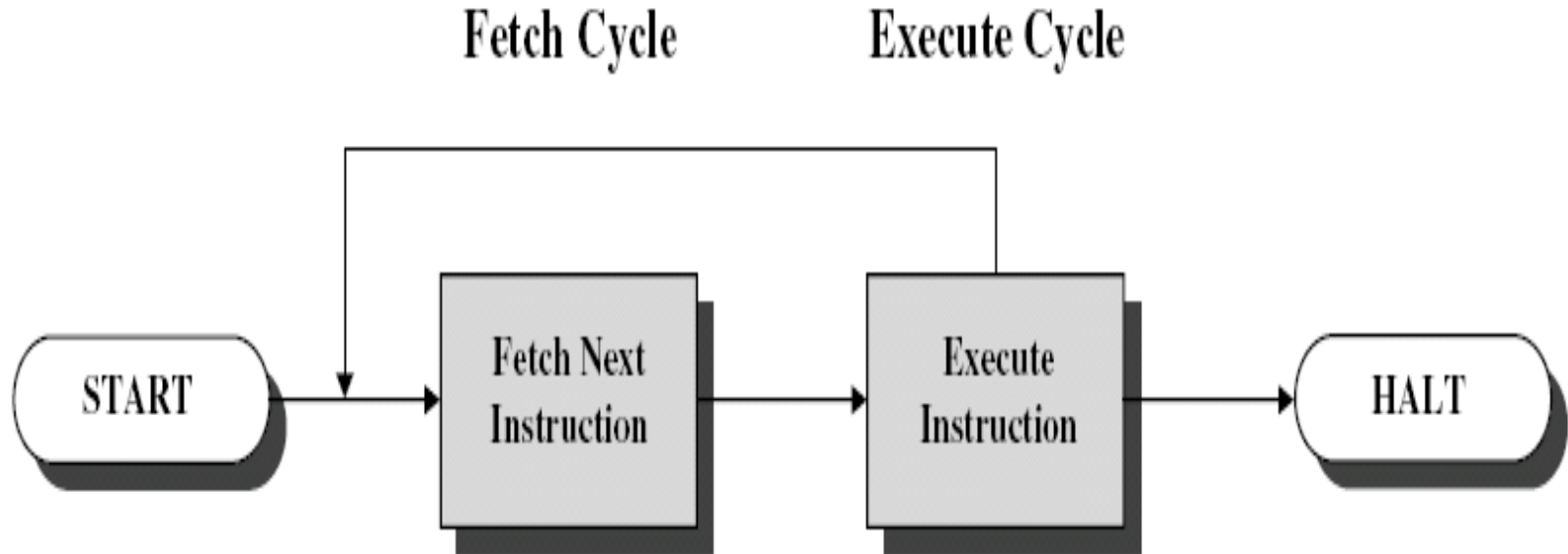
– Cor



érielle)

Introduction aux systèmes d'exploitation (SE)

- Organisation d'un système informatique (Aspect matériel)
 - Cycle d'exécution d'un instruction



Introduction aux systèmes d'exploitation (SE)

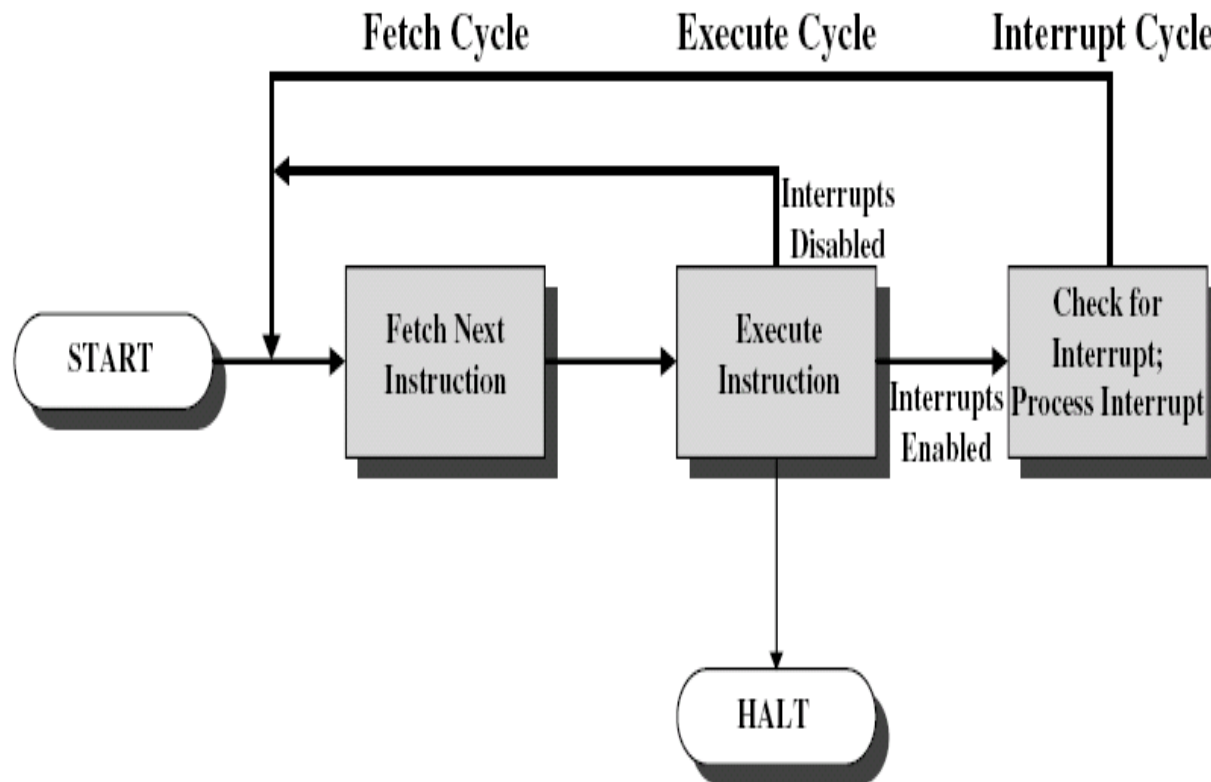
- Organisation d'un système informatique (Aspect matériel)
 - Génération d'interruptions

Program	Generated by some condition that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, and reference outside a user's allowed memory space.
Timer	Generated by a timer within the processor. This allows the operating system to perform certain functions on a regular basis.
I/O	Generated by an I/O controller, to signal normal completion of an operation or to signal a variety of error conditions.
Hardware failure	Generated by a failure, such as power failure or memory parity error.

Types d'interruption

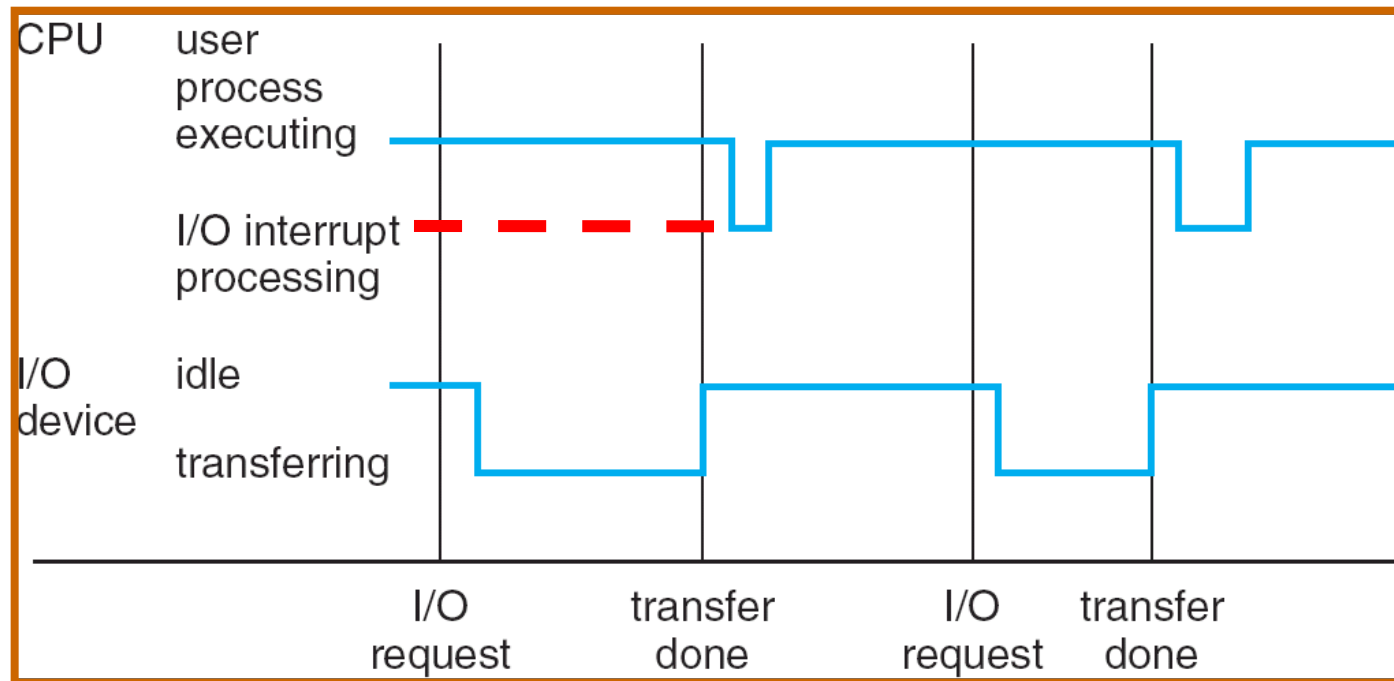
Introduction aux systèmes d'exploitation (SE)

- Organisation d'un système informatique (Aspect matériel)
 - Gestion des interruptions (Cycle d'exécution d'une instruction)



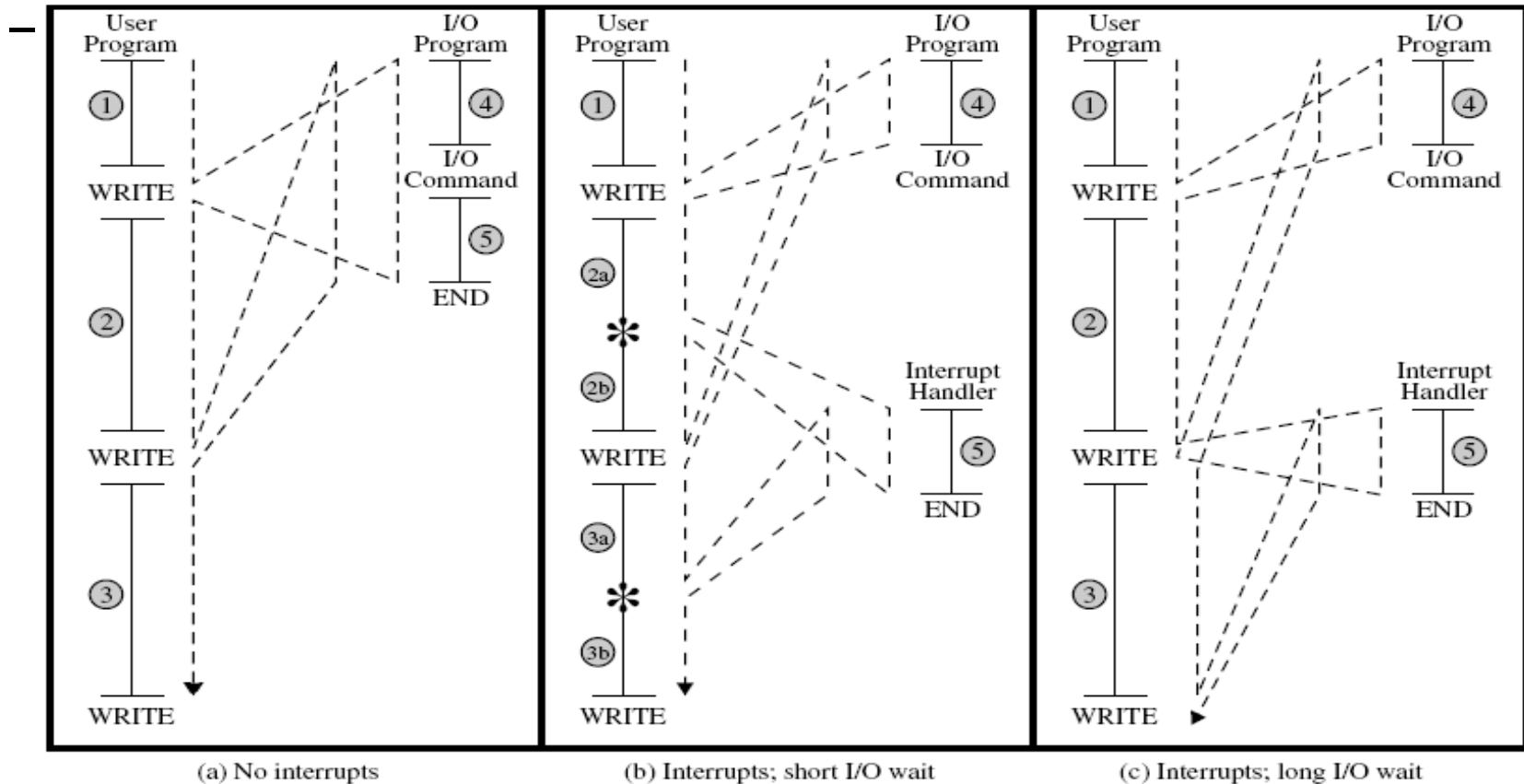
Introduction aux systèmes d'exploitation (SE)

- Organisation d'un système informatique (Aspect matériel)
 - Gestion des interruptions (Exemple: I/O)



Introduction aux systèmes d'exploitation (SE)

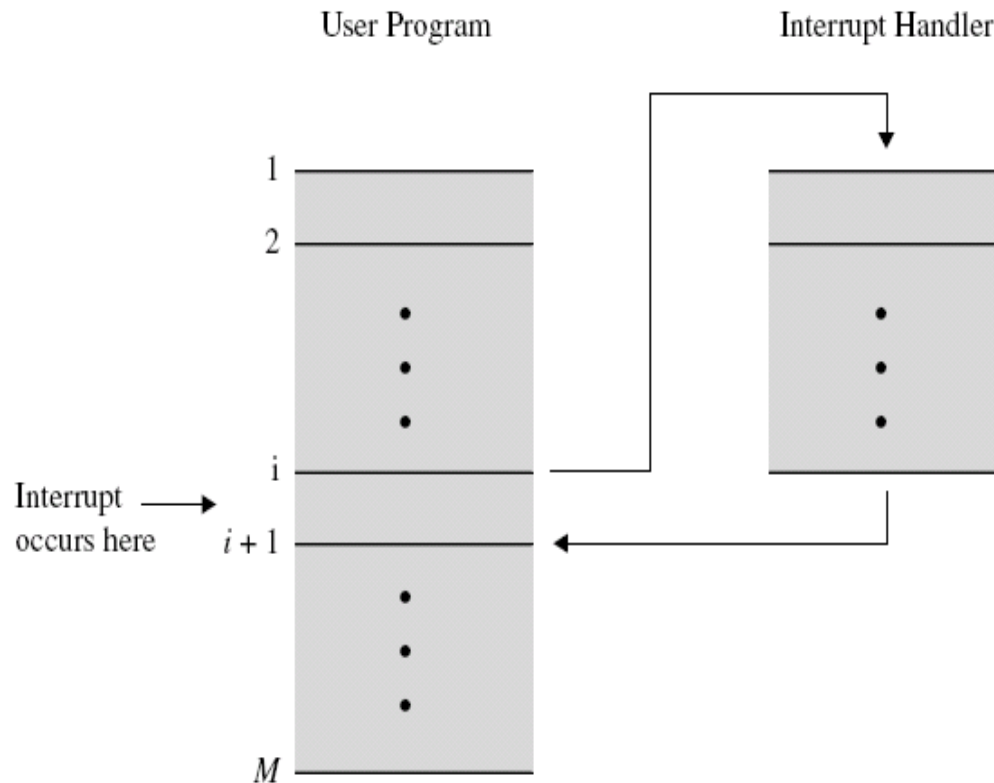
- Organisation d'un système informatique (Aspect matériel)



Exécution d'un programme avec ou sans interruption

Introduction aux systèmes d'exploitation (SE)

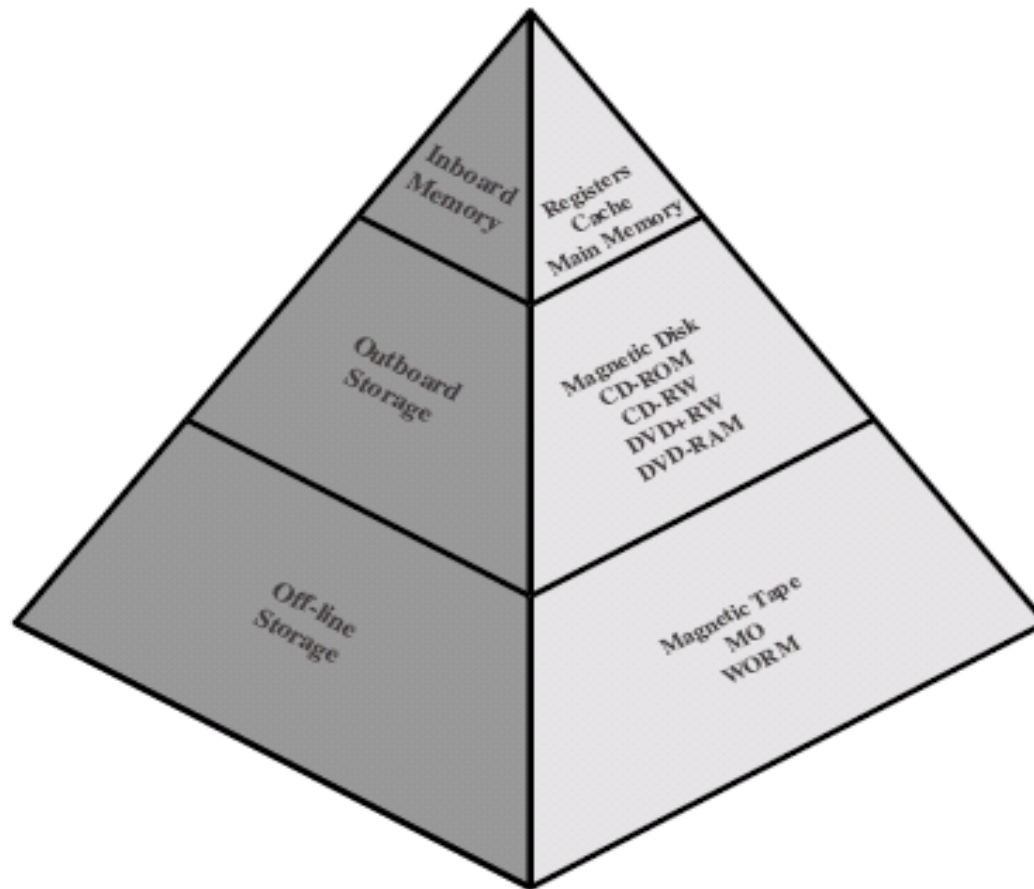
- Organisation d'un système informatique (Aspect matériel)
 - Gestic



Séquence de changements de contrôle d'un programme lors d'interruptions

Introduction aux systèmes d'exploitation (SE)

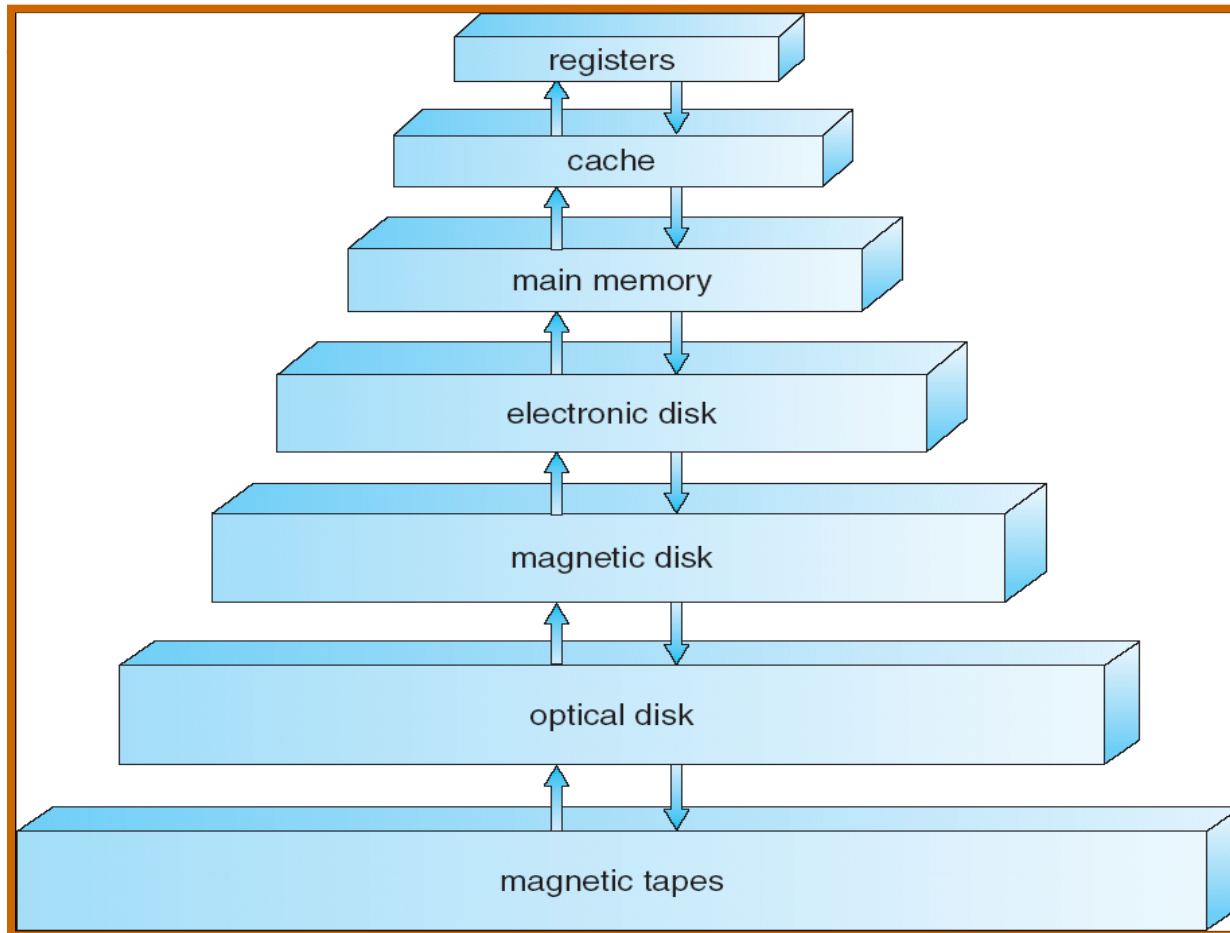
- Organisation d'un système informatique (Aspect matériel)
 - Gestion



Hiérarchie de la mémoire

Introduction aux systèmes d'exploitation (SE)

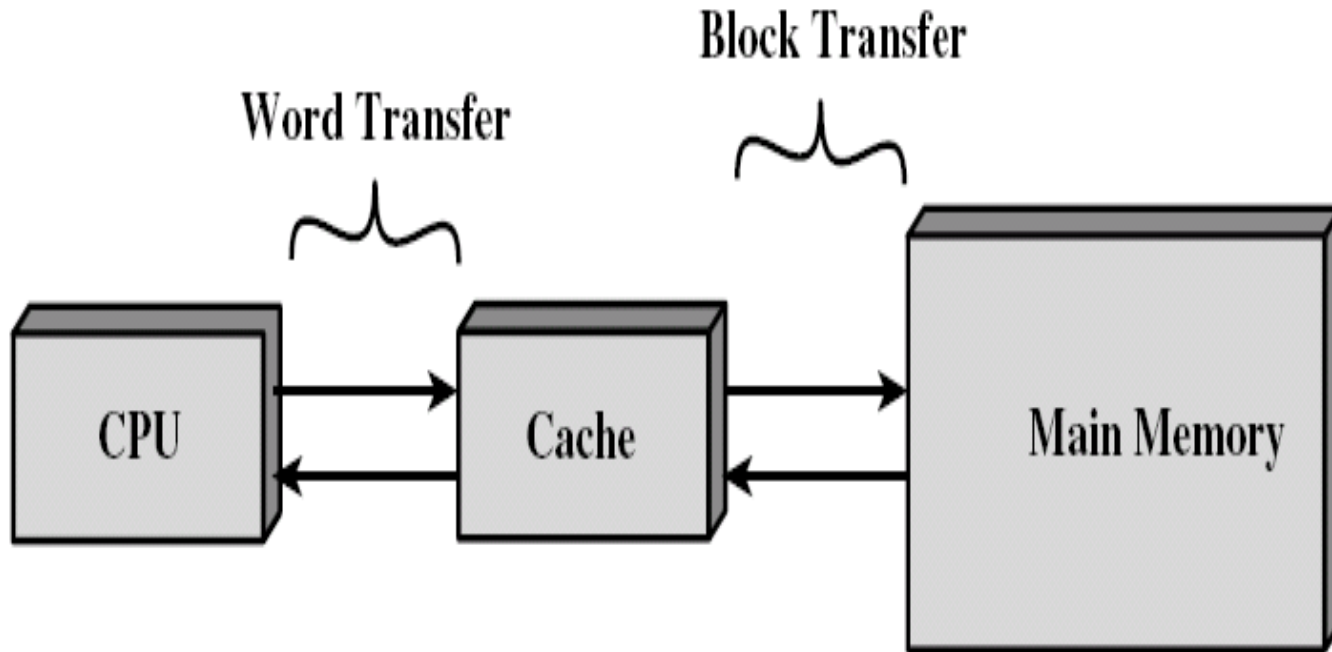
- Organisation d'un système informatique (Aspect matériel)
 - Gesti



Hiérarchie de la mémoire

Introduction aux systèmes d'exploitation (SE)

- Organisation d'un système informatique (Aspect matériel)
 - Gestion de la mémoire

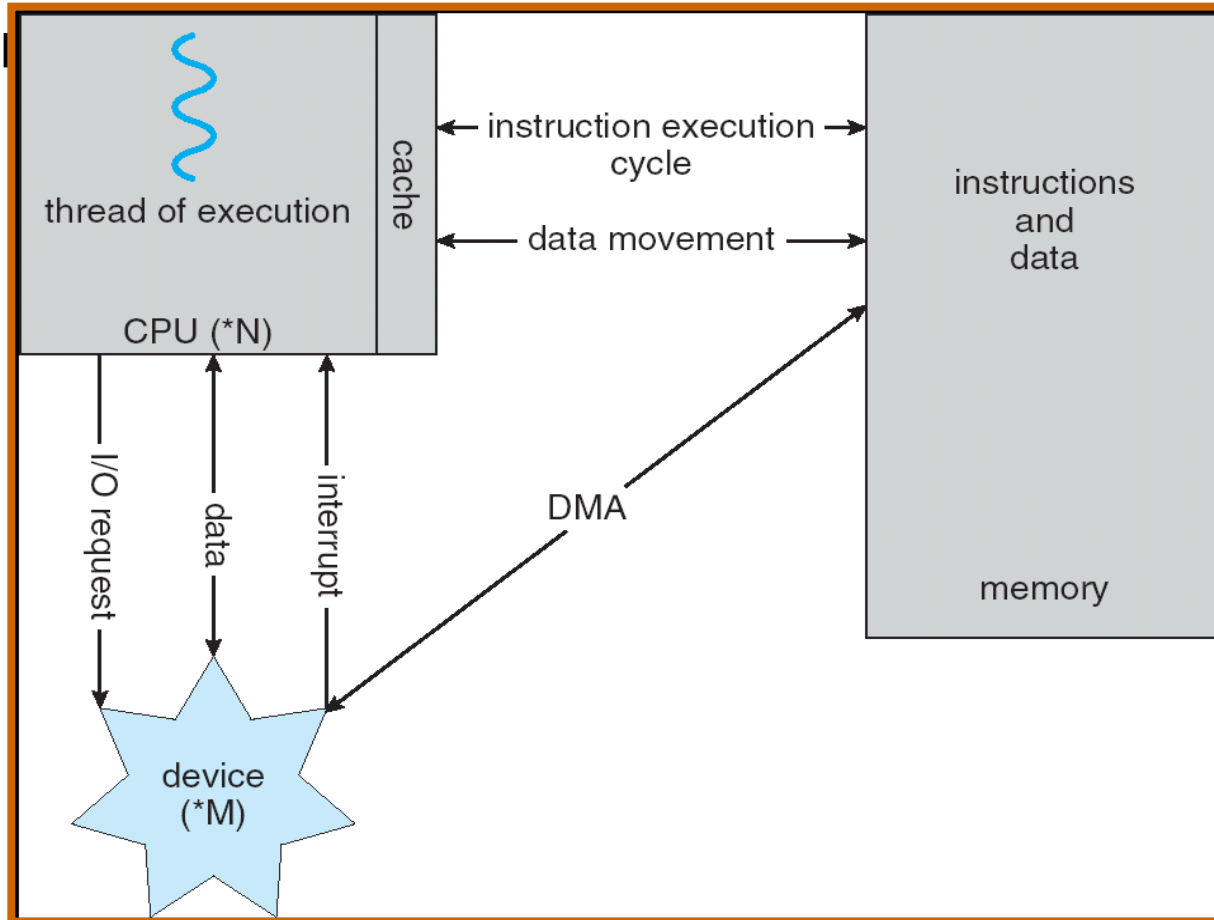


Mémoire cache et RAM

Introduction aux systèmes d'exploitation (SE)

- Organisation d'un système informatique (Aspect matériel)

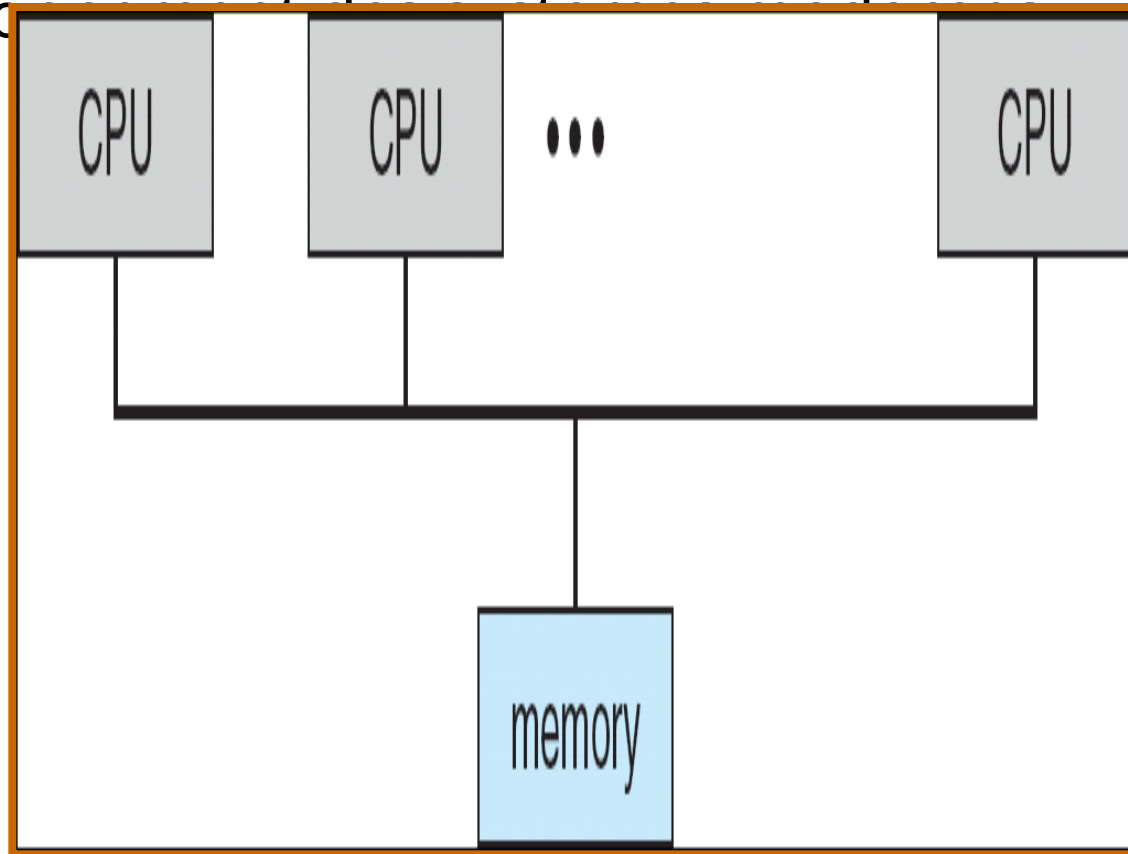
– For



Organisation Générale

Introduction aux systèmes d'exploitation (SE)

- Organisation d'un système informatique (Aspect matériel)
 - Fonctionnement d'un système à programmation symétrique



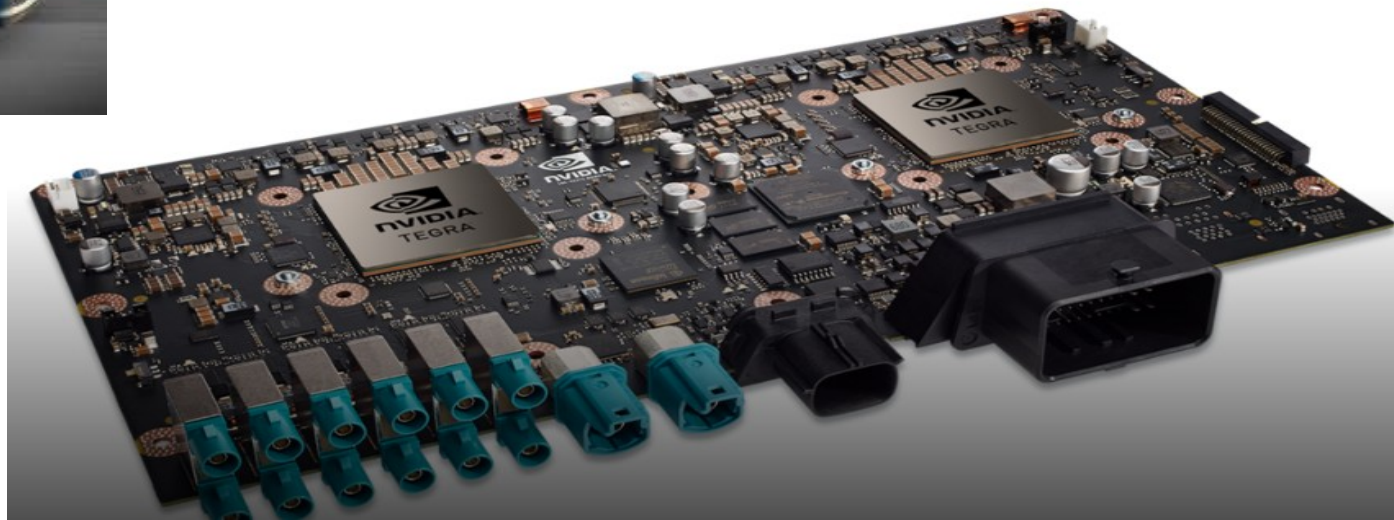
Architecture Symmetric Multi Programming

Introduction aux systèmes d'exploitation (SE)

- Organisation d'un système informatique (Aspect matériel)
- **Fonctionnement des systèmes modernes**
 - Processeurs pour les systèmes de conduite autonomes et ADAS
 - Voir le site Nvidia: <https://blogs.nvidia.com/blog/2016/08/22/parker-for-self-driving-cars/>



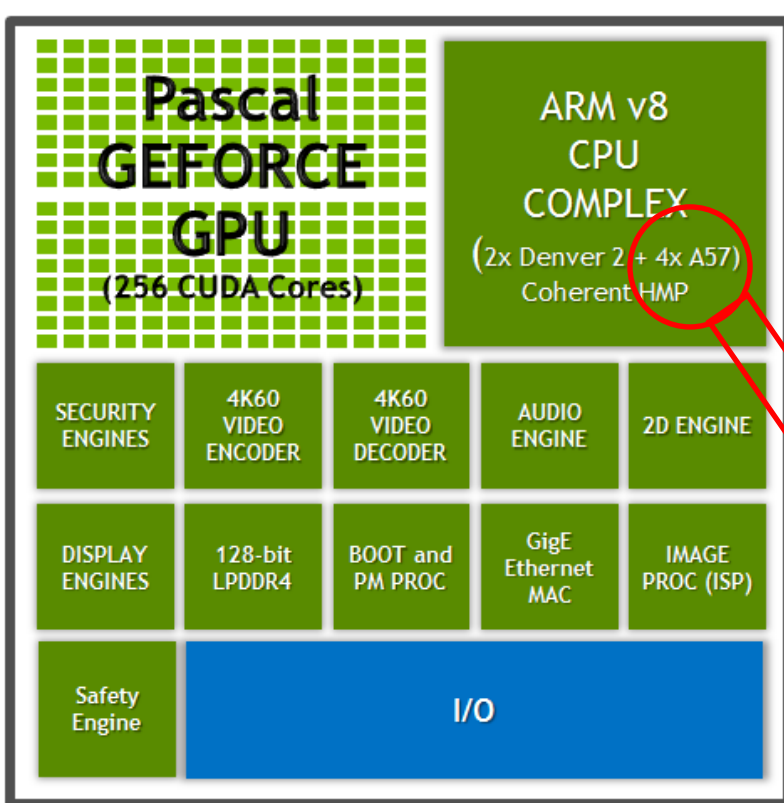
NVIDIA DRIVE PX 2 utilise un OS basé sur une distribution maison Vibrante Linux/QNX RTOS



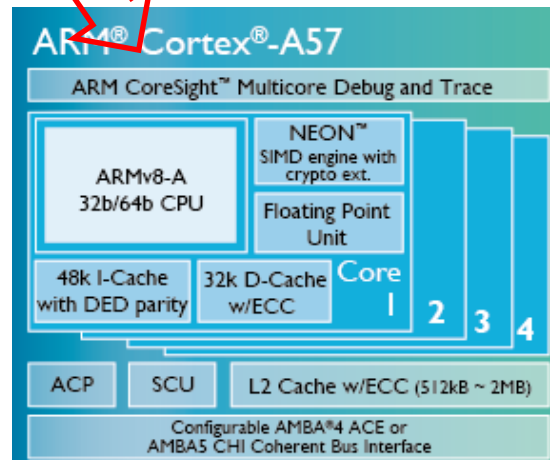
Introduction aux systèmes d'exploitation (SE)

- Organisation d'un système informatique (Aspect matériel)

- Processus de conception des systèmes d'exploitation modernes

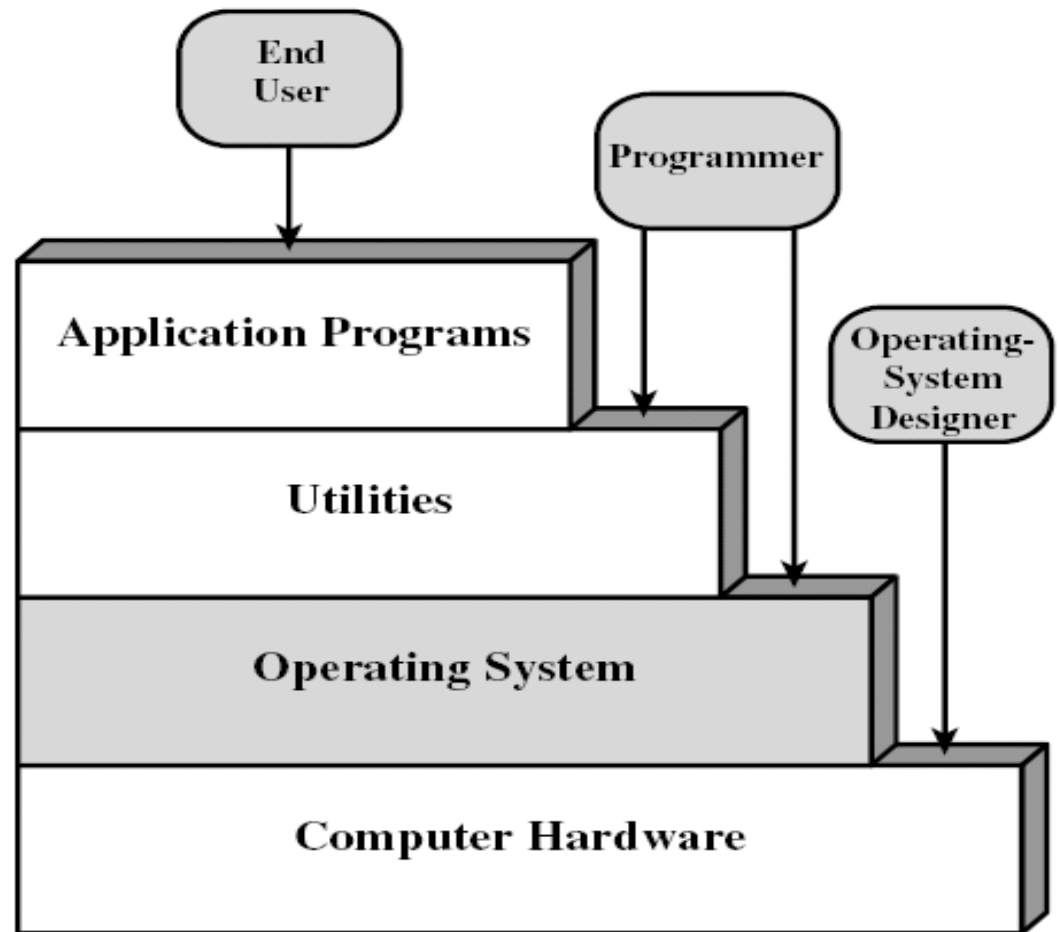
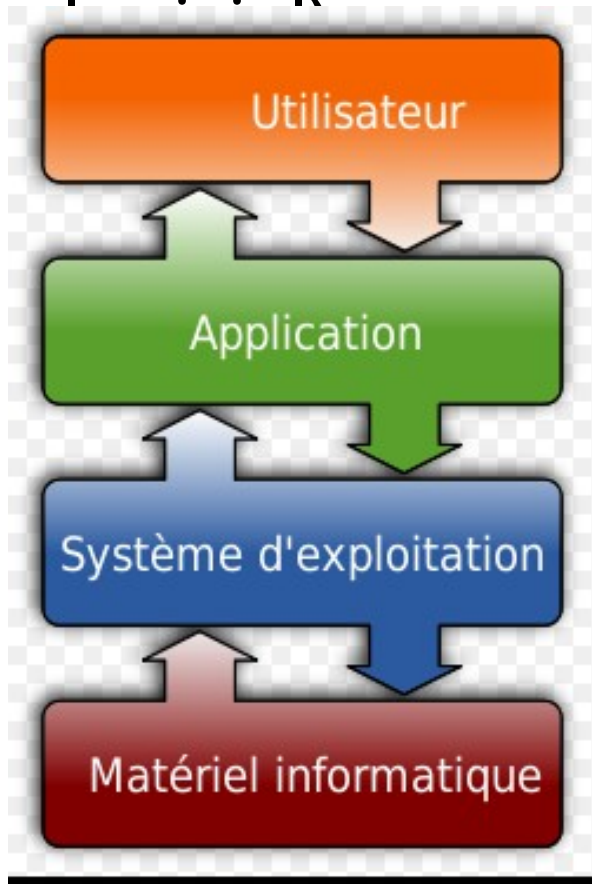


	Parker Specs
GPU	NVIDIA Pascal 256-core GPU DX12, OpenGL 4.5, NVIDIA CUDA 8.0, OpenGL ES 3.1, AEP and Vulkan
CPU	2 Denver 2.0 64b cores + 4 ARM Cortex A57 64b cores 2MB + 2MB L2 Cache, Coherent HMP architecture
Memory	128-bit LPDDR4 with ECC
Display	Triple Display Pipeline, each at up to 4K, 60fps
Video	Decode – H.265, VP9 up to 4K 60fps Encode – H.265, VP9 up to 4K 60fps
Process	16nm FinFet



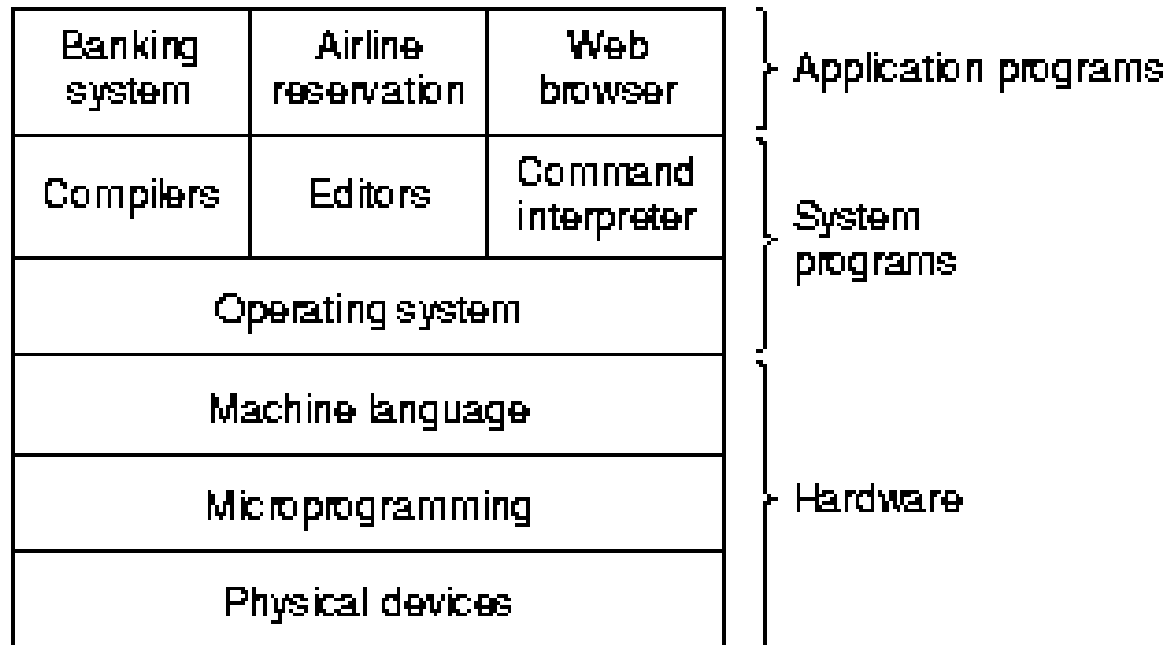
Introduction aux systèmes d'exploitation (SE)

- Organisation d'un système informatique (Aspect



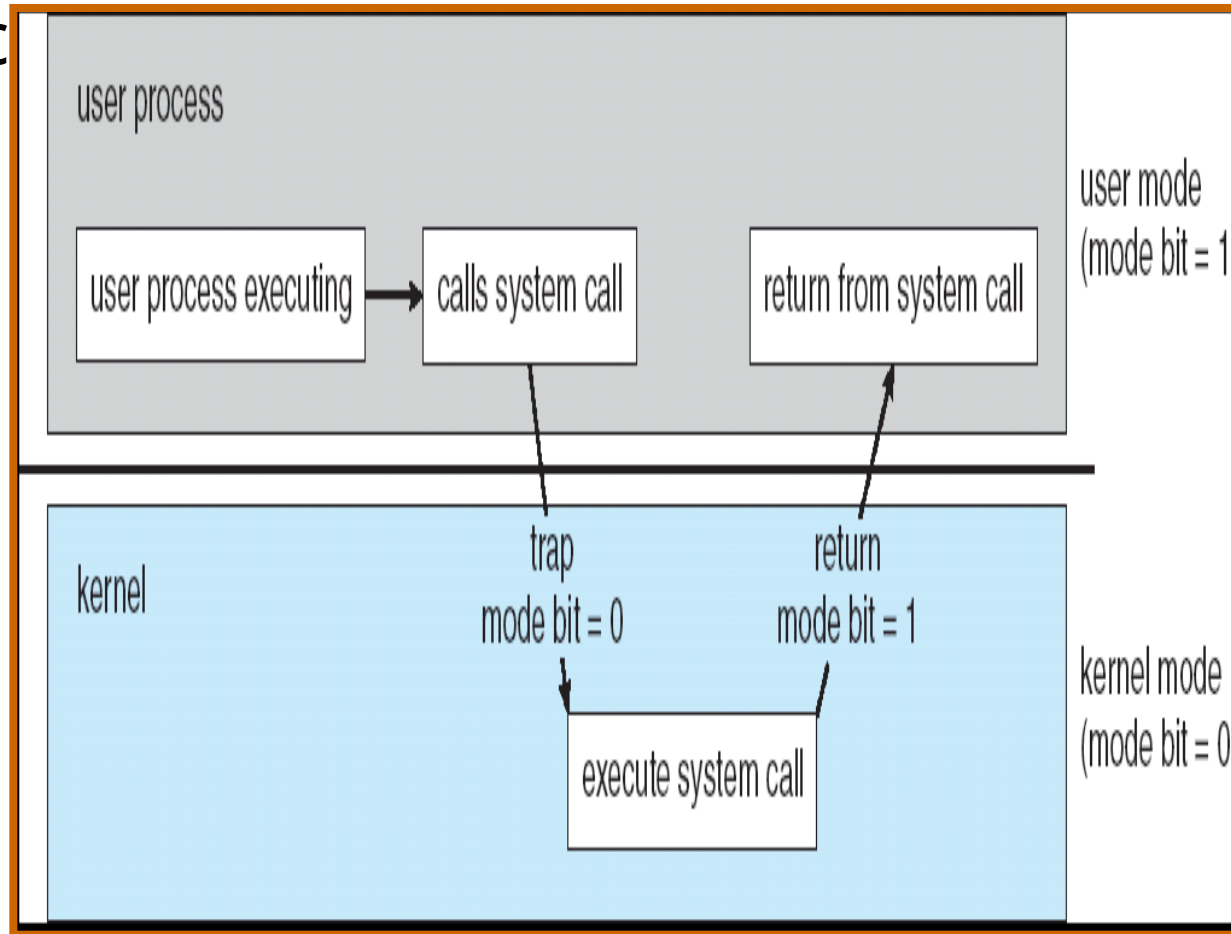
Introduction aux systèmes d'exploitation (SE)

- Organisation d'un système informatique (Aspect logiciel)



Introduction aux systèmes d'exploitation (SE)

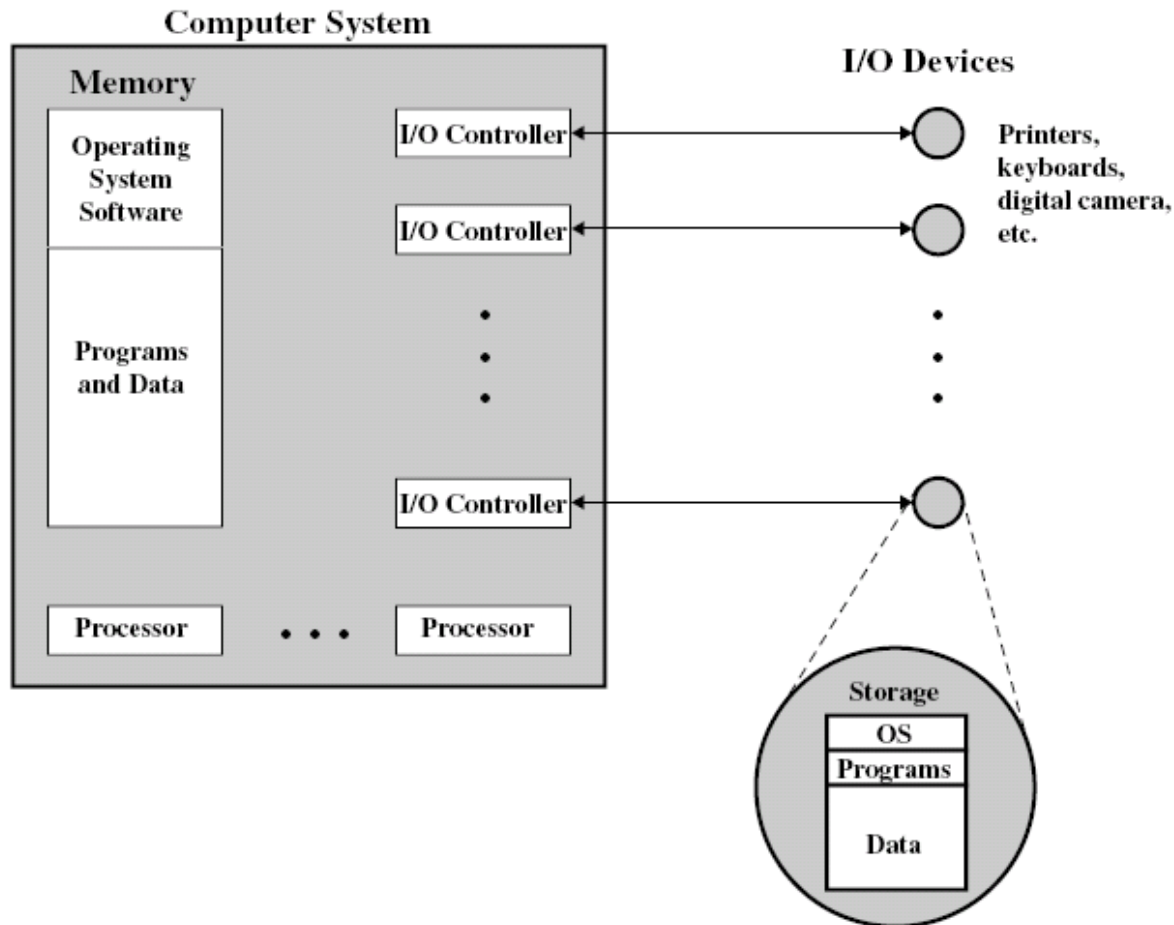
- Organisation d'un système informatique (Aspect logic)



Modes d'exécution: Utilisateur (USER), Noyau (KERNEL)

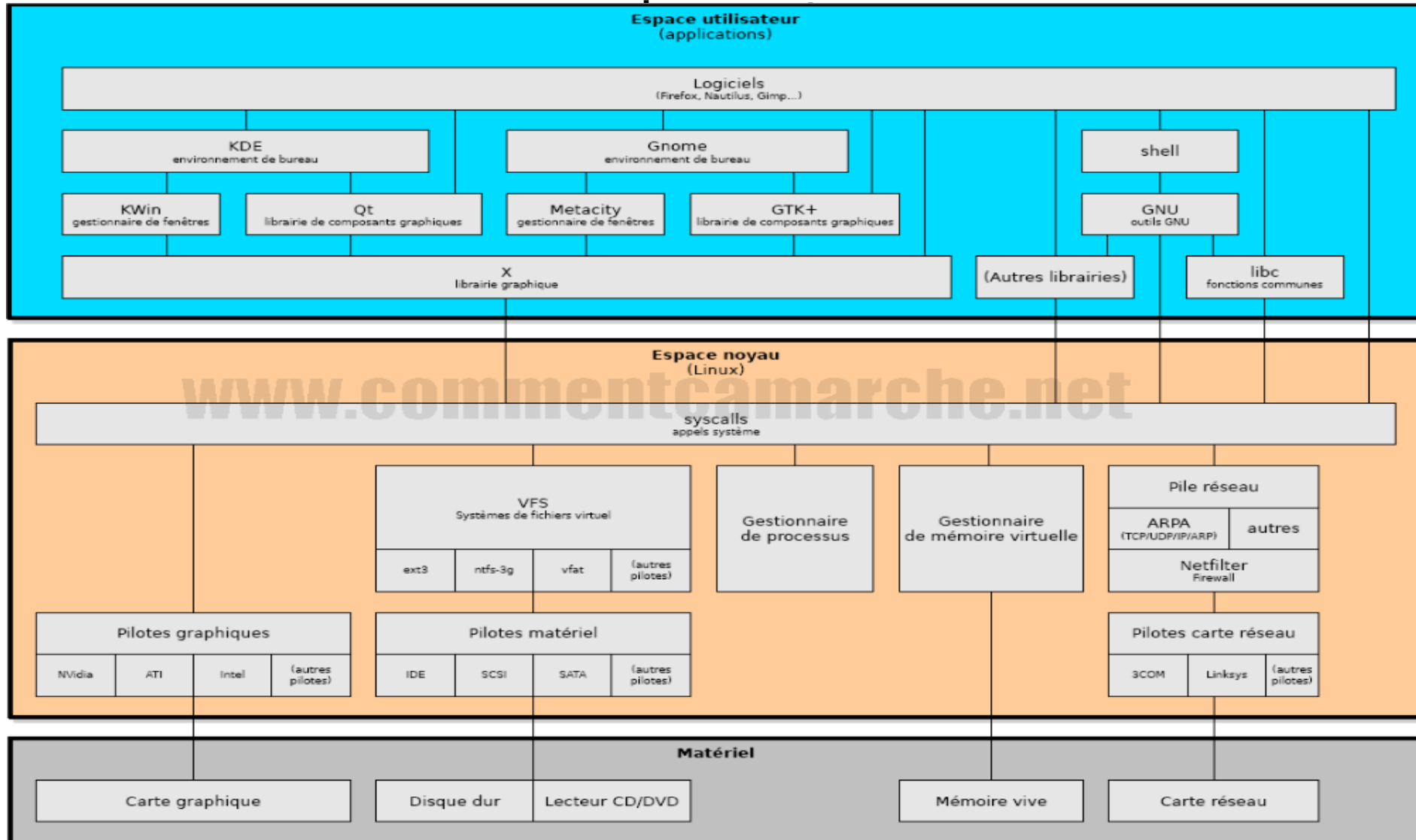
Introduction aux systèmes d'exploitation (SE)

- Fonctionnalités d'un système d'exploitation
 - Gestionnaire de ressources matérielles



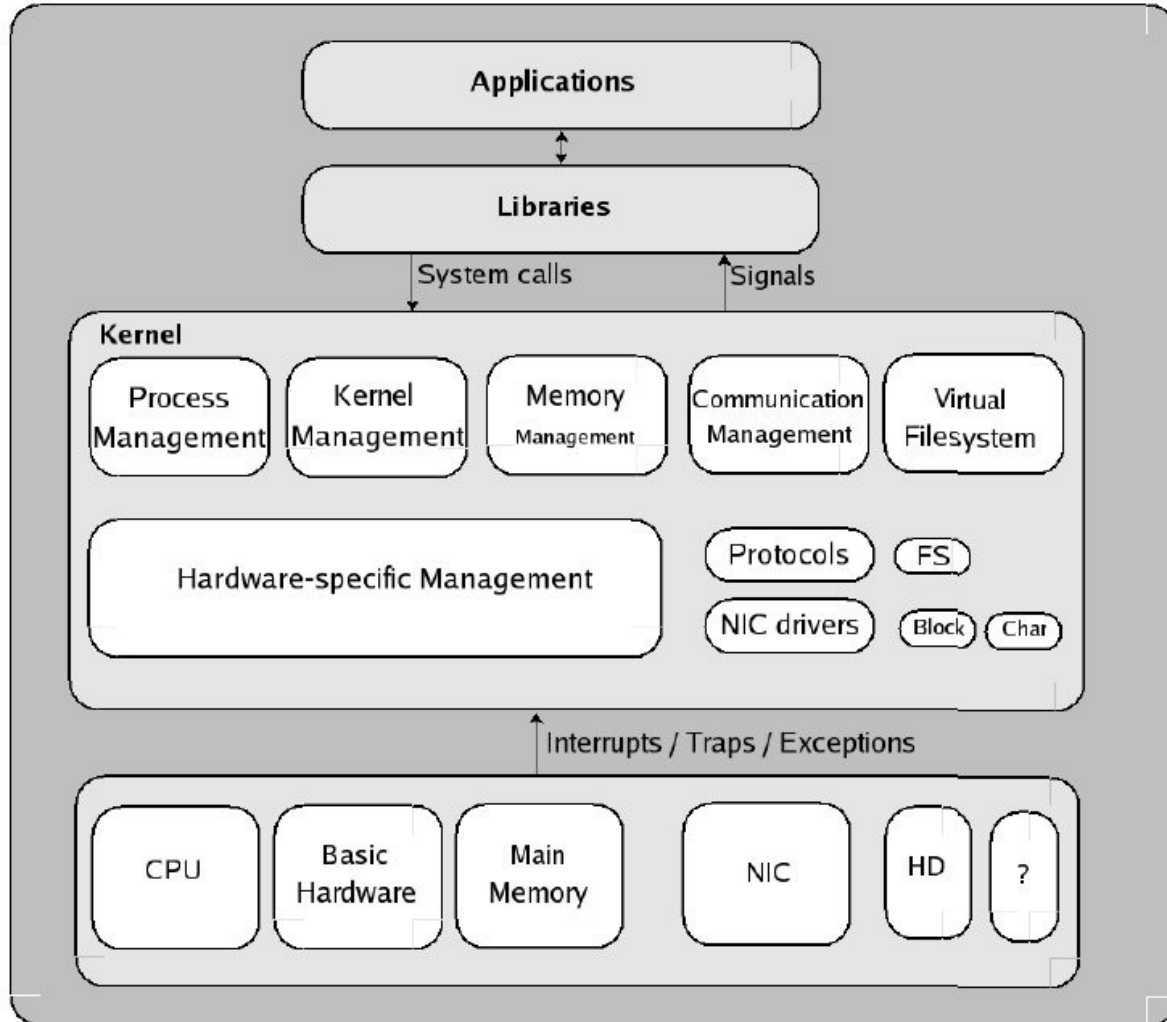
Introduction aux systèmes d'exploitation (SE)

- Architecture de LINUX (voir



Introduction aux systèmes d'exploitation (SE)

- Architecture de LINUX



Introduction aux systèmes d'exploitation (SE)

- Fonctionnalités de LINUX

LINUX est une implantation d'UNIX respectant les spécifications POSIX (Portable Operating System Interface), normes IEEE pour la standardisation du code source. LINUX possède aussi des extensions du SYSTEM V et BSD d'UNIX.

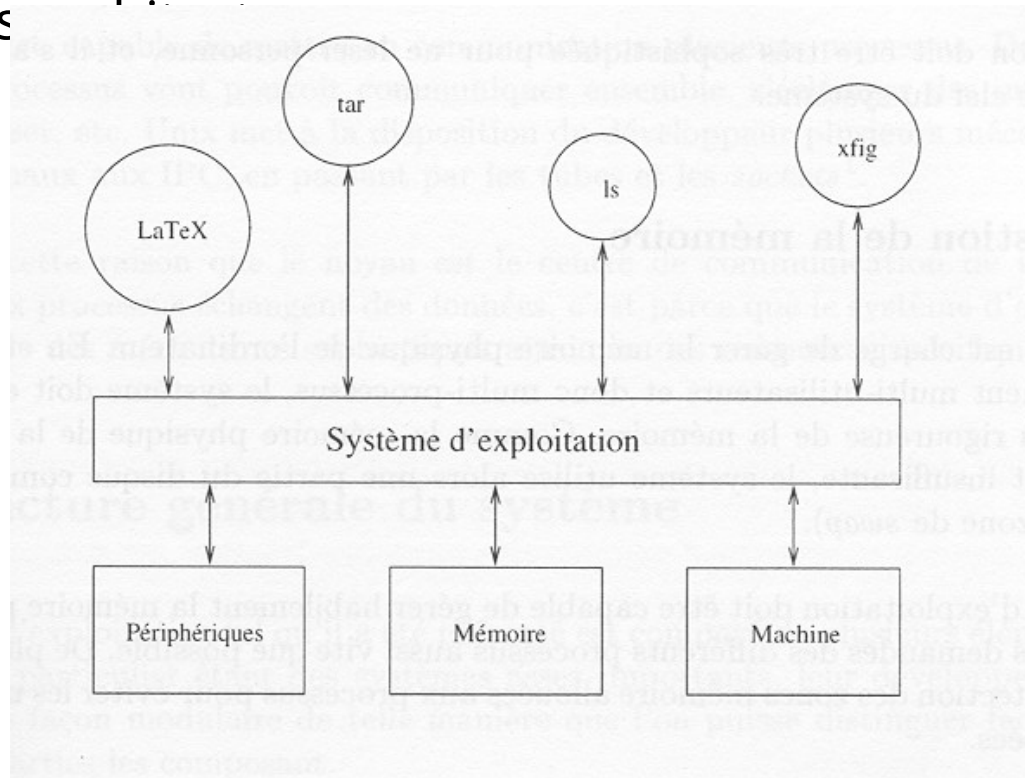
Introduction aux systèmes d'exploitation (SE)

- **Fonctionnalités de LINUX**

- **Multi-tâches, multi-processeurs**: exécution de plusieurs programmes en même temps sur un ou plusieurs processeurs
- **Multi-plate-formes**
- **Multi-utilisateurs**
- **Support des communications inter-processus** (tubes, IPC, socket)
- **Gestion des signaux**
- **Gestion des terminaux POSIX, pseudo-terminaux, contrôle de processus**
- **Support de périphériques** (carte de son, graphiques, réseaux, SCSI)
- **Buffer cache**: tampon mémoire des I/O par bloc des processus
- **Gestion de la mémoire** (paging)
- **Librairie partagée**
- **Système de fichiers**: partition LINUX avec le système de fichier EXT2 (EXT3)
- **Support réseaux** (TCP/IP)

Introduction aux systèmes d'exploitation (SE)

- Rôles d'un SE
 - **Offrir une machine virtuelle:** Le SE s'exécute sur une machine physique qui possède une interface de programmation de bas niveau, des abstractions de haut niveaux, une interface de programmation et d'utilisation évoluée. Standardisation de l'interface d'utilisation et de programmation donc d'utilisation facile sur d'autres

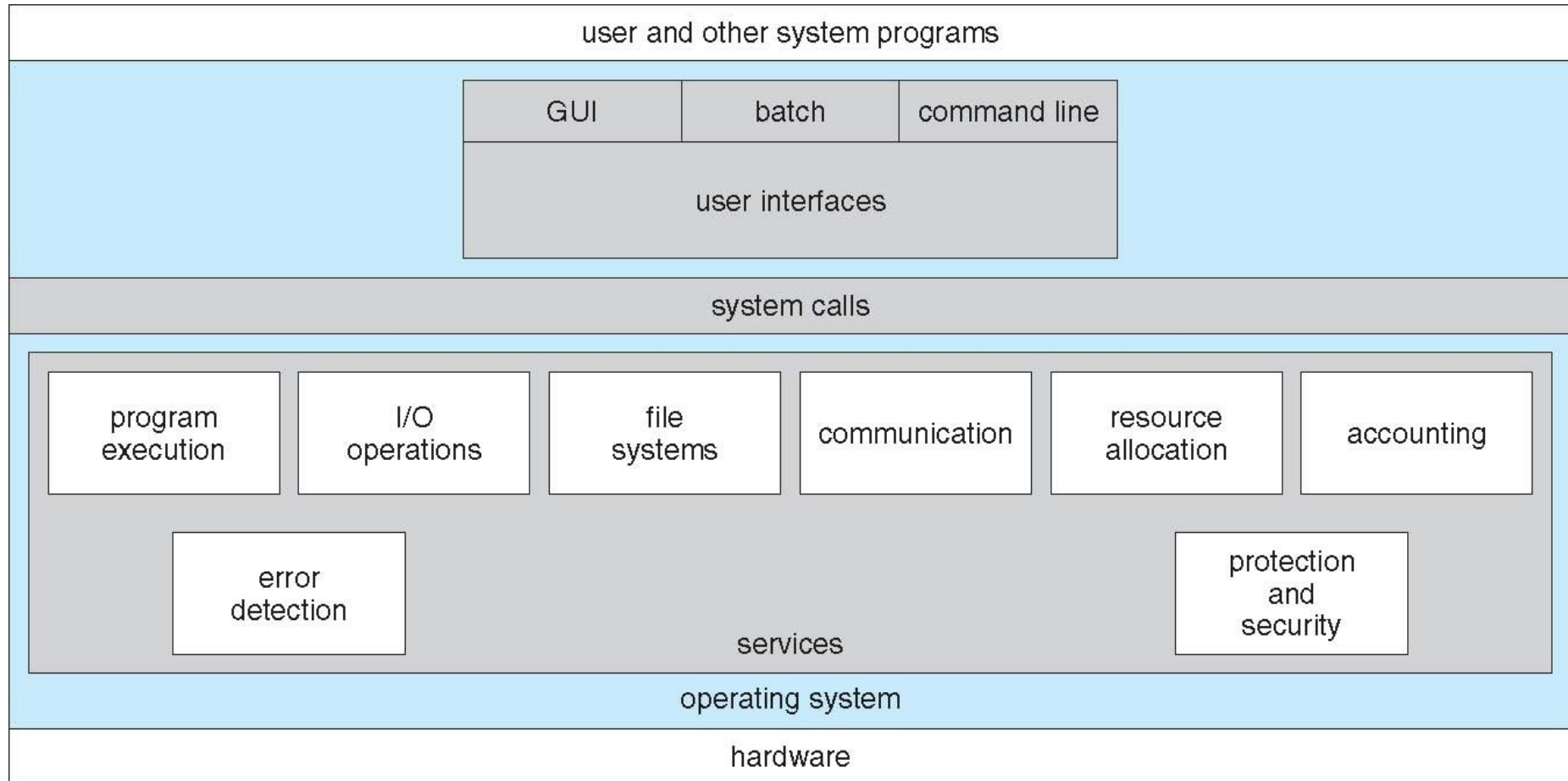


Introduction aux systèmes d'exploitation (SE)

- Un SE fournit des services aux programmes usagers
- Services accessibles par des **appels systèmes (system calls)**
- Appel système équivalent à un appel à une procédure quand il apparaît dans un programme
- **MAIS** transfert au SE (kernel) lors de l'exécution
- **SYSTEM CALLS** définissent les services et servent d'interface
- Accès aux services par l'appel de routines SE

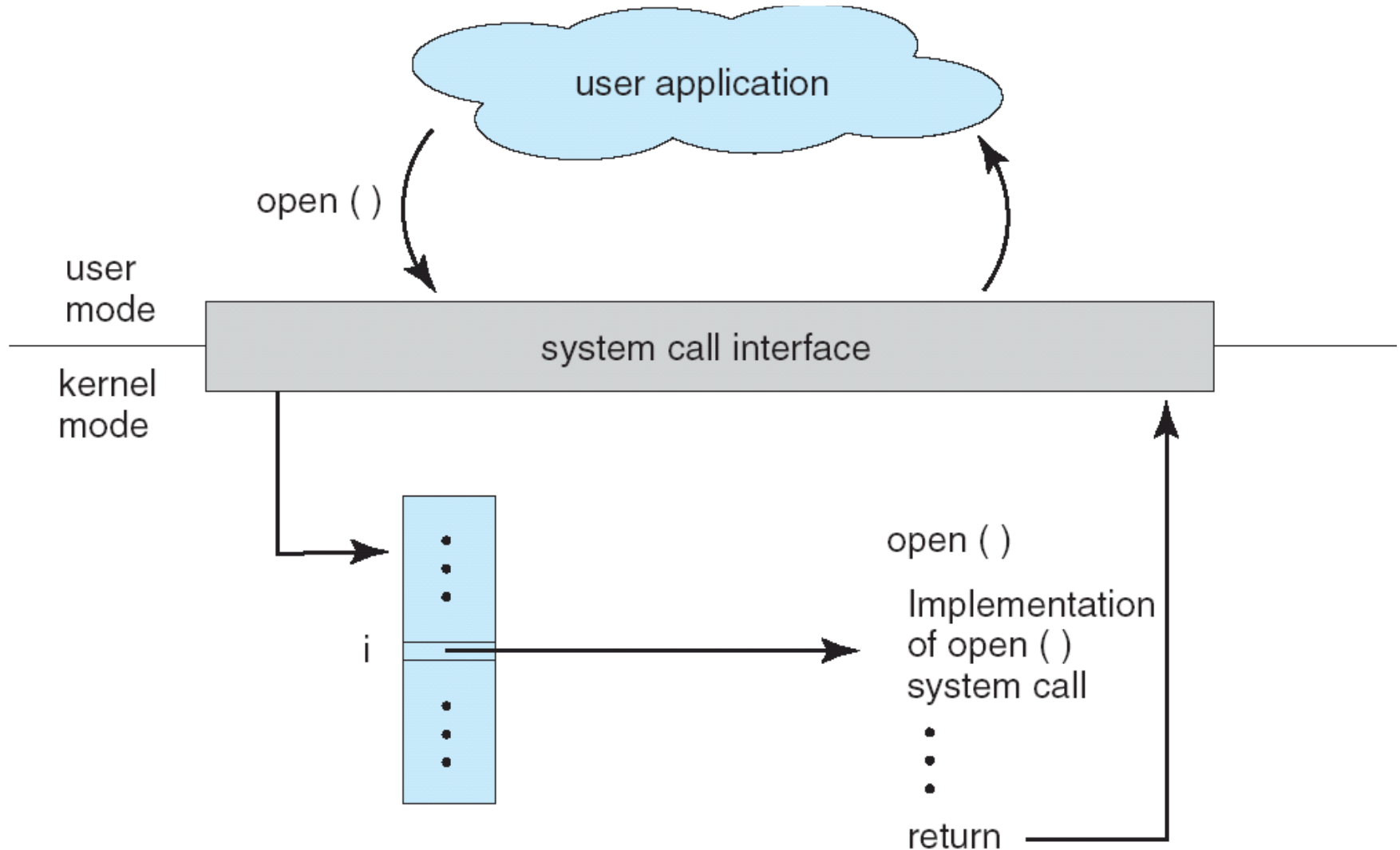
Introduction aux systèmes d'exploitation (SE)

- Vue des services offerts par un SE



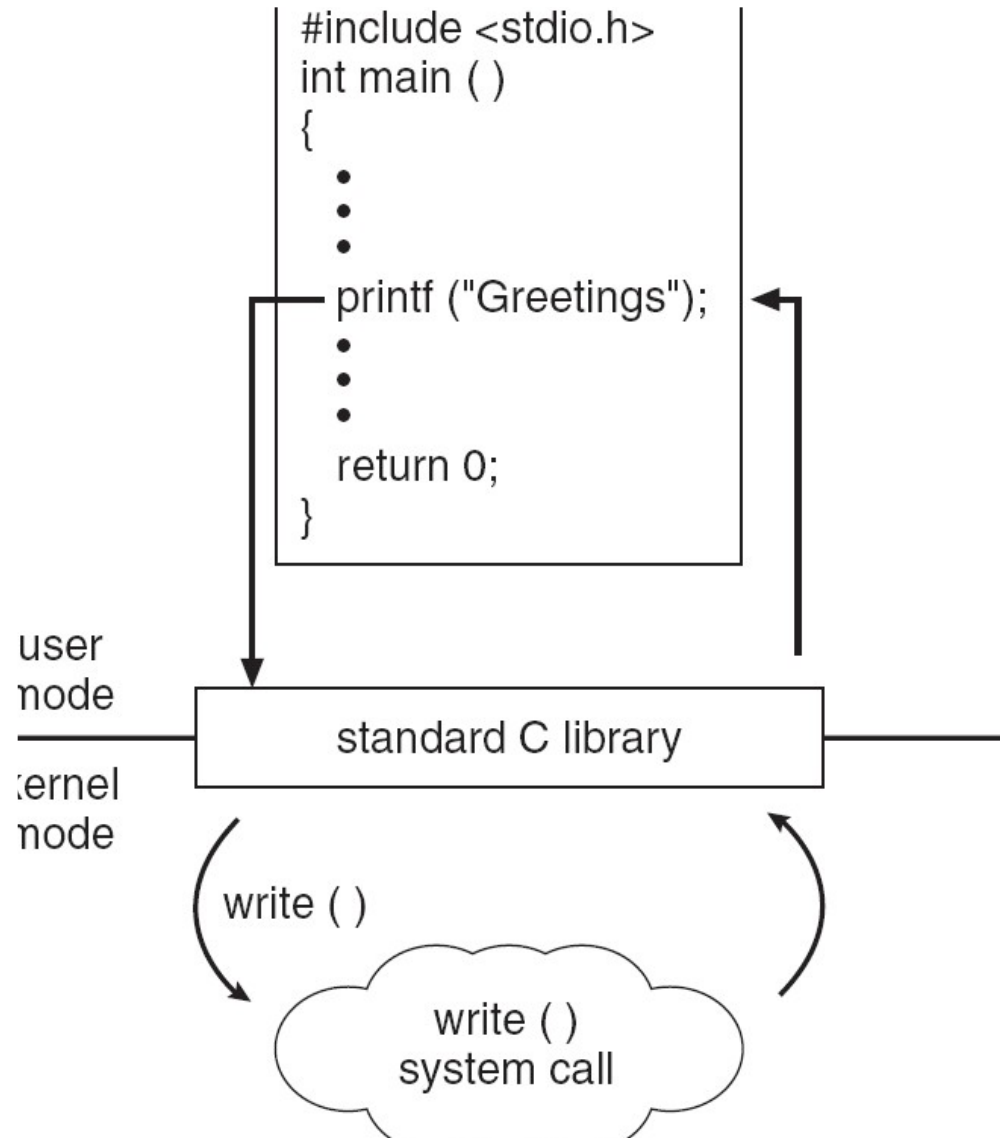
Introduction aux systèmes d'exploitation (SE)

- Exemple d'appel système (***open()***)



Introduction aux systèmes d'exploitation (SE)

- Programme en C et appel système



Introduction aux systèmes d'exploitation (SE)

- Types d'appels système (Windows/UNIX)

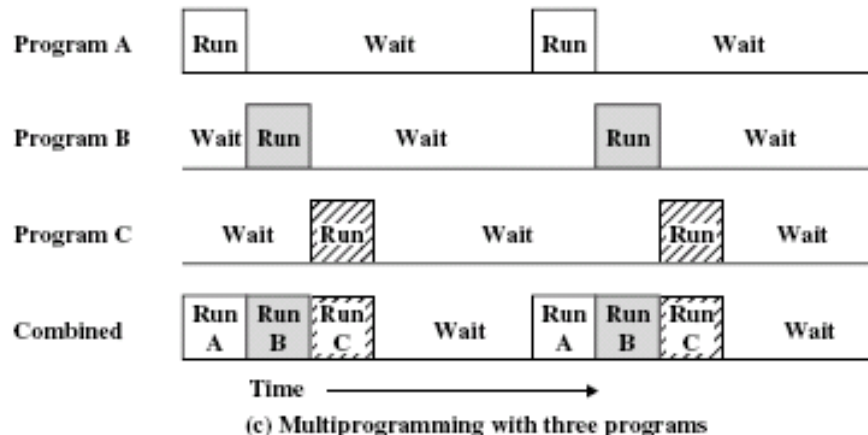
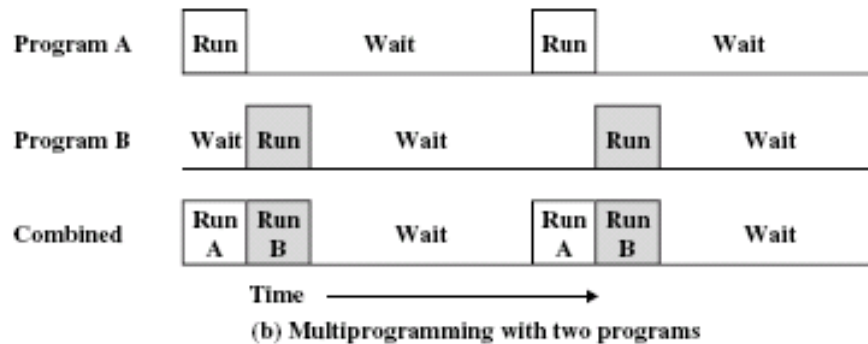
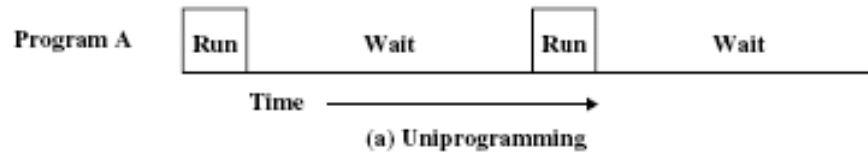
	Windows	Unix
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Manipulation	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device Manipulation	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()
Protection	SetFileSecurity() InitializeSecurityDescriptor() SetSecurityDescriptorGroup()	chmod() umask() chown()

Introduction aux systèmes d'exploitation (SE)

- Rôles d'un SE:
 - Partage du processeur
 - **Multi-tâches:** plusieurs processus s'exécute en même temps (ex: lancement d'une impression, formattage d'un disque, envoi/réception d'un mail peuvent s'exécuter en même temps)
 - Le SE coordonne les processus par un système d'ordonnan-cement qui permet de donner le processeur tour à tour à chacun des processus

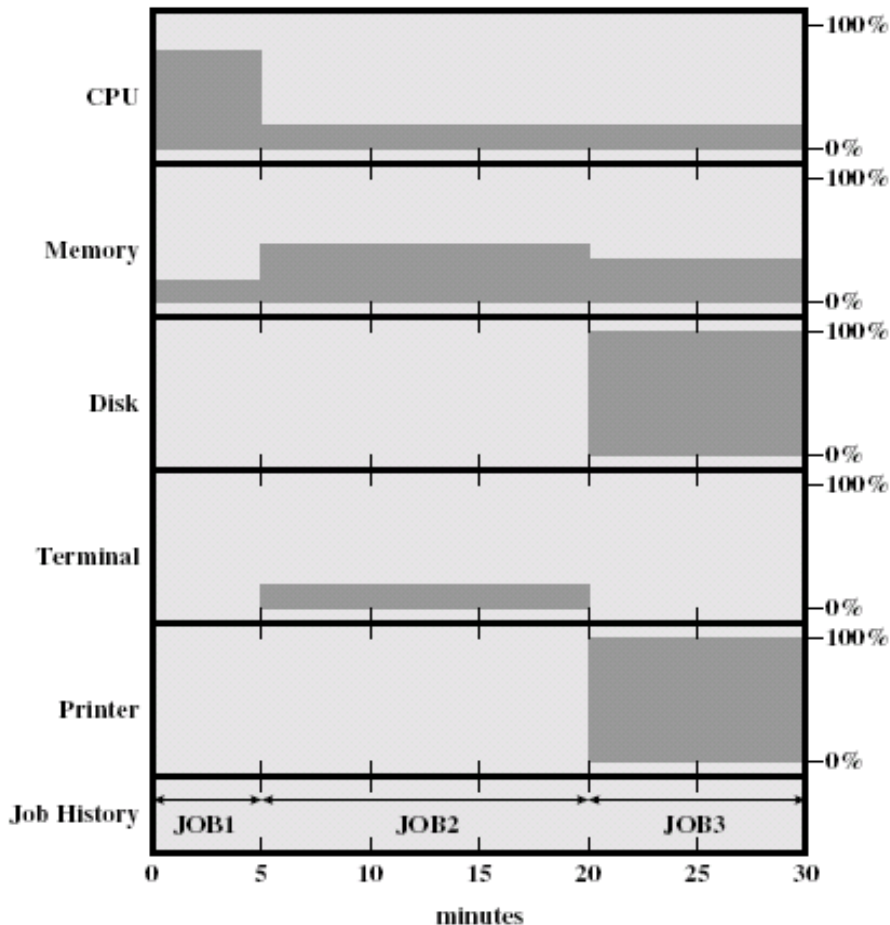
Introduction aux systèmes d'exploitation (SE)

- Rôles d'un SE:
 - Partage du processeur (Multi-tâches)

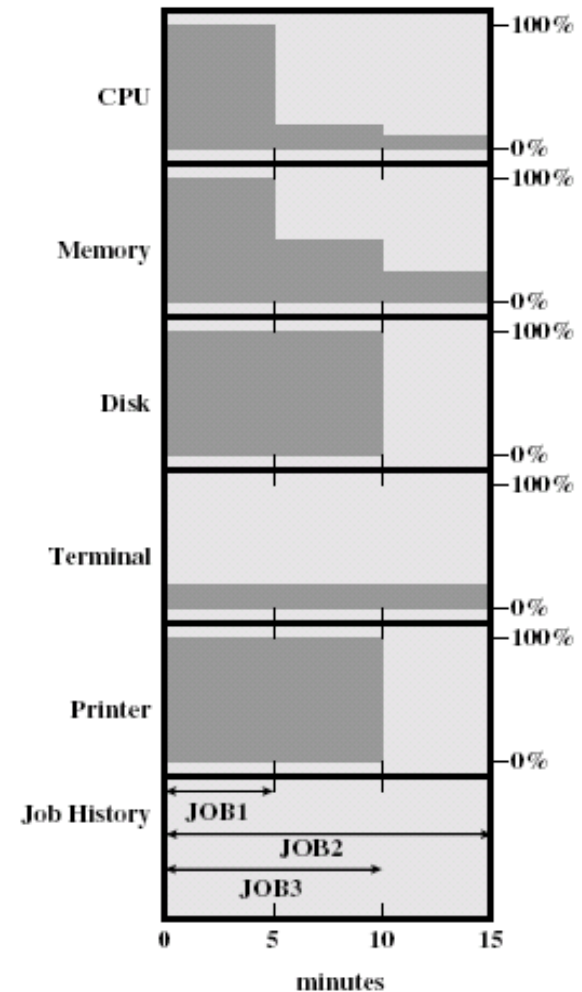


Introduction aux systèmes d'exploitation (SE)

- Rôles d'un SE:
 - Partage du processeur (Multi-tâches)



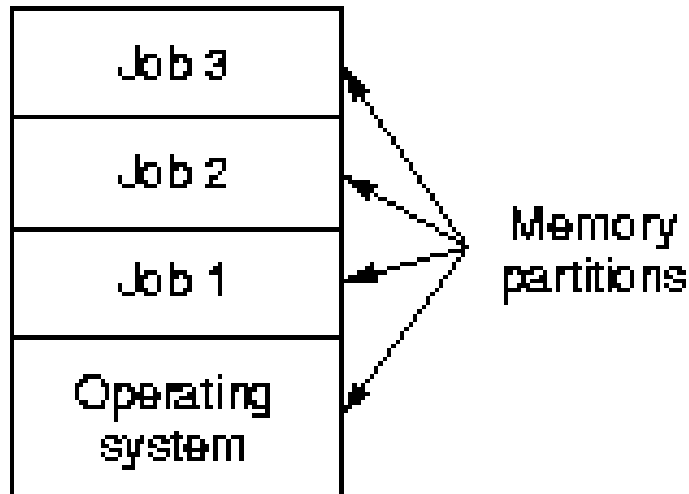
(a) Uniprogramming



(b) Multiprogramming

Introduction aux systèmes d'exploitation (SE)

- Rôles d'un SE:
 - **Partage du processeur** (traitement concurrent)
 - Le SE partage le processeur unique entre les multiples programmes (multiprogramming, multitasking, timesharing)
 - Le SE alloue le processeur tour à tour

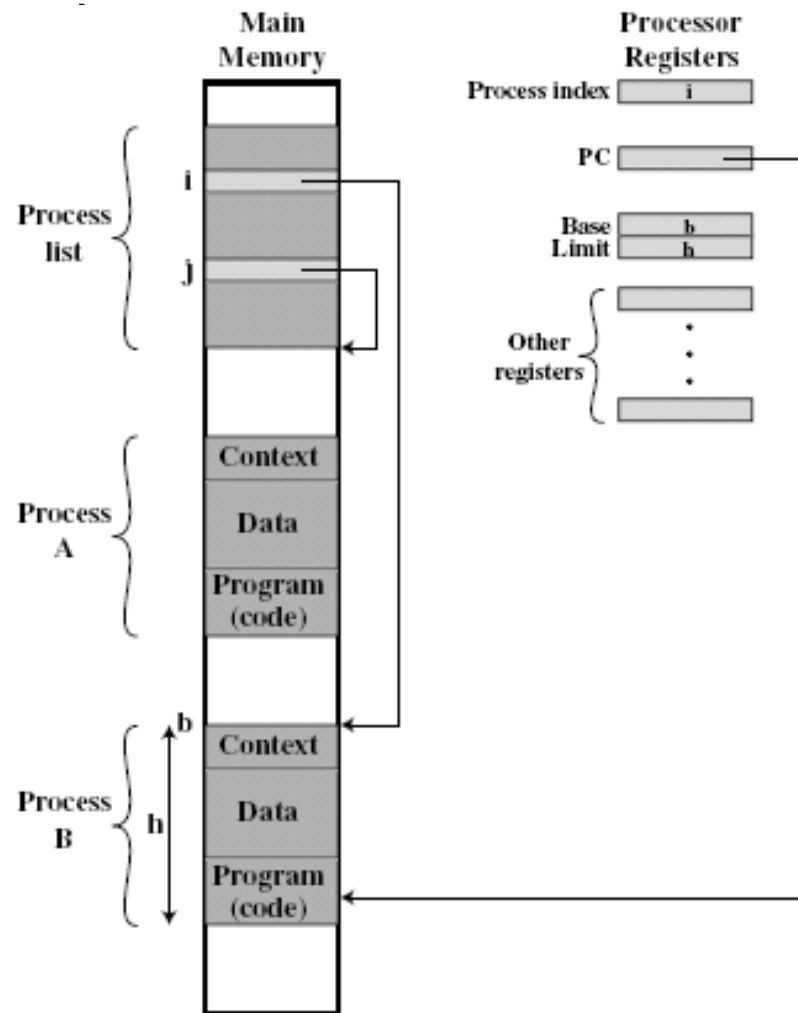


Introduction aux systèmes d'exploitation (SE)

- Rôles d'un SE:
 - Partage du processeur
 - **Distinction entre programme et processus**
 - Nous voyons un programme comme une séquence d'instructions s'exécutant sans interruptions
 - Dans un environnement concurrent, plusieurs processus peuvent s'exécuter
 - Le CPU est partagé par les processus
 - Des processus peuvent exécuter le même programme et parfois les mêmes instructions
 - Un appel de procédure normale ne retourne que lorsque la procédure est terminée
 - Plusieurs processus peuvent exécuter le même code simultanément
 - Le stockage des variables locales et des arguments de procédures est propre à chaque processus

Introduction aux systèmes d'exploitation (SE)

- Rôles d'un SE:
 - Partage du processeur (**Distinction entre programme et processeur**)



Introduction aux systèmes

d'exploitation (SE)

- Rôles d'un SE:
 - **Partage du processeur**
 - **Partage de la mémoire**
 - Chaque processus possède son ensemble de variables locales, de paramètres formels et d'appels de procédure **MAIS** peuvent aussi partager des variables globales
 - Comment gérer le partage des variables globales?
 - Par la **synchronisation** ou l'**exclusion mutuelle**
 - **Synchronisation**
 - Deux processus qui échangent des informations ordonnées doivent communiquer en synchronisme. Les **sémaphores** permettent la synchronisation des processus
 - **Exclusion mutuelle**
 - Plusieurs processus qui coopèrent sont dit en **exclusion mutuelle** si un seul d'entre eux a accès à une ressource à un temps donné
 - Pour protéger l'accès aux éléments d'une liste liée nous devons protéger et rendre exclusif l'accès à chaque élément (nœud) de la liste

Introduction aux systèmes d'exploitation (SE)

- Rôles d'un SE:
 - **Gestion mémoire**
 - Le SE doit gérer la mémoire physique de l'ordinateur
 - Avec de la mémoire physique restreinte, le SE utilise le disque comme mémoire auxiliaire (SWAP)
 - Le SE doit aussi protéger les zones de mémoires allouées aux processus
 - Le SE doit gérer l'ensemble de la mémoire (RAM et disque) en utilisant le concept de mémoire virtuelle

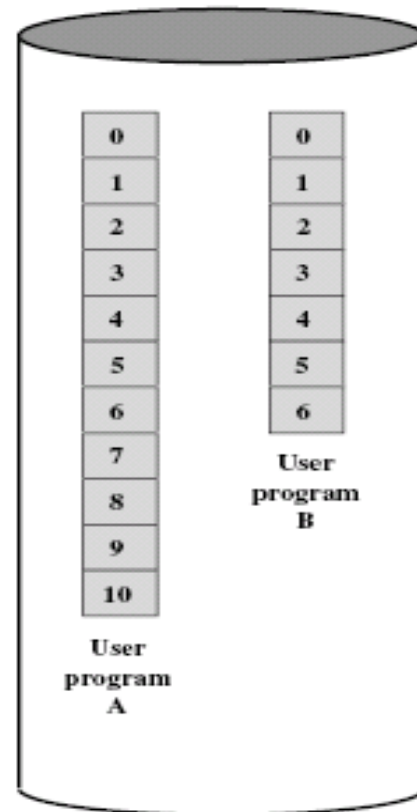
Introduction aux systèmes d'exploitation (SE)

- Rôles d'un SE:
 - **Gestion mémoire (Mémoire virtuelle)**

A.1			
	A.0	A.2	
	A.5		
B.0	B.1	B.2	B.3
		A.7	
	A.9		
		A.8	
B.4	B.5	B.6	

Main Memory

Main memory consists of a number of fixed-length frames, equal to the size of a page. For a program to execute, some or all of its pages must be in main memory.

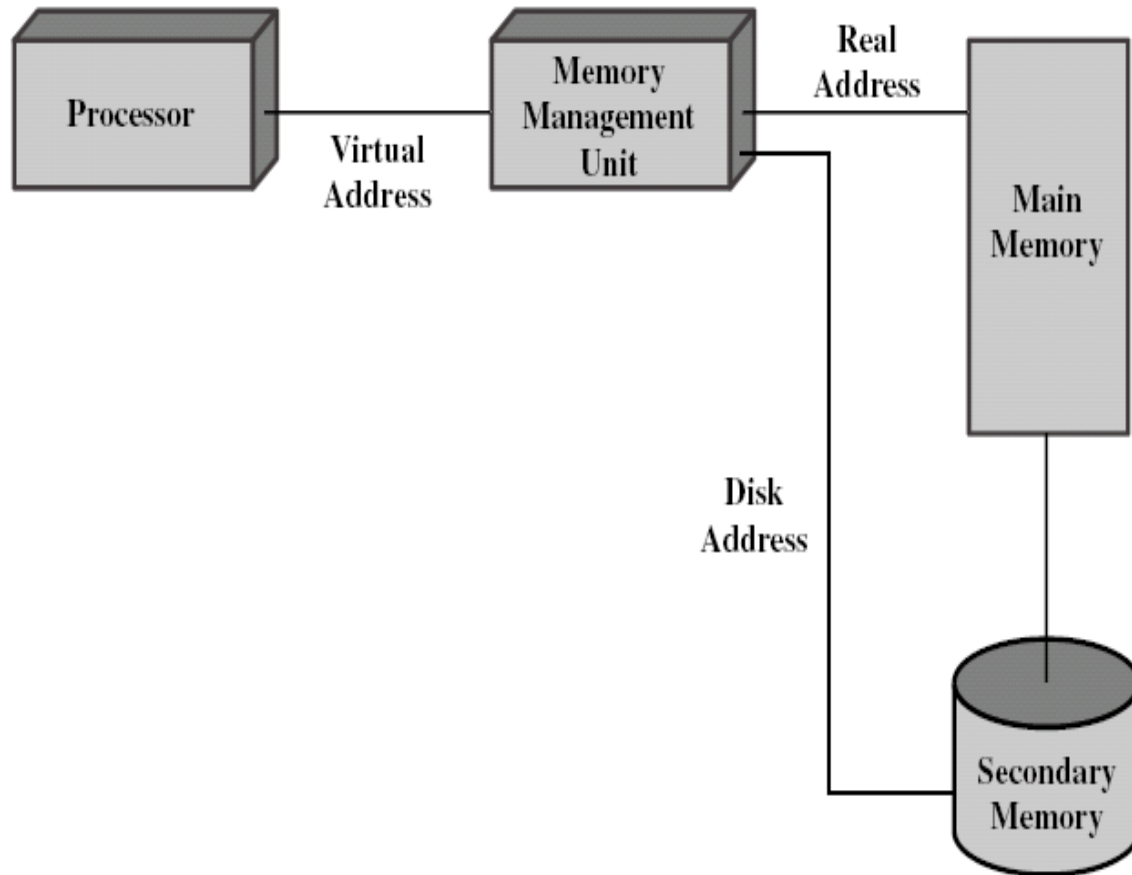


Disk

Secondary memory (disk) can hold many fixed-length pages. A user program consists of some number of pages. Pages for all programs plus the operating system are on disk, as are files.

Introduction aux systèmes d'exploitation (SE)

- Rôles d'un SE:
 - **Gestion mémoire (Mémoire virtuelle)**

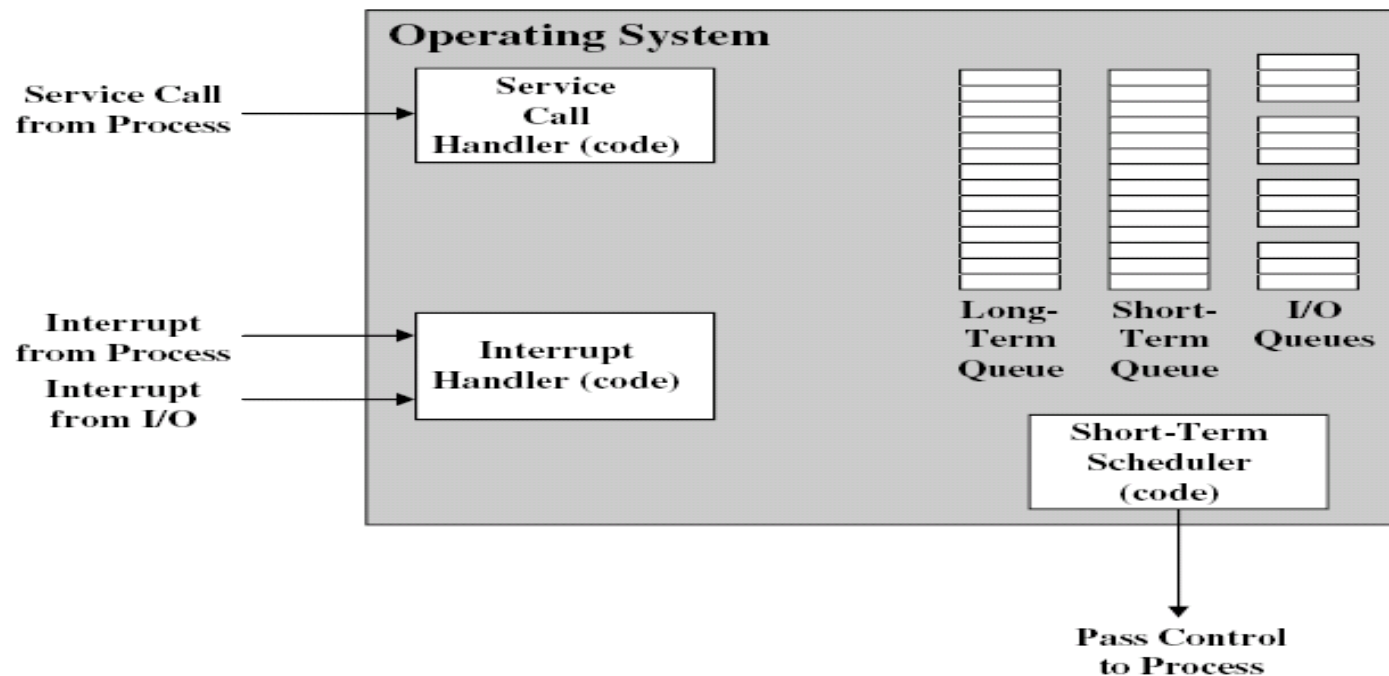


Introduction aux systèmes d'exploitation (SE)

- Rôles d'un SE:
 - **Gestion de ressources**
 - Le SE offre au processus une interface permettant de partager les ressources (disques, imprimantes, etc.) de la machine
 - Le SE maintient des listes de ressources disponibles et en cours d'utilisation, ce qui permet de les attribuer aux processus qui en ont besoin

Introduction aux systèmes d'exploitation (SE)

- Rôles d'un SE:
 - **Gestion de ressources**



Éléments importants d'un SE pour la gestion des ressources
(CPU: séquençement, dispositifs d'I/O)

Introduction aux systèmes d'exploitation (SE)

- Rôles d'un SE:
 - **Communication**
 - Gestion des événements provenant soit du matériel (interrup-tions), soit des applications (appels système). Le SE traite ces événements et les acheminent aux processus concernés
 - Le SE met en communication plusieurs processus sur la même machine ou distants qui peuvent échanger des informations, communiquer ensembles, se synchroniser (signaux, IPC, tubes, sockets)

Introduction aux systèmes d'exploitation (SE)

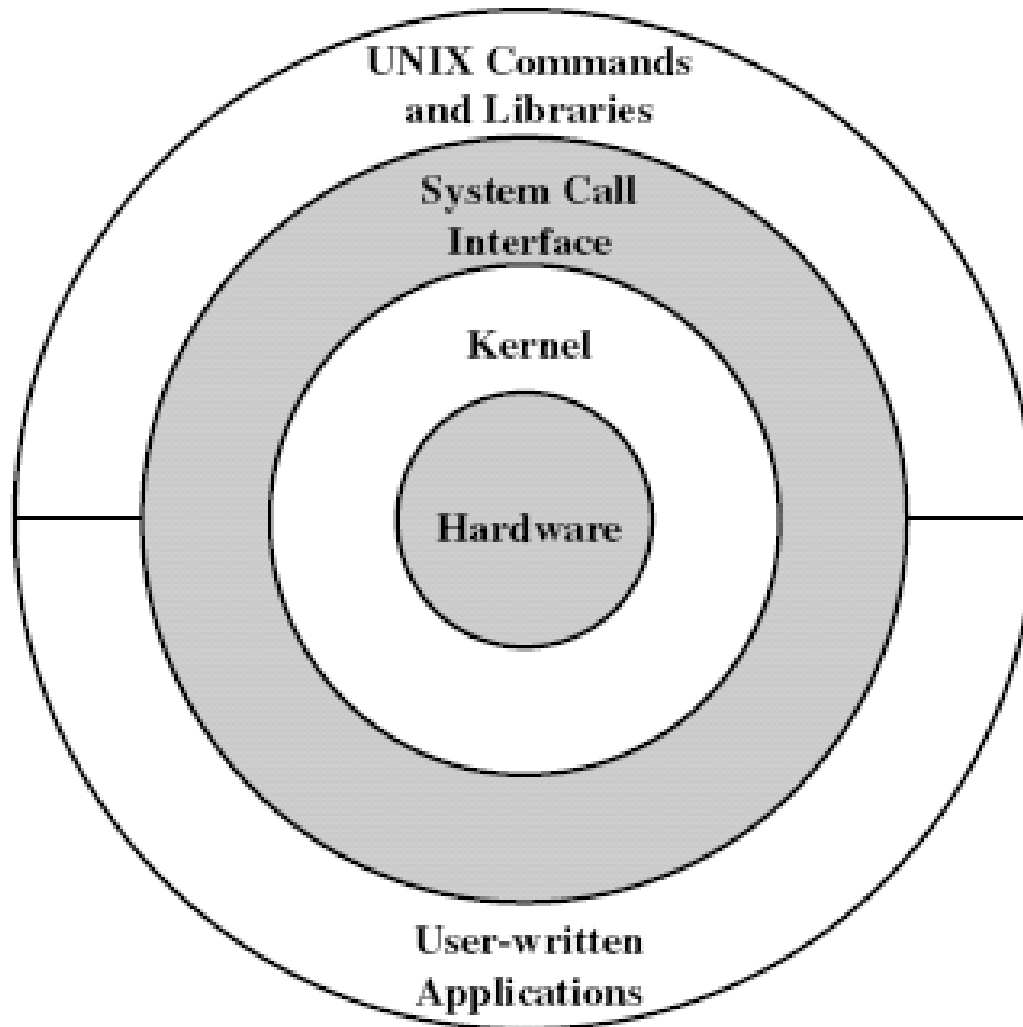
- Structures générales:
 - **Noyau du système UNIX**
 - **Appels système:** opérations exécutées en mode noyau
 - **Système de fichiers:** I/O sur les périphériques
 - **Buffer cache:** tampon pour I/O par bloc
 - **Gestionnaire de périphériques:** gestion de bas niveau des disques, des imprimantes etc.
 - **Gestion réseau:** protocoles de communication réseau
 - **Interface avec la machine:** code (assembleur) d'accès bas niveau au matériel de la machine

Introduction aux systèmes d'exploitation (SE)

- Structures générales:
 - Noyau du système UNIX
 - **Centre du noyau:**
 - Gestion des processus (création, duplication, destruction)
 - Gestion de l'ordonnancement des processus
 - Gestion des signaux
 - Gestion de mémoire (physique et virtuelle)

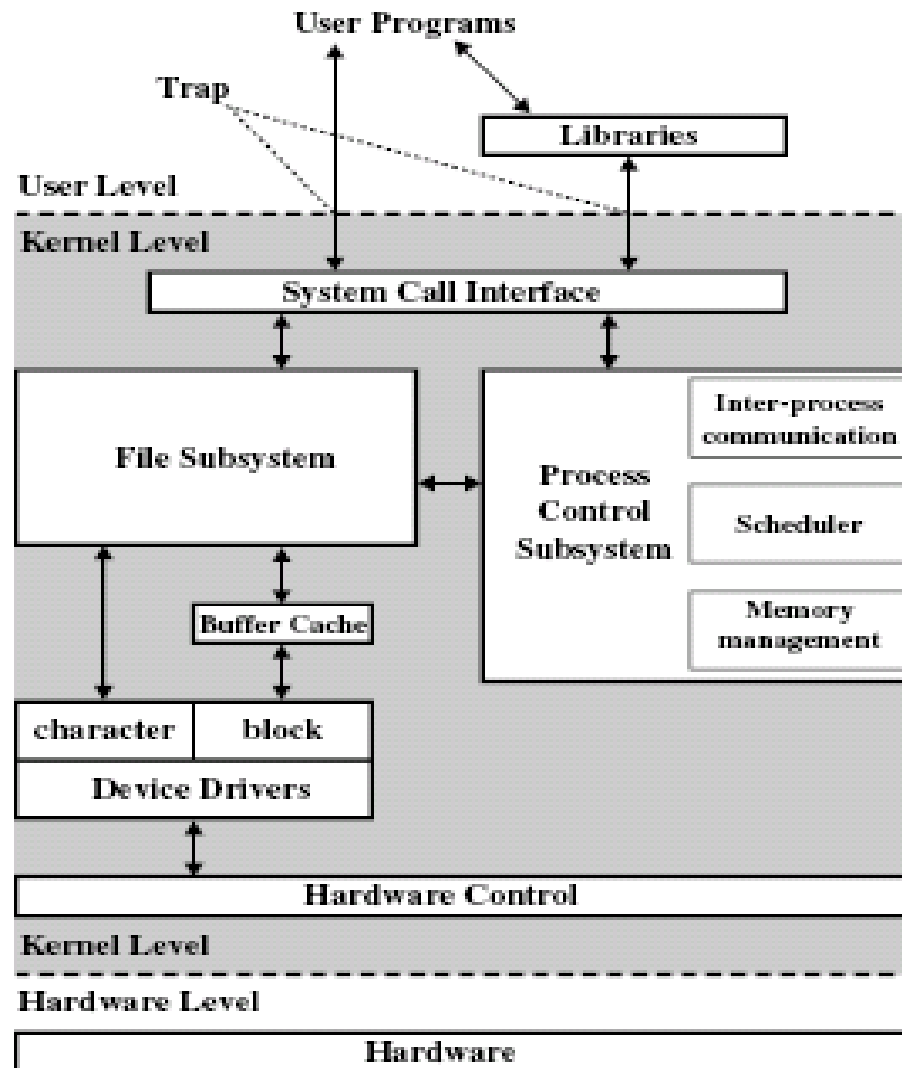
Introduction aux systèmes d'exploitation (SE)

- Structures générales: Noyau du système UNIX



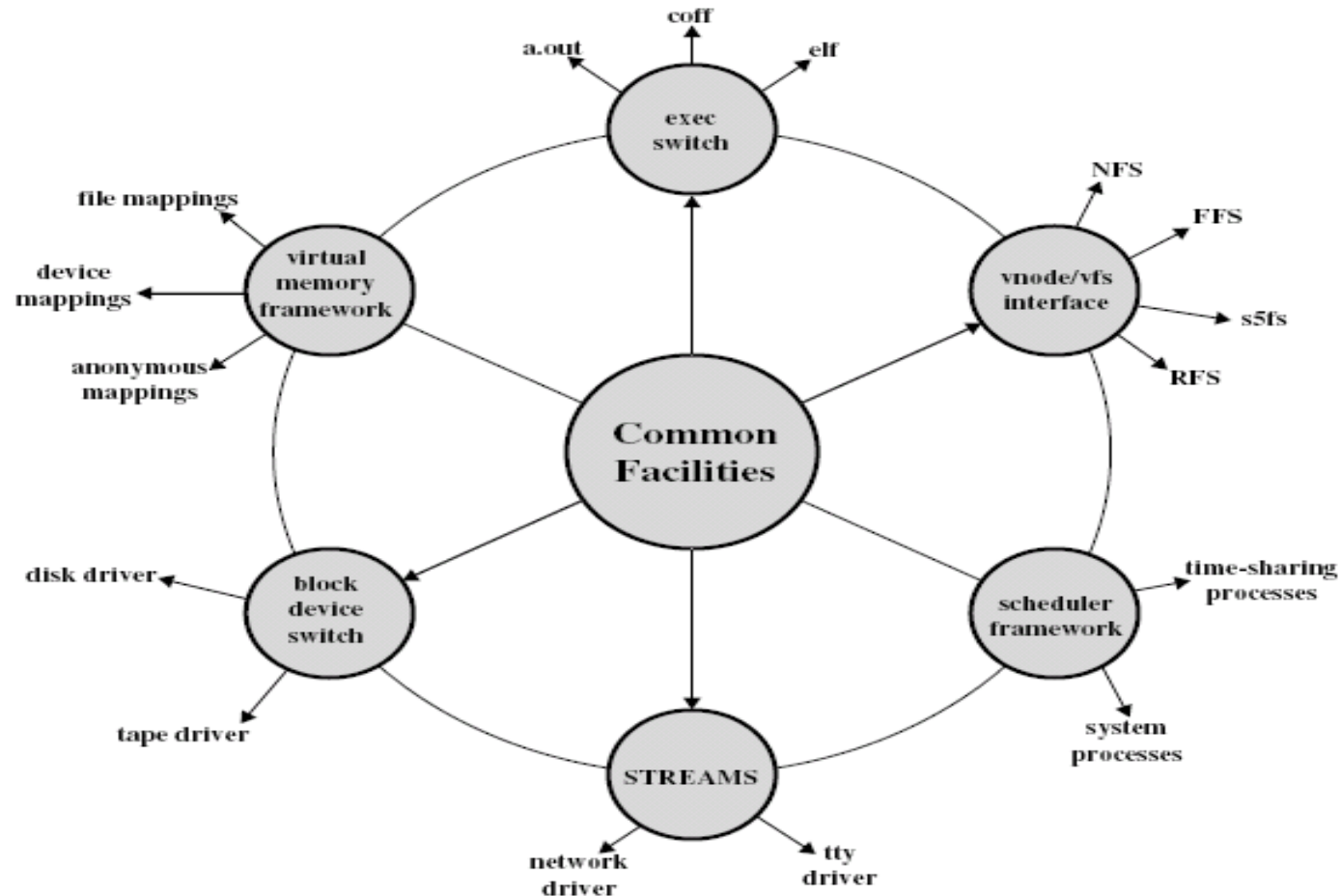
Introduction aux systèmes d'exploitation (SE)

- Structures générales: Noyau du système UNIX traditionnelle



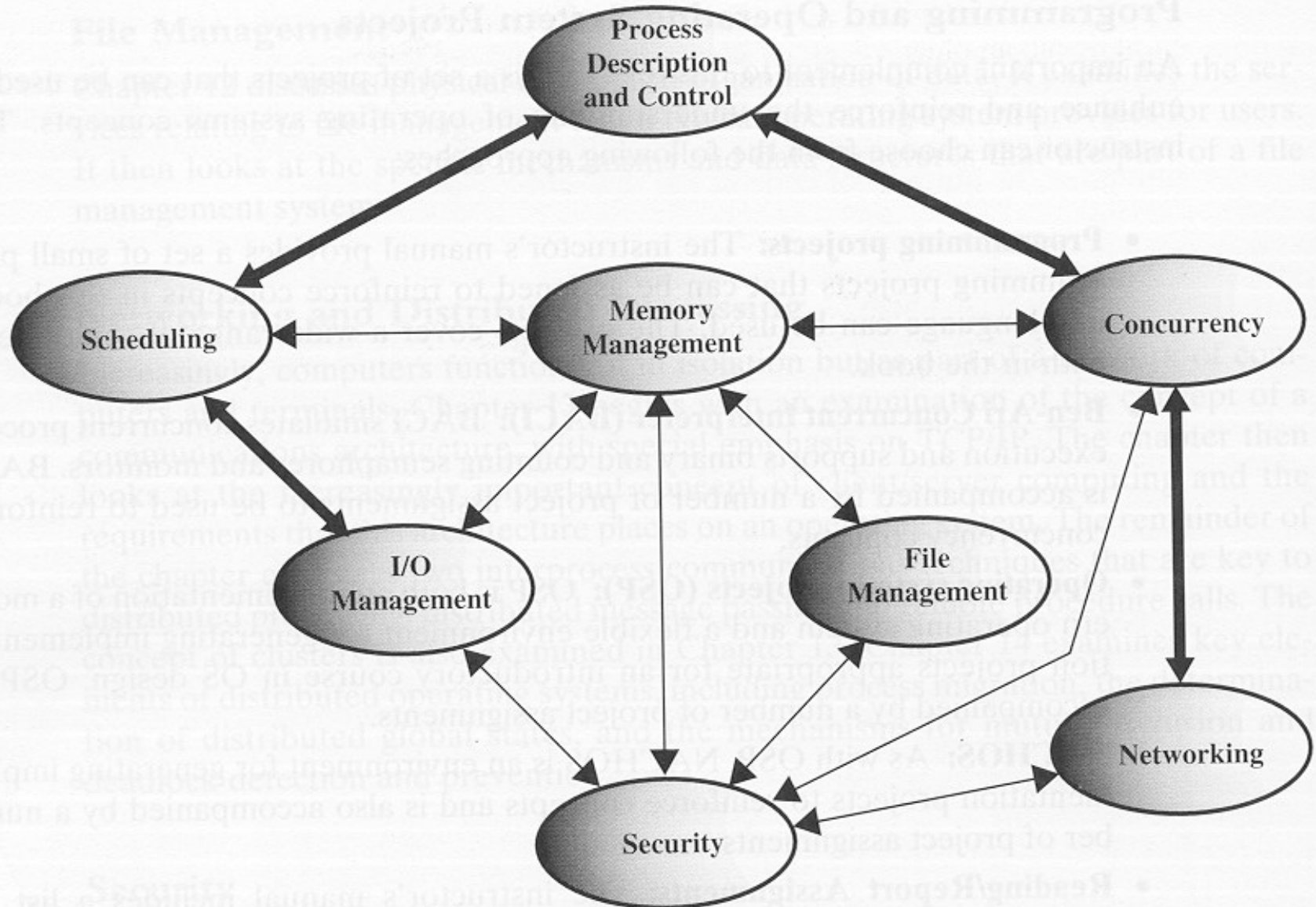
Introduction aux systèmes d'exploitation (SE)

- Structures générales: Noyau du système UNIX plus récent



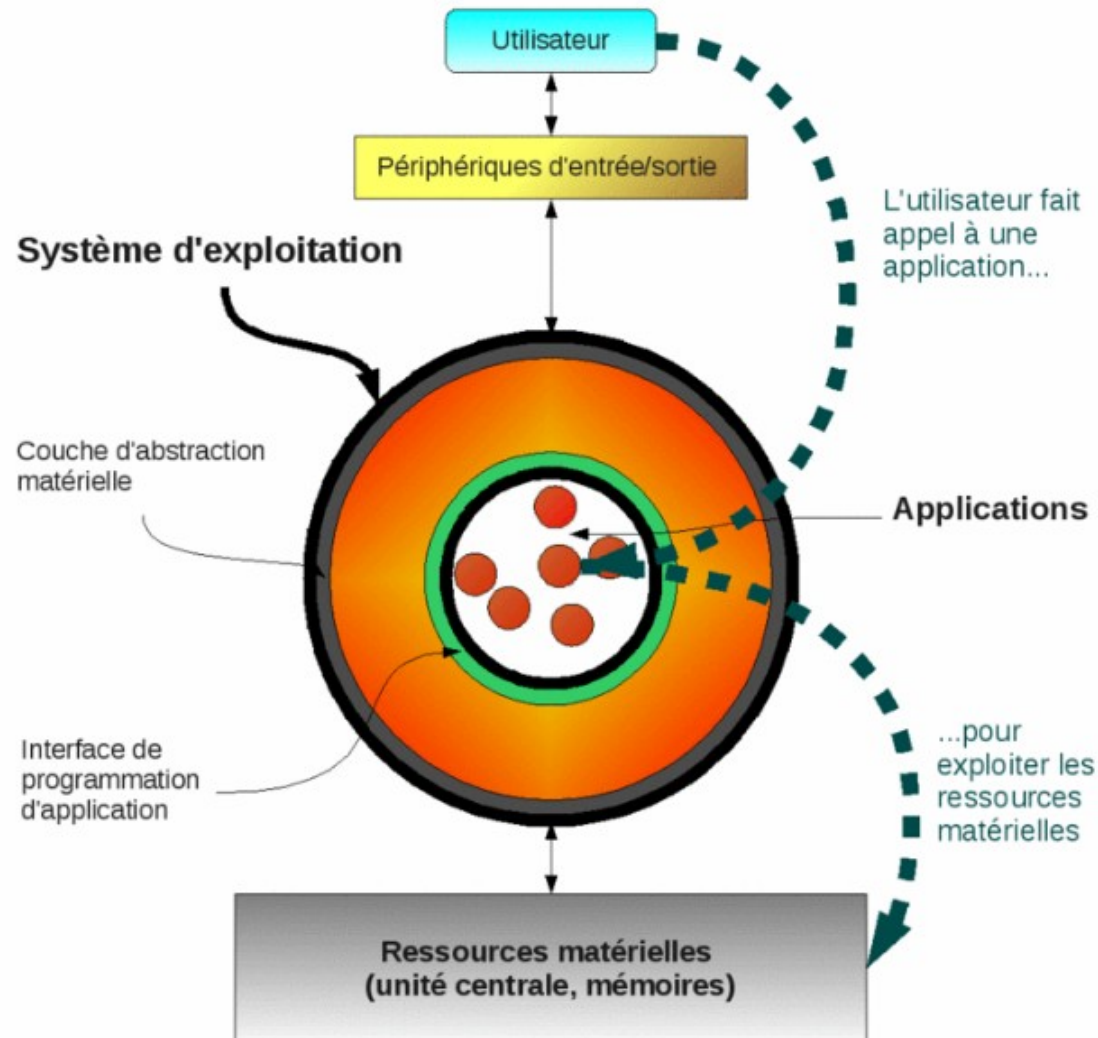
Introduction aux systèmes d'exploitation (SE)

- Structures générales:



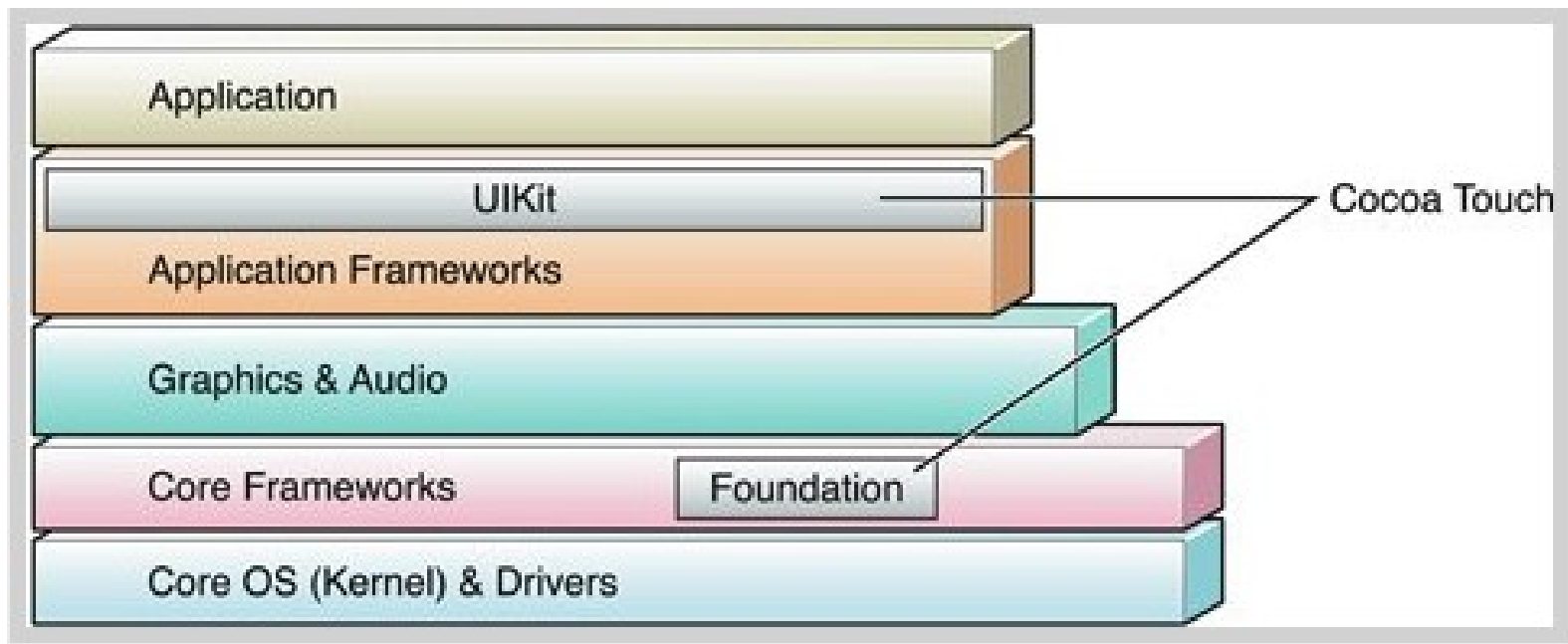
Introduction aux systèmes d'exploitation (SE)

- Structures générales:



Introduction aux systèmes d'exploitation (SE)

- Autres SE: Architectures Mac OS X, IOS, SE sur les Iphone, Ipad, Ipod Touch



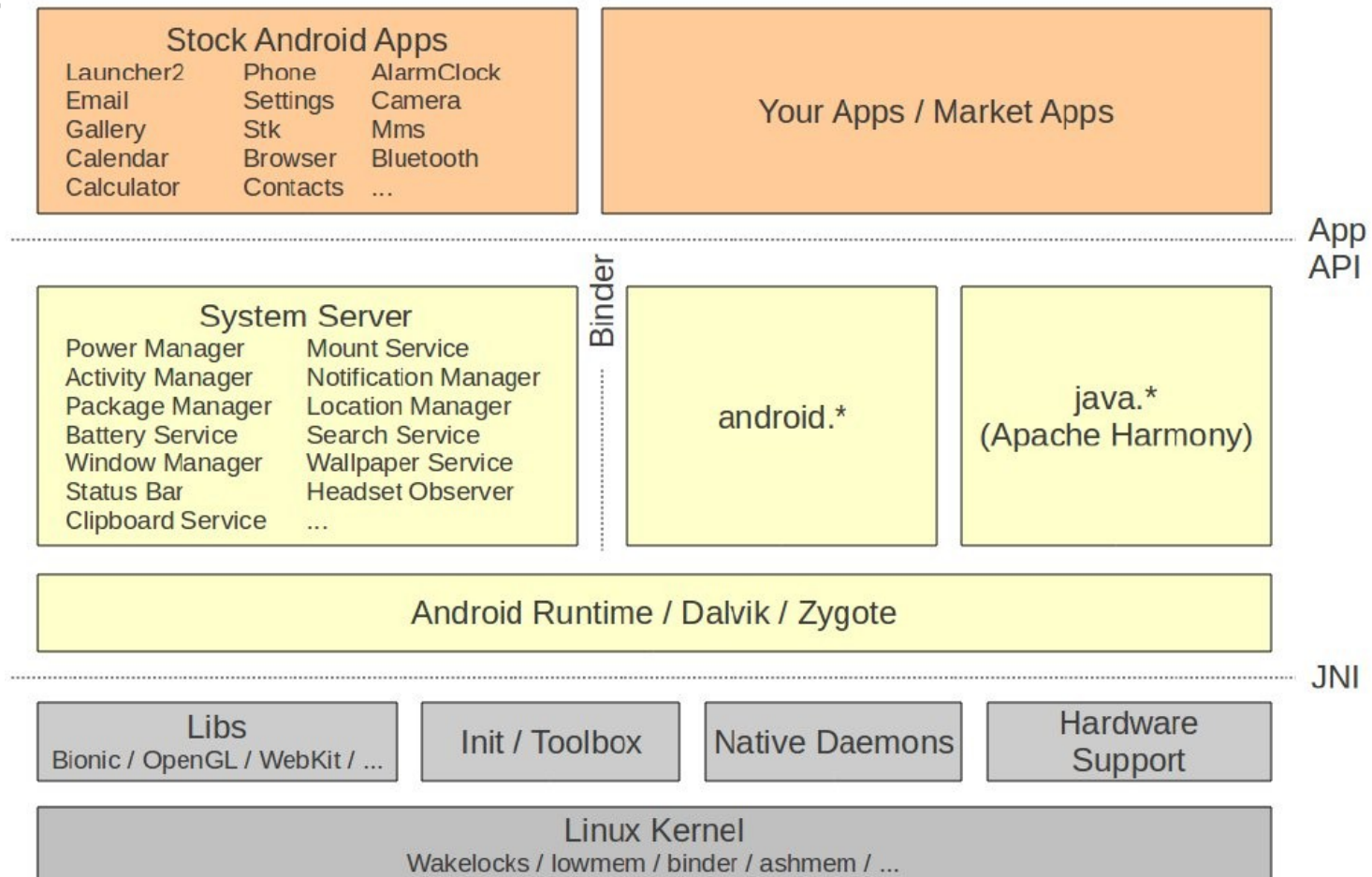
Introduction aux systèmes d'exploitation (SE)

- Autres SE: Architectures Google Android (Tablette Iconia Acer, Galaxy Tab Samsung, Tablette Xoom Motorola)



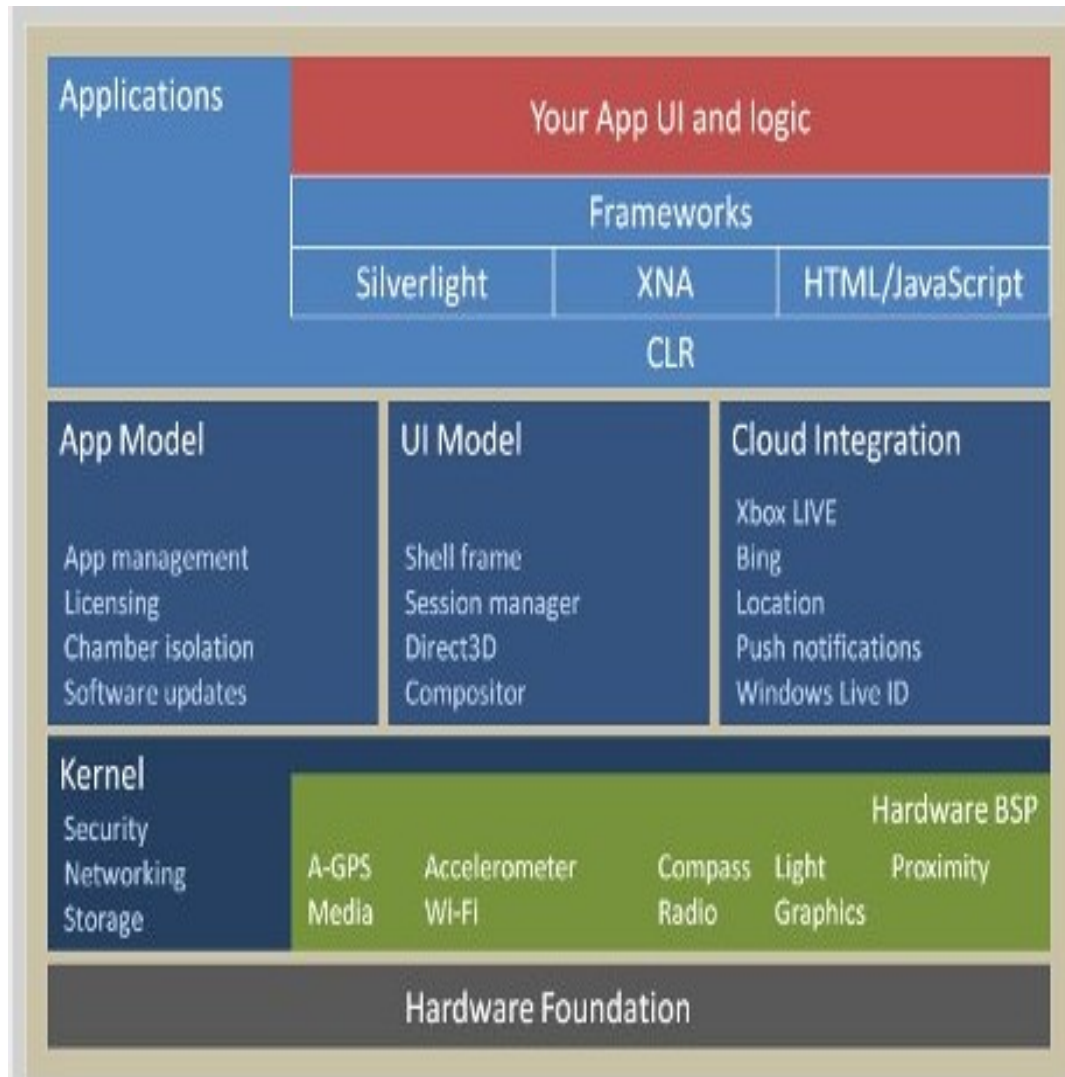
Introduction aux systèmes d'exploitation (SE)

- Autres SE: Architectures Google Android (Tablette Iconia Acer, Galaxy Tab Samsung, Tablette Xoom Motorola, etc)



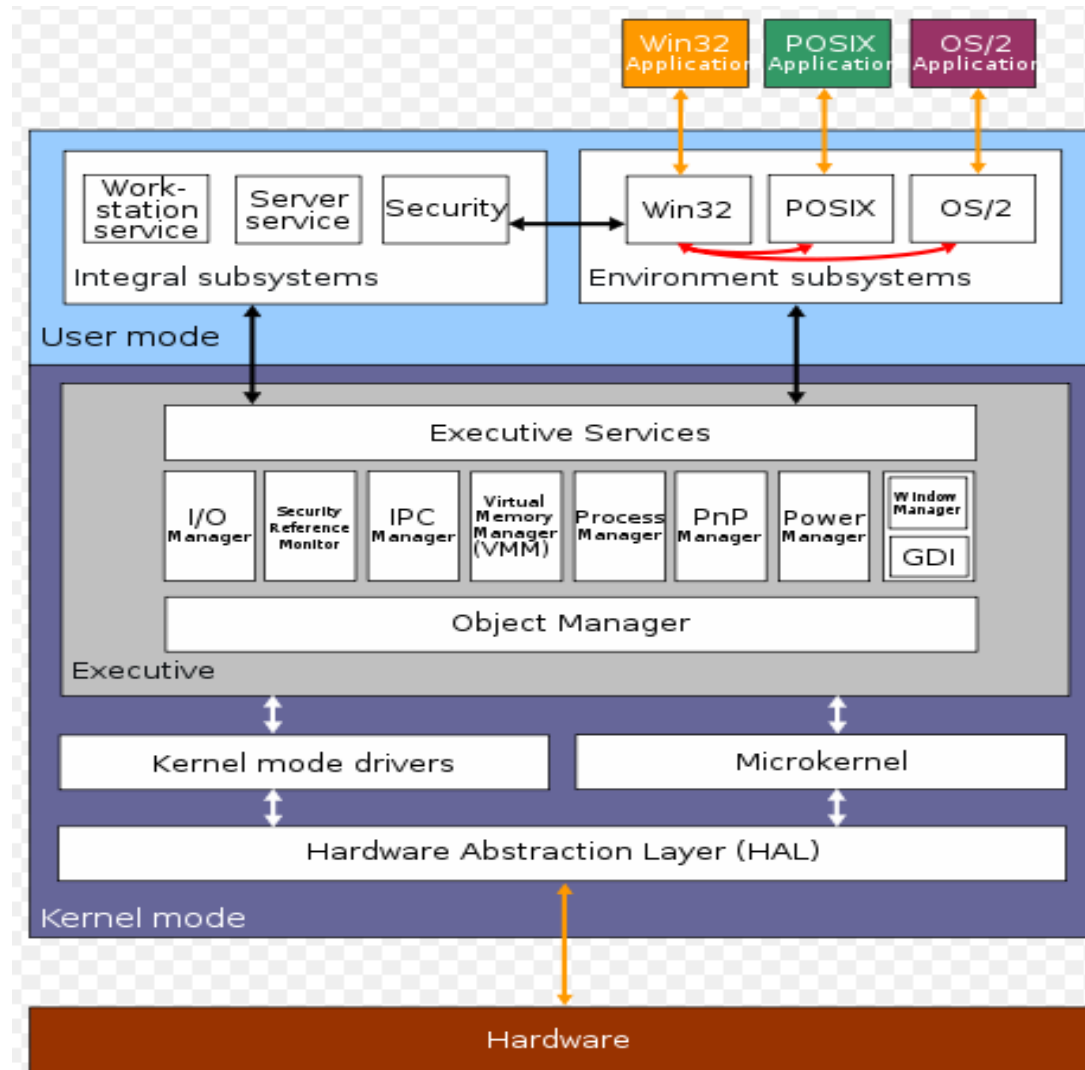
Introduction aux systèmes d'exploitation (SE)

- Autres SE: Architectures Windows Phone 7



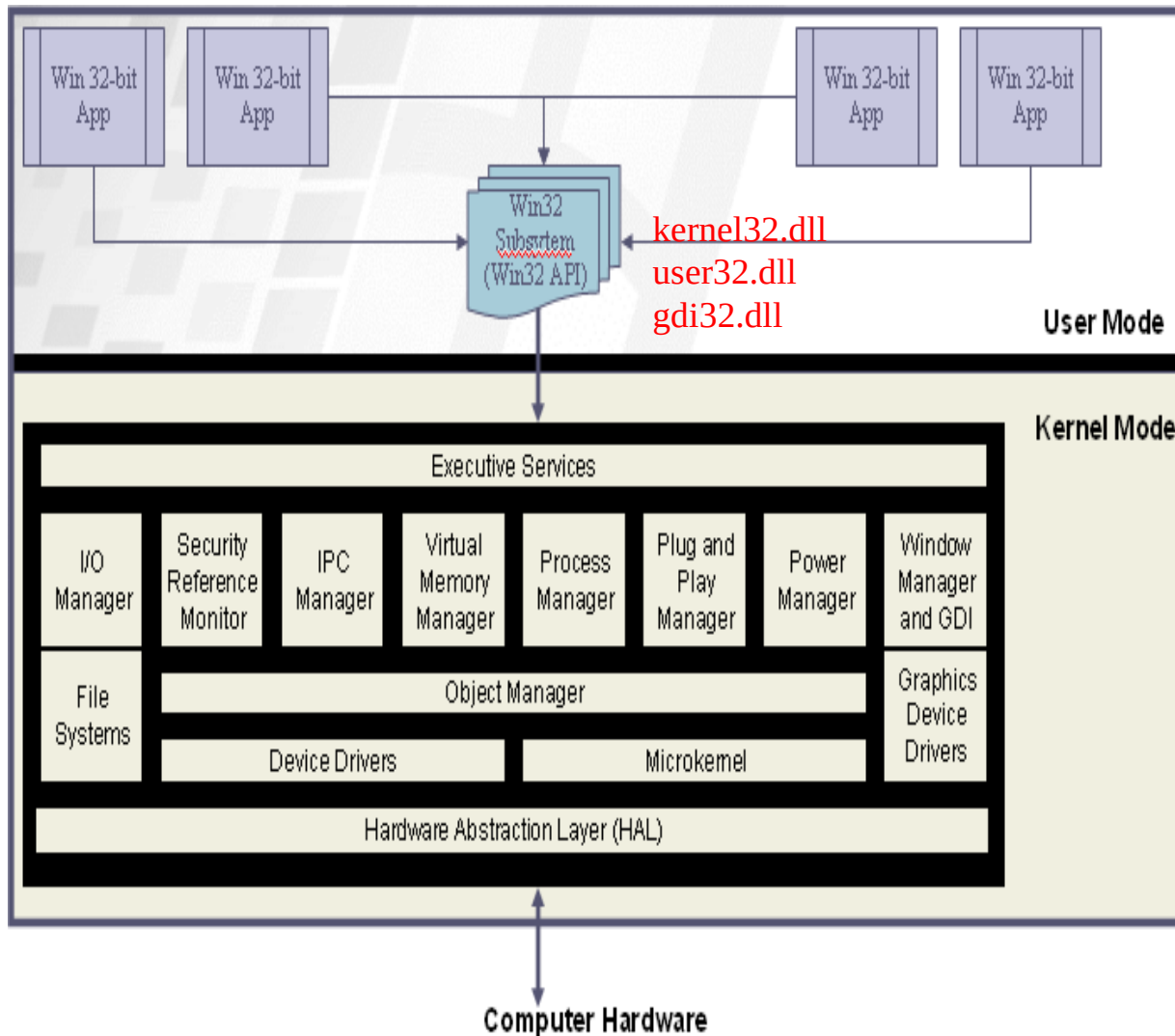
Introduction aux systèmes d'exploitation (SE)

- Autres SE: Architectures Windows 2000



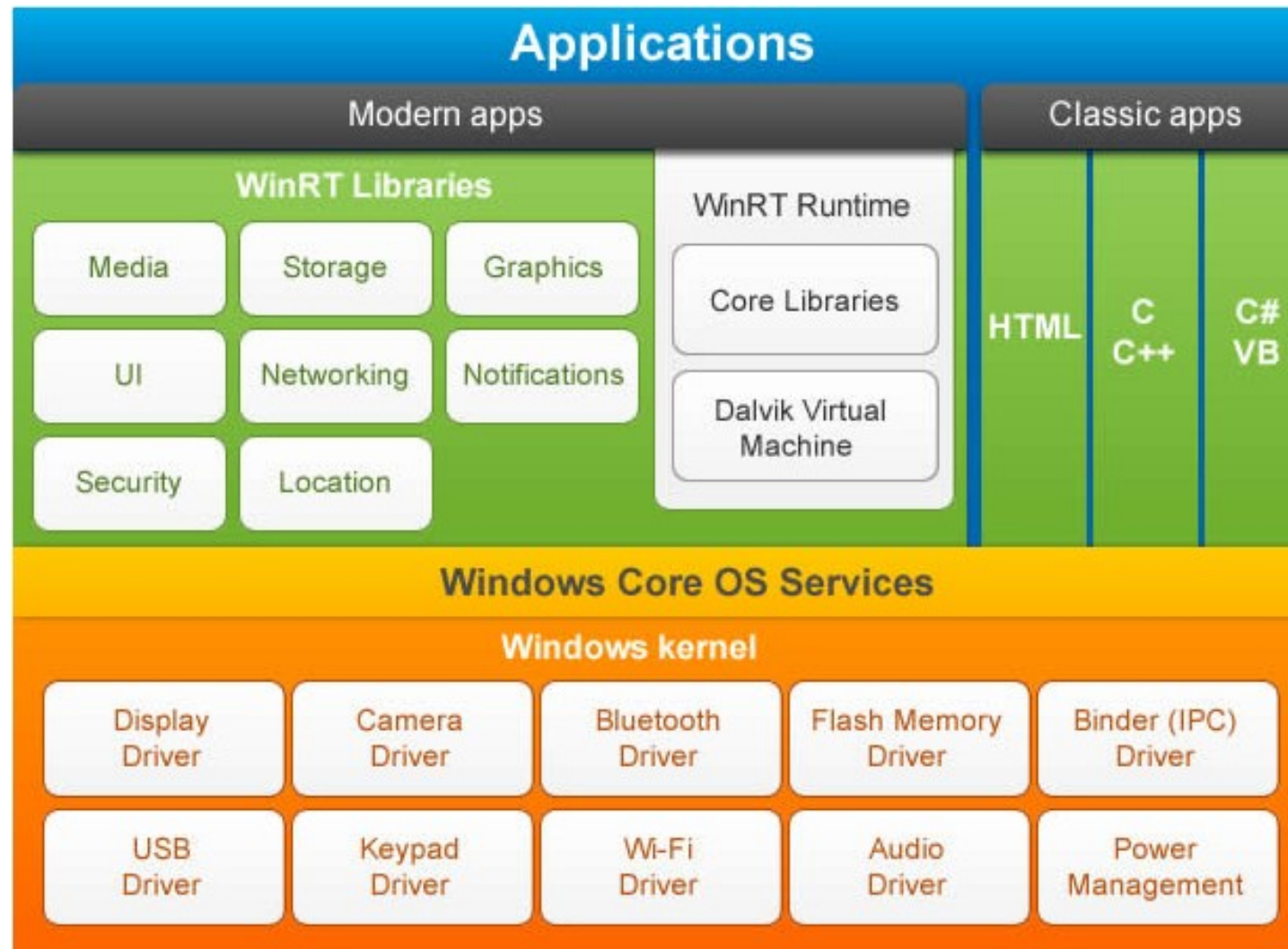
Introduction aux systèmes d'exploitation (SE)

- Autres SE: Architectures Windows XP



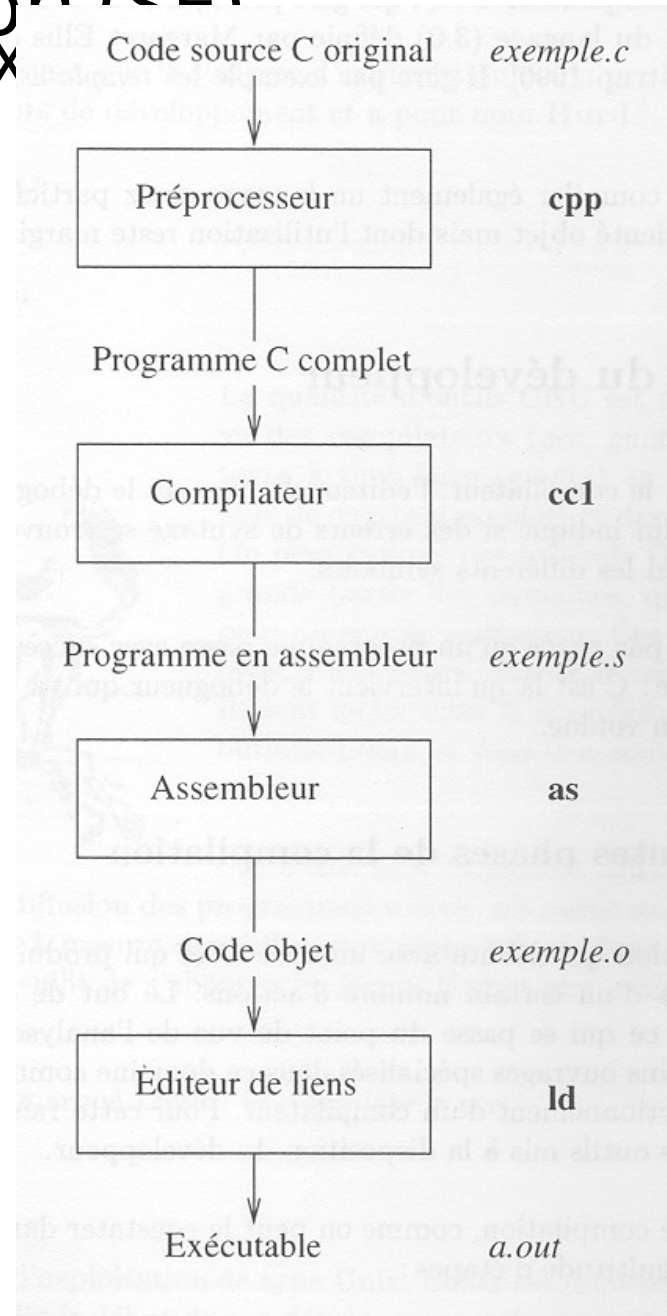
Introduction aux systèmes d'exploitation (SE)

- Autres SE: Architectures Windows 8



Introduction aux systèmes d'exploitation (CE)

- Développement sous LINUX
 - Phases de la compilation



Introduction aux systèmes d'exploitation (SE)

- Développement sous LINUX
 - Avec les distributions de LINUX, **gcc** permet d'enchaîner les phases de la compilation d'un programme
 - Options de **gcc**:
 - **-c**: générer que le code objet
 - **-o NOM_FICHIER**: générer un fichier exécutable
 - **-I REPERT**: répertoire ou **cpp** charge les fichiers en-tête
 - **-L REPERT**: répertoire ou **ld** charge les bibliothèques
 - **-l BIBLIOTHEQUE**: inclusion d'une bibliothèque lors de l'édition des liens
 - Make: outil de compilation de projets
 - Permet la compilation séparée des fichiers sources
 - Permet la compilation des fichiers sources modifiés
 - Permet de générer des fichiers exécutables

Introduction aux systèmes d'exploitation (SE)

- Environnement de travail LINUX et langage C
 - Utilitaire ***make***
 - Permet la compilation séparée des procédures et fonctions
 - Permet la création de bibliothèques
 - Permet de créer un fichier exécutable
 - Utilitaire ***make*** (utilisation)
 - Éditer le fichier ***Makefile*** dans le répertoire où sont les sources d'un programme
 - Pour lancer l'utilitaire make
 - taper ***make***

Introduction aux systèmes d'exploitation (SE)

- Environnement de travail LINUX et langage C

- Utilitaire ***make***

- Référence:

- <ftp://ftp-developpez.com/gl/tutoriel/outil/makefile/Makefile.pdf>

- Utilitaire ***make*** (exemple d'utilisation)

hello.c

```
#include <stdio.h>
#include <stdlib.h>

void Hello(void)
{
    printf("Hello World\n");
}
```

hello.h

```
#ifndef H_GL_HELLO
#define H_GL_HELLO

void Hello(void);

#endif
```

main.c

```
#include <stdio.h>
#include <stdlib.h>
#include "hello.h"

int main(void)
{
    Hello();
    return EXIT_SUCCESS;
}
```

Introduction aux systèmes d'exploitation (SE)

- Environnement de travail LINUX et langage C
 - Utilitaire ***make*** (exemple d'utilisation)
 - Format des énoncés dans le fichier ***makefile***

```
cible: dependance
commandes
```



Voir à insérer un tab au début de chaque ligne de commandes

Makefile

```
hello: hello.o main.o
    gcc -o hello hello.o main.o

hello.o: hello.c
    gcc -o hello.o -c hello.c -W -Wall -ansi -pedantic

main.o: main.c hello.h
    gcc -o main.o -c main.c -W -Wall -ansi -pedantic
```