Systèmes d'exploitation et Programmation système (RS)

Lucas Nussbaum <lucas.nussbaum@loria.fr>

Supports de cours, TD et TP largement basés sur ceux de
Martin Quinson <martin.quinson@irisa.fr>

Telecom Nancy – 2ième année

http://members.loria.fr/lnussbaum/rs.html

À propos de ce document

Document diffusé selon les termes de la licence

Licence Creative Commons version 3.0 France (ou ultérieure)

Attribution ; Partage dans les mêmes conditions

http://creativecommons.org/licenses/by-sa/3.0/fr/

Remerciements

- Sacha Krakoviack pour son cours et Bryant et O'Hallaron pour leur livre
- Les (autres) emprunts sont indiqués dans le corps du document

Aspects techniques

- Document LaTeX (classe latex-beamer), compilé avec latex-make
- Schémas : Beaucoup de xfig, un peu de inkscape

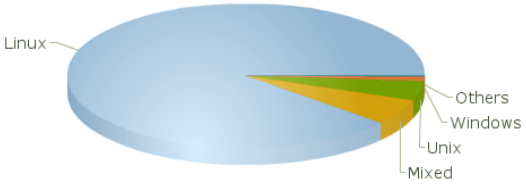
Site du cours : http://members.loria.fr/lnussbaum/rs.html

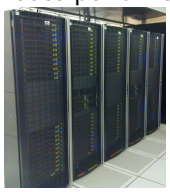
- TD/TP, exams et projets (sources disponibles aux enseignants sur demande)

À propos de moi...

Lucas Nussbaum

- Formation : ingénieur ENSIMAG (2005), Doctorat (2008)
- Depuis 2009 :
  - Enseignant-chercheur (Maître de conférences) à l'univ. de Lorraine
    - Principalement en licence professionnelle ASRALL (Administration de Systèmes, Réseaux et Applications à base de Logiciel Libre)
  - Chercheur dans l'équipe RESIST du LORIA
- Recherche : Systèmes distribués, calcul à haute performance, Cloud





- Contributeur au logiciel libre
  Debian (Project Leader 2013-2015, *Quality Assurance*), Ruby
- Plus d'infos :
  - http://members.loria.fr/lnussbaum/ (Lucas.Nussbaum@loria.fr)

Organisation pratique du module

Module en deux parties

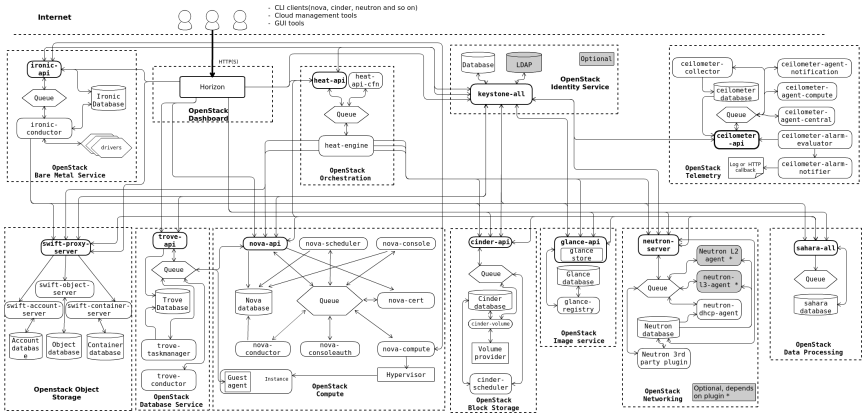
- Partie système (intervenant en cours : Lucas Nussbaum)
  - 5 cours, 3 TD, 3 TP
  - Examen sur table (mi octobre)
    - Documents interdits sauf un A4 recto/verso **manuscrit**
  - un projet (pour décembre)
    - Binômes et Git obligatoires
    - Le sujet arrive bientôt...
- Partie réseaux (intervenant en cours : Isabelle Chrisment)

Implication

- Manipulation : programmez ! Expérimentez !
- Questions bienvenues : pendant/après le cours, par mail, etc.

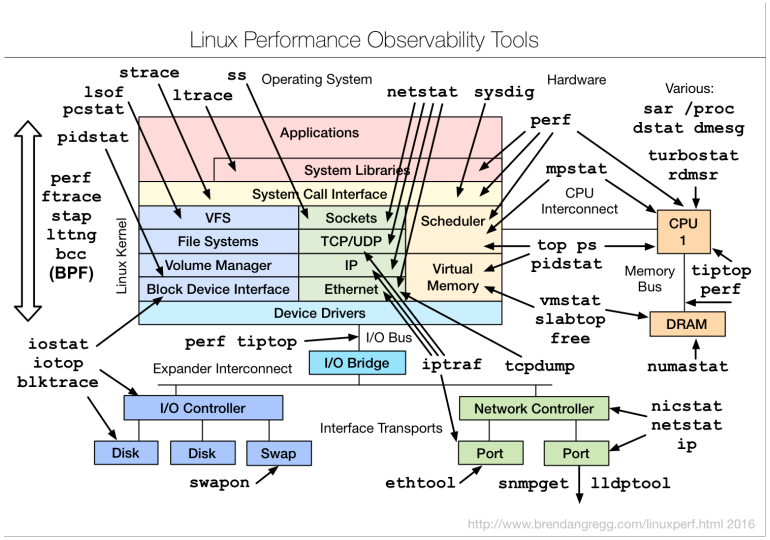
Pourquoi un cours de système ?

- Quatre concepts fondamentaux de l'Informatique (G. Dowek) : Information, **Machine**, Algorithme, Langage
- Les architectures et infrastructures modernes sont complexes
  - Wikipedia : 1145 serveurs, 30 900 CPUs
  - OVH : 250 000 serveurs
  - OpenStack (pile logicielle permettant de créer un *Cloud* privé)

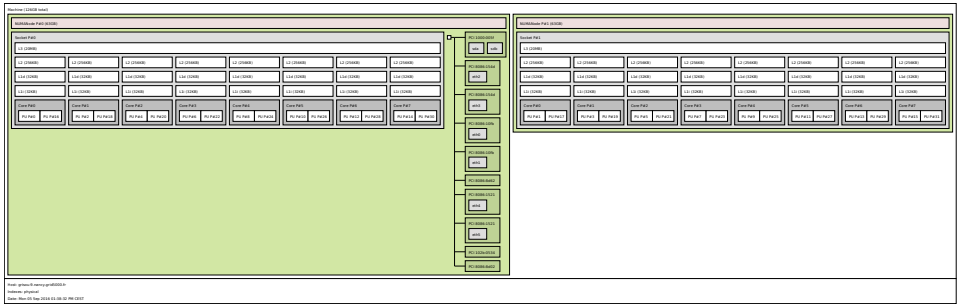


Pourquoi un cours de système ? (2)

- Noyau Linux (et outils d'observation)



Pourquoi un cours de système ? (3)



- Système dual-socket Intel récent (2x Intel E5-2630v3, 8 cœurs/CPU)
  - Hiérarchie de caches entre les processeurs et la mémoire
    - Partagés (L3) ou non (L1, L2) entre cœurs physiques
  - Plusieurs files d'attente pour le même cœur physique (*Hyperthreading*)
  - Mémoire séparée en deux, chaque moitié reliée à un processeur différent
    - Architecture NUMA : Non-Uniform Memory Access
  - Périphériques PCI reliés à un seul des deux processeurs
- Pour l'exploiter pleinement, il faut en comprendre les détails

Pourquoi un cours de système ? (4)

Comment les utiliser efficacement ? Comment les concevoir ?
↝ Performances, sécurité, résilience, efficacité énergétique

Métiers (à la sortie de TELECOM Nancy) :

- **IT Operations / Administration système et réseaux**
  ↝ Assurer le maintien en conditions opérationnelles d'une infrastructure (suivi des incidents, montées de version, etc.)
- Mouvement vers le modèle **DevOps** (≈Google Site Reliability Engineers)
  - Suppression des silos *software development* vs *operations*
  - Infrastructure as Code : cloud, *pet vs cattle*
  - Itérations rapides, tests automatiques, déploiement automatiques et continus

  Compétences nécessaires : développement logiciel, compréhension profonde des systèmes (combinaison très recherchée sur le marché du travail)
- **Systèmes embarqués / enfouis** : domotique, automobile, *appliances*
- **Sécurité informatique** (souvent lié à des aspects système/réseau)
- Même comme pur développeur, savoir ce qui se passe sous le capot est utile !

Objectif du module

Utiliser efficacement le système d'exploitation

Contenu et objectifs du module

- Grandes lignes du fonctionnement d'un système d'exploitation (OS)
  Focus sur UNIX (et Linux) par défaut, mais généralisations
- Concepts clés des OS : processus, fichier, édition de liens, synchronisation
- Utilisation des interfaces système : programmation pratique, interface POSIX
- Programmation système (et non programmation interne *du* système)
  Plutôt du point de vue de l'utilisateur (conception d'OS en RSA)

Motivations

- OS = systèmes complexes les plus courants ; Concepts et abstractions claires
- Impossible de faire un programme efficace sans comprendre l'OS
- Comprendre ceci aide à comprendre les abstractions supérieures

Prérequis : Pratique du langage C et du shell UNIX

Bibliographie succincte (pour cette partie)

Livres

- Bryant, O'Hallaron : Computer Systems, A Programmer's Perspective.

Autres cours disponibles sur Internet

- **Introduction aux systèmes et aux réseaux (S. Krakowiak, Grenoble)**
  Source de nombreux transparents présentés ici.
  `http://sardes.inrialpes.fr/~krakowia/Enseignement/L3/SR-L3.html/`
- **Programmation des systèmes** (Ph. Marquet, Lille)
  `http://www.lifl.fr/~marquet/cnl/pds/`
- Operating Systems and System Programming (B. Pfaff, Stanford)
  `http://cs140.stanford.edu/`

Sites d'information

- `http://systeme.developpez.com/cours/`
  Index de cours et tutoriels sur les systèmes

URL du cours : `http://members.loria.fr/lnussbaum/rs.html`

Plan de cette partie du module :

Systèmes d'exploitation et programmation système