

# Programmation système

mickael.hoerd@hesge.ch

# Objectifs du cours

- Distinguer les composants **internes** d'un système d'exploitation.
- Utiliser un système UNIX en ligne de commande.
- Ecrire des scripts système (bash/python).
- Programmer en langage C des applications faisant directement appel aux interfaces de base d'un système d'exploitation

# Bibliographie indicative

- Graham Glass et King Ables, *Unix for programmers and Users (3rd edition)*), Person Education 2003

# Contact et matériel de cours

- Par mail : [mickael.hoerdtd@hesge.ch](mailto:mickael.hoerdtd@hesge.ch)
- <https://hepia.infolibre.ch>

# Programme du semestre

- Introduction générale aux Systèmes d'exploitations
- Prise en main de la ligne de commande d'UNIX.
- Programmation de scripts système.
- Programmation système des entrées/sorties en langage C
- Programmation système des processus en langage C.

# Introduction générale aux systèmes d'exploitation

mickael.hoerd@hesge.ch

# Définition d'un système d'exploitation

## **Définition d'un système, tirée du dictionnaire :**

Un ensemble de composants coordonnés et conçus pour accomplir un but commun déterminé

## **Définition plus concrète, technique d'un système :**

Un ensemble de composants inter-connectés ayant un comportement attendu observés à l'interface de cette environnement.

## **Définition d'un système d'exploitation :**

Ensemble de programmes et de bibliothèques permettant de rendre (beaucoup) plus facile le travail des utilisateurs et des programmeurs d'une machine.

# Rôles principaux d'un SE (1)

Un système d'exploitation (Operating System) permet d'**exploiter** une machine c-à-d de:

- Simplifier la vie des utilisateurs/programmeurs.
- Gérer les ressources de la machine d'une manière efficace (accès partagé et concurrent).

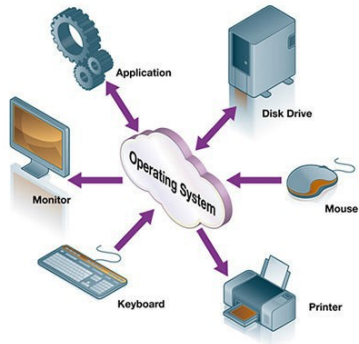




# Rôles principaux d'un SE (2)

Assurer la liaison entre

- Les ressources matérielles.
- les applications.



## Rôles principaux d'un SE (3)

- Exécuter d'autres programmes sur une machine.
- Fournir un accès programmatoire structuré, contrôlé et formalisé aux ressources d'une machine pour assurer la portabilité et la compatibilité binaire des applications développées sur une version précédente du système.
- Fournir une abstraction du matériel et des ressources aux applications.

# On en trouve partout et de tous les types



Il existe des **centaines de systèmes** :

- Windows
- Linux, Solaris, AIX, HP-UX, FreeBSD, Android, OSX
- GCOS, VMS, AS400

# Types de systèmes

## Classification

- **Spécialisés** pour un matériel : (système d'alarme, imprimante multi-fonction, console de jeux, magnétoscope...)
- **Généralistes** : permettent de faire tourner des programmes très divers sur des matériels différents

**Autres critères** : Mono/multi-tâches, postes de travail/serveur, embarqué ou non, Mono/Multi-utilisateurs, temps réel ou pas,...

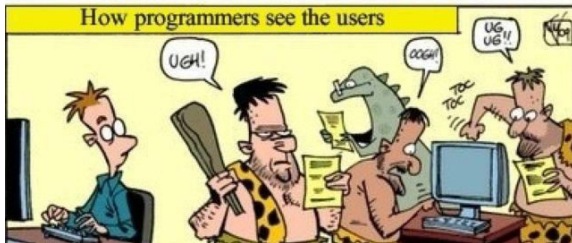
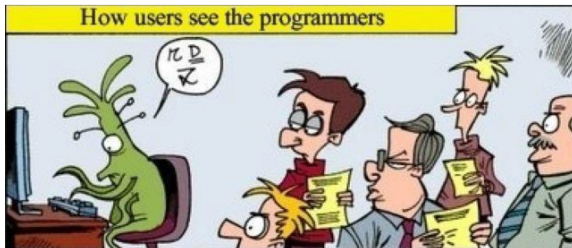
# Point de vue de l'utilisateur

- "Il faut que ça marche ! " (Comme j'en ai envie...)
- "ça n'imprime pas ... "
- **Conclusion** : La machine doit être utilisable pour le but fixé

# Point de vue du programmeur/ingénieur

- Simplifier l'accès aux ressources de la machine (Abstraction) :
  - Mémoire, processeur, périphériques, fichiers, processus, réseaux.
  - Modèle de programmation simple et unifié (Norme POSIX)
- Efficacité dans tous les cas : Accès partagé et concurrent du matériel par les applications.
- **Conclusion** : La machine doit être facile à programmer et gérer efficacement les ressources

# Utilisateurs et programmeurs ?



# Quelques fonctions plus précises d'un SE

- Initialiser et piloter les périphériques matériels (vidéo, audio, réseau, stockage).
- Gérer les fichiers (si présents) et les processus.
- Partager l'accès aux ressources du système.
- Assurer la gestion, l'ordonnancement et la communication des tâches.
- En cacher la complexité (Abstraction).
- ...



# Un bref historique

- 1945-55 : Tubes et interrupteurs
  - Pas de système d'exploitation
- 1945-65 : transistors, cartes perforées
  - traitement par lots
- 1965-80 : circuits intégrés, disques, réseaux
  - Multi-tâches, Multi-utilisateur, temps partagé, entrée/sorties : **Système d'exploitation nécessaire**
  - Unix, BSD, Interface POSIX
- 1980 - — : ordinateurs personnels (PC)
  - Interface graphique (concept créé vers 1960 à Stanford)
  - Réseaux et systèmes distribués
- 2000 - — : Multiplication des systèmes "intelligents"
  - Smartphone, Smartwatch, Smartpad,...