100 Introduction aux systèmes d'exploitation

INF3173

Principes des systèmes d'exploitation

Jean Privat

Université du Québec à Montréal

Hiver 2021

Plan

1. Préalables (et contexte)

2. Objectif du cours

Préalables (et contexte)

Ce qui est attendu des préalables

INF1070 Utilisation des systèmes informatiques

- Environnement Unix et shell
- Systèmes de fichiers (types, droits, etc.)
- Commandes, programmes et processus (redirection, tubes, etc.)
- `man` et RTFM

INF2171 Organisation des ordinateurs et assembleur

- CPU: Instruction, exécution, registres
- Mémoire: code machine, données, pile, tas, adressages, représentation de l'information

INF3135 Construction et maintenance de logiciels

- Programmation C: pointeurs, allocation, bibliothèques
- Qualité logicielle: exactitude, robustesse

INF1070 en très vite

Système de fichiers

- Fichier = forme libre de données stockées
- Indépendance au matériel et extensible
- Arborescence : chemins relatifs et absolus
- Sécurité : utilisateurs et droits
- `cd`, `ls`, `cat`, `>`, `rm`, `grep`, etc.

Commandes et processus

- Processus = programme en cours d'exécution
- PID, PPID, utilisateur, etc.
- Utilise mémoire et processeur (entre autres)
- `sh`, `ps`, `kill`, `|`, `&`, `uptime`, `free`, `PATH`, etc.

UNIX et Linux

INF2171 en très vite

CPU - Unité centrale de traitement

- Unité de contrôle + unité arithmétique et logique
- Exécute des instructions jusqu'à l'arrêt de l'ordinateur
- N'est qu'une machine

$\rightarrow$ Ne sait pas ce qu'est un SE, un programme ou un utilisateur

Registres (dans le CPU)

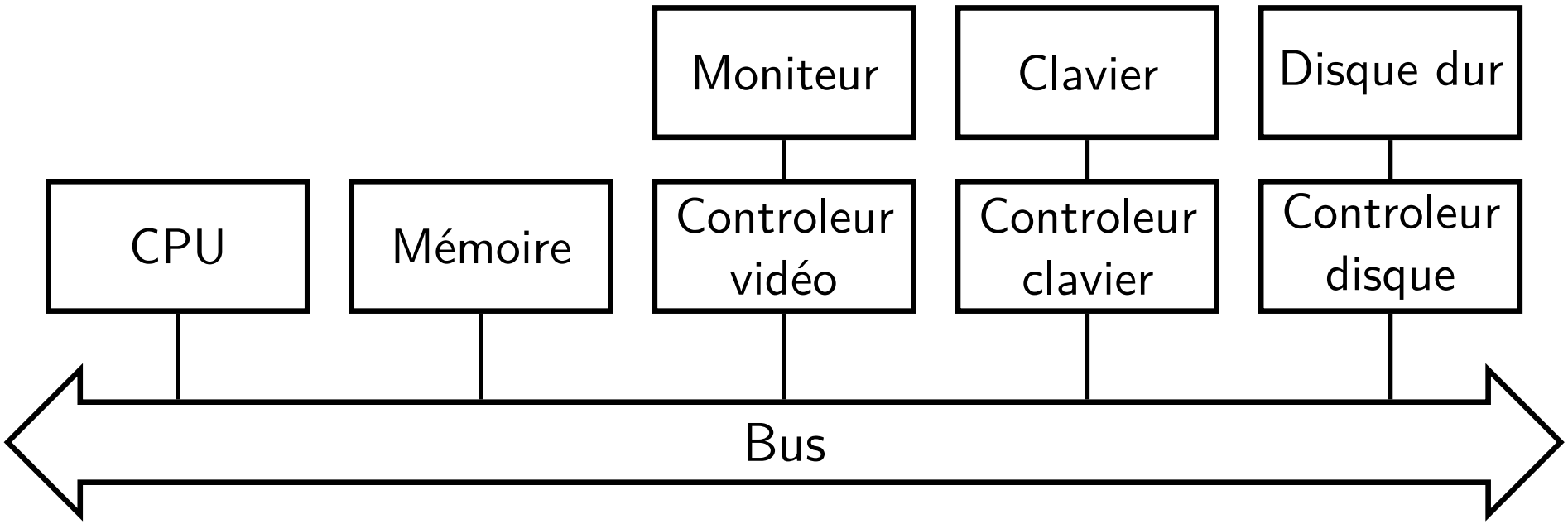
Généraux (pour les calculs) et spéciaux, dont :

- Compteur ordinal (PC, *program counter*)
- Pointeur de pile (SP, *stack pointer*)
- Mot d'état (flags d'états et de contrôle)

Mémoire volatile (RAM)

- Grand tableau d'octets adressables
- Contient code machine et données

Architecture à la von Neumann



INF3135 en très vite

Le logiciel c'est difficile

- Représentation des données, gestion de la mémoire, pointeurs, etc.
- Performance processeur, disque, réseau, énergie, etc.
- Débogage, portabilité, qualité logicielle, etc.

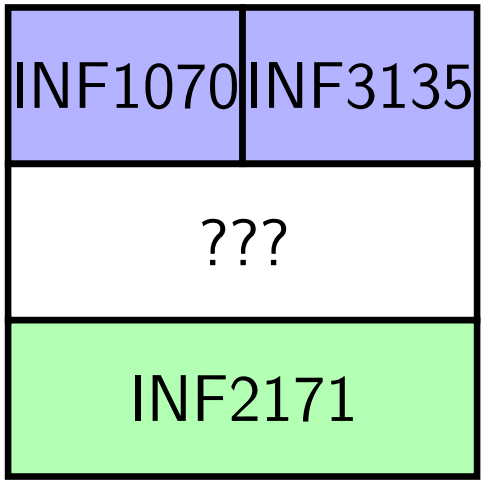
$\rightarrow$ Produire **le bon** code de **la bonne** façon, c'est difficile

Le logiciel c'est complexe

- Utilisation de bibliothèques
- Processus de compilation

Objectif du cours

Combiner les mondes



INF1070 : Des processus et utilisateurs

- Cohabitent pacifiquement
- Sur un même ordinateur

INF3135 : Des logiciels

- Complexes
- De qualité variable

INF2171 : Des ordinateurs et CPU

- Conceptuellement simples

Comment c'est possible que tout fonctionne ensemble ?

Exemples

Partage des ressources

- Plusieurs programmes s'exécutent en même temps
- Mais doivent se partager les ressources matérielles
- CPU, mémoire, fichiers, etc.

Isolation des processus

- Les bogues des programmes existent
- Mais affectent rarement les autres programmes directement
- Qui s'exécutent pourtant sur le même ordinateur

Sécurité des données

- Les utilisateurs malveillants (ou incompétents) existent
- Mais ne peuvent pas lire ou corrompre les données des autres
- Sauf avec un tournevis et un marteau

Objectifs du cours

- À quoi sert un SE
- Comprendre comment fonctionne un SE
- Savoir utiliser les services offerts par les SE

Beaucoup de rôles

Dans le cadre du cours, on va voir

- Gestion des processus et leur communication
- Gestion des fichiers et de l'espace disque
- Gestion de la mémoire
- Gestion des périphériques (entrées-sorties)

Attention

- Les rôles sont inter-reliés
- Le découpage des rôles est assez arbitraire

Beaucoup de points de vue

Utilisateur

- Humain (de base)
- Administrateur système

Développeur d'applications

- La plupart d'entre-vous

Développeur de systèmes d'exploitation

- Se mettre *dans ses souliers* peut aider à mieux comprendre

Constructeur de matériel

- On le prendra moins en compte

Beaucoup de niveaux

On raisonnera souvent à trois niveaux différents
Attention à pas les mélanger !

Niveau général

- Problèmes conceptuels et solutions générales
- Problèmes fréquents et solutions habituelles

Niveau Unix et POSIX

- Le mode Unix et sa philosophie
- Normes, appels système et API
- Avantage : répandu, (relativement) portable et documenté

Niveau Linux

- Le projet Linux (et son écosystème)
- Détails d'implémentation et de services spécifiques
- Avantage : versatile, libre, gratuit, ouvert, étudiable

Beaucoup d'histoire

Longue historique

Les systèmes d'exploitation existent depuis les années 1960
En 2020 les choses importantes ne sont pas **exactement** les mêmes qu'en 2010, 2000, 1980 ou 1960

Évolution des SE

- Monotâche $\rightarrow$ Multiprogrammation $\rightarrow$ Multitâche $\rightarrow$ Multiutilisateurs $\rightarrow$ Virtualisation et infonuagique

Difficulté

- La gomme a été mâchée par beaucoup de monde
- Le vocabulaire et les concepts sont parfois spécifiques

Beaucoup de spécialités

Difficultés autour des SE: plusieurs spécialités

Architecture matérielle

- Organisation matérielle des ordinateurs
- Elle est complexe et évolue chaque année

Programmation *bas niveau*

- À l'interne (dans le système lui-même)
- Dans les API offertes aux programmes

Algorithmique

- Recherche d'efficacité
- Taille croissante des systèmes

$\rightarrow$ impose l'utilisation d'algorithmes sophistiqués

Dans le cadre du cours on essaiera d'éviter ces difficultés