

Introduction aux systèmes GNU/Linux

Module IntroLinux – S25E01

Philippe Latu / Université de Toulouse

inetdoc.net



Objectif DevOps

De l'idée à la production.

→ Un flux continu de valeur !

Créer un pipeline fluide et continu de la conception à la mise en production, en mettant l'accent sur la livraison rapide de valeur.

Le plan

- Progression sur 2 ans
- Concepts Unix et GNU/Linux
- Modélisation d'un système GNU/Linux

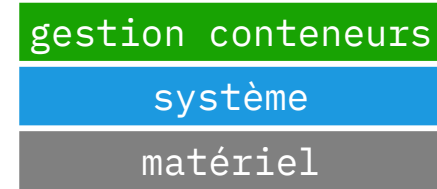
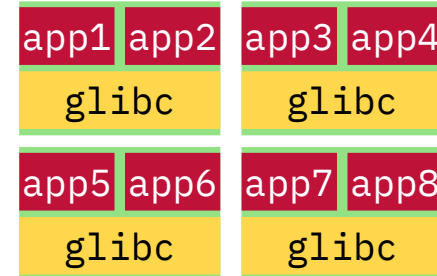
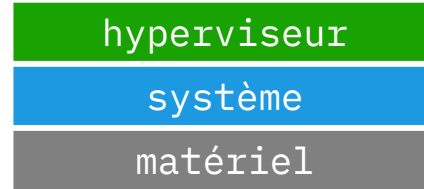
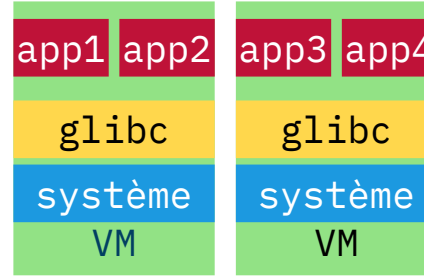


Progression sur 2 ans

3ème année de licence + 1ère année de master

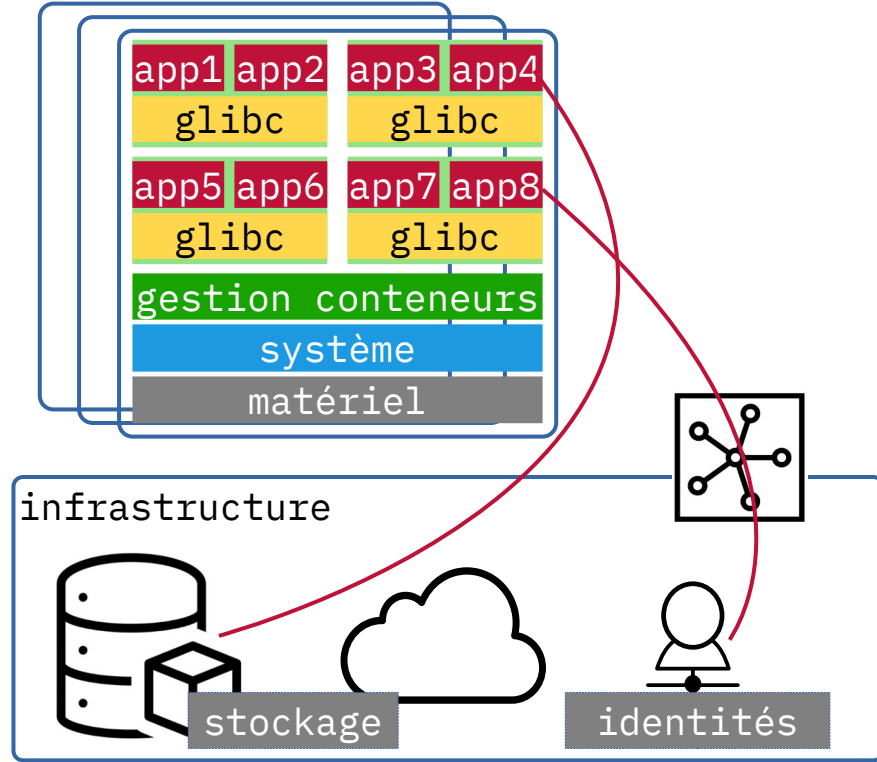
Progression sur 2 ans - L3

- Administrer un système
 - Identifier les composants d'une distribution Linux : Debian
 - Gérer des paquets
 - Gérer des conteneurs système
 - Gérer des machines virtuelles
 - Créer et personnaliser des comptes utilisateurs locaux
 - Identifier les ressources systèmes
 - Gérer les processus



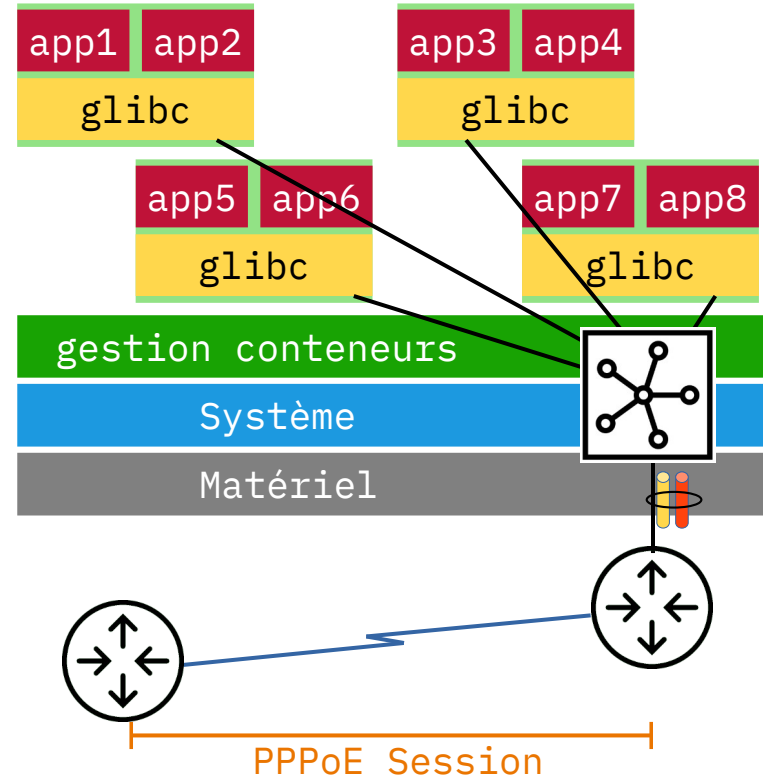
Progression sur 2 ans - M1

- Administrer des systèmes
 - Définir les types de stockage
 - Choisir et configurer une solution de stockage réseau iSCSI ou NFSv4
 - Installer et configurer un annuaire LDAP
 - Associer identité et système de fichiers



Progression sur 2 ans - M1

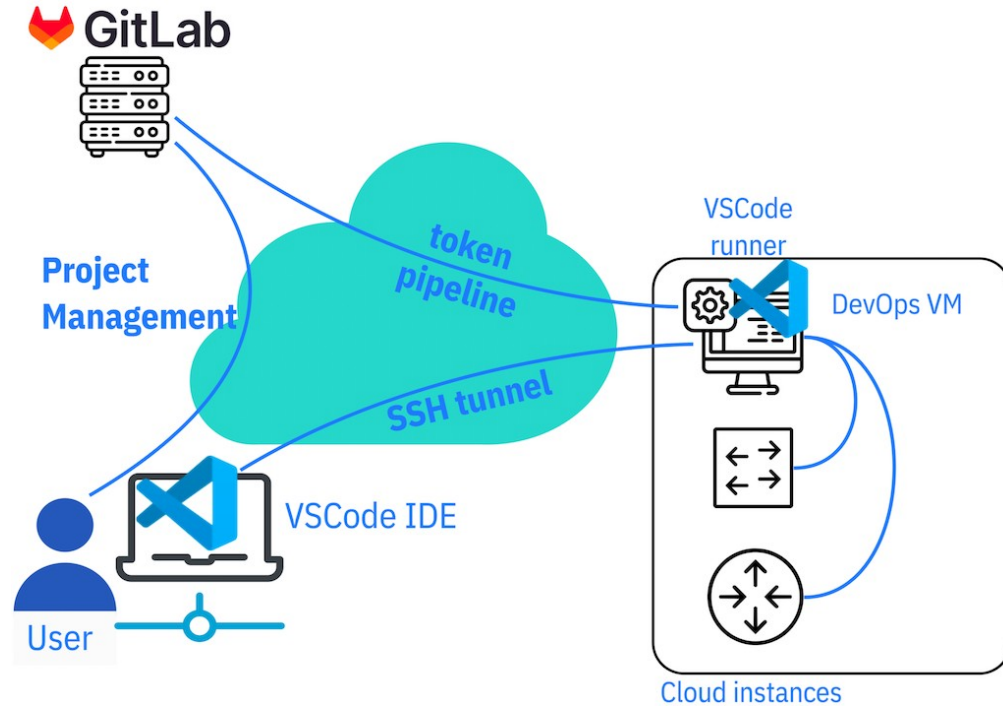
- Interconnecter des réseaux hétérogènes
 - Configurer une liaison WAN PPPoE
 - Utiliser le routage inter-VLAN
 - Configurer un réseau de conteneurs
 - Configurer le protocole de routage dynamique OSPF



Progression sur 2 ans - M1

- Automatisation

- Identifier les modèles de conception et de développement
- Utiliser des APIs REST en Python
- Déployer des conteneurs et des routeurs avec Ansible
- Construire des pipelines CI/CD avec GitLab



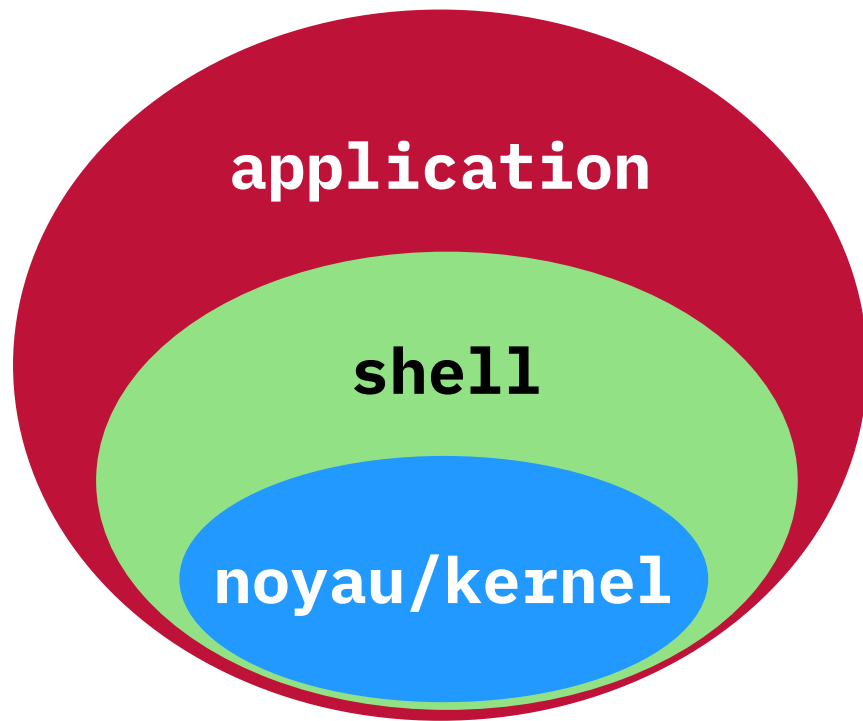
Progression de ce module

3ème année de licence

3 séances de cours

- Cours 1 & 2

- Définir les 5 fonctions de base des systèmes Unix
- Définir les 3 couches de la modélisation d'un système d'exploitation
- Définir les types de virtualisation et de conteneurs
- Définir une distribution GNU/Linux
- Identifier les licences

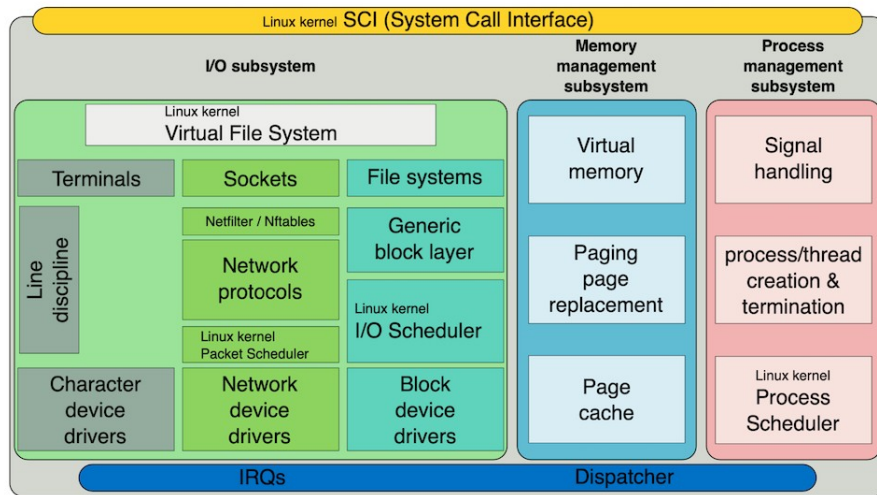


3 séances de cours

- Cours 3

- Décrire les protocoles des environnements graphique
- Définir les fonctions du shell BASH
- Définir les permissions sur les fichiers Unix
- Décrire les outils de l'initialisation système

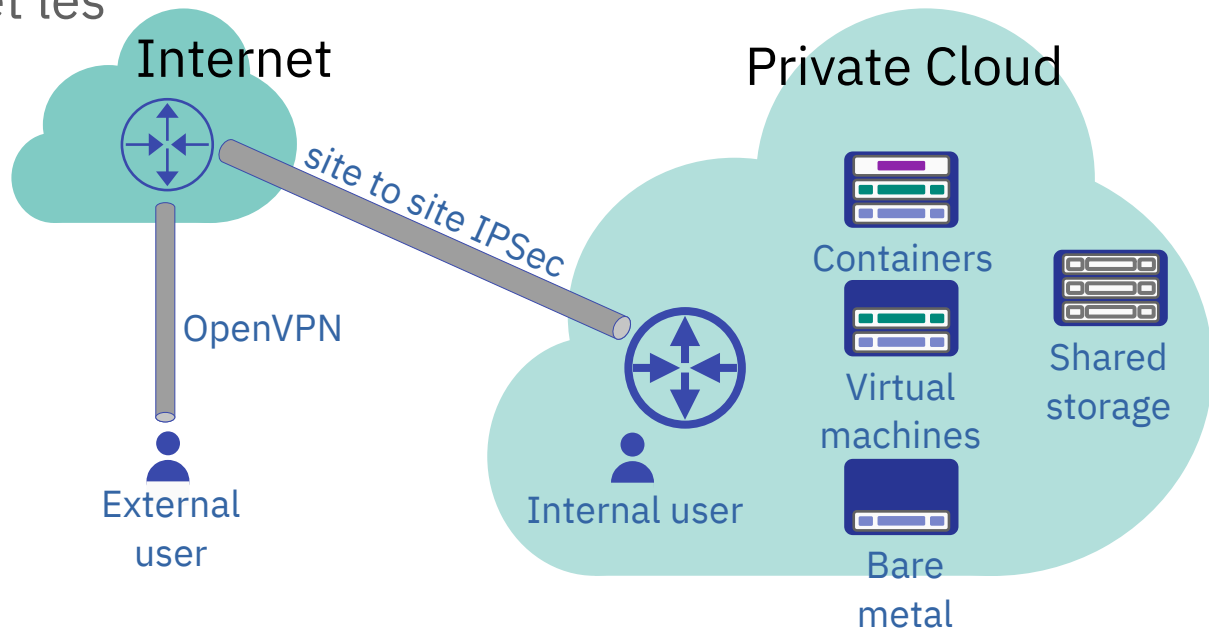
Initialisation système – Linux Kernel



Accéder au cloud privé de la formation

- Travaux Pratiques 1

- Identifier les types et les modes d'accès aux "Clouds"
- Utiliser un VPN
- Lancer une machine virtuelle
- Configurer un gestionnaire de conteneurs



Gérer les paquets et les conteneurs

- Travaux Pratiques 2

- Définir les fonctions clés de la gestion de paquets
- Utiliser l'Advanced Package Tool → apt
- Utiliser les outils de la gestion de paquets : dpkg, aptitude
- Installer et configurer le gestionnaire de conteneurs Incus

```
Actions Annuler Paquet Solutions Rechercher Options Vues Aide
C-T: Menu ?: Help q: Quit u: Update g: Preview/Download/Install/Remove Pkgs
aptitude 0.8.13 @ vm000
--\ Paquets pouvant être mis à jour (5)
--- libs      Ensembles de routines logicielles (4)
--- web       Navigateurs, serveurs, serveurs mandataires et outils pour le web (1)
--- Nouveaux paquets (384)
--- Paquets installés (495)
--- Paquets non installés (68166)
--- Paquets virtuels (69203)

Les paquets de la section « web » comprennent des navigateurs Web, des serveurs Web, des serveurs mandataires, des logiciels pour écrire des scripts CGI ou des programmes basés sur le Web, des programmes pré-écrits pour le Web et d'autres logiciels pour le World Wide Web.

Ce groupe contient 1 paquet.
```

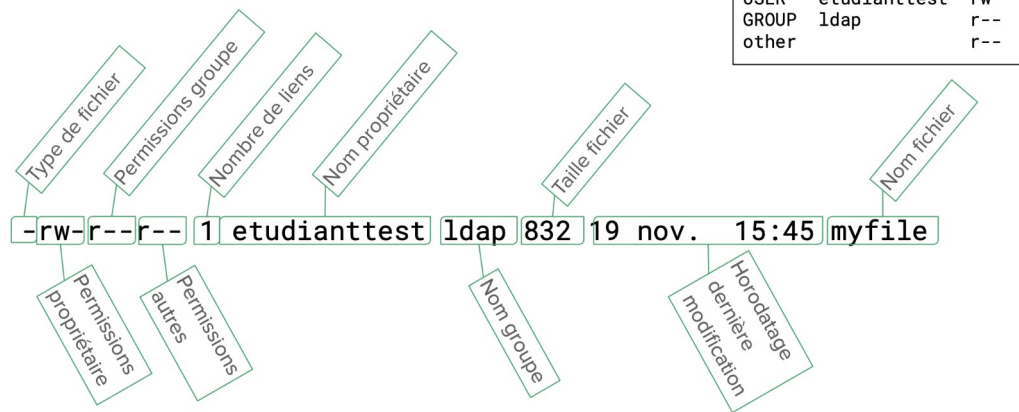
Gérer les processus et le permissions

• Travaux Pratiques 3

- Présenter les environnements graphiques
- Utiliser le shell Bash
- Gérer les processus
- Gérer les permissions Unix
- Définir les processus d'initialisation système

Permissions Unix – tous les attributs

Détail des attributs d'un fichier



Concepts Unix et GNU/Linux

« Unix est fondamentalement un système d'exploitation simple, mais il faut être un génie pour comprendre sa simplicité. »

Dennis Ritchie

Définir la notion de culture...

« un ensemble lié de manières de penser, de sentir et d'agir plus ou moins formalisées qui, étant apprises et partagées par une pluralité de personnes, servent, d'une manière à la fois objective et symbolique, à constituer ces personnes en une collectivité particulière et distincte. »

Guy Rocher – <https://fr.wikipedia.org/wiki/Culture>

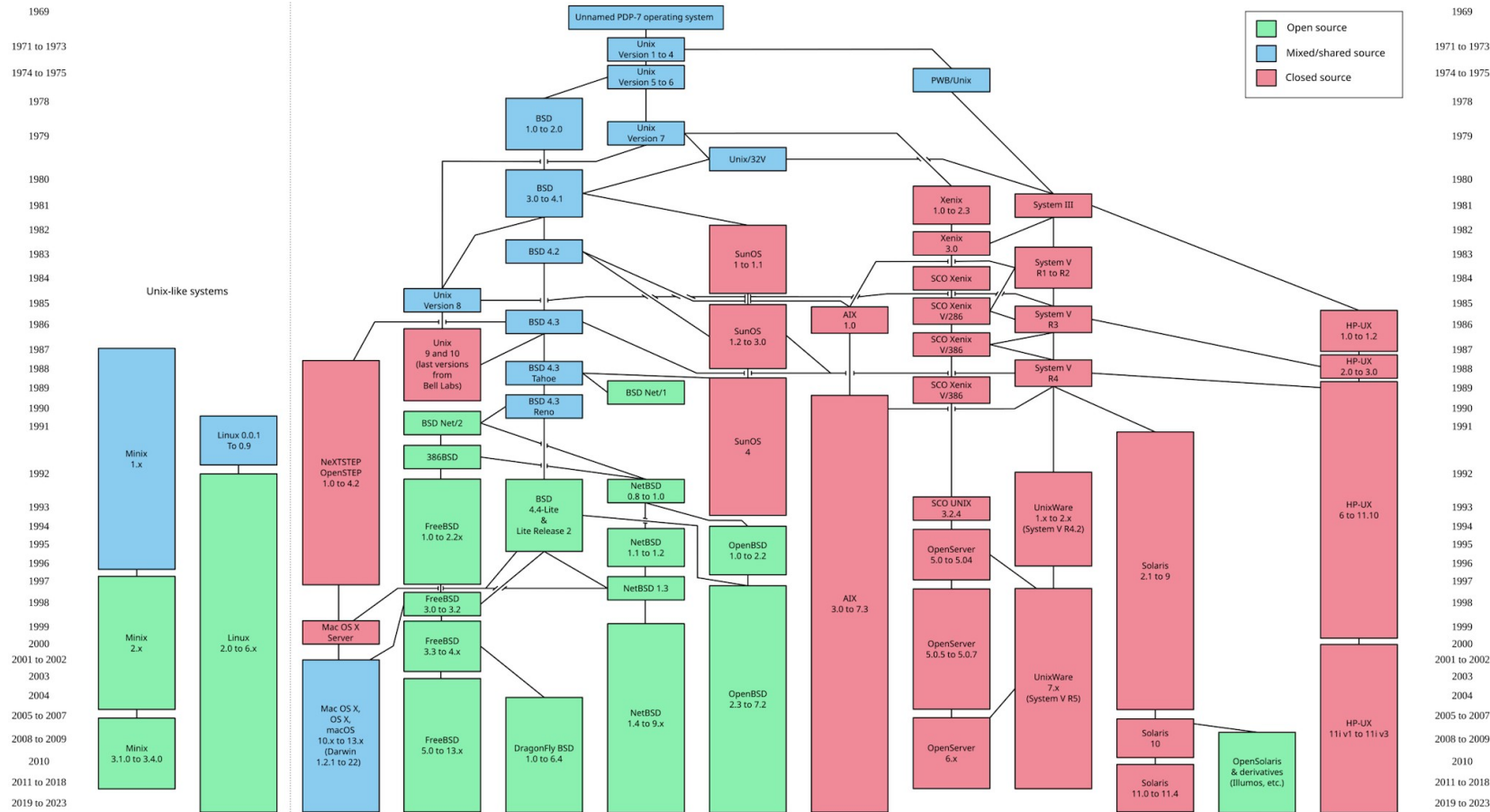
Développer une culture système, c'est...

- Identifier les étapes d'une histoire continue sur 5 décennies
- Comparer la genèse des systèmes Unix avec celle des services Internet
- Identifier les processus d'assurance qualité des distributions
- Décrire les nouveaux processus métier → **DevOps !**

Les 5 fonctions de base des systèmes Unix

- **Multi-tâches**
 - Temps processeur partagé entre plusieurs programmes
- **Multi-utilisateurs**
 - Ressources système partagées entre plusieurs utilisateurs
- **Portabilité**
 - Outils système partagés entre cibles matérielles différentes
- **Bibliothèques de développement standard**
 - Optimiser la qualité des développements en partageant le code source
- **Applications communes**
 - Services système, services Internet, etc.

L'histoire Unix



1969 – Unics – AT&T – System V

- Unix est un système «accidentel»
 - AT&T Bell labs – Ken Thompson - Dennis Ritchie
 - 1973 réécriture en Langage C
 - Diffusion sous licence AT&T incluant la totalité du code source
 - 1975 publication
RFC681 NETWORK UNIX
- Apparition des Unix constructeurs
 - Coût matériel + licence prohibitif
 - Versions constructeurs incompatibles entre elles



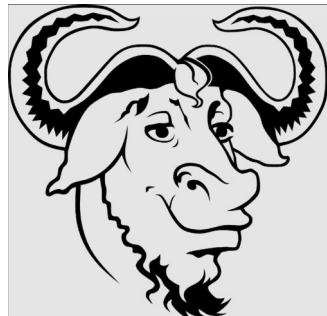
1977 – Berkeley University – BSD

- Branche Unix lancée à partir d'une licence AT&T
 - Nombreuses améliorations
 - Gestion mémoire
 - Sous-système réseau TCP/IP
 - Diffusion entre universités
 - Début de l'Internet universitaire
- Procès AT&T vs Berkeley
 - BSD → système complet autonome
 - Éclatement de la branche BSD
 - FreeBSD, NetBSD et OpenBSD



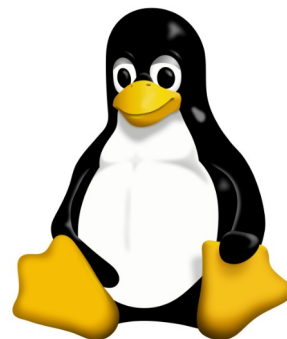
1984 – GNU – Not Unix

- Projet lancé par Richard Stallman → 2 objectifs
- Promouvoir les logiciels libres
 - Protection des travaux des développeurs à l'aide de licences
 - Fédérer les développements libres
 - Applications GNU
- Utiliser les fonctions Unix comme modèle
 - Fonctions de base déjà éprouvées
 - 1990 – chaîne de développement stable
 - GNU Compiler Collection



1991 – Début du noyau Linux

- Développement initié par Linus Torvalds
 - «divergences de vues» avec A.S. Tanenbaum
 - <http://www.oreilly.com/catalog/opensources/book/appa.html>
- Fonctions de base Unix + spécificités
 - Multi-tâches
 - Multi-utilisateurs
 - Gestion mémoire
 - Répétitive et étendue → mémoire virtuelle
 - Isolation entre les processus



Modéliser le système GNU/Linux

Le modèle en 3 couches - 1/2

Noyau/kernel → 3 fonctions

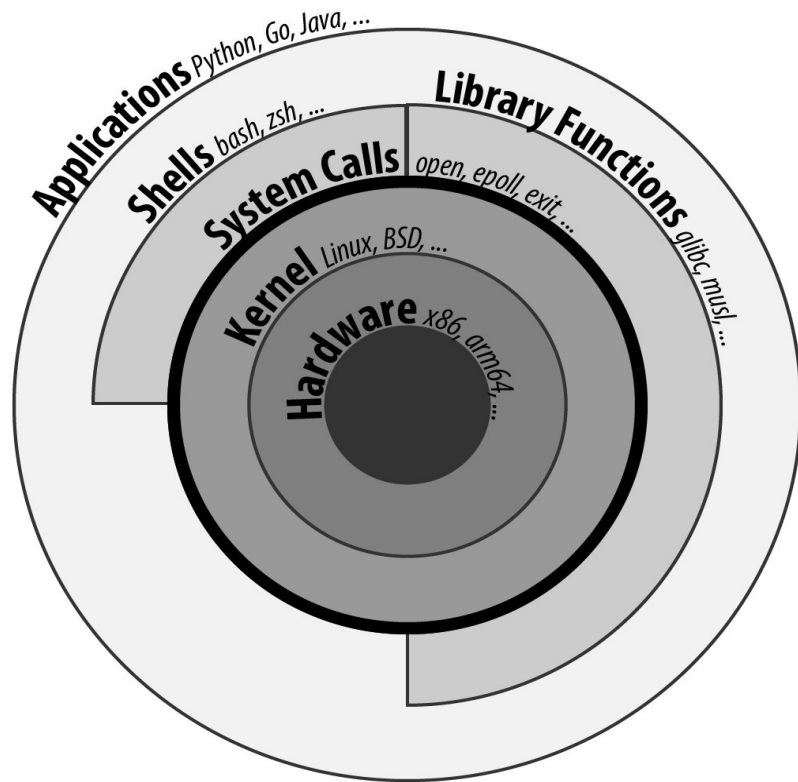
- Temps processeur
- Mémoire
- Entrées/Sorties

Shell → 2 fonctions

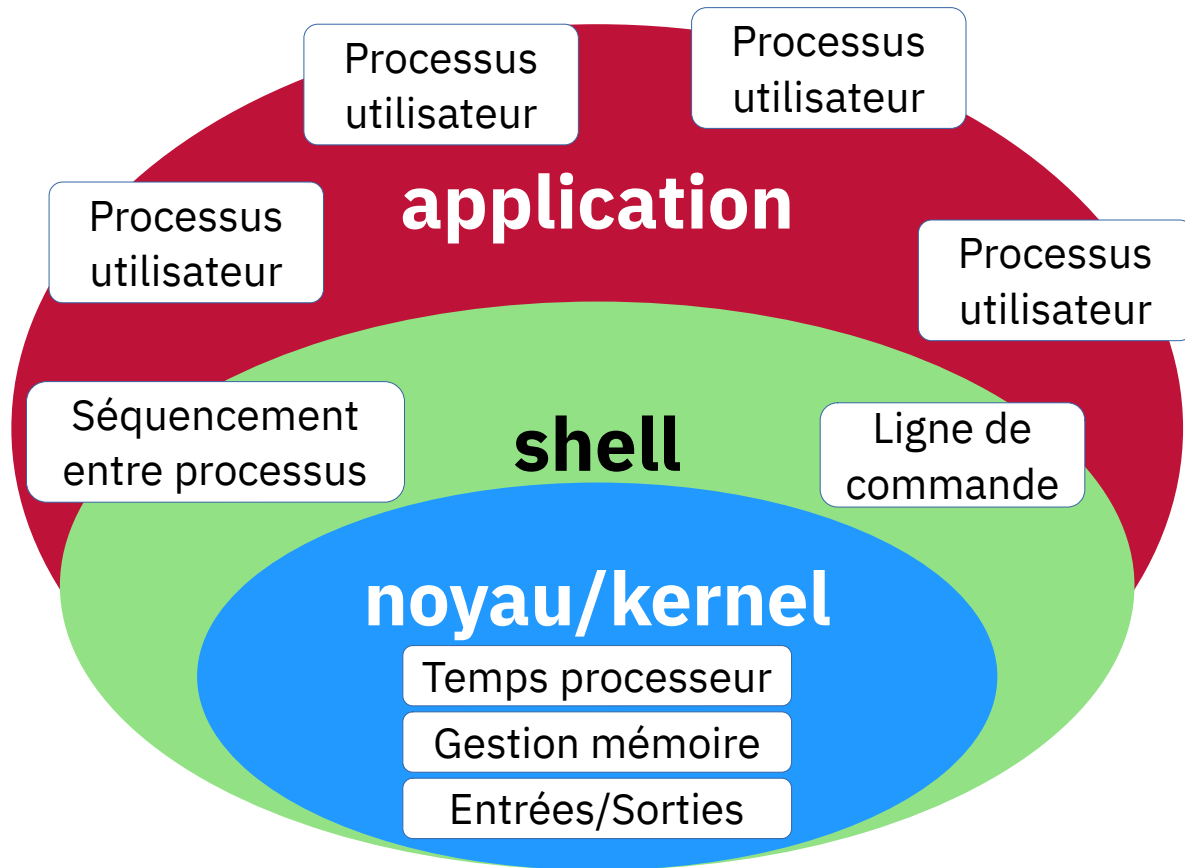
- Séquencement entre processus
- Ligne de commande

Application

- Tous les processus utilisateur

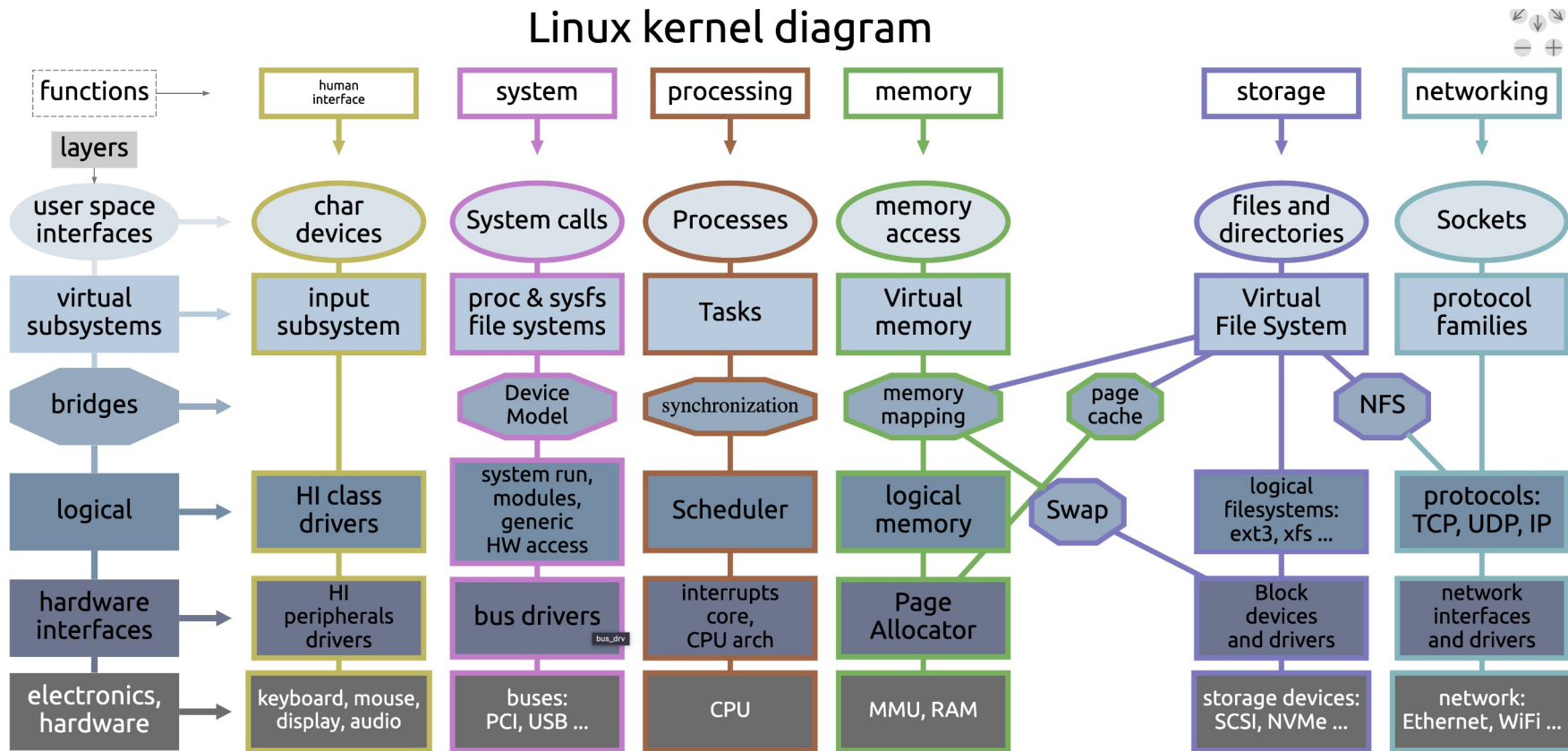


Le modèle en 3 couches - 2/2



Linux Kernel diagram

Linux kernel diagram



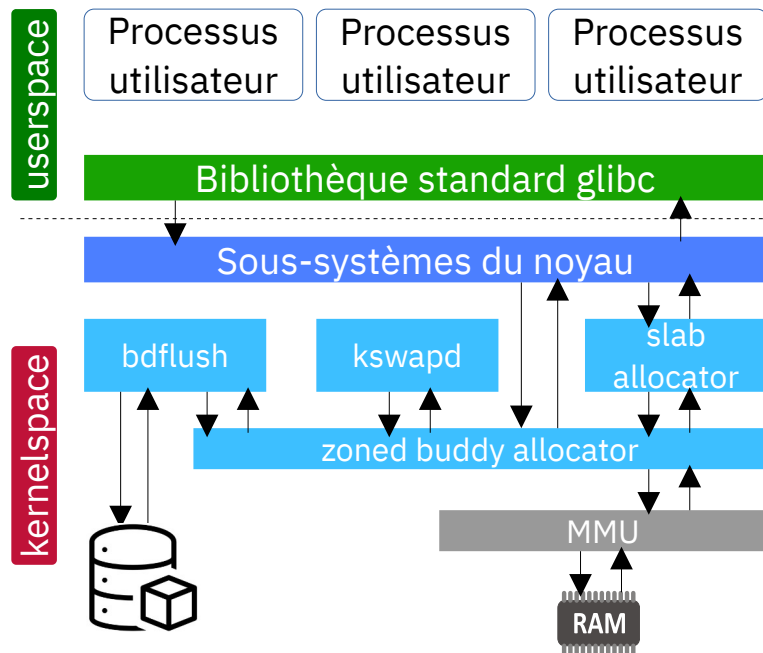
Définir la notion de mémoire virtuelle

userspace

- Programmes utilisateurs
- Bibliothèque standard glibc

kernelspace

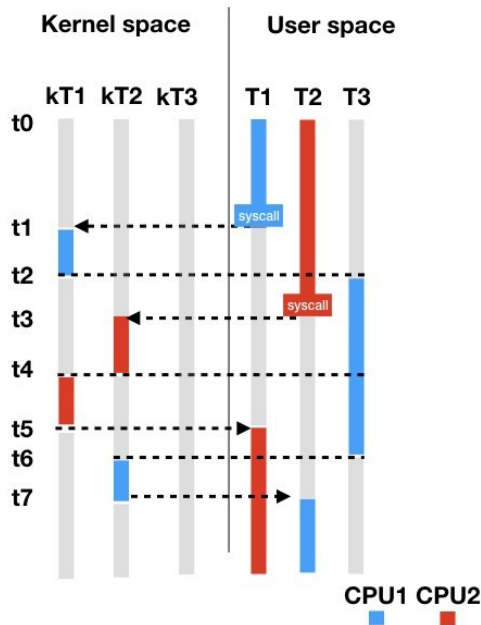
- Memory Management Unit (MMU)
- Zoned buddy allocator
 - Allocation pages mémoires
- Slab allocator
 - Cache dans les pages mémoire
- Kernel threads
 - Réutilisation de la mémoire



Définir la notion d'ordonnanceur (*scheduler*)

3 domaines ou types de tâches

- Domaine temps réel
 - Contraintes de temps d'exécution
 - Fréquence d'exécution garantie
- Domaine entrées/sorties
 - Attente de disponibilité des périphériques
- Domaine CPU
 - Temps consacré aux calculs
 - Tranche de temps CPU / time slice
 - Durée d'exécution d'un processus sur un cœur



Définir la notion d'ordonnanceur (*scheduler*)

Tranche de temps CPU (*time slice*)

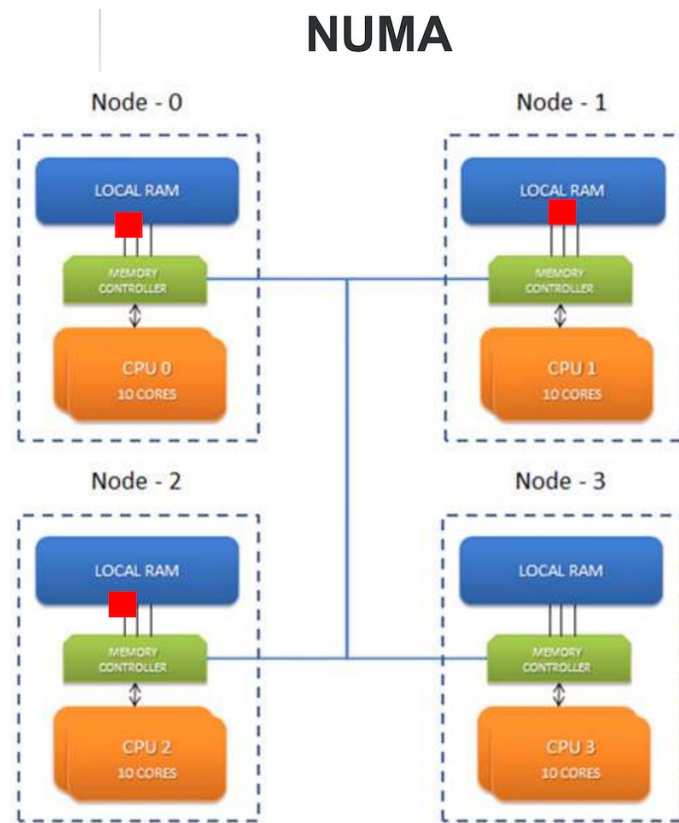
- Durée d'exécution d'un processus sur un cœur

Préemption

- Interruption d'un processus par un second de priorité plus élevée

NUMA (*Non-Uniform Memory Access*)

- Placer la mémoire avec les processeurs
- Éviter les variations sur les temps d'accès



Bilan du cours 1

Retenir les points clés

Les systèmes GNU/Linux → 5 fonctions de base Unix

- Multi-tâches
- Multi-utilisateurs
- Portabilité
- Bibliothèques de développement standard
- Applications communes

Retenir les points clés

Le modèle en 3 couches d'un système GNU/Linux

- Noyau/kernel (gestion des ressources)
- Shell (interface utilisateur)
- Applications (processus utilisateur)

Décrire les concepts clés

Les points forts du noyau Linux

- Mémoire virtuelle
- Ordonnanceur (scheduler)
- Prémption