

# 000 Présentation générale

## INF3173

### Principes des systèmes d'exploitation

Jean Privat

Université du Québec à Montréal

Été 2021

# Aujourd'hui

- Présentation du cours et des modalités
- Posez les questions au fur et à mesure
- Séance courte ?

## Cette séance est enregistrée (normalement)

- *Mutez* vos micros pour éviter les bruits de cuisine
- Allumez votre webcam, je me sentirai moins seul

# Osez l'interactivité

## Pas de protocole strict

- Intervenez oralement !
- Je ne regarde pas toujours le *chat* ou les mains virtuelles levées

# Je me présente, Jean Privat

## Langages, compilation, systèmes

- 07–...: Professeur, UQAM (QC)
- 06–07: Postdoc, Purdue (ÉU)
- 02–06: Ph.D, Montpellier (FR)

## Intérêts

- La programmation : de l'assembleur aux langages à objets
- L'info système : compilateurs, VM et systèmes d'exploitation

## Contactez-moi

- [privat.jean@uqam.ca](mailto:privat.jean@uqam.ca)

# Plan

- 1 INF3173, principes des systèmes d'exploitation
- 2 Formule pédagogique
- 3 Évaluations

# INF3173, principes des systèmes d'exploitation

# Principes des systèmes d'exploitation

## Cours « traditionnel »

- Existe dans la plupart des formations universitaires
- Depuis au moins 40 ans

## Approche dans INF3173 à l'UQAM

- Voir et comprendre les concepts, les besoins, les solutions
- Détailler un mise en œuvre (mécanique et politique)
  - On utilise Unix/Linux
- Expérimenter de façon programmatique (espace utilisateur)
  - Programmation système

## Exemple d'objectif

- Être capable de lire, de comprendre et d'appliquer une *manpage* de la section 2 en toute confiance

# Mise à jour INF3173

## Automne 2020 : ne jamais gâcher une bonne crise

Profiter des cours en ligne pour faire une grosse mise à jour

- Prise en compte de INF1070 (on ne part plus de zéro)
- Mise à jour du contenu, plus complet (on est dans les années 20)
- Production de matériel pédagogique (beaucoup d'énergie)
- Formule pédagogique: les cours en ligne c'est compliqué à monter autant que le jeu en vaille la chandelle

## 3e itération

- Contenu amélioré & disponible d'avance
  - On tente quand même de nouvelles choses !
  - Rythme plus régulier (réorganisation des semaines)
- Pardonnez néanmoins les problèmes techniques et les petites mises à jour inévitables



# Prérequis

- INF1070 : utilisation et administration des systèmes informatiques
  - Shell, fichiers et processus
- INF2171 : fonctionnement du processeur et de la mémoire
  - Rassurez-vous, on ne programmera pas (trop) en assembleur
- INF3135 : programmation *C de qualité*
  - Par contre on programmera *beaucoup* en C

# INF3173 difficile?

- INF3173 est un cours **avancé** (cours préalables nécessaires)
- INF3173 est un cours **dense** (plein de choses à voir)
- INF3173 est un cours de **programmation** (en grande partie)

## Difficultés du cours

- Les lacunes éventuelles dans les cours préalables
- Ne pas travailler au fur et à mesure et se laisser déborder par la matière
- Programmer sans comprendre et sans rigueur  
*stackoverflow* n'est pas la solution

# Formule pédagogique

# Classes normales et inversées

## Classe normale

- Je fais la leçon en classe
- Vous faites les exercices à la maison

## Classe inversée

- Vous faites la leçon à la maison
- On fait les exercices en classe

# Classe inversée INF3173

## Plusieurs étages

- Capsules + diapos + exemples + sujets d'exercices : à l'avance
  - Quiz d'autovalidation
- Laboratoires synchrones avec le démonstrateur
  - Vous faire pratiquer
- Travaux pratiques
  - Vous faire maîtriser

# Capsules

- Moins de 2h par semaine... en moyenne
- Semaines pas forcément égales : prenez de l'avance
- S'écotent bien en x1.5
- Et on peut réécouter !

## Faites à l'automne 2020

- Temps infini à faire
- Petits défauts techniques, restant de fautes et quelques lapsus
- Youtubeur c'est un métier

## Corrélation

- L'écoute des capsules vs. la réussite au cours
- Attention : corrélation  $\neq$  causalité

# Diapositives

- Spartiates (c'est beamer, pas keynote)
  - Support des capsules
  - Relativement détaillées (pas juste un support visuel)
  - Contient des hyperliens vers des ressources externes
- Les diapos ne sont pas des notes de cours !

## Différence avec les capsules?

- Révisées pour l'été 2021
  - Corrigées par rapport au capsules
- Ne contient pas les explications, exemples et expériences

## Marqueurs

- ★ Notion fondamentale
- 🔍 Notion avancée (ou parfois optionnelle)

# Semainier

## Nomenclatures des diapos et capsules

- Format: « ABC Titre »
- Exemple: « 120 Appels système »
- A : numéro de chapitre (0 pour présentation générale)
- B : numéro de section (0 pour intro du chapitre)
- C : numéro de partie (>0 quand la section est trop longue)

## Les semaines

- 15 semaines → 12 semaines de capsules
  - Le nombre de capsules par semaine est variable
- Point d'entrée unique <https://inf3173.uqam.ca/> maintenu à jour



# Ressources supplémentaires

## Les pages de man

- **RTFM** *read the fucking manual* (regarde ton fichu manuel)

## Deux ouvrages optionnels

- *Operating Systems Concepts*, Silberschatz, 10e édition (2018)
- *Modern Operating Systems*, Tanenbaum, 4e édition (2014)

Complets et intéressants !

# Mattermost

## Groupe privé dédié

- <https://mattermost.info.uqam.ca/inf3173-e21>
- Lien d'invitation envoyé par courriel et sur le site web

## Nétiquette

- Merci de garder les canaux propres
- Et de respecter les principes de base de la **nétiquette**
- Les discussions, les questions et les erreurs sont permises
- Les abus et la mauvaise foi ne seront pas tolérés  
Mais il n'y a pas de raison que ça arrive...

# Canaux

- `/var/log`
  - Discussions générales sur le cours, les questions, etc.
- `/dev/random`
  - Pour le social, l'actualité en informatique, les mèmes, etc.
- Des canaux spécifiques seront créés (Cours, Lab, TP, etc.)

# Séances magistrales synchrones via zoom ?

- Pas fameux à l'automne 2020
- Pas beaucoup mieux à l'hiver 2120

## Symptômes

- Peu de participants
- Peu d'interaction
- Peu de questions
- Peu de visionnement des séances enregistrées
- ... Peu utile?

# Été : modalité différente

- Visionnez les capsules et commencez les laboratoires quand vous voulez

## Sur mattermost

- Si vous avez des question, inutile d'attendre
- Posez-les dans le canal public dédié
- On essaiera de vous répondre rapidement
- Sauf pendant les périodes de quiz!

## Séance de disponibilité

- Durant la séance de cours (mardi de 9h30 à 12h)
- Je pourrais répondre plus rapidement
- Panopto-party
  - Regardez les capsules à ce moment-là
  - Posez vos question en direct

# Laboratoires

## Synchrone et par Zoom

- Exercices à faire ensemble
- Vous programmez et posez des questions
- On vous donne des indices, des solutions partielles puis des solutions plus complètes
- Une approche différente et complémentaire de la matière
- Interactivité !

## Les laboratoires sont obligatoires

- Exercices d'approfondissement et matière originale
- Qui sera éventuellement évaluée dans les devoirs/TP/quiz

# Votre travail **avant** le lab

- Écouter les capsules
  - Posez vos questions sur mattermost
  - En particulier le mardi matin
- Survolez les diapos
  - Cliquez sur les liens
  - Essayez de répondre aux questions
- Essayer les exemples
  - Compilez, exécutez, modifiez, comprenez
- Commencez le lab
  - Lisez l'énoncé
  - Faites la mise en place et les premiers exercices
  - Ce qui permet d'avoir des séances de lab plus utiles

# Linux

C'est le système qu'on utilisera

- En classe
  - Pour les labs
  - Pour les travaux
- Parce qu'il est bien, libre, gratuit, documenté et répandu

## Vous devez avoir accès

- À un système avec une distribution Linux récente
- Si vous pouvez être root dans l'espace de noms principal (pas dans un conteneur), c'est mieux, mais pas obligatoire

## Support Linux fourni « au mieux » et par les pairs

- Par les [auxiliaires](#)
- Par les [moniteurs de programme](#)
- Par l'[AGEEI](#)



# Ce qui est acceptable

## Sont acceptés

- Système natif (et double-boot)
- Machine virtuelle
- Serveur sur [labunix](http://labunix.uqam.ca), comme `java.labunix.uqam.ca`

## Ne sont pas acceptés

- Windows et WSL
  - macOS
  - Android
- On aura besoin des vrais appels système Linux spécifiques
- Et d'un environnement Unix traditionnel  
POSIX, shell, utilitaires, compilateurs, etc

# Évaluations

# Évaluations

- 12 quiz rapides
    - Mardi 12h à 23h55
    - 20 minutes max
    - 12% (1% chacun)
  - 3 travaux pratiques
    - Un petit (8%)
    - Un moyen (20%)
    - Un gros (20%)
    - Sur plusieurs semaines chacun
  - 2 devoirs à la maison
    - Asynchrones
    - Sur deux jours
    - 20% chacun
  - Validations orales en fin de session
- Travail individuel

# Quiz

- Choix de réponses (quiz moodle)
- 20 minutes chronométrées
- Ouvert le mardi de 12h00 à 23h55

## Autovalidation

- Après les capsules et la séance de disponibilité
  - Mais avant les labs
  - Questions souvent simples
- Objectif de 80% à 100% de bonnes réponses
- Moins de 80% : c'est inquiétant

# Travaux pratiques

- TP0 entraînement et/ou remise à niveau
- TP1 fichiers et/ou processus
- TP2 communication et/ou synchronisation

## Contenu

- Mise en pratique des concepts/services des systèmes d'exploitation
  - Programmation système (C&Linux) coté utilisateur (POSIX&Linux)
  - Exemple : développement de petits utilitaires
  - Objectifs : exactitude, robustesse, élégance
- Petits programmes mais de qualité

# Travaux pratiques

- INF3173, dernier cours de programmation obligatoire du BIGL?
  - Vous n'êtes plus des débutants en programmation
- Votre programme doit être fonctionnel
  - Le cahier des charges est précis et rigoureux
  - Un programme qui fonctionne presque  $\approx$   
Un parachute qui fonctionne presque
- Votre programme doit être robuste
  - En programmation système, on a rarement droit à l'erreur
  - La robustesse est la politesse des outils système
- Votre programme devrait être simple et élégant
  - Souvent les étudiants se compliquent la vie inutilement
  - Temps passé  $\neq$  qualité

# TP faisables

- Programmes courts
  - Quelques centaines de ligne
  - Un seul fichier source
- Pas ou peu de conception
  - Pas de domaine métier à modéliser
  - Pas de UML
- Pas ou peu d'algorithmes ou de structures de données
  - Et encore moins d'IA
- Pas ou peu de complexités non nécessaire
  - On essaye de limiter les difficultés liées aux C
  - Même si ça rend les programmes finaux moins intéressants

# Pour réussir les TP

- Lisez l'énoncé
- Respectez les exigences
- Profitez de l'infrastructure de test éventuelle
- Ayez fait (pour de vrai!) les laboratoires
- Programmez robuste et simple
- Ne commencez pas la veille
- Utilisez les services du monitorat de programme pour de l'aide à la programmation (INF3135, INF2120, INF1120, etc.)
- Relisez l'énoncé



# Par soucis d'équité dans les TP

- Posez vos questions publiquement sur le canal dédié mattermost
- Ne divulguez pas vos solutions sur le mattermost (ni ailleurs)
- Je ne réponds à aucune question sur un TP dans les quatre (4) jours précédents la remise
- Aucun rendu hors consigne ou délais ne sera évaluée

# Devoirs à la maison

## Modalités

- 48h, mardi et mercredi
- Pas de cours ni de lab les semaines des devoirs
- 3h de temps (max estimé) pour le faire
- N'utilisez pas les 48h à temps plein
- Le reste de la semaine pour étudier la matière suivante

## Contenu

- Questions ouvertes et de de réflexion
- Étude des cas ou de programmes
- Mise en situation
- Etc.

# Devoirs à la maison

## Pour réussir

- Visionnez les capsules, posez des questions, participez aux labs
- Ne commencez pas à visionner la veille

## Par soucis d'équité

- Aucune négociation de note
- Si vous vous estimez lésé dans l'évaluation de votre travail
  - Procédures de modification de note et/ou révision de notes
  - Gérés au niveau du départementale
  - Instruit de manière équitable

# Validations orales

- Objectif: s'assurer de l'intégrité des évaluations
  - Prise de rdv préalable: je vous contacterai
  - Pas de note: c'est soit OK, soit **infraction académique**
- Pas besoin de paniquer pour autant

# Liens

- [Plan de cours](#)
- [Site web du cours](#)
- [Mattermost](#)
- [Moniteurs de programme](#) (support académique)