

[inetdoc.LINUX]

<http://www.linux-france.org/prj/inetdoc>

Exploration GNU/Linux - Séance 1

Historique Unix, GNU et Linux

Définitions et Fonctions d'un noyau

Projets et Méthodes Open Source



Philippe Latu

philippe.latu@linux-france.org

IUT 'A' Paul Sabatier - STRI

Présentation

- **Fonctions et Responsabilités**
 - 1992 - Enseignant IUT 'A' Paul Sabatier
 - ▶ Département Génie Électrique & Informatique Industrielle
 - 1996 - Chargé de mission système & réseau IUT
 - 2000 - Premiers enseignements STRI
- **Expérience GNU/Linux**
 - 1996 - Premiers serveurs GNU/Linux IUT
 - ▶ Services Internet : dns, smtp, http
 - 1997 - Documentation RNIS/ISDN Linux France
 - 2000 - Projet inetdoc.LINUX
- **Expérience Cisco Networking Academy**
 - 2001 - Instructeur CCNA
 - 2005 - CCNA CCAI Certified 640-801
 - 2008 - Instructeur CCNP BSCI

Progression pédagogique STRI

- Initiation au système GNU/Linux
 - L3 - semestre 1 - 19 heures
- Administration système en réseau
 - L3 - semestre 2 - 24 heures
- Interconnexion réseau
 - M1 - semestre 1 - 24 heures
- Sécurité des systèmes d'information
 - M2 - semestre 1 - 24 heures
- Ressources en ligne
 - <http://www.linux-france.org/prj/inetdoc/cours/>

Objectifs

- Installer et utiliser un système GNU/Linux
 - mode console et mode graphique
- Exploiter les ressources documentaires
- Rechercher et choisir des (services|applications)
- Maintenir une collection de paquets
- Consulter les messages systèmes

Planning

- Séance 1 - Initiation au système GNU/Linux
 - Présentation progression
 - Concepts Unix, GNU/Linux, projets Open-Source
 - Paquets applicatifs & Distributions
 - Méthode de travail
- Séance 2 - Installation du Système GNU/Linux
 - Partitionnement disque dur
 - Installation du système de base
 - choix des paquets d'applications

Planning (suite)

- Séance 3 - Configuration du Système GNU/Linux
 - Interface graphique KDE
 - Gestion des paquets
 - Droits sur les fichiers et répertoires
- Séance 4 - Administration premier niveau
 - Ressources du shell BASH
 - Compilation des sources d'une application
 - Systèmes de fichiers et montages

Planning (suite)

- Séance 5 - Services du systèmes GNU/Linux
 - Gestion et personnalisation des comptes utilisateurs
 - Messages systèmes (syslog)
 - Exécution périodique (cron)
- Séance 6 - Initialisation du système GNU/Linux
 - Lancement du système
 - Niveaux de démarrage
 - Détection des périphériques
 - Ajout/retrait de pilote de périphérique
- Évaluation
 - Questionnaire à choix multiple + test d'aptitude pratique

Concepts Unix & GNU/Linux

- Pourquoi étudier le logiciel libre ?
 - Histoire cohérente et continue
 - ▶ incompatible avec le marketing propriétaire !
 - Alternative et analyse comparative critique
 - ▶ transparence et démarche qualité
 - Acquisition de méthodes et de compétences originales
 - ▶ administration et volumes
 - ▶ architectures «multi-tier»
 - Autonomie et indépendance
 - ▶ maîtrise des coûts
 - ▶ maîtrise de la conception

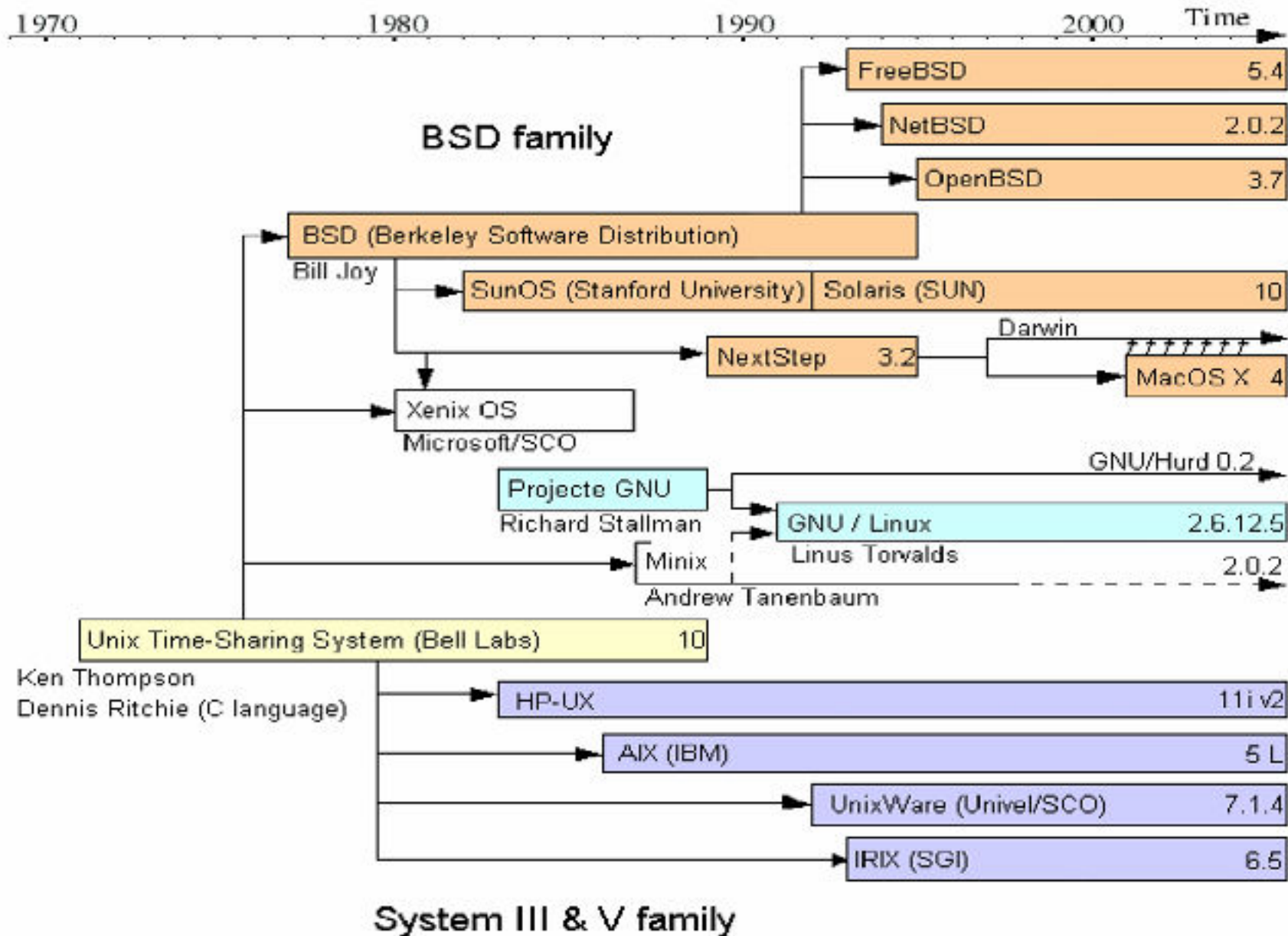
Concepts Unix & GNU/Linux

- Objectifs.
 - Identifier les fonctions de base des systèmes Unix
 - Connaître les étapes majeures du développement des systèmes Unix
 - Différencier les fonctions du système GNU/Linux
 - Argumenter sur le choix des différentes licences

Concepts Unix & GNU/Linux

- 5 fonctions de base des systèmes Unix
 - Fonctionnement multi-tâches
 - exécution simultanée de plusieurs programmes
 - Fonctionnement multi-utilisateurs
 - système partagé entre plusieurs utilisateurs
 - Portabilité
 - utilisation des mêmes outils sur différentes machines
 - Applications
 - outils, services Internet, etc.
 - Partage des bibliothèques de développement d'applications
 - optimisation des développements en partageant le code

Concepts Unix & GNU/Linux



1969 - AT&T = Système V

- Unix est un système «accidentel»
 - Laboratoires Bell d'AT&T
 - Réglementation U.S. antitrust avant 1980
 - Vente de logiciel interdite pour AT&T
- AT&T a fourni des licences pour le code source
 - Utilisation du nom Unix autorisée
- Unix propriétaires
 - Segmentation et parts de marché captives
 - Versions Unix incompatibles entre elles
 - Discontinuités nuisibles à la croissance d'Unix

1973 - Berkeley University = BSD

- Branche BSD lancée à partir d'une licence AT&T
- Accès libre au code source
 - Nombreuses améliorations
 - Dynamique de développement
 - Services de communication réseau de l'ARPANET
 - Branche système V d'AT&T dépassée par les développements universitaires
- Université de Berkeley attaquée par AT&T
 - Procès entre 1984 et 1994
 - Éclatement de la branche BSD sous la pression judiciaire
 - FreeBSD, NetBSD et OpenBSD

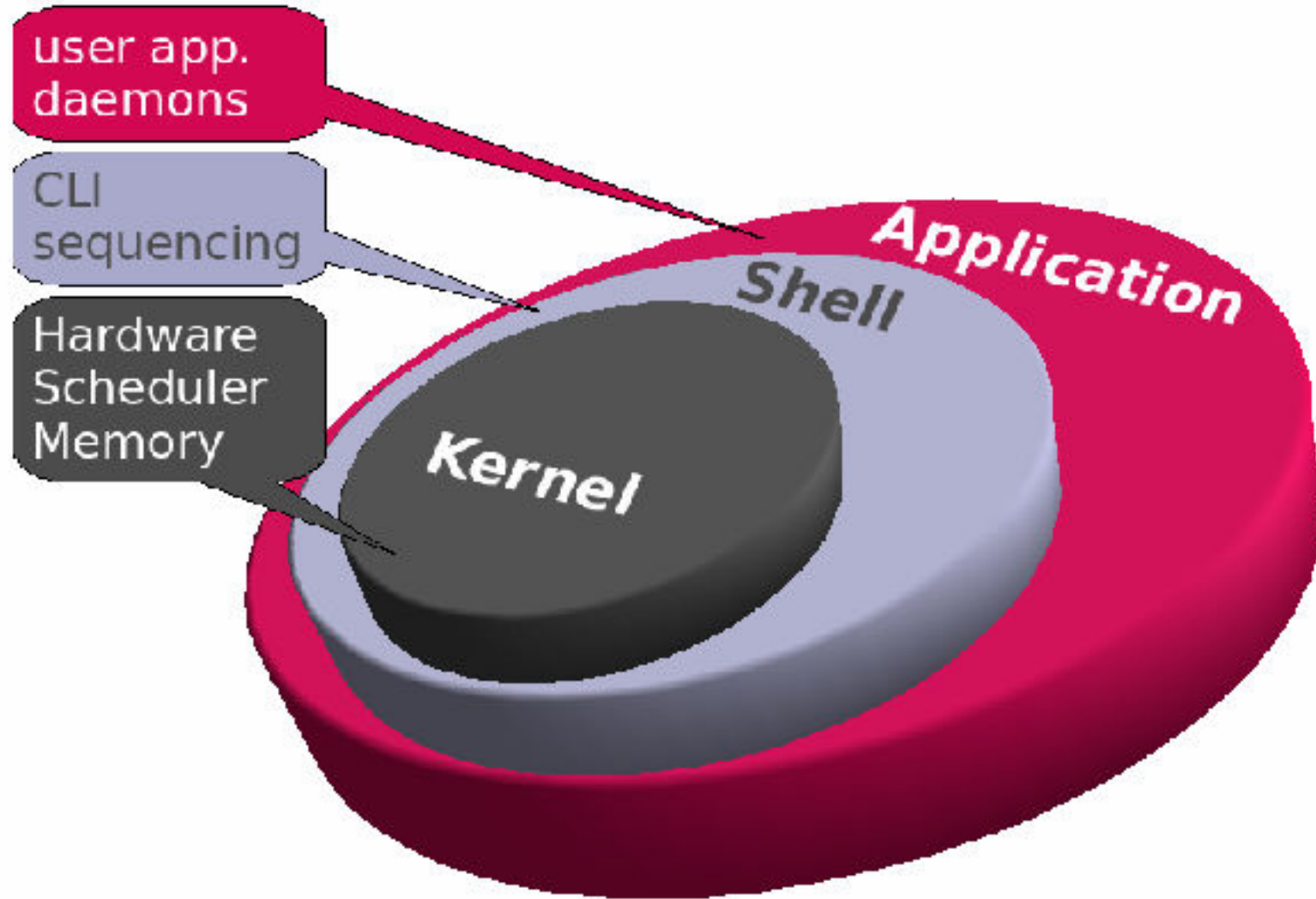
1984 - GNU = Not Unix

- Projet lancé par Richard Stallman (MIT/Emacs)
- 2 objectifs :
 - Promouvoir le développement coopératif
 - ▶ protéger les droits des développeurs
 - ▶ licences GPL, LGPL et GFDL
 - Fédérer les développements libres : applications GNU
- Unix choisi comme modèle
 - Fonctions de base déjà opérationnelles
 - Chaîne de développement stable vers 1990
 - ▶ GNU Compiler Collection : glibc, cpp, gcc, make
 - Pièce essentielle manquante en 1990 : le noyau
 - Arrivée d'un projet de noyau Unix libre sur machines Intel
 - ▶ Linux de Linus Torvalds

1991 - Début du noyau Linux (kernel)

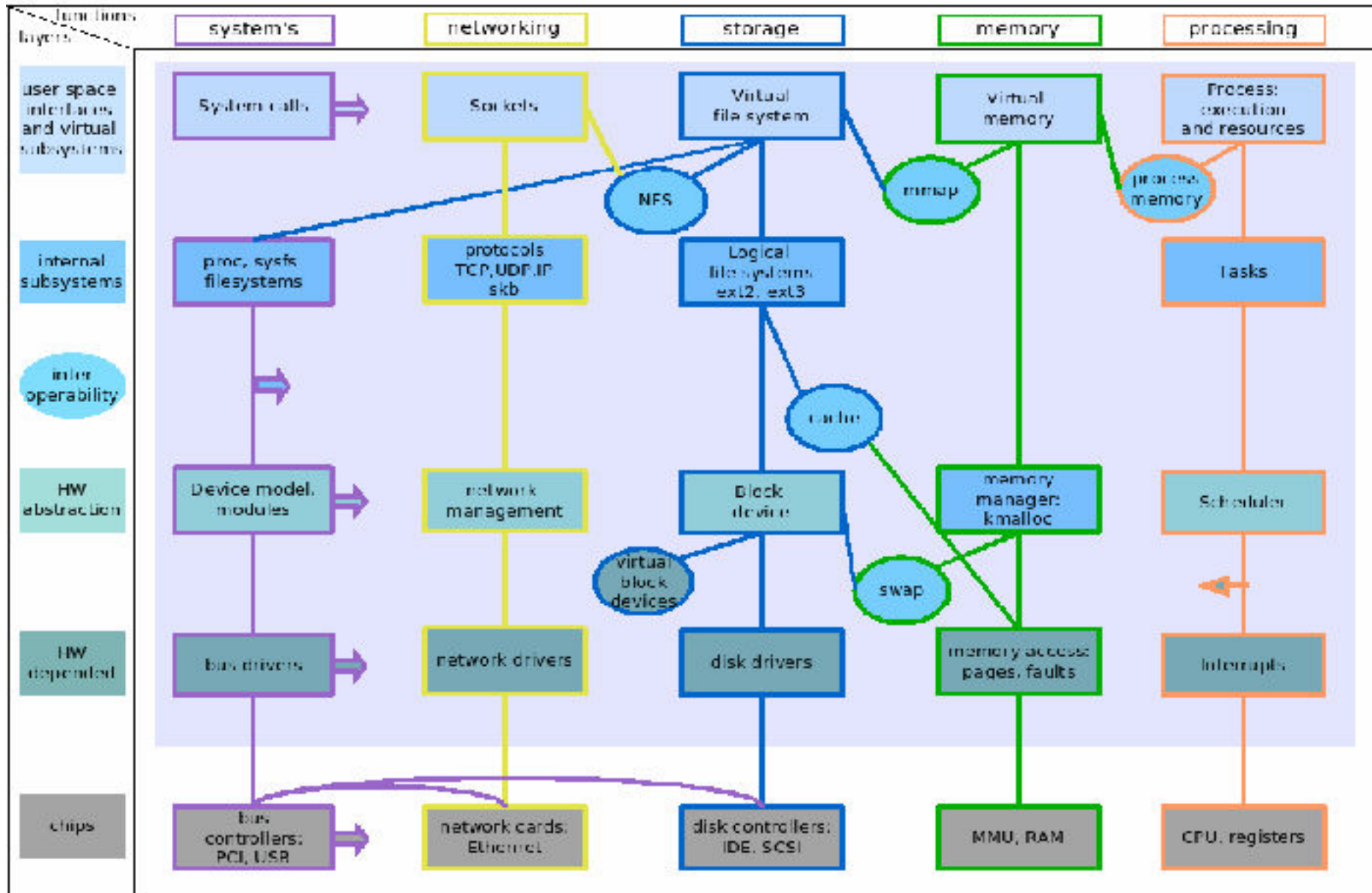
- Développement initié par Linus Torvalds
 - «divergences de vues» avec A.S. Tanenbaum
 - ▶ correctifs sur le noyau Minix refusés
 - ▶ <http://www.oreilly.com/catalog/opensources/book/appa.html>
- Fonctions de base Unix plus quelques spécificités
 - Fonctionnement multi-tâches
 - Fonctionnement multi-utilisateurs
 - Gestion de mémoire virtuelle
 - ▶ utilisation RAM répétitive et étendue
 - Système d'interconnexion réseau
 - Mode protégé
 - ▶ contrôle des accès mémoire
 - Bibliothèques partagées
 - ▶ code partagé par les applications

Noyau Linux : OS



Noyau Linux : architecture

Simplified Linux kernel diagram in form of a matrix map



Noyau Linux : Ordonnanceur/Scheduler

- 3 domaines ou types de tâches
 - Domaine temps réel
 - Contraintes de temps élevée
 - Garantie sur la fréquence d'exécution
 - Priorité la plus élevée
 - Domaine Entrées/Sorties
 - Temps consommé à attendre les périphériques
 - Domaine CPU
 - Consommation en puissance de calcul
- Tranche temps (time slice)
 - Durée d'exécution d'un processus / processeur
- Prémption
 - Interruption d'un processus
par un autre de priorité plus élevée

Noyau Linux : débuts

- 1994 - 1996 = Série 1.0
 - Extension du catalogue des processeurs
 - Alpha, Sparc, Mips et PowerPC
- 1996 - 1999 = Série 2.0
 - Reproduction des fonctions Unix de base
 - Support de nouveaux processeurs
 - Plus de pilotes de périphériques
 - Interconnexion réseau 1ère génération
- 1999 - 2001 = Série 2.2
 - Stabilisation des fonctions Unix de base
 - Toujours plus de pilotes de périphériques
 - Interconnexion réseau 2ème génération

Noyau Linux : 2001 - 2003 = Série 2.4

- **Nouveautés orientées grands systèmes**
 - **Support de la répartition de charge**
 - ▶ Architectures multiprocesseurs SMP et NUMA
 - ▶ Stockage : LVM & RAID
 - **Nombre de processus simultanés / taille mémoire RAM**
 - ▶ Services Web & transactionnels
 - **Support mémoire plus important**
 - ▶ RAM jusqu'à 64 GB et fichiers de taille > 1 To
 - **Support des systèmes de fichiers journalisés**
 - ▶ évolution ext2 (ext3), IBM Journalling Filesystem (JFS), SGI (XFS), etc.
 - **Système d'interconnexion réseau original**
 - ▶ Traitements des flux réseaux modulaires et génériques
 - ▶ QOS, filtrage de trames et de paquets

Noyau Linux : 2001 - 2003 = Série 2.4

- Nouveautés orientées petits systèmes

- Memory Technology Devices

- ▶ accès mémoires ROM ou FLASH sur systèmes embarqués
- ▶ Stockage équivalent disque dur

- Systèmes de fichiers CRAMFS

- ▶ Optimisation systèmes embarqués limités en RAM

- Bus I2O, USB, FireWire, etc.

- Multimédia

- ▶ vidéo, audio et systèmes de codage originaux

- Pour en savoir plus

- «Wonderful World of Linux 2.4»

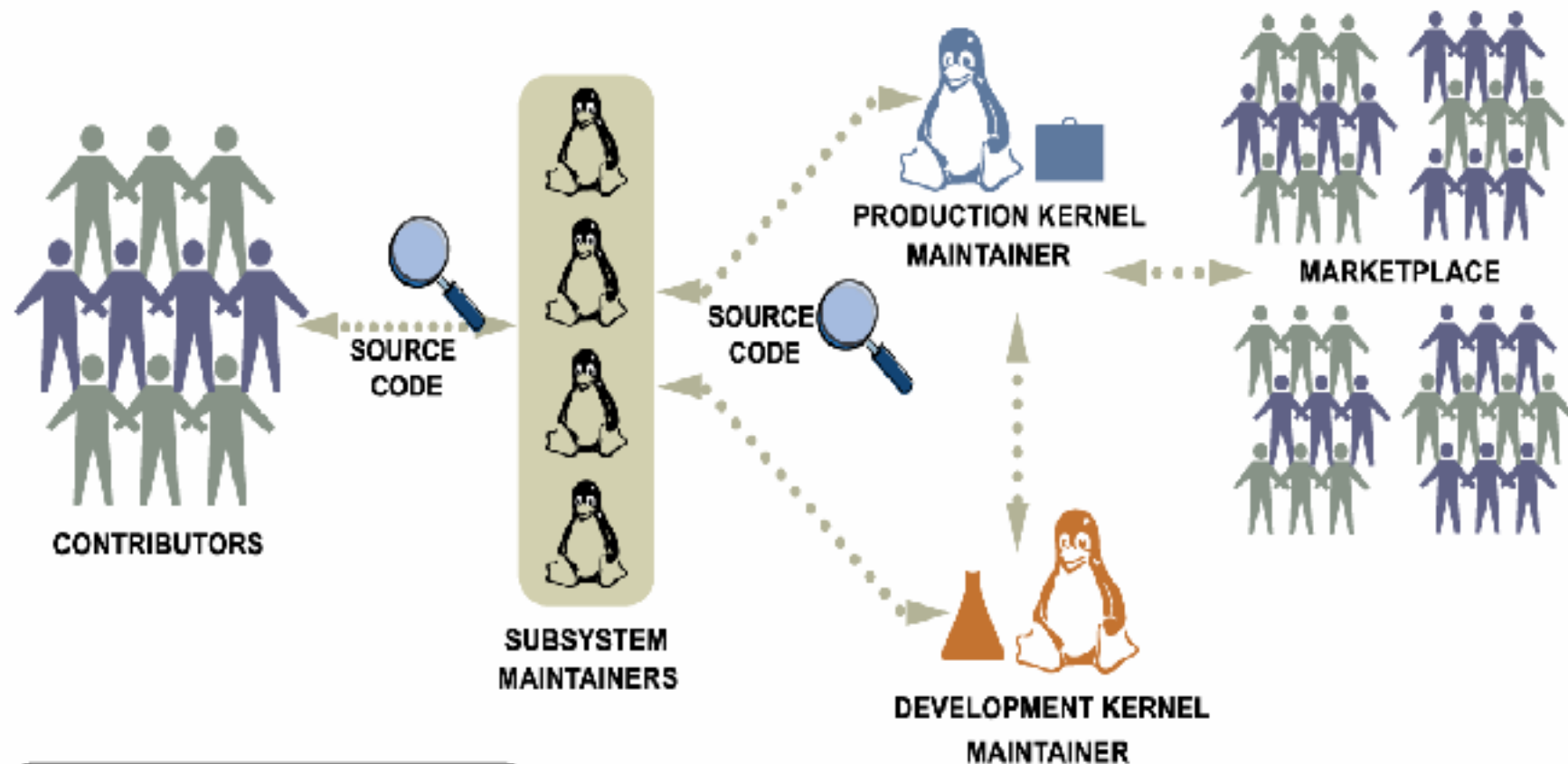
- ▶ <http://kniggit.net/wwol24.html>


Noyau Linux : 2004 - 200x = Série 2.6

- Les nouveautés orientées grands systèmes
 - Complete Fair Queuing I/O scheduler
 - Contrôle de priorité sur les E/S des processus
 - Native POSIX Thread Library (NTPL)
 - Partage des ressources entre processus
 - O(1) process scheduler
 - Affectation temps processeur / processus
 - Non-Uniform Memory Access (NUMA)
 - Optimisation temps d'accès mémoire sur systèmes multiprocesseurs
- Les nouveautés orientées petits systèmes
 - Intégration des travaux du projet uClinux
 - <http://www.uclinux.org>
 - Introduction de nouveaux processeurs spécialisés
- <http://kernelnewbies.org/LinuxChanges>

Noyau Linux : 2004 - 200x = Série 2.6

LINUX KERNEL DEVELOPMENT PROCESS



 Ongoing peer review of code
Continuously available online
for public review

Licences logiciels libres

- Licence BSD = restrictions possibles
 - Création de versions propriétaires autorisée
 - Restrictions possibles sur les droits de redistribution
 - Restrictions rarement appliquées dans les faits
- Licence GNU = Copyleft
 - Copyright != Copyleft
 - Principe de protection du Logiciel libre et des concepteurs
 - Restrictions interdites sur les conditions de redistribution

Logiciel Libre : libertés des utilisateurs

- Usage des logiciels
- Étude du fonctionnement et adaptation
- Redistribution
- Amélioration et distribution de ces améliorations pour le bénéfice de toute la communauté

Logiciel Libre : application du Copyleft

- 1. Appliquer un copyright sur le logiciel
- 2. Fixer les conditions de distribution
 - Donner à tout utilisateur le droit d'utiliser, de modifier et de redistribuer le programme sans changer les conditions de distribution
- Le code source et les libertés associées deviennent légalement inséparables
 - <http://www.gnu.org/copyleft/copyleft.fr.html>

Concepts Unix & GNU/Linux

- Quizz

- Qui est à l'origine de la branche Système V d'Unix ?
- Qui est à l'origine de la branche BSD d'Unix ?
- Quels sont les objectifs du projet GNU ?
- Qu'est-ce que Linux ?
- Qu'est-ce qui différencie les systèmes GNU/Linux des systèmes propriétaires ?
- Qu'est-ce qui différencie les licences GNU et BSD ?
- Quel est le principe du copyleft ?

Projets Open Source : Objectifs.

- Identification des caractéristiques d'un projet de logiciel libre
- Argumentation sur le choix d'une application de logiciel libre

Projets Open Source

- Applications et systèmes Unix
 - Mêmes évolutions
 - Mêmes processus
 - Mêmes méthodes
- Services Internet
 - Plus grosse base installée de logiciels libres
 - <http://survey.netcraft.com>
 - Noms de domaines : bind
 - Courrier électronique : sendmail, postfix
 - Web: Apache

Projets Open Source

- Terminologie «Open Source»
 - Plus que la simple diffusion du code source
 - Libre redistribution
 - ▶ Licence toujours distribuée avec le code
 - ▶ Pas d'interdiction de vente/don
 - ▶ Pas de limite sur le nombre de copies
 - Travaux dérivés
 - ▶ Modifications distribuées dans les mêmes conditions que l'original
 - Intégrité du code source de l'auteur
 - ▶ Restrictions possibles sur la redistribution des correctifs
 - Pas de discrimination
 - ▶ Sur l'utilisateur
 - ▶ Sur le domaine d'application
 - Pas de contamination d'autres logiciels
 - ▶ Licence sans restrictions sur d'autres logiciels associés

Projets Open Source

- **Modèle de développement**
 - Même mode depuis plus de 30 ans
 - Masse critique de développeurs / vie d'un projet
 - Systèmes GNU/Linux | Unix
 - Plus grande population de développeurs
- **Modèle économique**
 - Modèles d'affaires pour les grands acteurs
 - Situation délicate pour les sociétés moyennes
 - Beaucoup de succès côté fondations
 - Mozilla
- **Références**
 - <http://opensource.org/>
 - <http://groklaw.net/>

Projets Open Source

- Outils communautaires de travail coopératif
 - Modèle de développement
 - ▶ Dépôts de code
 - ▶ Suivis de versions
 - ▶ Suivis de bugs
 - Fermes de compilation
 - ▶ <http://alioth.debian.org/>
 - ▶ <http://savannah.gnu.org/>
 - ▶ <http://sourceforge.net/>
- Répertoires de projets Open Source
 - Mesures de qualité du développement
 - ▶ notation des utilisateurs
 - ▶ vitalité du projet
 - ▶ popularité
 - <http://freshmeat.net/>

Distributions

- Distributions GNU/Linux et BSD
 - Canaux de diffusion du logiciel libre
- Distribution GNU/Linux = association
 - Un noyau
 - Plusieurs shells
 - Cocktail d'applications
- Composants distribués sous forme de paquets
 - Code binaire exécutable
 - Configuration type
- Gestion des paquets
 - Principal enjeu dans la vie d'une distribution

Distributions

- 2 logiques s'opposent
 - Vente = Publier
 - Maximum de nouveautés en un minimum de temps
 - Qualité = Garantir
 - Cohérence
 - Continuité
 - Évolutivité
- Importance équivalente entre
 - Expertise de la mise en paquet
 - Choix de l'application
- Rôle essentiel du responsable d'un paquet
 - Capitalisation des compétences d'exploitation
 - Qualités humaines dans la coordination
 - Démarche qualité lors des évolutions

Distribution : choix

- Deux critères essentiels pour l'exploitation
- Facilité d'adaptation
 - Un contexte = une configuration
 - On ne maîtrise pas la totalité des éléments du système
 - On s'appuie sur l'expérience des développeurs de paquets
- Continuité lors des mises à jour
 - Mises à jours ou corrections sans réinstallation
- Adaptation + évolution continue = garantie
 - Continuité de service
 - Haute disponibilité

Distribution : Debian GNU/Linux

- Contrat social Debian
 - Règles à suivre pour garantir le logiciel libre
 - ▶ http://www.debian.org/social_contract.html
- Principes du logiciel libre selon Debian
 - Définition des caractéristiques des logiciels libres
- Charte Debian
 - Procédures qualité du projet
 - ▶ <http://www.debian.org/devel/index.fr.html>
- Gestionnaire de paquets APT
 - Synthèse de toutes les caractéristiques
 - ▶ Continuité indépendante des versions
 - ▶ Adaptabilité en séparant la configuration de l'application
 - ▶ Automatisation de la publication des correctifs de sécurité

Distribution : Debian GNU/Linux

- Choix pédagogique
 - Mille et une distributions spécialisées
 - Très peu de distributions généralistes
- Debian GNU/Linux = modèle
 - Transparence des processus
 - Qualité
 - Sécurité
- Processus métiers difficiles à illustrer
 - Coût d'accès au support
 - Information cloisonnée
 - Communauté réduite

Synthèse projets Open Source

- Logiciel libre présent pour longtemps
- Pas de modèle de développement supérieur
 - «la cathédrale» vs. «le bazar»
- Projets Open Source
 - Ressources humaines
 - Associations sans équivalent propriétaires
 - ▶ <http://www.linuks.mine.nu/debian-worldmap/>
- Tentatives de discontinuités = échecs
- Baromètre fiable de l'industrie
- L'Internet s'est construit avec le logiciel libre

Méthode de travail

- Préparation des séances de travaux pratiques
- Lire la présentation de la séance à venir
 - ▶ <http://www.linux-france.org/prj/inetdoc/cours/explore.gnulinux/>
- Consulter le manuel d'installation
 - ▶ <http://www.debian.org/releases/stable/i386/index.html.fr>
 - ▶ <file:/cdrom/install/doc/fr/install.fr.html>
- Réviser les commandes essentielles d'Unix
 - ▶ <http://www.linux-france.org/article/debutant/debutant-linux.html>

Synthèse

- Étapes du développement des systèmes Unix
 - Histoire «continue» sur plus de 30 ans
 - Mémoire des évolutions
- Histoire + Mémoire = culture
 - Par opposition à l'obscurantisme
 - Compétences != «recettes de cuisine»
- Noyau Linux
 - Laboratoire libre d'accès
- Coût d'acquisition important
 - Il faut être motivé
- Capitalisation des connaissances
 - Autonomie