

## Introduction aux Systèmes d'Exploitation

---



Laurent Réveillère

Enseirb-Matmeca  
Département Télécommunications

Laurent.Reveillere@ipb.fr  
<http://uuu.enseirb-matmeca.fr/~reveille/>



## Introduction aux Systèmes d'Exploitation

---

### □ Objectifs

- Présentation des principaux concepts des systèmes d'exploitation (*Operating Systems – OS*)
- Bases du système Unix
- Point de vue de l'utilisateur
  - » Introduction au Shell
  - » Initiation à la programmation Shell

### □ Organisation

- Cours
  - » Supports de cours disponibles en ligne
- Enseignements intégrés sur machine et TPs

## Ce cours a été conçu à partir de ...

- ❑ Livres (cela existe encore ...)
  - Systèmes d'exploitation – *A. Tanenbaum*
  - Le système Linux – *M. Welsh, M. Kalle Dalheimer*
- ❑ Cours en ligne de collègues
  - Neilze Dorta Santiago Pereira
    - » <http://www.mi.parisdescartes.fr/~dorta/>
  - Fabienne Boyer
    - » <http://sardes.inrialpes.fr/~boyer/cours/SE>
  - Charles Consel
    - » <http://uuu.enseirb.fr/~consel>
  - Sara Bouchenak
    - » <http://sardes.inrialpes.fr/~bouchena/teaching/SR>
  - Sacha Krakowiak
    - » <http://sardes.inrialpes.fr/~krakowia>
- ❑ Autre ressources disponibles ...

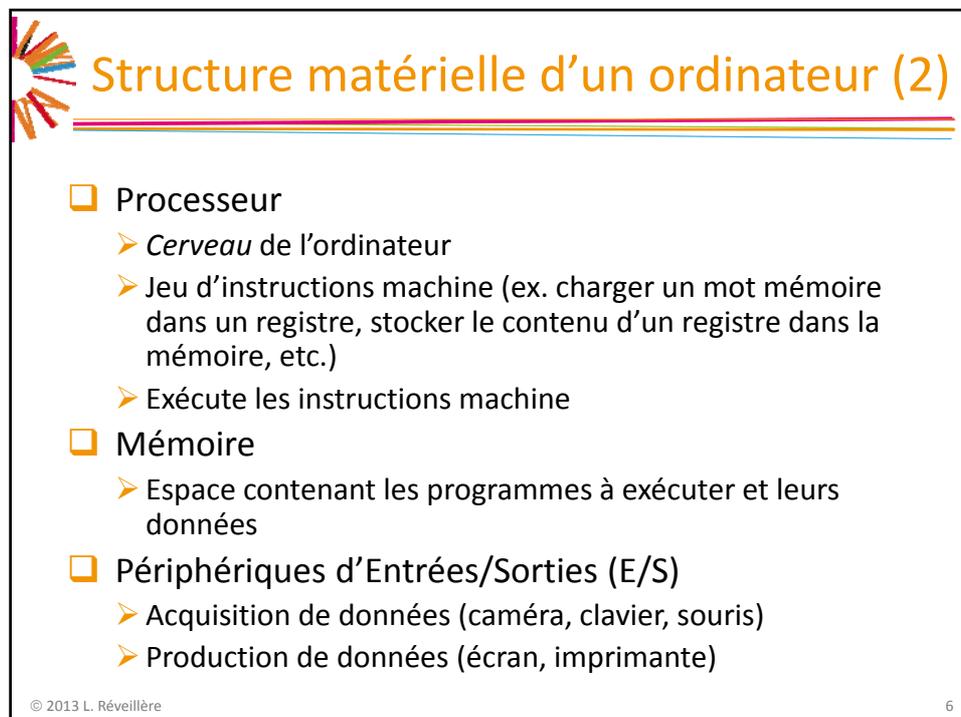
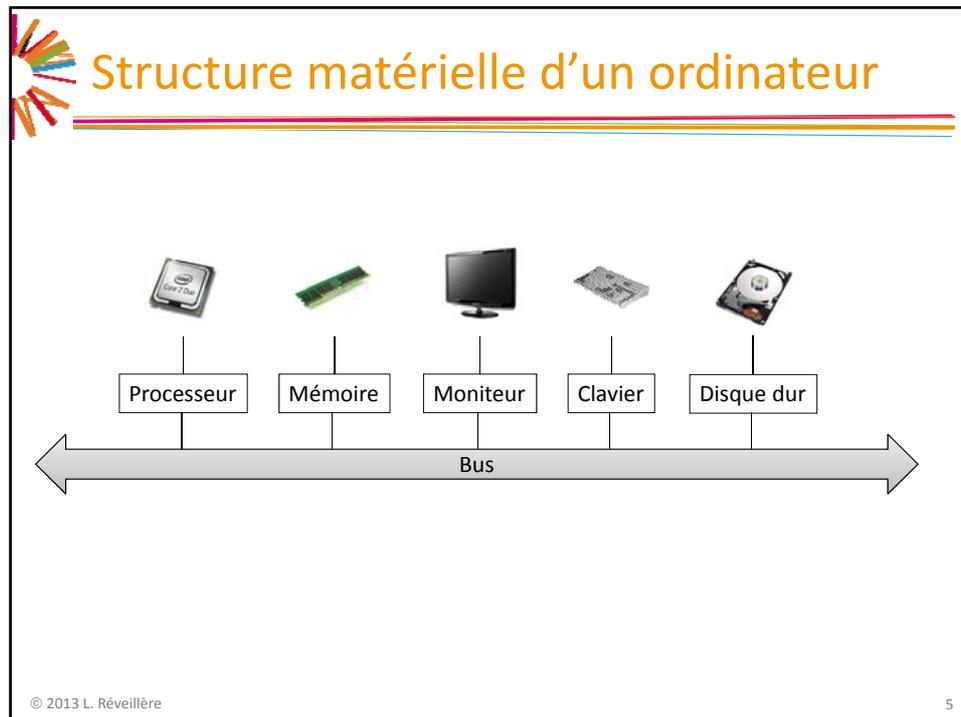


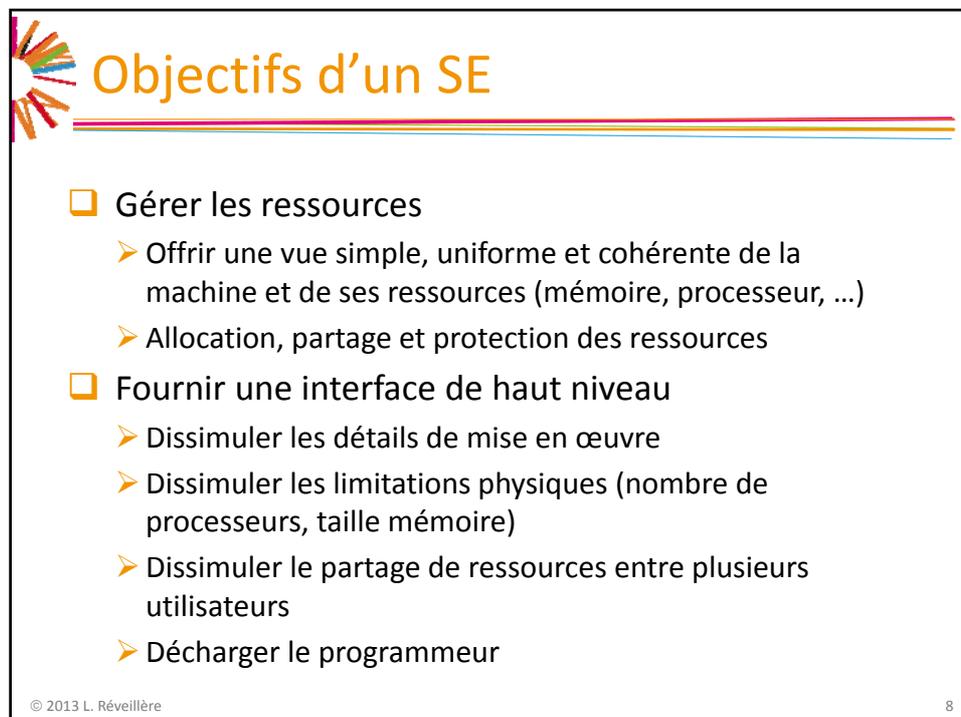
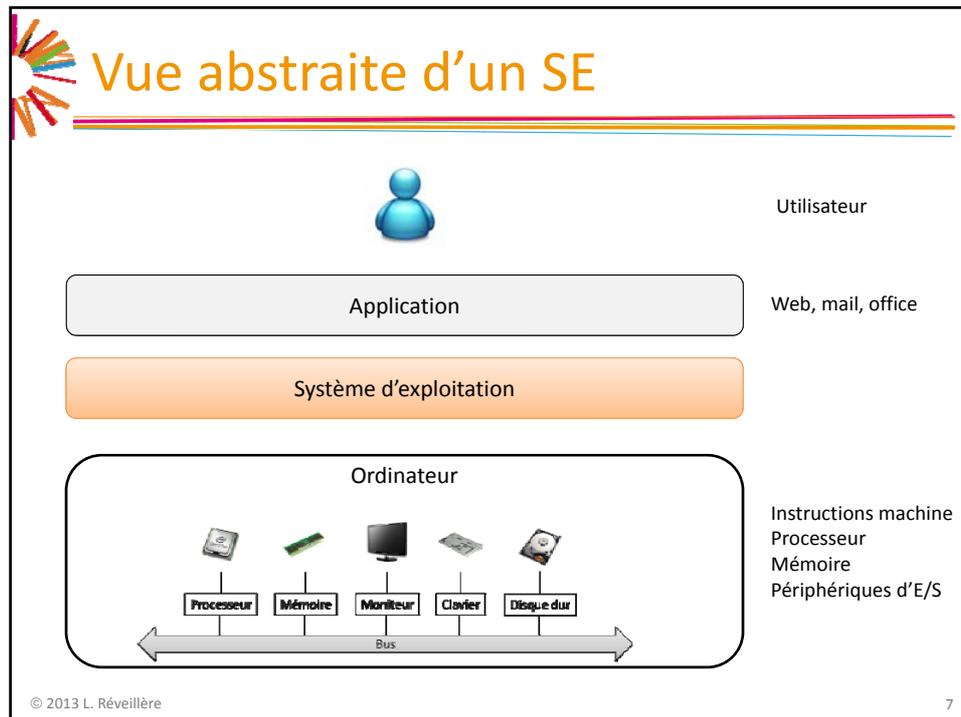
© 2013 L. Réveillère 3

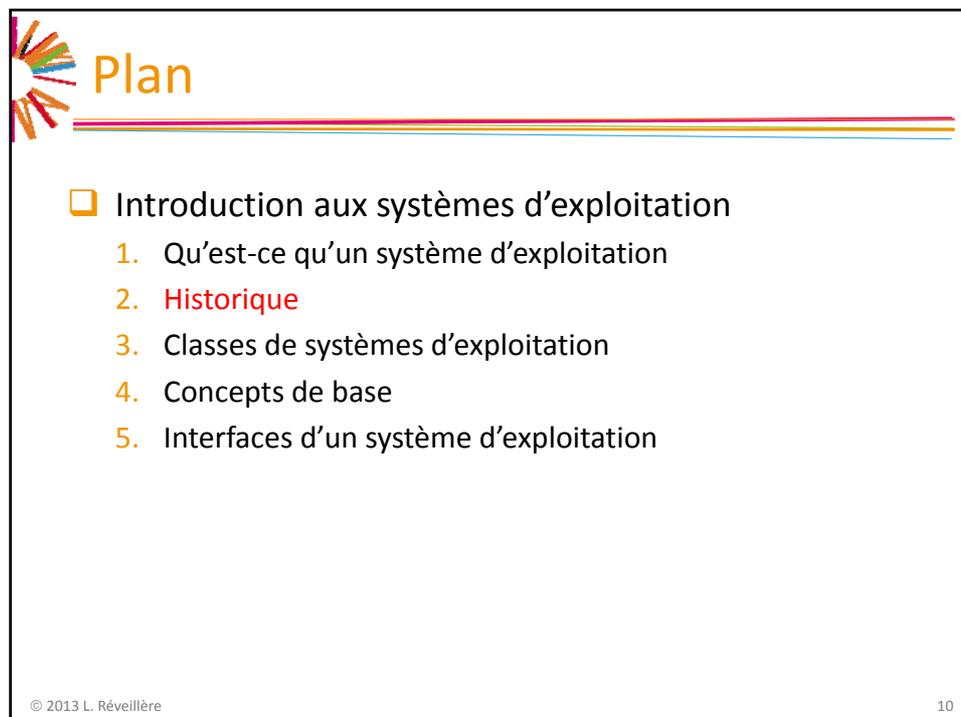
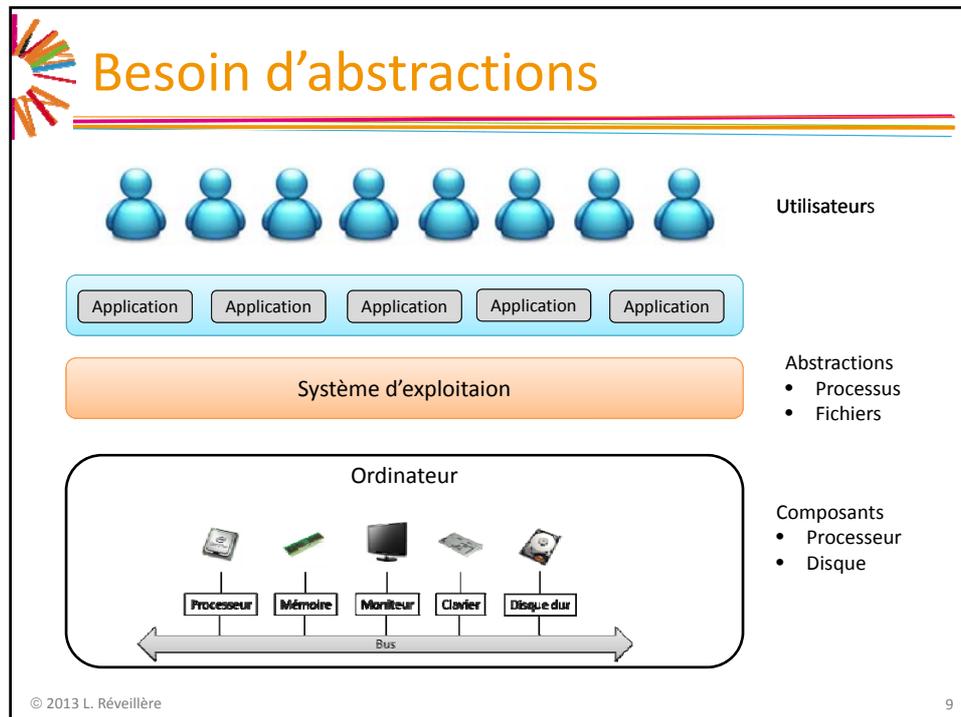
## Plan

- ❑ Introduction aux systèmes d'exploitation
  1. Qu'est-ce qu'un système d'exploitation
  2. Historique
  3. Classes de systèmes d'exploitation
  4. Concepts de base
  5. Interfaces d'un système d'exploitation

© 2013 L. Réveillère 4







## Historique des Ordinateurs et des Systèmes d'exploitation

1<sup>ère</sup> génération  
1938 - 1955

2<sup>ème</sup> génération  
1955 - 1965

3<sup>ème</sup> génération  
1965 - 1980

4<sup>ème</sup> génération  
1980 – aujourd'hui

© 2013 L. Réveillère 11

## Première génération (1938 - 1955)

- ❑ Tubes à vide et tableaux d'interrupteurs
  - Moteurs de calcul à relais mécaniques
    - » Très lents (temps de cycles mesurés en secondes)
  - Tubes à vide (lampes) à partir de 1943
    - » Relais mécaniques remplacés par des tubes à vide
    - » Machines énormes remplissant des pièces entières (dizaines de milliers de tubes à vide)




© 2013 L. Réveillère 12



## 1<sup>ère</sup> génération

---

- ❑ La même équipe concevait, construisait, programmait, administrait et maintenait la machine
- ❑ Tout programme était conçu en langage machine
  - les autres langages n'existaient pas
- ❑ Un programme était conçu en basculant des tableaux d'interrupteurs pour contrôler les fonctions de base de la machine

© 2013 L. Réveillère 13



## 1<sup>ère</sup> génération (suite)

---

- ❑ Protocole classique d'utilisation de la machine
  - réservation de la machine pour une certaine durée
  - Programmation de la machine par manipulation du tableau d'interrupteurs
  - Exécution du programmes pendant plusieurs heures
    - » Si un seul des 20.000 tubes grille → tout est à refaire
- ❑ Protocole amélioré avec les cartes perforées
  - Programme écrit avec des cartes à trous
  - Cartes lues par la machine
    - » Plus besoin de manipuler les tableaux d'interrupteurs

© 2013 L. Réveillère 14

## 1<sup>ère</sup> génération (fin)

- ❑ Types de programmes
  - Simples calculs numériques
- ❑ Et le système d'exploitation ?
  - Il n'existait pas encore

© 2013 L. Réveillère 15

## Deuxième génération (1955 - 1965)

- ❑ Transistors et systèmes par lots
  - Transistors
    - » Ordinateurs plus fiables
    - » Séparation nette entre concepteurs, constructeurs, programmeurs, opérateurs et personnel de maintenance
- ❑ Les programmes étaient écrits en FORTRAN puis codés sur des cartes perforée

```

C AREA OF A TRIANGLE - HERON'S FORMULA
C INPUT - CARD READER UNIT 5, INTEGER INPUT
C OUTPUT - LINE PRINTER UNIT 6, REAL OUTPUT
C INPUT ERROR MESSAGE OUTPUT CODE 1 IN JOB
CONTROL LISTING
INTEGER A,B,C
READ(5,501) A,B,C
501 FORMAT(3I5)
IF(A<0 .OR. B<0 .OR. C<0) STOP 1
S = (A + B + C) / 2.0
AREA = SQRT( S * (S - A) * (S - B) * (S - C))
WRITE(6,601) A,B,C,AREA
601 FORMAT(4E15.5E) B= .15,SH C= .15,SH
AREA= .15,SH C= .15,SH
STOP
END

```

© 2013 L. Réveillère 16



## 2<sup>ème</sup> génération

---

- ❑ Protocole d'utilisation de la machine
  - Programmeur fournit son paquet de cartes perforées
  - L'opérateur fait lire et exécuter les cartes par la machine
  - L'opérateur récupère la trace d'exécution sur une imprimante
  - Si le compilateur FORTRAN est nécessaire
    - » Besoin de le charger dans la machine

© 2013 L. Réveillère 17



## 2<sup>ème</sup> génération (suite)

---

- ❑ Protocole lent
  - Un opérateur humain traite séquentiellement un job et gère la soumission :
    - » des entrée (lecture de cartes perforée)
    - » des sorties (résultat sur imprimante) une fois le calcul terminé
- ❑ Amélioration du protocole
  - Découpage du traitement d'un *job* en plusieurs étapes
    - » Lecture des entrées, calcul, production des sorties
    - » Entrées/sorties d'un job traitées par une machine peu onéreuse et performante pour les opération d'E/S
    - » Calcul effectué par une machine performante (calculateur)

© 2013 L. Réveillère 18



## 2<sup>ème</sup> génération (fin)

---

- ❑ Types de programmes
  - Calculs scientifiques et d'ingénierie
    - » résolution d'équations aux dérivées partielles
  - Programmés en FORTRAN ou en langage assembleur
  
- ❑ Système d'exploitation
  - FMS (*Fortran Monitor System*)
  - IBYS (1<sup>er</sup> système d'exploitation de l'IBM 7094)

© 2013 L. Réveillère 19



## Troisième génération (1965 - 1980)

---

- ❑ Circuits intégrés
  - Avantage prix/performance
  
- ❑ Une seule gamme de produits
  - Début 60
    - » machines spécialisées dans les E/S (IBM 1401)
    - » machines spécialisées dans le calcul (IBM 7094)
  - Puis une série de machines (taille entre l'IBM 1401 et l'IBM 7094) avec la même architecture matérielle et le même jeu d'instructions, ex. system/360
  - Un même programme peut en principe tourner sur toutes ces machines

© 2013 L. Réveillère 20



## 3<sup>ème</sup> génération

---

- ❑ Système d'exploitation
  - OS/360
    - » devait tourner sur tous les modèles (du plus petit au plus grand)
- ❑ Nouvelles techniques dans les systèmes d'exploitation
  - Multiprogrammation
    - » Plusieurs programmes pouvant s'exécuter en parallèle sur une même machine
    - » Mémoire partagée
    - » Mécanismes de protection
  - Spool (*Simultaneous Peripheral Operation On Line*)
  - Lectures et écritures simultanées sur le disque
  - IBM 1401 n'est plus nécessaire
  - Manipulations de bandes ont disparu

© 2013 L. Réveillère 21



## 3<sup>ème</sup> génération (fin)

---

### Exemples de systèmes d'exploitation

- ❑ MULTICS : 1964
  - MIT + Bell Labs + General Electric
  - Système capable de supporter des centaines d'utilisateurs simultanés
- ❑ Systèmes pour mini-ordinateurs, DEC PDP
  - PDP-1 ... PDP-11
  - Peu d'espace mémoire (4 K-mots de 18 bits)
  - Performants pour des tâches non scientifiques
- ❑ UNIX : 1969
  - Ken Thompson de Bell Labs, qui a travaillé sur MULTICS, a écrit une version allégée de MULTICS pour un PDP-7
  - Norme POSIX : standardisation des API de logiciels pour systèmes UNIX

© 2013 L. Réveillère 22



## Quatrième génération (1980 - aujourd'hui)

- ❑ Ordinateurs personnels (PC : *Personal Computer*)
- ❑ Circuits intégrés à haute densité
  - Puces contenant des milliers de transistors sur 1mm<sup>2</sup> de silicium
  - Micro-ordinateurs, très peu onéreux comparés aux mini-ordinateurs de type PDP-11

© 2013 L. Réveillère 23



## 4<sup>ème</sup> génération

- ❑ Exemple de systèmes d'exploitation
  - CP/M (Control Program for Microprocessors)
  - MS-DOS (MicroSoft-Disk Operating System)
  - Macintosh d'Apple
  - Windows de MicroSoft
  - UNIX
  - Linux

© 2013 L. Réveillère 24



## 4<sup>ème</sup> génération - CP/M

---

- ❑ Développé en 1973 par Gary A. Kildall
  - Consultant pour Intel, auteur du langage PL/M pour l'Intel 8080
  - CP/M intègre un contrôleur pour le tout récent lecteur de disquettes 8 pouces
  - Intel pense que les  $\mu$ -ordinateurs équipés de disque ont peu de chance de prospérer
    - » cède à G. Kildall les droits du CP/M
    - » G. Kildall fonde Digital Research en 1976
- ❑ 1985
  - » Dominait le marché → installé sur environ 4 millions de postes

© 2013 L. Réveillère 25



## 4<sup>ème</sup> génération – MS-DOS

---

- ❑ En 1980, IBM propose l'IBM PC
  - IBM demande à Digital Research qui refuse
- ❑ IBM demande MicroSoft (B. Gates) de lui fournir un système d'exploitation
  - B. Gates achète QDOS à un petit constructeur de Seattle
  - QDOS est un clone à bas prix de CP/M
  - QDOS est adapté à l'IBM PC et renommé en MS-DOS
- ❑ B. Gates décide de vendre MS-DOS aux constructeurs
  - G. Kildall (Digital Research) vendait CP/M aux utilisateurs

© 2013 L. Réveillère 26



## 4<sup>ème</sup> génération – Macintosh d'Apple

---

- ❑ Xerox
  - Invente le concept d'IHM graphique
  - Steve Jobs (co-inventeur d'Apple) visite un centre de recherche Xerox
- ❑ 1983 - Lisa
  - Premier PC à posséder une souris et une interface graphique
  - Trop cher → échec commercial
  - Caractéristiques
    - » Prix: 9 995\$, proc Motorola 68000 à 5MHz, 1Mo RAM, DD externe de 5Mo, 2 FD de 5,25" de 875Ko.
- ❑ 1984 - Macintosh
  - Beaucoup moins cher, très convivial → succès commercial
  - Destiné à des utilisateurs lambda
  - Version allégé du Lisa (128Ko de RAM, pas de multitâche, ...)

© 2013 L. Réveillère 27



## 4<sup>ème</sup> génération – fin

---

- ❑ Windows de MicroSoft
  - Successeur de MS-DOS
  - Intègre une IHM graphique (influencé par le mac)
  - De nombreuses versions et évolutions
- ❑ UNIX
  - Intégration d'IHM graphique
  - Système de fenêtres appelé X-Window (MIT) - 1984
  - IHM complète telle que Motif au-dessus de X-Window

© 2013 L. Réveillère 28





## Unités de mesures

---

Multiples d'octets tels que définis par IEC 60027-2					
Préfixe SI			Préfixe binaire		
Nom	Symbole	Valeur	Nom	Symbole	Valeur
kiloctet	ko	$10^3$	kibioctet	Kio	$2^{10}$
mégaoctet	Mo	$10^6$	mébioctet	Mio	$2^{20}$
gigaoctet	Go	$10^9$	gibioctet	Gio	$2^{30}$
téraoctet	To	$10^{12}$	tébioctet	Tio	$2^{40}$
pétaoctet	Po	$10^{15}$	pébioctet	Pio	$2^{50}$
exaoctet	Eo	$10^{18}$	exbioctet	Eio	$2^{60}$
zettaoctet	Zo	$10^{21}$	zébioctet	Zio	$2^{70}$
yottaoctet	Yo	$10^{24}$	yobioctet	Yio	$2^{80}$

Multiples d'octets Usage traditionnel		
Nom	Symbole	Valeur
kilo-octet	ko	$2^{10}$
méga-octet	Mo	$2^{20}$
giga-octet	Go	$2^{30}$

© 2013 L. Réveillère 31



## Plan

---

- Introduction aux systèmes d'exploitation
  1. Qu'est-ce qu'un système d'exploitation
  2. Historique
  3. Classes de systèmes d'exploitation
  4. Concepts de base
  5. Interfaces d'un système d'exploitation

© 2013 L. Réveillère 32



## Classes de SE

---

- ❑ Systèmes temps réel
- ❑ Systèmes mobiles
- ❑ Systèmes parallèles multiprocesseurs
- ❑ Systèmes clusterisés
- ❑ Grilles de calculs
- ❑ Cloud computing (*Informatique en nuage*)

© 2013 L. Réveillère 33



## Systèmes temps réel

---

- ❑ Contraintes temporelles
  - Temps d'exécution bornés
- ❑ Utilisation dans des situations spécifiques
  - Systèmes d'imagerie médicale
  - Systèmes de contrôle industriel
  - Robotique
  - Aéronautique
  - ...

© 2013 L. Réveillère 34



## Systèmes temps réel

---

- ❑ Temps réel dur (*Hard real-time*)
  - Peu ou pas de mémoire secondaire
  - Pas de temps-partagé
  - Temps de commutation très court
  - SE complètement spécifique
  
- ❑ Temps réel mou (*Soft real-time*)
  - Utilisés dans les applications multimédia, réalité virtuelle
  - Contraintes de temps plus souples
  - Gestion de priorités de tâches
  - Gestion spécifique de la mémoire

© 2013 L. Réveillère 35



## Système mobiles

---

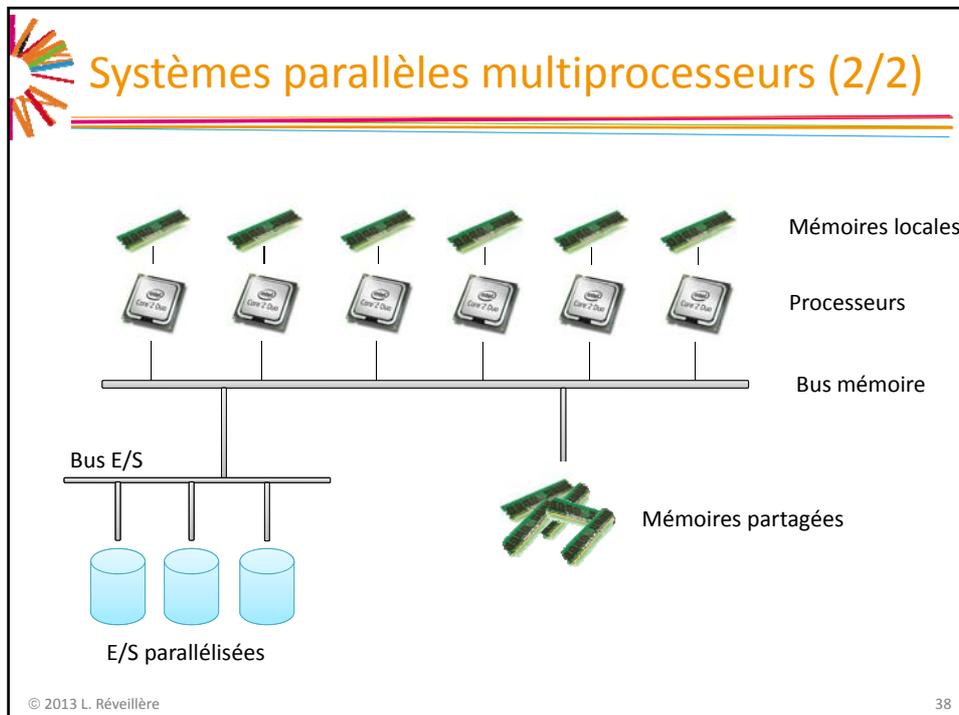
- ❑ Téléphones mobiles
  - android, iOS, Windows phone, ...
- ❑ Systèmes d'exploitation spécifiques
- ❑ Contraintes
  - Mémoire limitée
  - Processeurs lents
  - Petits écrans
  - Faible consommation d'énergie
  - Réactivité

© 2013 L. Réveillère 36

## Systèmes parallèles multiprocesseurs (1/2)

- ❑ 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)

© 2013 L. Réveillère 37





## Systèmes clusterisés

---

- ❑ Autres appellations
  - Grappe de serveurs
  - Ferme de calcul
- ❑ Plusieurs nœuds
  - Homogènes
  - Partageant un disque (*share disk*) ou bien ne partageant rien (*share nothing*)
- ❑ Connectés par un réseau spécialisé
  - SCI, InfiniBand , Myrinet ...
  - Echanges rapides de données de grandes taille
- ❑ Trois caractéristiques :
  - Montée en charge
  - Passage à l'échelle (*scalability*)
  - Haute disponibilité (serveurs back-up)

© 2013 L. Réveillère 39



## Grilles de calculs

---

- ❑ Infrastructures réparties composées d'un ensemble de machines hétérogènes et potentiellement distantes
- ❑ Exemple
  - Grid5000
  - Environ 3000 processeurs sur 9 sites en france
- ❑ Calcul intensif
- ❑ Garanties de disponibilité
- ❑ Système de réservation des ressources matérielles

© 2013 L. Réveillère 40



## Cloud computing

- ❑ Services déportés sur un ou plusieurs serveurs distants
- ❑ Les services proposés par le Cloud peuvent s'exécuter au sein d'architectures cluster
- ❑ Trois modèles principaux
  - IaaS (Infrastructure as a Service)
  - PaaS (Platform as a Service)
  - SaaS (Software as a Service)

© 2013 L. Réveillère 41



## Plan

- ❑ Introduction aux systèmes d'exploitation
  1. Qu'est-ce qu'un système d'exploitation
  2. Historique
  3. Classes de systèmes d'exploitation
  4. **Concepts de base**
  5. Interfaces d'un système d'exploitation

© 2013 L. Réveillère 42



## Concepts de base des systèmes d'exploitation

---

- ❑ Processus
- ❑ Interblocage
- ❑ Gestion de la mémoire
- ❑ Entrées/sorties
- ❑ Fichiers

© 2013 L. Réveillère 43



## Processus

---

- ❑ Processus = programme en cours d'exécution
- ❑ Chaque processus est pourvu d'un espace d'adressage en mémoire et d'un ensemble de registres
- ❑ Cet espace contient :
  - programme exécutable
  - données
  - pile d'exécution
- ❑ Ex. de registres :
  - compteur ordinal, sommet de pile

© 2013 L. Réveillère 44



## Interblocage entre processus

- ❑ Interblocage = deux ou plusieurs processus se retrouvent dans une situation dont ils ne peuvent plus s'échapper
- ❑ Exemple d'interblocage entre deux processus
 

P1	P2
ouvrir (F, écriture)	ouvrir (G, écriture)
ouvrir (G, écriture)	ouvrir (F, écriture)
...	...
fermer (F)	fermer (G)
fermer (G)	fermer (F)
- ❑ Interblocage : si on alloue F à P1, puis G à P2, alors les deux processus sont bloqués.

© 2013 L. Réveillère 45



## Gestion de la mémoire

- ❑ Tout ordinateur possède une mémoire principale qui contient le programme en cours d'exécution
- ❑ Monoprogrammation (cas simple)
  - Seul un programme à la fois peut se trouver en mémoire
  - Pour exécuter un second programme, on doit d'abord décharger le 1er programme de la mémoire puis charger le second

© 2013 L. Réveillère 46



## Gestion de la mémoire (2)

---

- ❑ Multiprogrammation (cas courant)
  - Plusieurs programmes peuvent cohabiter en même temps en mémoire
  - Mécanisme de protection qui empêche deux programmes d'interférer entre eux
  - Mécanisme de gestion de l'espace d'adressage
    - » Un processus possède un espace d'adressage supérieur à celui de la mémoire principale (mémoire virtuelle)
    - » Une partie de l'espace d'adressage en mémoire principale et une partie sur le disque
      - Système d'exploitation: gestion des allers-retours entre les deux

© 2013 L. Réveillère 47



## Entrées/sorties

---

- ❑ Les ordinateurs sont munis de périphériques
  - acquisition de données en entrée
    - » Clavier, souris
  - production de données en sorties
    - » Écran, imprimante
- ❑ Le système d'exploitation possède un sous-système d'entrées/sorties responsable de la gestion des périphériques

© 2013 L. Réveillère 48



## Entrées/sorties (2)

---

- ❑ Le sous-système d'entrées/sorties dans un système d'exploitation contient :
  - Une partie générique, indépendante du périphérique et s'appliquant donc à plusieurs périphériques
  - Une partie spécifique à des périphériques particuliers
    - » Exemple: le pilote (driver) d'une imprimante HP Laser Jet

© 2013 L. Réveillère 49



## Fichiers

---

- ❑ Les fichiers sont un mécanisme d'abstraction (ou de virtualisation) du disque
- ❑ Les fichiers permettent d'écrire/lire des données sur un disque
- ❑ L'utilisateur ne voit pas :
  - comment sont stockées les données
  - où sont stockées les données
  - comment les disques fonctionnent

© 2013 L. Réveillère 50



## Fichiers (2)

---

- ❑ Exemple d'abstraction fournie par les fichiers
  - Lecture d'un fichier vs. localisation d'un bloc sur un secteur sur le disque
- ❑ Système de gestion de fichiers (SGF) = sous-système d'un système d'exploitation
  - Exemple: NFS (*Network File System*) dans UNIX

© 2013 L. Réveillère 51



## Plan

---

- ❑ Introduction aux systèmes d'exploitation
  1. Qu'est-ce qu'un système d'exploitation
  2. Historique
  3. Classes de systems d'exploitation
  4. Concepts de base
  5. Interfaces d'un système d'exploitation

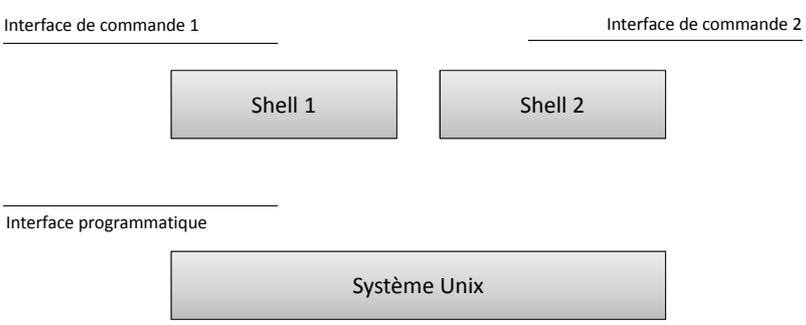
© 2013 L. Réveillère 52

## Interfaces d'un système d'exploitation

- ❑ L'interface d'un système d'exploitation permet aux utilisateurs du système de faire appel aux fonctions du système
  
- ❑ Un système d'exploitation présente en général deux interfaces
  - Une interface programmatique
  - Une interface de commande

© 2013 L. Réveillère 53

## Types d'interfaces ...



```
graph TD; I1[Interface de commande 1] --- S1[Shell 1]; I2[Interface de commande 2] --- S2[Shell 2]; IP[Interface programmatique] --- SU[Système Unix];
```

© 2013 L. Réveillère 54



## Interfaces d'un système d'exploitation (2)

---

- ❑ Interface programmatique (API)
  - Dédicée aux programmes s'exécutant au-dessus d'un système d'exploitation
  - Composée d'un ensemble de fonctions système avec leurs paramètres
- ❑ Interface de commande
  - Dédicée aux utilisateurs humains d'un système d'exploitation
  - Composée d'un ensemble de commandes
    - » Commandes textuelles (shell)
    - » Commandes graphiques (glisser-déplacer de fichiers)

© 2013 L. Réveillère 55



## Interface de commandes dans Unix – shell

---

- ❑ Le shell est un programme qui interprète des commandes
  - Entrée standard du shell : clavier
  - Sortie standard du shell : écran
- ❑ Exemple de commande de shell
  - Commande date affiche la date courante à l'écran  
\$ date

© 2013 L. Réveillère 56