

Généralités sur Linux

Fabien Lahoudère

Consultant Linux embarqué chez Open Wide
fabienlahoudere.pro@gmail.com
aragua sur github et IRC

INSA Linux embarqué, Novembre 2015

Sommaire

- 1 Introduction
 - Historique
 - Linux dans l'industrie
- 2 Concepts
 - Basé sur UNIX
 - Fonctionnalités
- 3 Licence
- 4 Architecture GNU/Linux
 - Noyau
 - Rootfs
- 5 Execution d'un système GNU/Linux
 - Démarrage
 - Le processus init

1

Introduction

- Historique
- Linux dans l'industrie

2

Concepts

- Basé sur UNIX
- Fonctionnalités

3

Licence

4

Architecture GNU/Linux

- Noyau
- Rootfs

5

Execution d'un système GNU/Linux

- Démarrage
- Le processus init

Linux

Histoire

Créé par Linus Torvalds en 1991, Linux est un noyau (logiciel gérant les ressources d'une machine). Il est devenu libre en 1992.

Il n'est pas un système d'exploitation. De ce fait il est très lié au projet GNU. On parle de distribution GNU/Linux.

Internet a beaucoup contribué à son succès. Il est développé par les communautés de hackers et d'activiste de l'open source.

1

Introduction

- Historique
- **Linux dans l'industrie**

2

Concepts

- Basé sur UNIX
- Fonctionnalités

3

Licence

4

Architecture GNU/Linux

- Noyau
- Rootfs

5

Execution d'un système GNU/Linux

- Démarrage
- Le processus init

Linux

Linux dans l'industrie

Linux s'est en premier imposé dans le monde de l'entreprise sous forme de systèmes d'exploitations sur serveur.

Peu de temps après Linux fut adapté aux mondes des systèmes embarqués.

L'intérêt pour les entreprise est:

- de maîtriser les sources de leur OS.
- d'économiser le prix des licences.
- de bénéficier du support d'une communauté importante de développeur.

Seule la partie Desktop a du mal à émmérgé face à ses concurrents.

1

Introduction

- Historique
- Linux dans l'industrie

2

Concepts

- Basé sur UNIX
- Fonctionnalités

3

Licence

4

Architecture GNU/Linux

- Noyau
- Rootfs

5

Execution d'un système GNU/Linux

- Démarrage
- Le processus init

Concepts

Basé sur UNIX

Implémentation libre d'UNIX diffusé sous licence GPL

Inspiré des deux versions: AT&T et BSD

Les principes d'UNIX sont respectés:

- Simplicité, modularité, respect des standards, ouverture
- Deux espaces de mémoire: noyau / utilisateur
- Le noyau permet d'accéder au matériel (pilotes, appels système)
- Tout composant est un fichier: répertoire, périphérique, élément de communication, etc. (organisation arborescente)
- Puissance de la «ligne de commande» (shell et regexpr)

1

Introduction

- Historique
- Linux dans l'industrie

2

Concepts

- Basé sur UNIX
- **Fonctionnalités**

3

Licence

4

Architecture GNU/Linux

- Noyau
- Rootfs

5

Execution d'un système GNU/Linux

- Démarrage
- Le processus init

Concepts

Fonctionnalités

- Noyau monolithique (en un seul fichier) + modules dynamiques
- Création des processus via `fork()` et `exec()`
- Multi-threading
- Nombreuses piles réseau (IPv4, IPv6, Ethernet, etc.)
- Organisation des fichiers arborescente à partir de la racine (`/`), montage et démontage logique (`mount`)
- Notion de super-utilisateur (`root`), groupes, et utilisateurs

GPL en bref

GPL General Public License

On la surnomme également copyleft

La GPL v2 (1991) est la plus répandue (ex: noyau Linux)

La licence s'applique uniquement en cas de redistribution

Un code source utilisant du code GPL est du travail dérivé et doit être publié

Publication: celui qui reçoit la version binaire peut obtenir le code source

Pas de lien (ld) possible entre du code GPL et du code propriétaire

LGPL

La GPL est complexe à gérer dans l'industrie → création de la LGPL

Le lien avec du code propriétaire est possible avec la LGPL (Lesser/Library GPL) !

En majeure partie, les bibliothèques système sont diffusées sous LGPL (exemple: GNU-libc)

Dans le cas d'une application propriétaire il faut donc vérifier qu'aucune bibliothèque liée n'est GPL

Le lien dynamique n'affranchit pas de la licence sauf dans des cas très particuliers

GPL uniquement pour LINUX

Dans l'espace noyau (pilotes), SEULE la GPL s'applique (en théorie) !

En théorie: On ne peut utiliser les headers du noyau Linux pour créer des binaires non GPL

Certaines fonctions ne sont pas disponibles si la licence n'est pas GPL

En pratique: tolérance si le pilote n'a pas été créé pour Linux (cas du portage) => nVidia, Broadcom, ...

Cependant les pilotes binaires posent des soucis techniques vu qu'un pilote fonctionne pour la version de noyau utilisée pour la compilation

GPL V3

Nouvelle version sortie en 2007

Oblige à fournir les éléments pour construire un logiciel fonctionnel => réponse à la Tivoisation

La GPL v2 demande uniquement la publication des sources à celui qui a reçu le binaire

La GPL v3 ne sera pas utilisée pour le noyau Linux.

Voir: <http://www.gnu.org/licenses/quick-guide-gplv3.fr.html>

Architecture

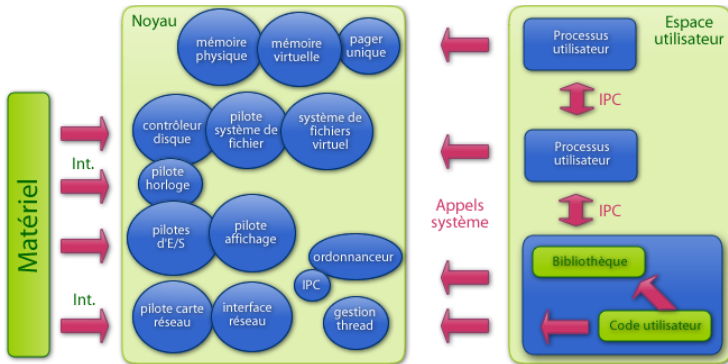
Système GNU/Linux

Linux est un noyau. Seul il ne sert à rien. On parlera donc de système GNU/Linux. Il est en général composé de:

- Un bootloader (Rare sont les carte capable de booter un noyau Linux)
Il est trop gros et doit etre chargé en RAM (elle doit etre initialisé par un microcode)
- Le noyau Linux (gérant les ressources de la machine)
- Un système de fichier contenant à minima un programme de démarrage (rootfs)

Architecture

Système GNU/Linux



- 1 Introduction
 - Historique
 - Linux dans l'industrie
- 2 Concepts
 - Basé sur UNIX
 - Fonctionnalités
- 3 Licence
- 4 Architecture GNU/Linux
 - **Noyau**
 - Rootfs
- 5 Execution d'un système GNU/Linux
 - Démarrage
 - Le processus init

Architecture

Noyau

Le noyau Linux est un binaire de type ELF.

Son image est souvent compressé pour gagner en taille lors du déploiement et de la copie en RAM

Il contient un auto extracteur qui va le décompresser

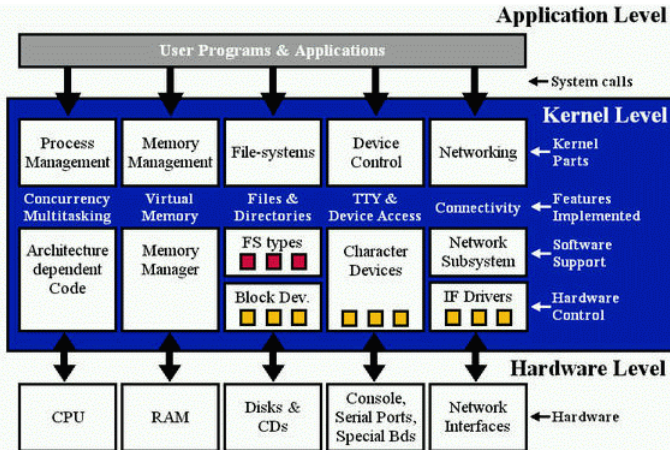
Fournit des services nécessaires à sa fonction:

- ordonnanceur
- gestion mémoire, disque, interface réseaux
- service abstraits (systeme de fichier, pile réseaux ...)

Une grande partie du noyau peut etre déporté en module chargé dynamiquement.

Architecture

Noyau



Architecture

Module noyau

Modules noyau: .ko (Kernel Object)

Les modules binaires sont liés à la version du noyau !

Peut se compiler après Nécessite les entetes du noyau

Mécanisme système pour charger les modules

dynamiquement: modprobe, insmod, rmmod

- 1 Introduction
 - Historique
 - Linux dans l'industrie
- 2 Concepts
 - Basé sur UNIX
 - Fonctionnalités
- 3 Licence
- 4 **Architecture GNU/Linux**
 - Noyau
 - **Rootfs**
- 5 Execution d'un système GNU/Linux
 - Démarrage
 - Le processus init

Architecture

Organisation d'un rootfs

Organisation commune à 90% entre les UNIX Quelques spécificités GNU/Linux et distribution Les fichiers communs:

- /bin,/sbin,/usr/bin,/usr/sbin: binaires communs et systèmes
- /lib,/usr/lib: bibliothèques et modules noyau
- /etc: fichiers de configuration
- /dev: nœuds d'accès aux périphériques (nodes)
- /var: fichiers variables: log, spool, mail, ...
- /opt: pour les programmes externes (ex: OpenOffice)
- /home: accueille les répertoires des utilisateurs

Architecture

Organisation d'un rootfs

Quelques répertoires spéciaux:

- `/lib/modules`: contient les modules du noyau
- `/root`: home-directory de l'utilisateur root (pas dans OE)
- `/media`: point de montage des volumes amovibles
- `/proc`: système de fichier virtuel (état du système)
- `/sys`: idem pour les périphériques connectés (2.6)
- `/boot`: noyau statique (vmlinuz, ulmage, ...)

Architecture

/proc

Système de fichier virtuel (lecture/écriture) géré par le noyau.

(Réponse sur sollicitation, pas d'écriture sur un support)

Intérêt: manipuler les variables systèmes comme de fichiers
(cat, echo, grep)

Exemples:

- /proc/version: version du noyau
- /proc/cpuinfo: type(s) de processeur(s)
- /proc/interrupts: interruptions
- /proc/pid: répertoire décrivant le processus associé au pid
- /proc/mounts: partitions montées
- /proc/modules: liste des modules noyau chargés

Nombreuses commandes systèmes basé sur /proc : lsmmod, lspci, top, mount, ... Il est aisé de proposer une entrée dans

Architecture

/sys

Introduit dans le noyau 2.6 (2003) => sysfs
Vue synthétique des périphériques connectés

- /sys/class (utilisé par UDEV)
- /sys/modules
- /sys/bus

But: mieux gérer l'ajout/suppression dynamique des périphériques (hotplug) Utilisé par UDEV pour créer dynamiquement les entrées dans /dev Quelques recouvrements avec /proc (bus PCI, USB, ...)

- 1 Introduction
 - Historique
 - Linux dans l'industrie
- 2 Concepts
 - Basé sur UNIX
 - Fonctionnalités
- 3 Licence
- 4 Architecture GNU/Linux
 - Noyau
 - Rootfs
- 5 Execution d'un système GNU/Linux
 - Démarrage
 - Le processus init

Execution

Démarrage

- 1 Le matériel lance le bootloader
- 2 Le bootloader configure la RAM et le support contenant le noyau
- 3 Le bootloader copie le noyau en RAM et lui donne la main
- 4 Le noyau s'auto extrait en RAM
- 5 Le noyau initialise les périphériques et les services (ordonnanceur, pile réseaux, système de fichier...)
- 6 Le noyau récupère dans le rootfs le binaire d'initialisation (/sbin/init par défaut) et crée le processus 1
- 7 Ce processus est chargé de démarrer tous les services en espaces utilisateurs.

- 1 Introduction
 - Historique
 - Linux dans l'industrie
- 2 Concepts
 - Basé sur UNIX
 - Fonctionnalités
- 3 Licence
- 4 Architecture GNU/Linux
 - Noyau
 - Rootfs
- 5 Execution d'un système GNU/Linux
 - Démarrage
 - Le processus init

Execution

Le processus init

Le père des processus du système

Plusieurs systèmes possible:

- system5:
 - facile à configurer car basé sur des scripts shell
 - architecture vieille et dépassé. Moins performant.
- systemd:
 - Parallélise l'initialisation du système. Centralise de nombreux services (logs,
 - Plus complexe à prendre en main et moins modulaire pour certain.
- custom:
 - Linux permet d'utiliser son propre binaire init.
 - Il faut gérer soit même le système.