

# Programmation Système

---

Guillaume Chanel - [guillaume.chanel@hesge.ch](mailto:guillaume.chanel@hesge.ch)

Florent Glück - [florent.gluck@hesge.ch](mailto:florent.gluck@hesge.ch)

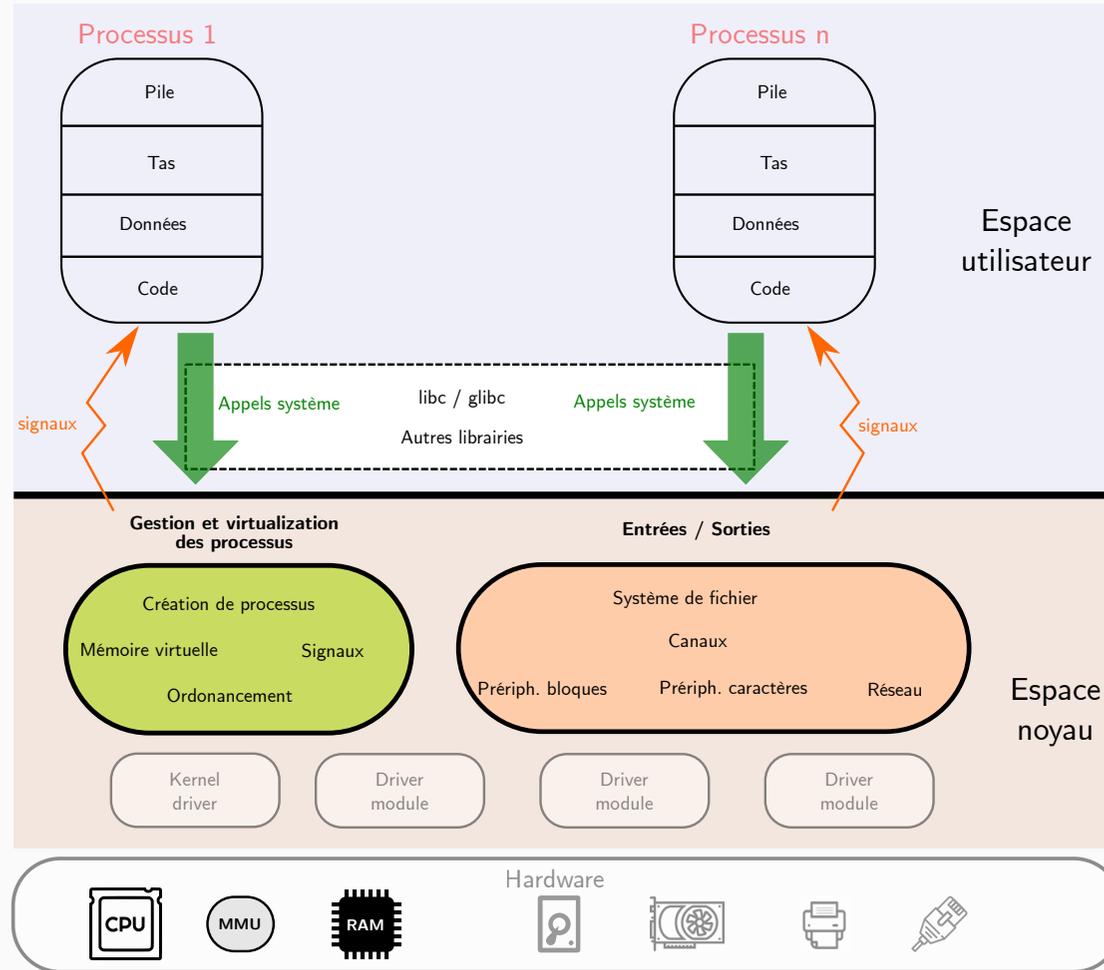
February 12, 2025

ISC - HEPIA

# Sujets abordés

1. Historique d'UNIX
2. Introduction à UNIX par le shell Bash
3. Réaliser des scripts en Bash
4. Le langage C
5. Librairies et édition de liens
6. Les appels systèmes
7. Les entrées / sorties (I/O)

# Concepts



# Organisation

---

- Portail du cours **HEPIA ISC Programmation Système** sur Cyberlearn
  - <https://cyberlearn.hes-so.ch/course/view.php?id=13665>
  - clés d'inscription : **rtfm25**
- Lien du dépôt git sur Cyberlearn
- Lien d'inscription à Mattermost sur Cyberlearn
- Tous les contenus (cours + TPs) sont disponibles sur **git**
- Questions/discussions liées au cours sur **Mattermost**
- **N'oubliez pas de vous inscrire à Cyberlearn et Mattermost !**
- Amenez **toujours** votre PC

Durant le cours, plusieurs exercices et TPs sont à réaliser

- Travail en cours **et à la maison**
- TPs et exercices ne sont pas à rendre et ne sont pas évalués directement
  - examens pratiques portent directement sur TPs et exercices
  - **impératif de réaliser** TPs et exercices **vous-mêmes** pour **très fortement augmenter** vos chances de **réussir les examens**
    - forte probabilité **d'échouer** dans le cas contraire !

# Méthode de travail (1/2)

- **Soyez présent et attentif** : des informations seront souvent données sur le moment
  - manquer des cours signifie manquer des explications et des conseils utiles !
- **Prenez des notes** car les slides sont **incomplètes**
- **Travaillez régulièrement** : à la dernière minute ne suffira pas 😨

# Méthode de travail (2/2)

- **Soyez proactif** et posez des questions, car il n'y a pas de questions bêtes 😊
  - pas de questions = on suppose que tout est compris...
- Ne copiez-collez pas **aveuglément** du code trouvé ailleurs (ChatGPT, Stack Overflow, etc.)
  - l'objectif est que vous **compreniez** réellement ce que vous faites !
  - meilleur moyen pour comprendre : **prendre le temps** de réaliser et réfléchir par **vous-mêmes**

# Notation

---

# Modalités évaluations

- **QCMs notés pendant tous les semestre**
  - contrôles réguliers, par exemple chaque deux semaines
  - assurez-vous donc de toujours être présents
- **Examen de programmation shell** : réaliser des scripts
- **Examen de programmation C**, méthode d'évaluation parmi :
  - un code à trou à compléter
  - un code de taille moyenne à écrire entièrement
  - plusieurs petits codes à écrire (style exercices)
-  **Dans tous les cas, absence non justifiée, triche ou plagiat = 1**

# Note semestrielle

Note semestrielle basée sur trois notes :

- Note moyennant tous les QCMs : 20%
- Note pour partie shell (bash, scripts, etc.) : 40%
- Note pour partie programmation C (appels systèmes, libs, etc.) : 40%

# Examens sur machines virtuelles

- Les QCMs seront réalisés sans documents ou outils à disposition (shell, etc. si sur machine)
- Les examens de programmation shell et C seront réalisés sur machines virtuelles :
  - sans accès internet (ou accès limité)
  - accès aux slides de cours (parfois partiellement)
  - accès au manuel système

# Objectifs du cours

---

# 1. Utiliser un shell

- Utiliser le manuel pour obtenir de l'aide
- Utiliser une ligne de commande
- Configurer son shell bash
- Écrire et lire des scripts bash

## 2. Introduire les API POSIX

- Identifier un appel système
- Programmer en C, analyser et comprendre un code existant
- Reconnaître un type opaque
- Utiliser des bit fields
- Entrées/sorties (I/O)
  - connaître les appels systèmes liés aux I/O
  - connaître les différences entre I/O par appels systèmes et I/O bufferisées

### 3. Librairies, édition des liens et chargement

- Lier, utiliser des librairies et comprendre leur fonctionnement
  - pour l'exécution de programmes
  - pour le développement
- Identifier les différences entre librairies statiques et dynamiques

# Questions

