

# Introduction aux systèmes d'exploitation



UNIX PEOPLE ARE HAPPY

GIF-1001 Ordinateurs: Structure et Applications, Hiver 2016  
Jean-François Lalonde

# Questions

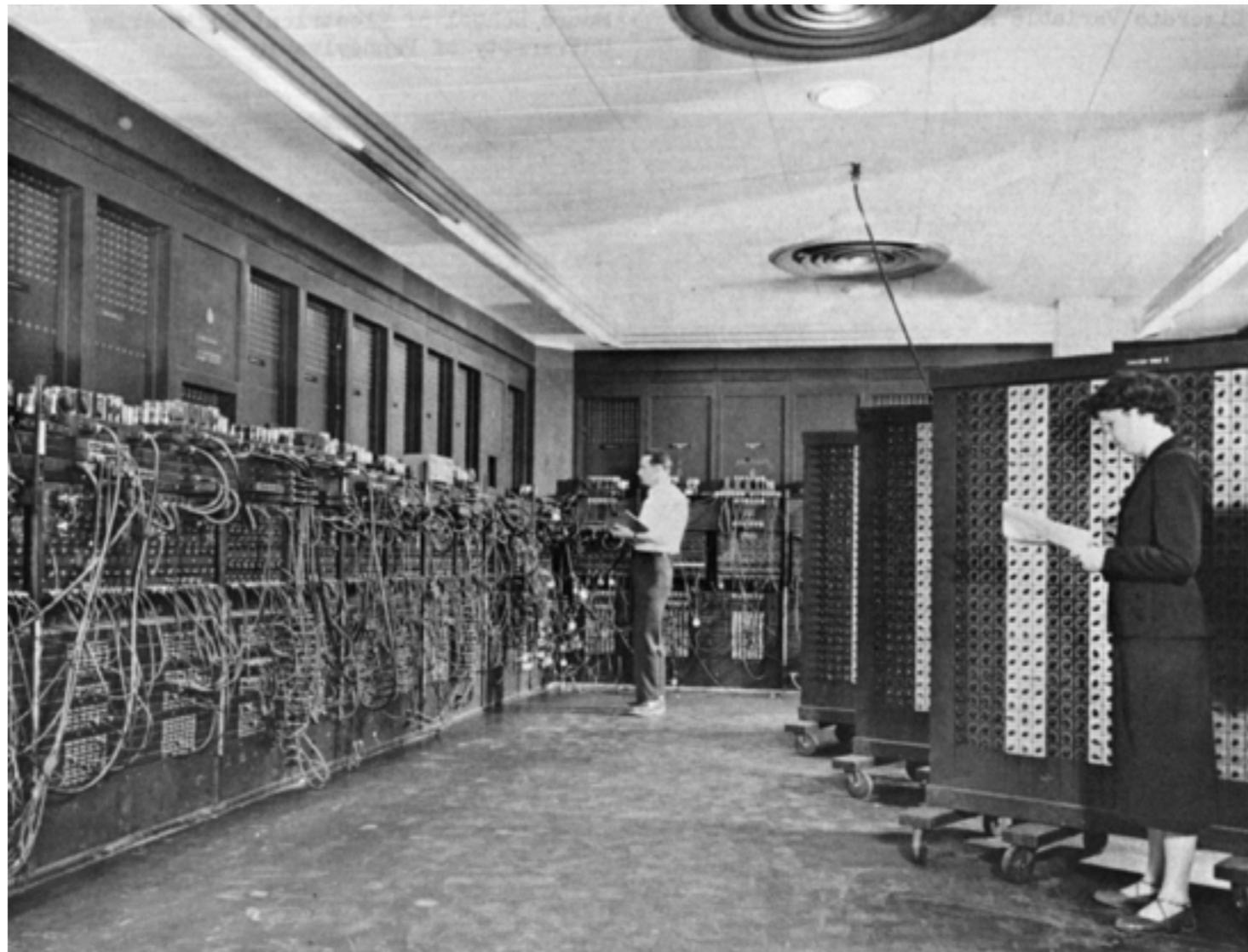
- Comment le programme a-t-il été chargé en mémoire?
- Comment faire pour dire à l'ordinateur de débiter l'exécution d'un programme?
- Comment peut-on interagir avec un programme?
- Que doit-on faire quand un programme est terminé?
- Comment peut-on exécuter plus d'un programme?

# ENIAC (Maulchy & Eckert, 1946)

calculs de balistique durant la 2e Guerre Mondiale

les panneaux à gauche étaient les programmes: il fallait programmer manuellement en branchant et débranchant ces connexions

18,000 tubes à vide, 15,000 pieds carrés, 30 tonnes, 140 kwatts puissance



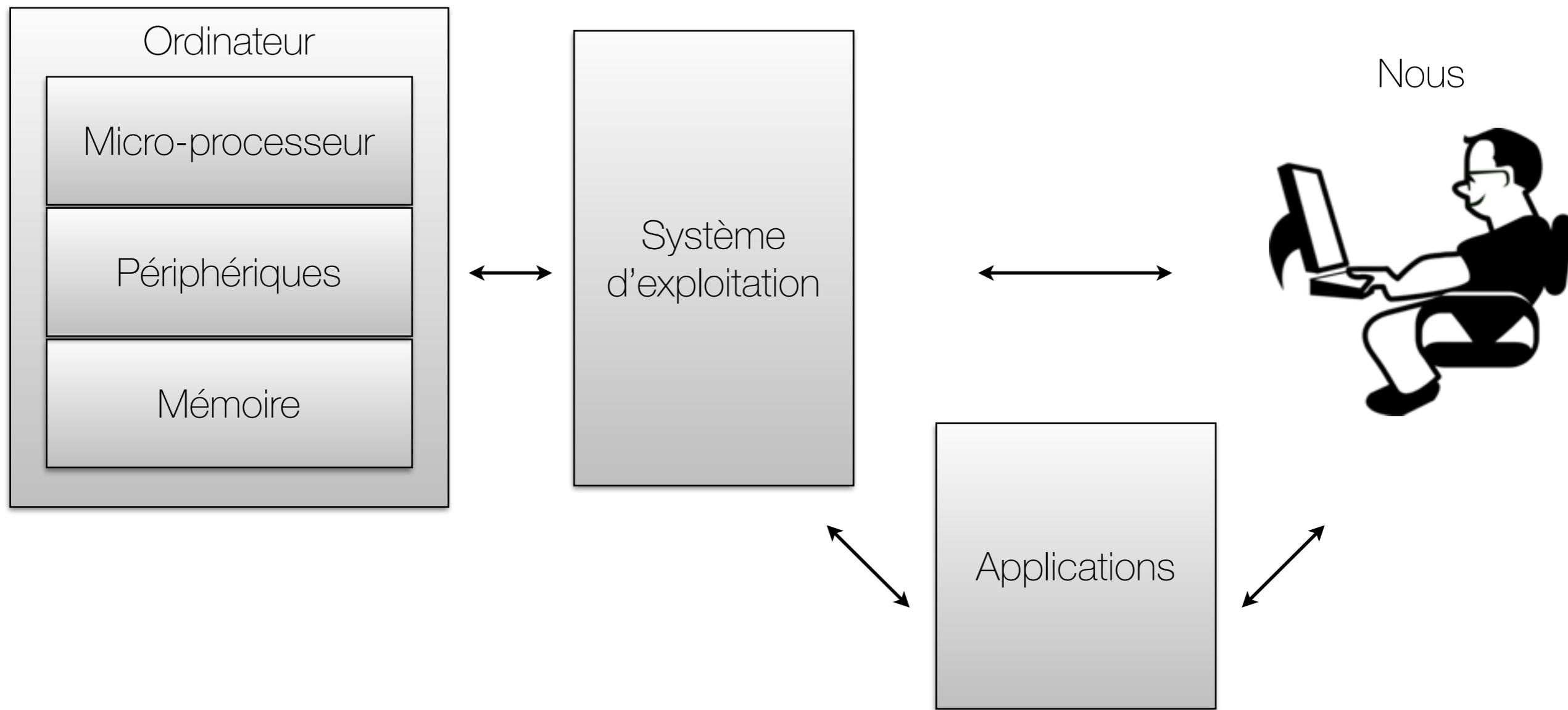
# Les premiers ordinateurs

- Les premiers ordinateurs étaient constitués d'un microprocesseur, une mémoire, un périphérique d'entrée, quelques voyants/bascules et une imprimante...
- Le programmeur introduisait le programme à exécuter dans la mémoire du microprocesseur à partir du périphérique d'entrée (lire lecteur de carte perforée...). Le microprocesseur était déconnecté pendant ce temps!
- Des bascules permettaient de lancer l'exécution du programme par le microprocesseur.
- Des voyants donnaient les statuts d'erreurs et les résultats du programme. En cas d'erreur, le programmeur regardait directement la mémoire...
- Un seul programme était exécuté à la fois. Il fallait s'inscrire sur des listes afin de pouvoir utiliser l'ordinateur.

# Systeme d'exploitation (SE) — définition générale

- Un système d'exploitation est
  - un ensemble de programmes
  - qui gèrent les ressources matérielles d'un ordinateur et qui rendent ces ressources disponibles à l'utilisateur,
  - de telle sorte que l'utilisateur peut utiliser efficacement son ordinateur.

# Systeme d'exploitation



# Analogies



- Un illusionniste :
  - Fait disparaître certaines limites du matériels
  - Donne l'illusion que la machine a une mémoire infinie et une infinité de processeurs



- Un gouvernement :
  - Protège les utilisateurs les uns des autres
  - Partage des ressources de façon efficace et équitable

# Restrictions vs. possibilités

- S.E. est l'équivalent de vivre en société
  - Partage des ressources, incluant CPU
  - Règlement à suivre : perte de liberté
  - Augmente les possibilités



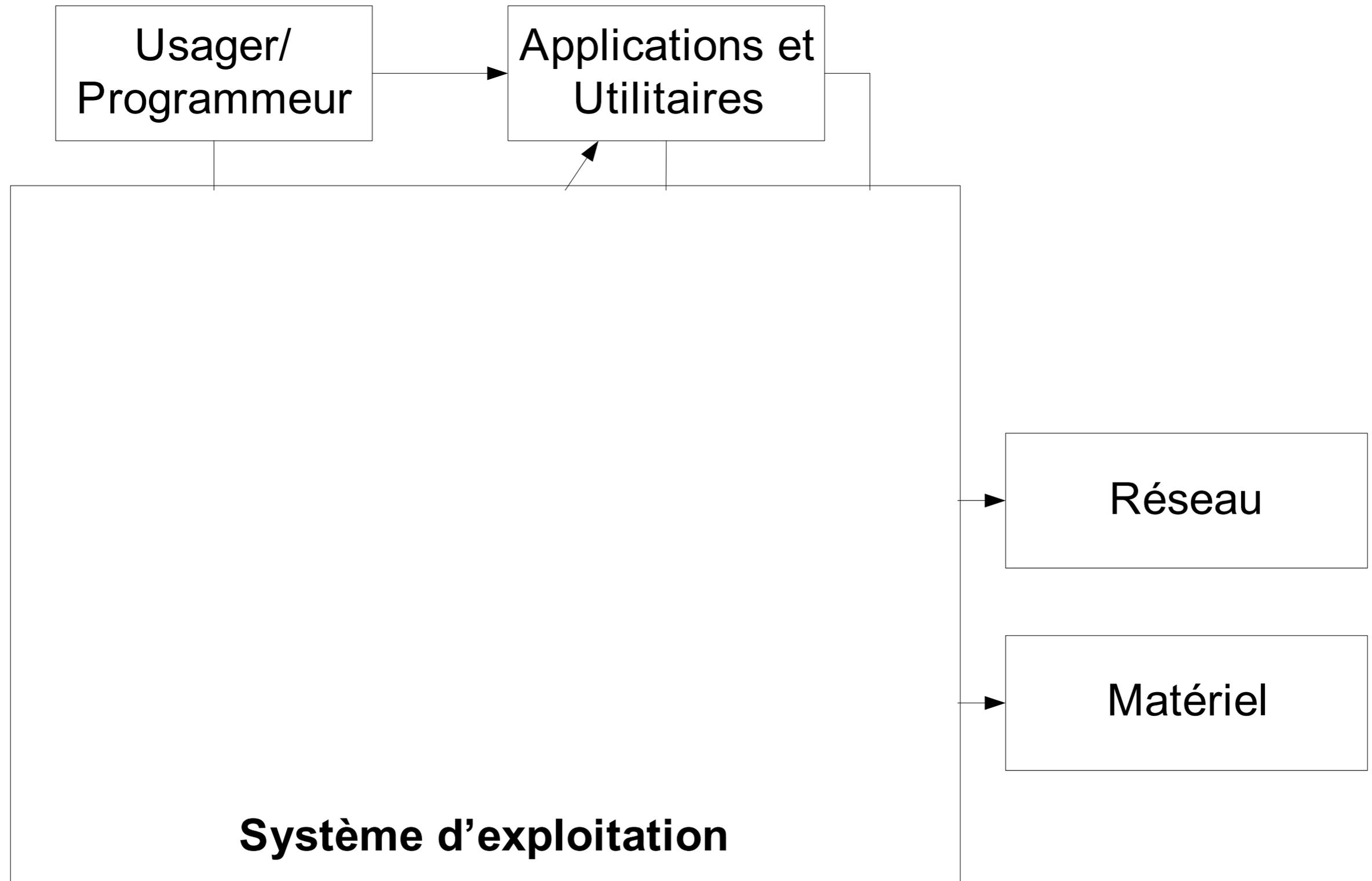
# Rôles du système d'exploitation

- Les principaux rôles d'un système d'exploitation sont:
  - Fournir une interface usager conviviale.
  - Gérer, lire et exécuter des programmes.
  - Gérer les ressources matérielles de l'ordinateur.
- Le but d'un système d'exploitation est de faciliter l'utilisation de l'ordinateur et de rendre son utilisation efficace.

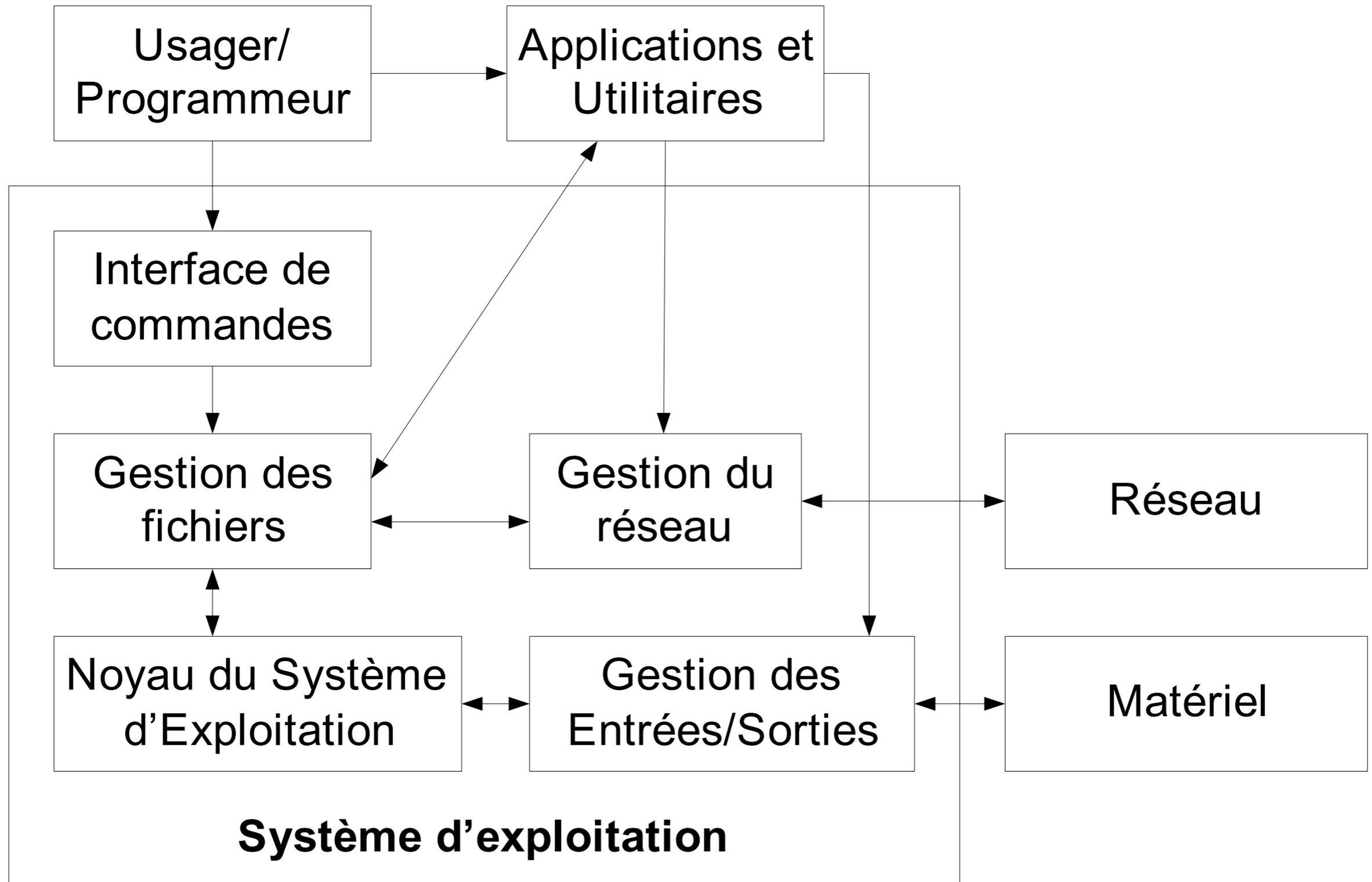
# Le système d'exploitation: un programme

- Le système d'exploitation est un programme.
  - Il occupe une partie de la mémoire de l'ordinateur.
  - Il est chargé en mémoire par le BIOS.
  - Le rôle du système d'exploitation est de lancer d'autres programmes. Il doit « abandonner » le contrôle du microprocesseur aux autres programmes pour que ceux-ci puissent être exécutés.

# Vision simplifiée des services d'un SE



# Vision simplifiée des services d'un SE



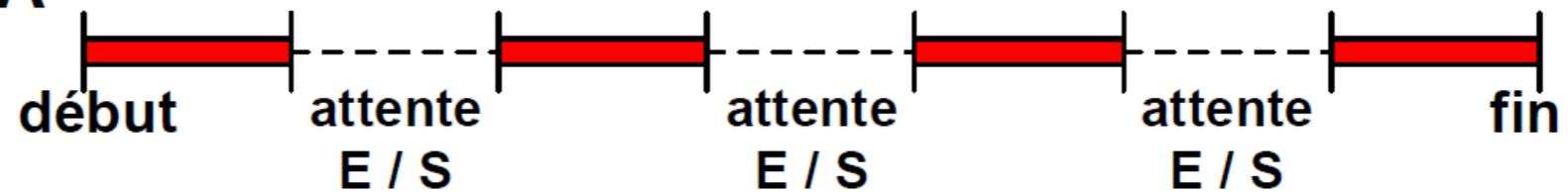
# Vision simplifiée des services d'un SE

- Un système d'exploitation doit fournir les services suivants:
  - Une interface de commandes pour l'utilisateur
  - Un support pour la création de programmes
  - Une gestion des fichiers
  - Une gestion des programmes et applications
  - Une façon de poursuivre le démarrage de l'ordinateur amorcé par le BIOS
  - Un service des interruptions (Software et Hardware)
  - Une gestion des entrées/sortie et du matériel qui y est relié.
  - Des services pour les réseaux et pour le matériel qui y est associé.
  - Une mesure des statistiques et performances de l'ordinateur.

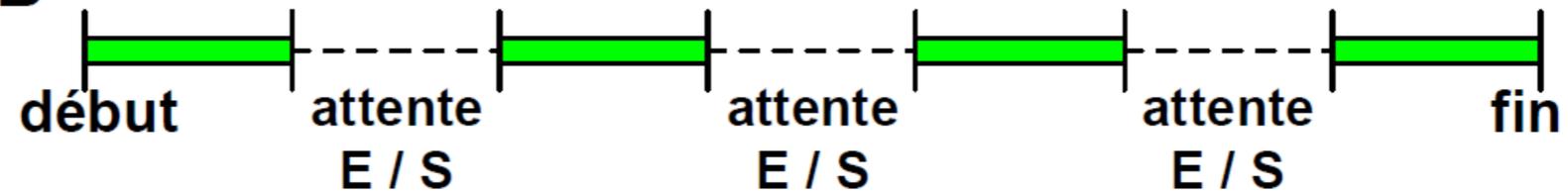
# Processus

- E/S = entrée-sortie
  - ex: lecture ou écriture sur le disque

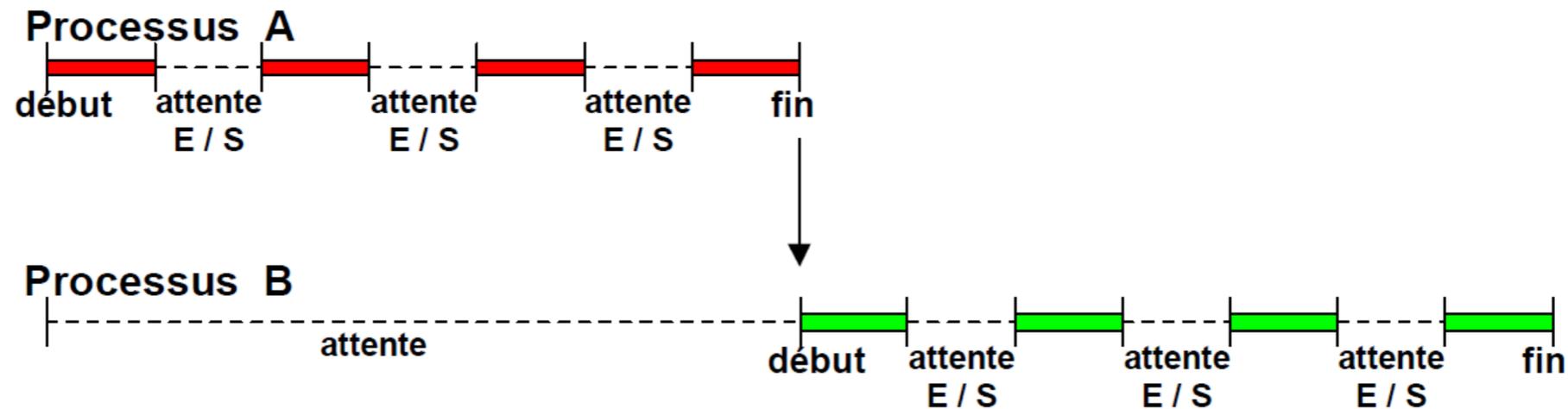
## Processus A



## Processus B

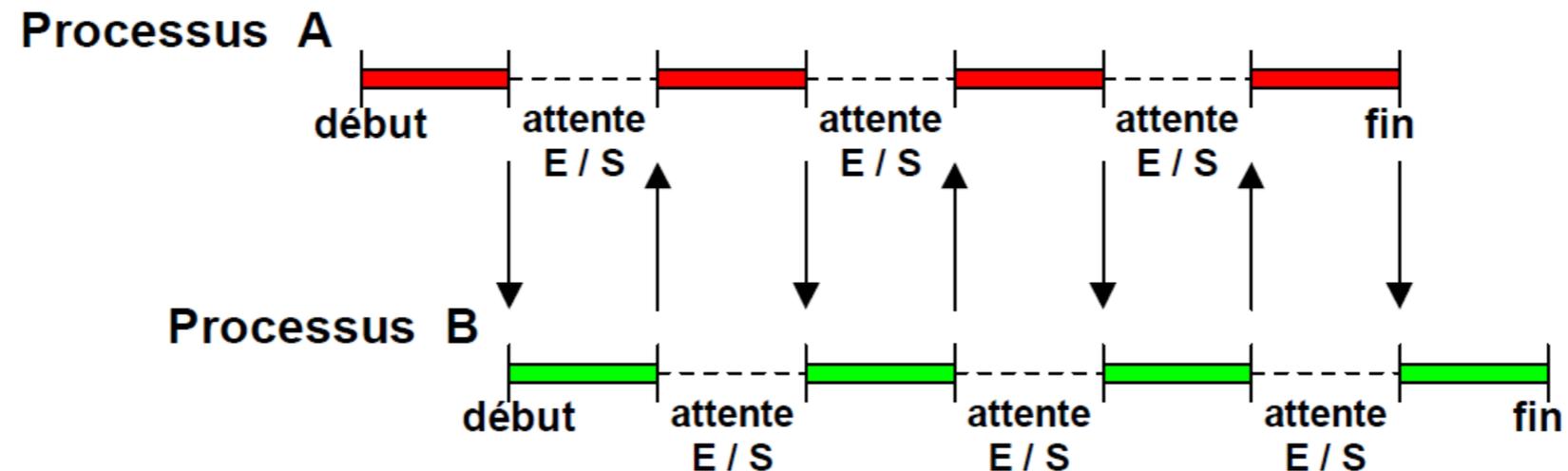


# SE “simples” (e.g. DOS)



- Un seul programme peut être exécuté à la fois
- Tout de même pas si simple! Il peut:
  - gérer des périphériques.
  - traiter les interruptions du système.
  - fournir un interpréteur de commandes pour traiter les requêtes de l'utilisateur
  - charger un programme en mémoire, puis de l'exécuter
  - fournir un ensemble de routines disponible pour la création de programmes.
  - gérer les fichiers et les accès disques.

# SE “multi-tâches” (e.g. UNIX)



- détermine la séquence de programmes à exécuter selon plusieurs facteurs (priorité, utilisation d'entrées/sorties, conflit de ressources, etc.)
- offre des protections et des niveaux de sécurité pour les accès aux ressources de l'ordinateur.
- gère une mémoire et des ressources pouvant être communes à plusieurs processus.
- gère la communication avec d'autres ordinateurs.

Types de systèmes d'exploitation?

# Type: SE dans les PCs

- Certainement les plus familiers
  - Monotâche — multitâches
  - Importance GUI conviviale
  - Support pour nombreux périphériques
  - Nombreux logiciels disponibles
- Exemples:
  - Windows XP/Vista/7/8
  - MacOS
  - Ubuntu (Linux)



# Type: Serveurs

- Application typiques :
  - web
  - fichiers – email
- Grand nombre de petites tâches
- Exemples de S.E. :
  - Solaris
  - FreeBSD (Berkeley Software Distribution)
  - Linux
  - Windows Server 20xx



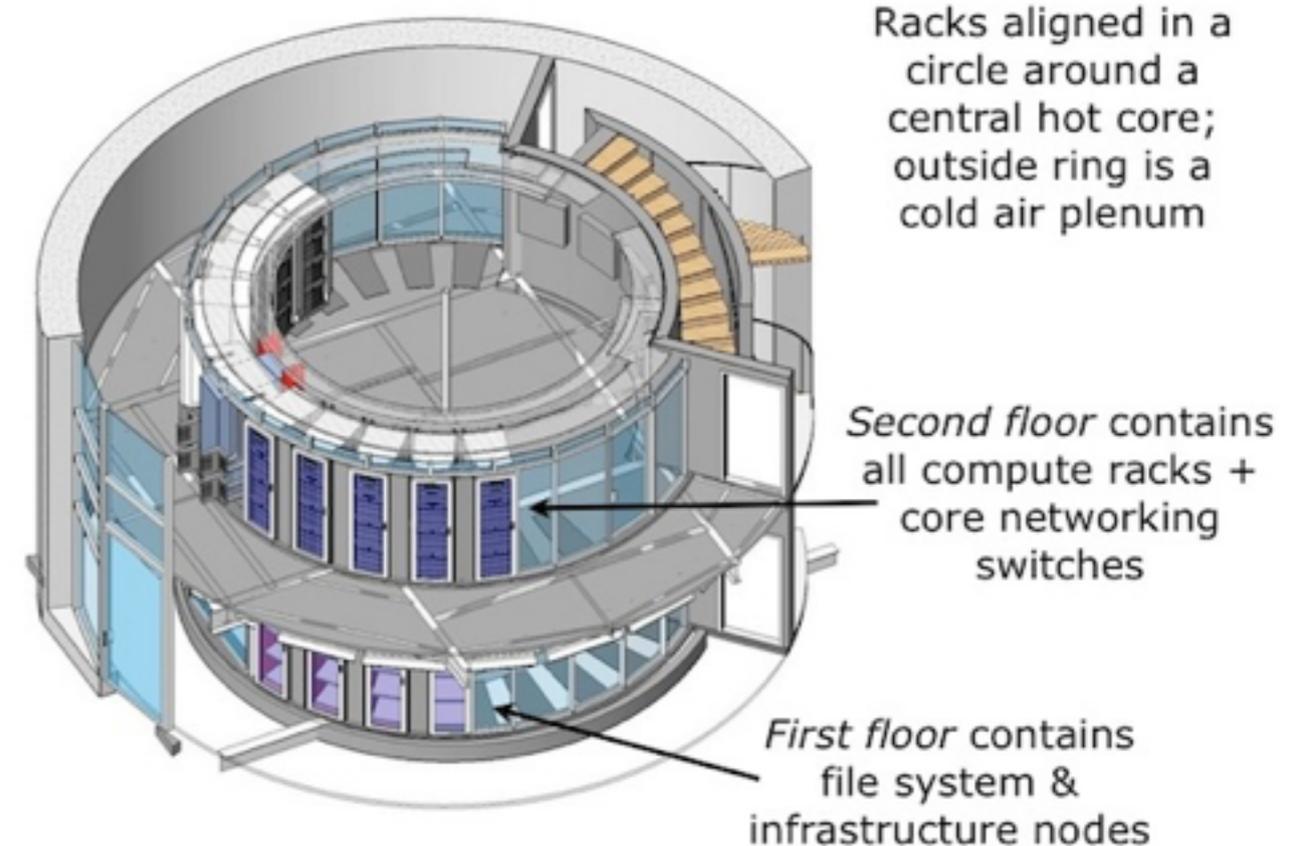
# Type: Serveurs à grande échelle

- Google: >450,000 serveurs
- Facebook: >60,000 serveurs
- Utilisent souvent des versions propriétaires comme S.E. (développent leurs propres S.E.)
- 1.3% de l'électricité mondiale est utilisée pour les "data centers"



# Type: Super-calculateurs

- Grosses tâches
  - calcul scientifiques: météo, séquençage de génôme, etc.



# Type: Systèmes embarqués

- S.E. simplifiés
  - peu ou pas d'interface usager
  - taille mémoire + CPU réduite
- Peuvent être mono-tâches
- Pas de démarrage de nouveaux processus par l'utilisateur : statique
- Exemples :
  - QNX, VxWorks
  - Kernel Linux + Busybox



# Type: Systèmes temps-réel

- Met l'accent sur la prédictibilité des temps d'exécution
- e.g. exécuter tâche exactement à chaque milliseconde
- Exemples:
  - QNX
  - VxWorks
  - FreeRTOS
  - Real Time Linux

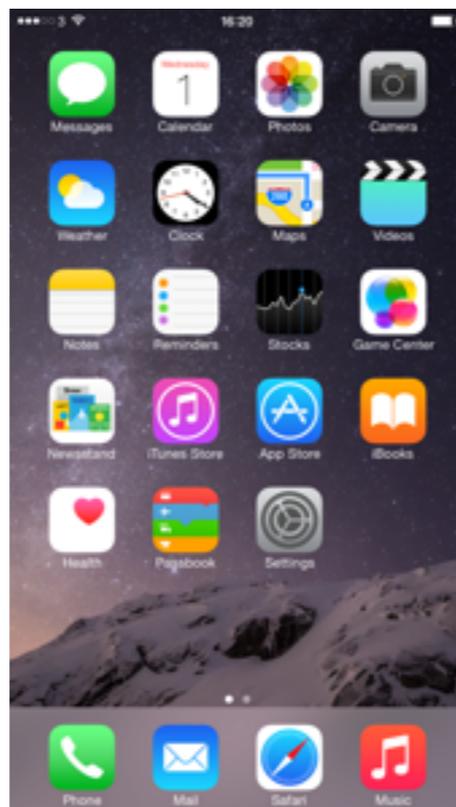
<https://www.youtube.com/watch?v=rVlhMGQgDkY>



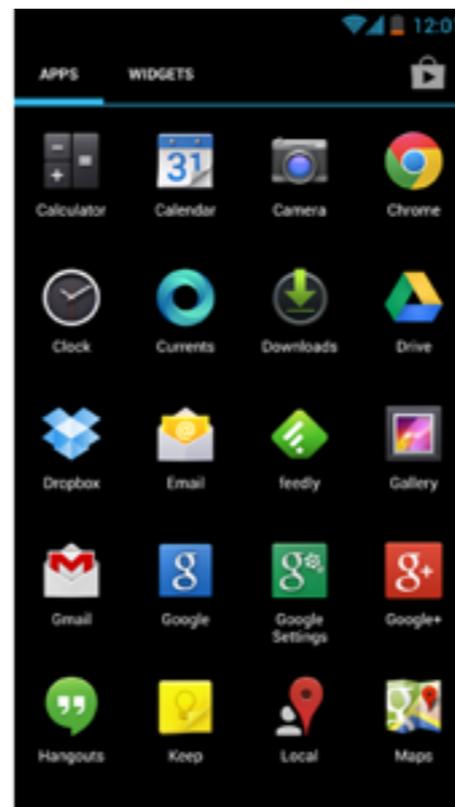
# Type: Systèmes mobiles

- Systèmes fermés (pas d'ajout périphérique)
- Petit / CPU limitée / téléphonie / réseau / GPS
- Faible consommation électricité
- Emphase sur l'interface graphique, pas la performance calcul
- Démarrage rapide, utilisation courte

Apple iOS



Android OS



Windows Phone OS

