Développer sous Linux

Gilles Maire

2017

Plan de la formation

Introduction

Rubriques

- Préliminaires
- Licences Libres
- Différences entre les licences

Préliminaires

Objet de la formation

- La formation détaille les différents outils de développement couramment utilisés dans le monde open source et notamment sous Linux
- La première partie présente la philosophie et le mode de développement de Linux
- La seconde présente une introduction à linux à destination des développeurs ne connaissant pas Linux
- La troisième présente les grands outils de développement et leur utilisation

Histoire

- **1983 :** Richard Stallman propose le projet GNU. Début du développement gcc, gdb, glibc
- **1991 :** Linus Torvalds propose la première version du noyau Linux
- **1993 :** Première distribution Slakware puis Debian
- **1995 :** début des sites web sous Linux
- **2000 :** début de Linux embarqué
- **2008 :** début des applications mobiles sous Linux
- **2010 :** début des application Android sous Linux

Avantages de Linux

- Sources des logiciels permettant une meilleure compréhension des mécanismes
- Sources permettant des modifications
- Développeurs facilement joignables
- Outils collaboratifs de développement offrant une bonne qualité logicielle
- Très bonne réactivité de maintenance par les SSLL (Sociétés de Service Logiciel Libre) comparativement aux supports propriétaires
- Possibilité de rejoindre une communauté pour rendre ses propres développements ou adaptations pérennes
- Les briques logicielles sont testées sur beaucoup d'équipements dans des conditions variés

Modèle Open Source

- Open Source :
  - les sources sont disponibles, modifiables et en principe maintenus
- les différentes licences
  - Gratuit : gratuit avec parfois des versions de logiciels payants pour les entreprises souhaitant disposer d'un support officiel (Vmware, VirtualBox, Qt)
- Services payants proposés par des sociétés (SSLL) :
  - formations
  - développement / extension
  - maintenance
  - support

Licences Libres

Free vs Open Source

- **Free** veut dire gratuit
- **Open source** veut dire que vous mettez à disposition les sources
- Mais attention :
  - Free et Open Source ne veulent pas dire que vous avez le droit de distribuer le logiciel
  - Le mode de distribution se fait généralement dans le cadre d'une licence commerciale ou d'une licence dite libre
- Certains logiciels libres ne sont régis par aucune licence ce qui rend floue leur utilisation dans un cadre professionnel
- Un auteur peut très bien inclure des clauses fantaisistes en lieu et place de licences
- Les licences sont faites pour éviter de convoquer un juriste pour étudier les clauses de chaque logiciel open source utilisé.

Avertissement

- Les définitions des diapositives suivantes présentent les concepts pour aider le développeur à comprendre les champs d'utilisation des licences.
- L'exposé ne couvre pas :
  - le droit de propriété industrielle
  - la notion de brevet
  - les preuves d'antériorité
- En outre, des jurisprudences par pays rendent certaines licences plus vulnérables dans certains pays
- Il existe des juristes spécialisés dans les droits des logiciels libres.

L'objectif

- **Licence :** est un contrat «par lequel le titulaire des droits du logiciel autorise un tiers à poser des gestes qui autrement les enfreindraient.»
- Le terme Contrat de Licence Utilisateur Final (**CLUF**) est une traduction du terme anglais **EULA**, End User License Agreement. C'est un contrat liant une personne installant un logiciel affecté par ce type de licence sur un/son ordinateur et l'éditeur du logiciel
- Protection contre les vices cachés : **As is** L'éditeur du logiciel décline donc toute garantie en cas de fonctionnement défectueux et se réserve le droit de faire payer les corrections.

Les formes de licence

- **Licence fixe :** est conçue pour être installée sur un ordinateur particulier
- **Licence nominative :** est attribuée à un utilisateur particulier, qui peut l'installer sur tout ordinateur, mais est le seul utilisateur agréé à l'utiliser.
- **Licence flottante :** fonctionne avec un ordinateur serveur de licence(s) : celui-ci décompte le nombre de licences utilisées à un instant T sur le réseau
- **Shareware :** (ou partagiciel) attribue un droit temporaire et/ou avec des fonctionnalités limitées d'utilisation
- **Licences libres :** sont une forme particulière de licence

Les principales licences de logiciels libres

- GPL
- LGPL
- Apache
- Licence X11
- Licence Publique Eclipse
- BSD
- Licences sur les contenus

Différences entre les licences

GPL

Nom	GPL
Appellation	GNU Public Licence
Version	3
Modification	**oui sous licence GPL**
Diffuser	**oui**
Version dérivée	**oui sous licence GPL**
Remarque	la plus répandue
Livraison sources	**obligatoire**
Tarif	**Frais d'acte de fabrication des supports**
Contrat de garantie	OK par une société de logiciel libre

LGPL

Nom	LGPL
Appellation	Lesser GNU Public Licence
Version	3
Modification	oui + adjonction parties non GPL
Diffuser	**oui**
Version dérivée	**oui**
Remarque	répandue
Livraison sources	**obligatoire**
Contrat de garantie	OK par une société de logiciel libre

Apache

Nom	Apache
Appellation	Software Foundation
Version	2
Modification	oui + ajout autre licence
Diffuser	**oui**
Version dérivée	on garde + adjonction
Remarque	Ok pour adjonctions commerciales
Livraison sources	**obligatoire**
Tarif	Libres
Contrat de garantie	Peut être fourni par une société de logiciel libre

X11

Nom	X11 ou MIT
Appellation	Licence MIT
Version	2
Modification	oui + ajout autre licence
Diffuser	**oui**
Version dérivée	oui en changeant la licence
Remarque	Ok pour adjonctions commerciales
Livraison sources	**obligatoire**
Tarif	Libres
Contrat de garantie	Peut être fourni par une société de logiciel libre

Eclipse

Nom	EPL
Appellation	Public Licence
Version	1
Modification	oui + ajout autre licence
Diffuser	oui
Version dérivée	oui en gardant la licence
Remarque	Ok pour adjonctions commerciales
Livraison sources	**obligatoire**
Tarif	Libres
Contrat de garantie	Peut être fourni par une société de logiciel libre

BSD

Nom	BSD
Appellation	Berkeley Software Distribution
Version	1999
Modification	oui + modification licence
Diffuser	oui
Version dérivée	oui sans respect de la licence
Remarque	Ok pour adjonctions commerciales
Livraison sources	Non obligatoire
Tarif	Libres
Contrat de garantie	Peut être fourni par une société de logiciel libre

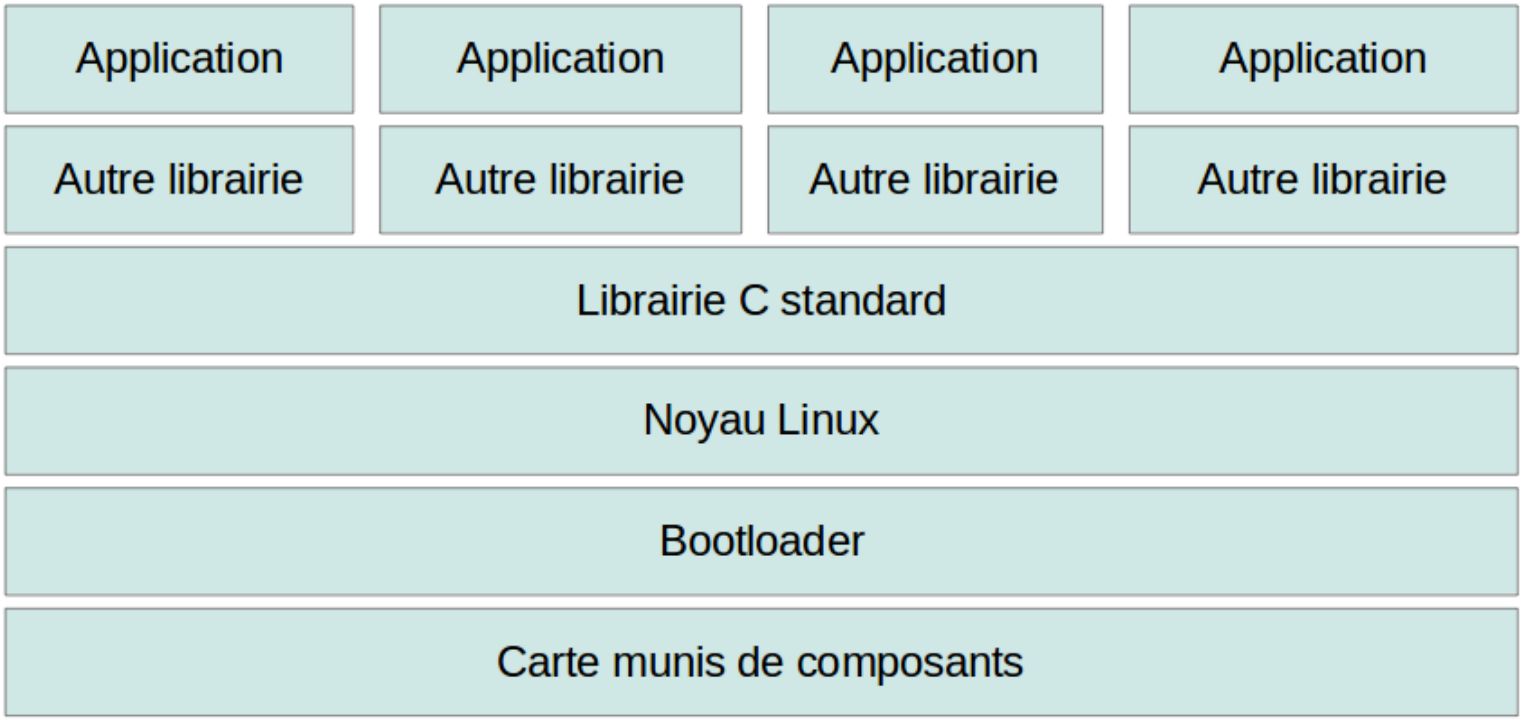
Architecture

Rubriques

- Les composants
- Init et démons

Les composants

Architecture Linux



Les différents éléments d'un système Linux

- Le Bios / le Boot
- Le noyau
- La librairie libc
- L'init
- Le shell
- Le gestionnaire de fenêtres
- Les distributions
- Les outils de mise à jour

Le BIOS et le boot

- Tous les PC commencent par exécuter un programme appelé BIOS présent dans une EEPROM ou CMOS c'est à dire en mémoire ROM.
- Ce Bios démarre puis appelle un bootloader sur un des disques. Ce disque sera appelé le BOOT DISK
- Ce Bios se configure en tapant sur F1 ou ESC au démarrage (notamment pour indiquer le disque sur le quel le bootloader sera installé)
- Il existe plusieurs distributions de de BIOS : AMI, AWARD, MSI ou les BIOS des constructeurs
- Cette partie est très spécifique au monde PC, en embarqué nous n'utiliserons pas le même mécanisme.

Le boot

- Le BIOS appelle un média : disque, clé USB, CDROM, Disquette
- Au secteur 0 de ce média se trouve le bootloader qui est en charge de l'amorce de l'OS c'est à dire du chargement du noyau Linux
- Chacun de ces médias est organisé suivant un système de fichiers
- doit savoir ouvrir le système de fichier et les médias pour charger le noyau
- Cette partie est très spécifique au monde PC, en embarqué nous n'utiliserons pas le même mécanisme.

Introduction au Noyau

- Le noyau ou kernel est responsable du fonctionnement des couches basses de Linux.
- C'est ce noyau qui a été développé en 1991 par Linus Torvalds qui en est toujours le mainteneur en 2017
- Des milliers de personnes contribuent via un serveur git à l'amélioration de chaque release et à l'ajout des drivers des nouvelles cartes
- Le noyau est en licence GNU GPL V2 (sources disponibles avec droit de modification pour toute vente d'un système contenant Linux)
- Une fois le noyau lancé, le programme init est appelé
- En embarqué, le noyau utilisé est le même que sur les postes PC avec une compilation pour un processeur adapté.

La librairie LibC

- Très proche du noyau cette librairie permet aux programmes systèmes fonctionnant sous Linux d'appeler des primitives du noyau
- Elle permet les **appels systèmes** suivants :
  - malloc, free, memset, strcpy, strlchr, strlen
  - gestion des Threads & gestion des Mutex
  - gestion de l'encodage décodage UTF8
  - stdio : getc, printf etc
  - regex
  - qsort
  - calculs mathématiques flottants
  - gestion des timezones
- Il existe plusieurs souches de la librairie : glibc, uclibc, eglibc, dietlibc, newlibc
- La libc est la librairie contenant tous les appels système
- La commande `time` précédent une commande donne le temps passé par une commande à exécuter des appels systèmes, autrement dit des primitives lancées par la libc.

Init et démons

Inittab et init

- Une fois le noyau chargé en mémoire, le programme `/sbin/init` est appelé (`ps awx` donne le numéro de processus 1 pour l'init)
- Le fichier de configuration de l'init est /etc/inittab remplacé par /etc/init/rc-sysinit.conf sous Ubuntu. Ce fichier donne le niveau d'utilisation de l'équipement
- `/etc/init.d` contient les programmes de démarrage de chacun des démons
- Chaque niveau d'utilisation est matérialisé par un répertoire `/etc/rc0.d` `/etc/rc1.d ... /etc/rc6.d` chacun de ces répertoires comprenant un lien symbolique vers le programme de démarrage contenu dans `/etc/init.d`
- Les choix de ces différents programmes dépend de l'utilisation que l'on souhaite faire de son équipement

Niveau d'exécution

- Dans le fichier `/etc/inittab` on trouve des lignes sous la syntaxe:
  - `id : niveau(x)_exécution` : action : processus
  - une étiquette indiquant le niveau par défaut
- Les niveaux sont les suivants :
  - **Niveau 0 :** arrêt du système,
  - **Niveau 1 :** mode de maintenance (single user)
  - **Niveau 2 :** souvent non utilisé
  - **Niveau 3 :** système réseau sans interface graphique
  - **Niveau 4 :** souvent non utilisé
  - **Niveau 5 :** avec interface graphique
  - **Niveau 6 :** redémarrage du système
- Pour passer à un niveau n, il faut être root et entrer la commande

```
init n
```

Runlevel

- Lorsque la commande init est lancée avec un niveau n, le système exécute essentiellement les scripts d'initialisation comme suit.
  - Les noms de scripts de `/etc/rc<n>.d/` commençant par un `K` sont exécutés dans l'ordre alphabétique avec le paramètre unique `stop` (arrêt des services).
  - Les noms de scripts de `/etc/rc<n>.d/` commençant par un `S` sont exécutés dans l'ordre alphabétique avec le paramètre unique «start» (lancement des services).
- Dans les systèmes modernes on lanche un service par

```
service machin start
```

- Sur fedora pour rendre le service cron activable :

```
# en niveau 3 et 5
/sbin/chkconfig --level 35 crond on
```

Exemple fichier /etc/inittab

```
# Le niveau d'exécution par défaut
id:2:initdefault:
# Initialisation du système
si::sysinit:/etc/rc.d/rc.sysinit
# Les différents niveaux d'exécution
l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
# Intercepter les touches CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
1:2345:respawn:/sbin/mingetty tty1
```

Démons

Rubriques

- Les démons classiques
- Init et shell

Les démons classiques

Les différents types de démon

- Les classiques
  - Gestion des fichiers
  - Exécution des tâches
  - Protocole réseau
  - Gestion de l'heure et des logs
  - Système
- Les desktop
  - Gestion l'énergie
  - Gestion du multimédia
  - Gestion de la communication
  - Gestion de l'affichage

Gestion des fichiers

- **bootmisc :** Prépare la machine au démarrage.
- **checkfs :** Vérifie les systèmes de fichiers autres que /.
- **checkroot :** Vérifie le système de fichiers de /.
- **evms :** Initialise les différents disques et raids
- **hdparm :** Règle certains paramètres sur les disques
- **vm :** Démarre les périphériques disques gérés via LVM
- **mdadm-raid :** Gère le raid logiciel.
- **mdadm :** Gère le raid logiciel.
- **mountall :**Monte les partitions locales.
- **mountvirtfs :**Monte les File Systems virtuels (`/proc`)
- **umountfs :** Démonte les file systems locaux.

Exécution des tâches

- **atd :** Démon qui gère des tâches périodiques par commande `at`.
- **cron :** Démon qui gère des tâches périodiques pour les équipement tournant 24h/24
- **anacron :** Démon qui permet d'effectuer des tâches périodiques pour des équipements ne fonctionnant pas en permanence
- **halt :** Arrête l'équipement
- **rc :** Gère le démarrage et l'arrêt des services.
- **rcS :** Gère le démarrage des scripts de `/etc/rcS.d`
- **reboot :** Redémarre l'équipement
- **sendsigs :** Arrête les process lors de l'arrêt de la machine.
- **single :** Prépare la machine à passer en mode mono utilisateur

Protocoles réseaux

- **mountnfs :** Monte les partitions NFS.
- **rsync :** Gère la synchronisation de répertoires locaux avec des répertoires distants.
- **samba :** Gère le partage de fichiers avec les machines Windows.
- **fetchmail :** Ramène les e-mails distants en tant que mail locaux.
- **ssh :** Permet les connections SSH sur la machine.
- **umountnfs :** Démonte les file systems NFS.

Gestion de l'heure et des log

- Heure
  - **hwclock :** Synchronise l'heure système avec celle du BIOS.
  - **wclockfirst :** Synchronise l'heure système avec celle du BIOS.
  - **ntpdate :** Synchronise l'horloge locale avec l'horloge d'Ubuntu sur Internet.
- Log
  - **bootlogd :** Il enregistre les évènements liés au démarrage.
  - **klogd :** Note les informations fournies par le kernel dans des fichiers de logs.
  - **stop-bootlogd :** Stoppe le démon de logging des évènements du noyau.
  - **sysklogd :** Démon qui loggue les évènements systèmes.

Init et shell

Gestion autres périphériques

- **cupsys :** Gère les imprimantes.
- **console-screen :** Charge les paramètres pour les consoles textes.
- **hplip :** Gère les imprimantes et scanner HP.
- **keymap :** Charge la langue du clavier pour la console texte.
- **hotkey-setup :** Permet l'utilisation des touches «spéciales» de certains portables.
- **Linux-restricted-modules-common :** gère les modules propriétaires
- **pcmcia :** Gère les périphériques PCMCIA.
- **ppp :** Démarre les connexions sur ligne série ou modem.
- **pppd-dns :** Récupère le fichier `/etc/resolv.conf` après un crash de la ligne série ou du modem.

Les commandes de base et programme

- Une fois l'init appelée elle peut à son tour appeler un simple shell script ou un serveur Web
- On peut décider d'adjoindre d'autres programmes personnalisés
- En général en embarqué on utilise une source appelée Busybox qui permet de compiler l'ensemble des outils que l'on souhaite fournir à son équipement
- On fournit un shell permettant des commandes de maintenance

Le Shell

- Le shell est un programme qui interprète des commandes lancées par un opérateur
- Ces commandes peuvent être également empilées dans un fichier qui les lancera sous contrôle de conditions (if while etc) appelé ShellScript
- Il existe plusieurs shell : sh (bourne shell), bash, ksh (korn shell), csh, tcsh
- Chacun a ses avantages : rappels des dernières commandes, commandes internes évoluées etc..
- Le plus courant en environnement Linux est bash
- Si on programme des ShellScript on doit choisir entre un développement en sh qui est le plus répandu et bash qui est le plus puissant

Les gestionnaires de fenêtre

- s'appuient sur la couche X11
- Il en existe des légers comme XFCE ou des plus lourds comme KDE
- Le choix médian sur les distribution pour PC se porte souvent sous gnome
- En embarqué, on n'utilise pas de couche X11 et si on possède un écran on l'adressera en mode framebuffer soit avec Qt soit avec Android

Qt

- est une librairie C++ fournissant 1200 classes et fonctionnant sous Windows, Linux, Windows et aussi en mode framebuffer
- fonctionne également en mode Android et en mode IOS, permettant avec un même développement de fournir un logiciel dans tous ces environnements.
- existe depuis 1993 et un gestionnaire graphique basé sur Maemo est moins utilisé qu'Android mais est très présent en embarqué graphique.

Distributions

Rubriques

- Les interface graphiques
- Les Distributions

Les interface graphiques

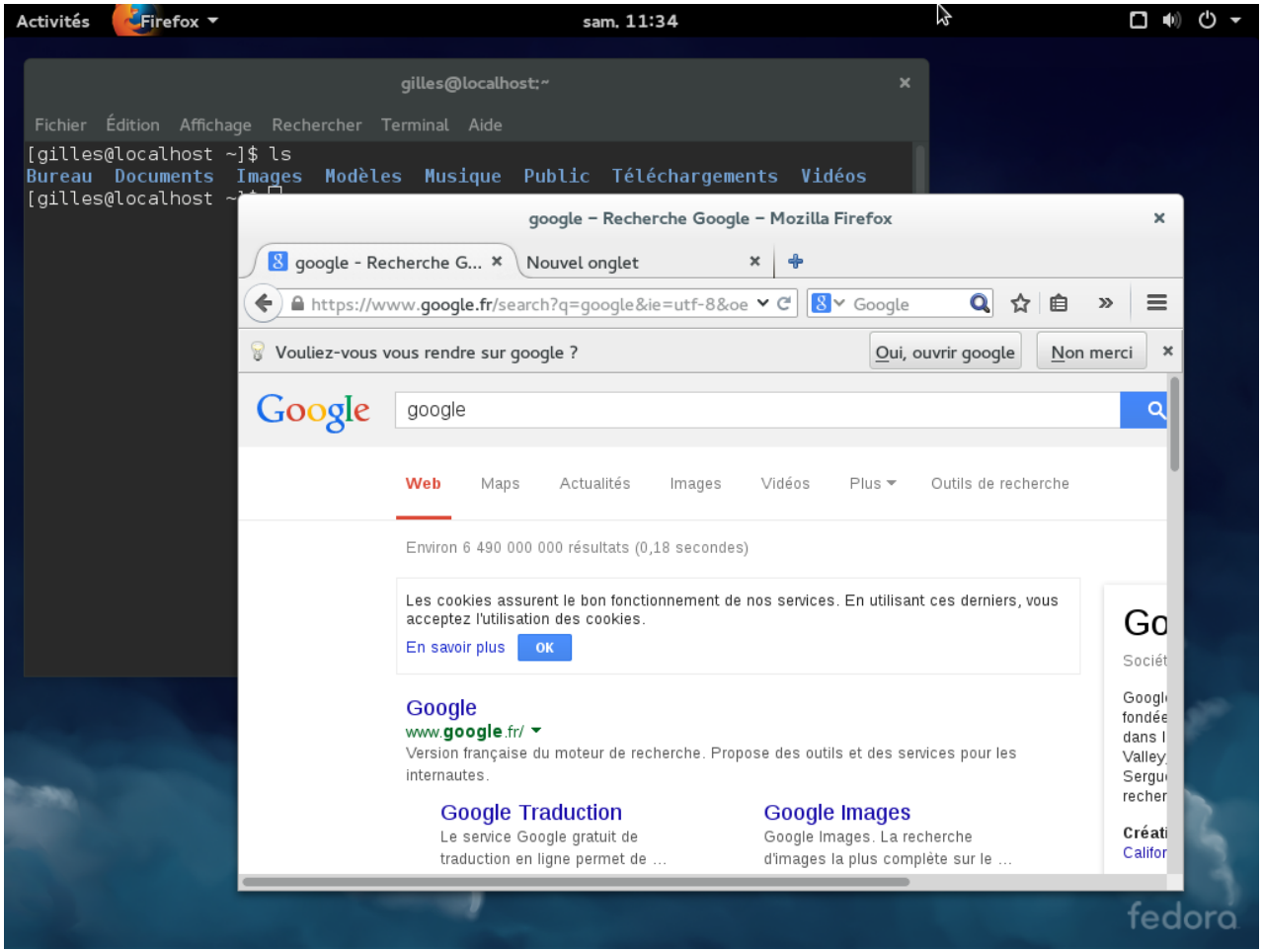
Les interfaces graphiques

- Trois environnements graphiques.
- Chaque distribution choisit son propre environnement.
- KDE :
  - utilise des plasmoïdes
  - assez lourds
  - beaucoup d'effets
  - basé sur la librairie Qt
- Gnome
  - interface unifiée pour mobiles, poste de travail
  - moins fourni en terme de réglage
- XFCE
  - très léger et utilisé pour les configurations peu performantes
  - look simple
  - encore moins fourni en terme de réglages

KDE

Distribution Mageia française, avec Kde4

Fedora


Activités
Firefox
sam. 11:34
gilles@localhost:~
Fichier Édition Affichage Rechercher Terminal Aide
[gilles@localhost ~]$ ls
Bureau Documents Images Modèles Musique Public Téléchargements Vidéos
google – Recherche Google – Mozilla Firefox
google - Recherche G...
Nouvel onglet
https://www.google.fr/search?q=google&ie=utf-8&oe
Google
Vouliez-vous vous rendre sur google ?
Oui, ouvrir google
Non merci
Google
google
Web Maps Actualités Images Vidéos Plus Outils de recherche
Environ 6 490 000 000 résultats (0,18 secondes)
Les cookies assurent le bon fonctionnement de nos services. En utilisant ces derniers, vous acceptez l'utilisation des cookies.
En savoir plus
OK
Google
www.google.fr/
Version française du moteur de recherche. Propose des outils et des services pour les internautes.
Google Traduction
Le service Google gratuit de traduction en ligne permet de ...
Google Images
Google Images. La recherche d'images la plus complète sur le ...
fedora

GNOME Unity

XFCE

Autres interfaces

- Il existe d'autres systèmes d'interface graphiques, certains évitant la souris et fonctionnant uniquement aux raccourcis claviers
- Sur un serveur on ne met pas d'interface graphique et on se sert :
  - de lignes de commande qu'il faut connaître
  - d'interface HTML permettant de piloter le serveur Linux
- Il existe quelques différences de choix de logiciels entre chacune des distributions, mais tous les logiciels présents sur une distribution peuvent être installés sur n'importe quelle autre.

Les Distributions

Les distributions Linux

- 500 distributions
- chacune va retenir une interface privilégiée parmi les différentes interfaces présentées
- Certaines distributions ne retiennent que certains types de licences
- Les logiciels empaquetés par défaut ne sont pas toujours les mêmes
- Mais les logiciels d'installation permettent d'installer les logiciels qui ne sont pas présents sur une distribution

Les types de distribution

- Les distributions compilées
  - prêtes à l'emploi
  - contiennent le plus grand nombre de pilotes possible
  - deux grandes familles :
    - issues de Debian
    - issues de Redhat
- Les distributions à compiler
  - doivent être compilées avec des options très proches des composants
  - très légères
- Des distributions très orientées
  - montage video
  - media server
- Les embarquées
  - Raspbian
  - Routeurs
  - Set Top Box

Les distributions compilées

- Debian : communautaire - paquetage deb - pas de logiciel non GPL
  - Sous famille : Ubuntu/Kubuntu/Xubuntu : entreprise
  - sous famille MINT
- Redhat (entreprise) /Fedora (communautaire) : paquetage rpm
  - Suse : entreprise
  - Mandriva/Mageia
- ArchLinux : sans version - communautaire - gestionnaire de paquet pacman

Les distributions à compiler

- Bureautiques
  - LFS : communautaire
  - Gentoo : communautaire
- Embarquées
  - Raspbian
  - BuildRoot
  - Yokto

Noyau

Rubriques

- Téléchargement des sources
- Paramétrage du noyau
- Détails
- Compilation et installation du noyau

Téléchargement des sources

Présentation

- On peut compiler le noyau avec un maximum de modules drivers destinés à fonctionner sur toutes les plate-formes : c'est l'optique retenue par les distributions comme Ubuntu, Mandriva, RedHat etc.
- Le noyau sera de taille acceptable et sera fourni avec un grand nombre de modules correspondant aux drivers
- Les modules se chargent en mémoire dynamiquement donc ils ne prennent pas beaucoup de place mémoire, mais ils en prennent beaucoup sur le périphérique de stockage
- On peut compiler le noyau avec un nombre minimum de drivers correspondants aux caractéristiques de l'équipement.
- Ceci engendre un noyau prenant moins de place, plus rapide à compiler et également plus rapide à l'exécution.
- On se passe des modules dans ce cas là, sauf si certains drivers ne sont fournis que sous forme de module par les constructeurs

Versions

- Sur les systèmes de version antérieurs à 3
  - les versions de décimale paires sont stables (2.6)
  - les versions de décimales impaires ne le sont pas (2.7)
- Sur les systèmes embarqués, la version encore rencontrée est la version 2.6
- Sur les systèmes actuels
  - sur les systèmes les plus modernes la version est 4.xx (commande uname -r)
  - Les versions rc sont les releases candidates c'est à dire les versions non encore officielles

Les sources

- les sources sont disponibles sur le site http://www.kernel.org
- on peut récupérer les sources via ftp http ou par une commande git.

```
git clone git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git \
linux-git
```

- Si vous souhaitez contribuer, la mise à jour se fait par des patches qui sont envoyés et pris en compte par les lieutenants chargés du contrôle des versions
- la liste des nouveautés se lit dans la liste des logs de git

```
git log
```

- Il n'est pas recommandé d'utiliser les versions en cours de développement
- La dernière version stable est la version 4.9
- les sources compressés font une taille approchant 100 Mo, le noyau comprend les drivers de toutes les cartes reconnues par Linux

Paramétrage du noyau

Configuration du noyau

- On décompresse le noyau par la commande tar idoine de préférence dans /usr/src

```
cd linux-version
vi README
make mrproper
```

- Ensuite au choix (ces différents choix seront explicités un peu plus loin)

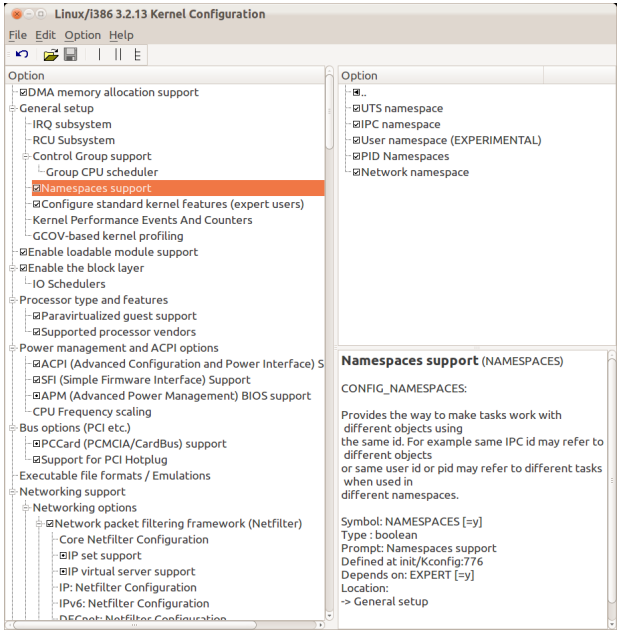
```
make config
make nconfig
make menuconfig
make xconfig
make gconfig
```

- on verra qu'on peut aussi modifier le fichier `.config`, mais avant nous devons parler des modules
- **NB :** Tout ce qui est explicité dans la configuration du noyau, sera utile pour la compilation de Busybox et de BuildRoot

Module / Kernel / None

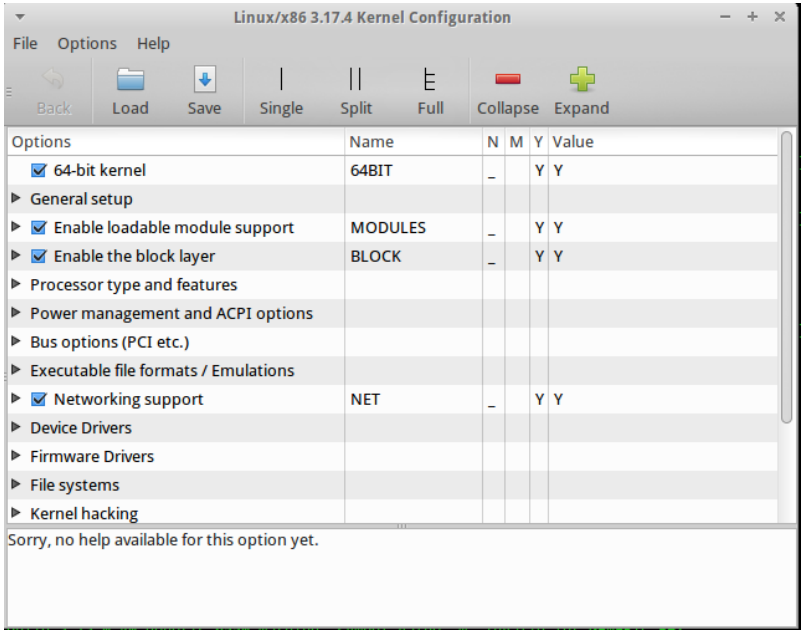
- Pour chaque option de driver rencontrée dans la configuration du noyau on a le choix entre :
  - mettre le driver dans le noyau ;
  - le mettre en module ;
  - ne pas le mettre du tout.
- Il faut faire attention car un driver supprimé peut entraîner des erreurs s'il est nécessaire pour un autre driver
- On peut répondre à chacune des options par :
  - **true** (dans le kernel)
  - **m** (en module)
  - **false** (ne pas inclure)

make xconfig



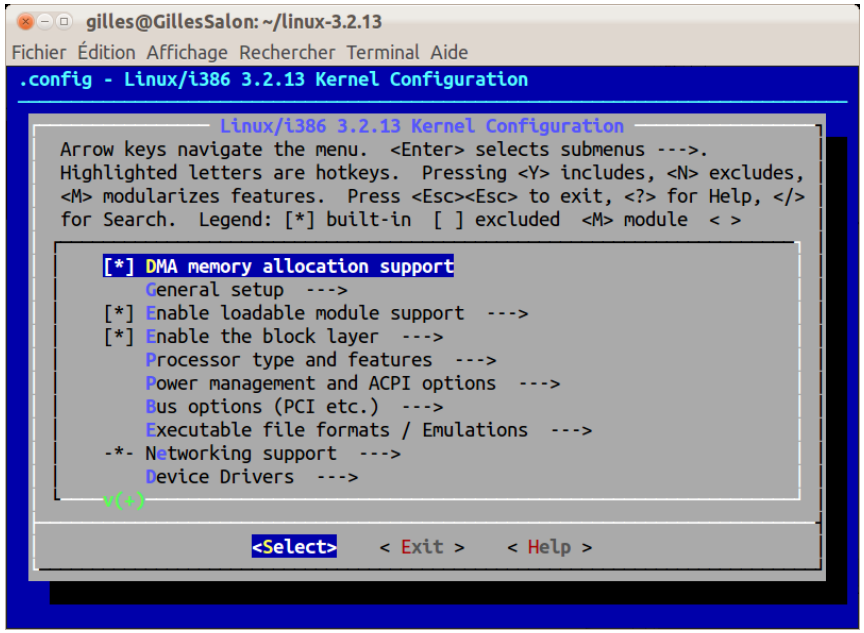
- est la façon la plus courante de configurer un noyau
- permet la recherche par mot clé d'un module ou driver ce qui est plus pratique que de rechercher dans l'arbre des catégories
- requiert la librairie libqt4-dev

make gconfig



- pas de recherche
- similaire à xconfig
- requiert libglade2-dev

make menuconfig



- quand on n'a pas x11 et requiert libncurses-dev
- La recherche se fait par la touche /

make nconfig

- ne requiert pas x
- requiert libncurses-dev
- la recherche d'un module se fait par la touche /

Détails

Autres possibilités

- .config : les options sont rangées dans le fichier .config qui est modifiable via un éditeur (dans les distributions, on trouve le fichier configure qui a servi à construire le noyau dans /boot)
- Pour positionner toutes les options du fichier .config et demande uniquement les nouvelles options qui n'existaient pas auparavant

```
make oldconfig
```

- Pour positionner toutes les options du fichier .config et positionner les nouvelles à leurs valeurs par défaut

```
make olddefconfig
```

- Pour positionner toutes les options aux valeurs par défaut

```
make defconfig
```

- Pour positionner toutes les options à oui (respectivement à non ou au maximum à module)

```
make allyesconfig
make allnoconfig
```

Les options de compilation du noyau

- **Prompt for development/incomplete code :** affiche ou masque les drivers qui ne sont pas considérés comme stables
- **Local version - append to kernel release :** permet d'ajouter une chaîne de caractères arbitraire au nom du noyau qui sera affichée avec l'option uname -r
- **Support for swap :** le swap est un mécanisme qui utilise le disque pour augmenter la mémoire. Doit être mis à NO sur les systèmes embarqués.
- **Configure standard kernel features (for small systems) :** permet de supprimer des options du noyau pour réduire sa taille. A ne pas trop utiliser
- **Loadable module support :** peut être supprimé pour avoir un noyau monolithique sans module dynamique
- **Enable block layer :** en mode CONFIG_EXPERT peut être supprimé si on utilise le mode FLASH NAND
- **Processor type and features :** à mettre à ARM avec les variantes correspondant au processeur
- **Preemptible Kernel :** (ON)
- **Network Support :** Unix sockets, TCP/IP, DCCP, SCTP, TIPC, ATM, Ethernet ;Bridging, QoS
- **SCSI :** est utile pour USB et pour les drivers IDE SATA et PATA
- **Input device support :** Mice, joystick, touchscreen, tablets, keyboard

Options de compilation (fin)

- **Characters devices :** ports séries, PTY driver pour SSH
- **I2C, SPI, 1Wire :** sont utiles
- Thermal sensor, Watchdog support, Multifonction drivers
- **Multimedia Drivers :** V4L, DVB, ALSA et OSS pour le son
- **Graphics supports :** Framebuffer
- **HID devices :** utiles
- **USB Support :** infrastructure, Host controller, devices drivers, gadget
- MMC/SD/SDIO, LED, RealTime Clock driver, Voltage an current régulator

Compilation et installation du noyau

La compilation

- **make :** compile le noyau pour générer :
  - **vmlinux :** noyau au format RAW au format ELF utile pour débuguer mais ne fonctionne pas
  - **arch/arm/boot/Image :** l'image finale au format uimage
  - tous les modules au format *.ko
- Installation du noyau

```
make install
```

- Installation des modules

```
make modules_install
```

- Supprimer les fichiers générés :

```
make clean
```

- Supprimer également le .config

```
make mrproper
```

-Supprimer les backup produits par les patches

Exercice : compiler un noyau

- Récupérer un noyau sur le site http://www.kernel.org
- Le décompresser
- Lire le fichier README.txt
- Récupérer l'ensemble des modules nécessaires via lsmod
- Lancer make xconfig
- Régler les options
- Lancer make

Installation du noyau

- Sur un système hôte pour installer un nouveau noyau on se contente de faire

```
make install
```

- quand on charge le noyau via le gestionnaire de paquetages
- Pour installer le noyau dans le répertoire cible en embarqué ou sur un média autre que /boot doit faire des opérations manuelles :

```
cp arch/i386/boot/bzImage /boot/vmlinuz-version
```

Modules

Rubriques

- Compilation et installation de modules
- Charger / décharger un module

Compilation et installation de modules

Avantage / Désavantage des modules

- Il existe des milliers de modules que l'on met en œuvre sur les distributions «tout terrain» (Ubuntu etc).
- Mais leur temps de compilation est important (>1 heure)
- Pour une version Linux propre à une carte embarquée on aura tendance à minimiser l'espace du noyau en flash et à refuser un maximum de modules pour ne garder que ceux vraiment utiles.
- Les noyaux sans module sont plus rapides à l'exécution

Installation des modules

- sur un système hôte :

```
make modules_install
```

- sur un système cible :

```
make modules_install INSTALL_MOD_PATH="
    ~/busybox-version/_install"
```

- Les modules de votre noyau seront installés dans le répertoire ~/busybox-1.19.4/_install/modules
- Les firmwares de votre noyau seront installés dans le répertoire ~/busybox-1.19.4/_install/firmware

Disque minimal en RAM

- Dans certains système Linux on créé un disque minimal contenant les drivers permettant de monter les périphériques nécessaires SCSI.
- on peut éviter cela en fabriquant ces périphériques dans le noyau et non pas en module (surtout vrai pour les modules SCSI)
- Sinon il faut créer un fichier initrd.img-3.2.9 par la commande

```
mkinitramfs  -o initrd-3.2.13 3.2.13
```

- En général on ne créé pas de disque RAM en embarqué.

Fichier config

- Une fois le noyau testé et installé, on copie le fichier .config qui a servi à la compilation dans le répertoire /boot avec
  - le nom du noyau.
  - ceci afin de pouvoir reprendre une compilation avec adjonction d'un driver

Charger / décharger un module

Commandes sur les modules

- La gestion des modules est automatisée par le système via la signature de chaque matériel
  - par son identifiant constructeur
  - son modèle
- On peut gérer les modules de façon manuelle pour des besoins de débugage:
  - **lsmod :** affiche la liste des modules montés
  - **modprobe module :** charge un module identifié par son nom partiel ou son nom avec chemin
  - **insmod :** installe un module identifié par son chemin
  - **rmmod :** désinstalle un module
  - **modinfo :** donne une information sur un module

Compilation

Rubriques

- Compilation des sources
- Les librairies

Compilation des sources

Vérification d'intégrité

- Lorsqu'on récupère des sources on se verra proposer :
  - un fichier d'archive au format tgz bz2 ou xz
  - un fichier signature permettant de contrôler l'intégrité du fichier
- Une fois les deux fichiers chargés :
  - On lance la commande

```
md5sum fichier.tgz
```

- On peut le comparer avec l'étiquette md5sum contenue dans le fichier signature
- La signature sha1sum s'obtient de façon analogue à la signature md5sum

Décompression des sources

- On peut être amené à récupérer des sources sous plusieurs formats
  - TGZ : fichier.tgz (ou fichier.tar.gz) : se décompresse par

```
tar zxvf fichier.tgz
```

```
- BZ2 :  fichier.bz2 (fichier.tar.bz) : se décompresse par
```

```
tar jxvf fichier.bjz
```

```
- XZ : fichier.xz : se décompresse par
```

```
tar Jxf fichier.xz
```

- Sur les versions récentes on peut utiliser le paramètre automatique pour la décompression

```
tar axvf fichier.xz
```

Git SVN CVS

- **Git**, **SVN** ou **CVS** sont des gestionnaires de version qui permettent au développeurs de travailler ensemble sur les mêmes sources :
  - en gérant les modifications des autres développeurs
  - en pouvant récupérer la version d'une certaine date pour la tester
  - les serveurs git, svn ou cvs servent également de backup des sources
- On peut récupérer également les dernières sources en cours de développement avec une commande svn, cvs ou git
  - avantage : avoir des sources up to date
  - inconvénient : avoir des sources moins testés

Commandes de base CVS/SVN/git

- Avec chacun des gestionnaires, vous aurez à récupérer les sources et éventuellement de temps à autres à updater les sources ou au moins à vérifier qu'il n'y a pas de nouveautés
  - CVS :

```
cvs checkout adresse
cvs update
```

```
-  SVN :

svn checkout adresse
svn update
```

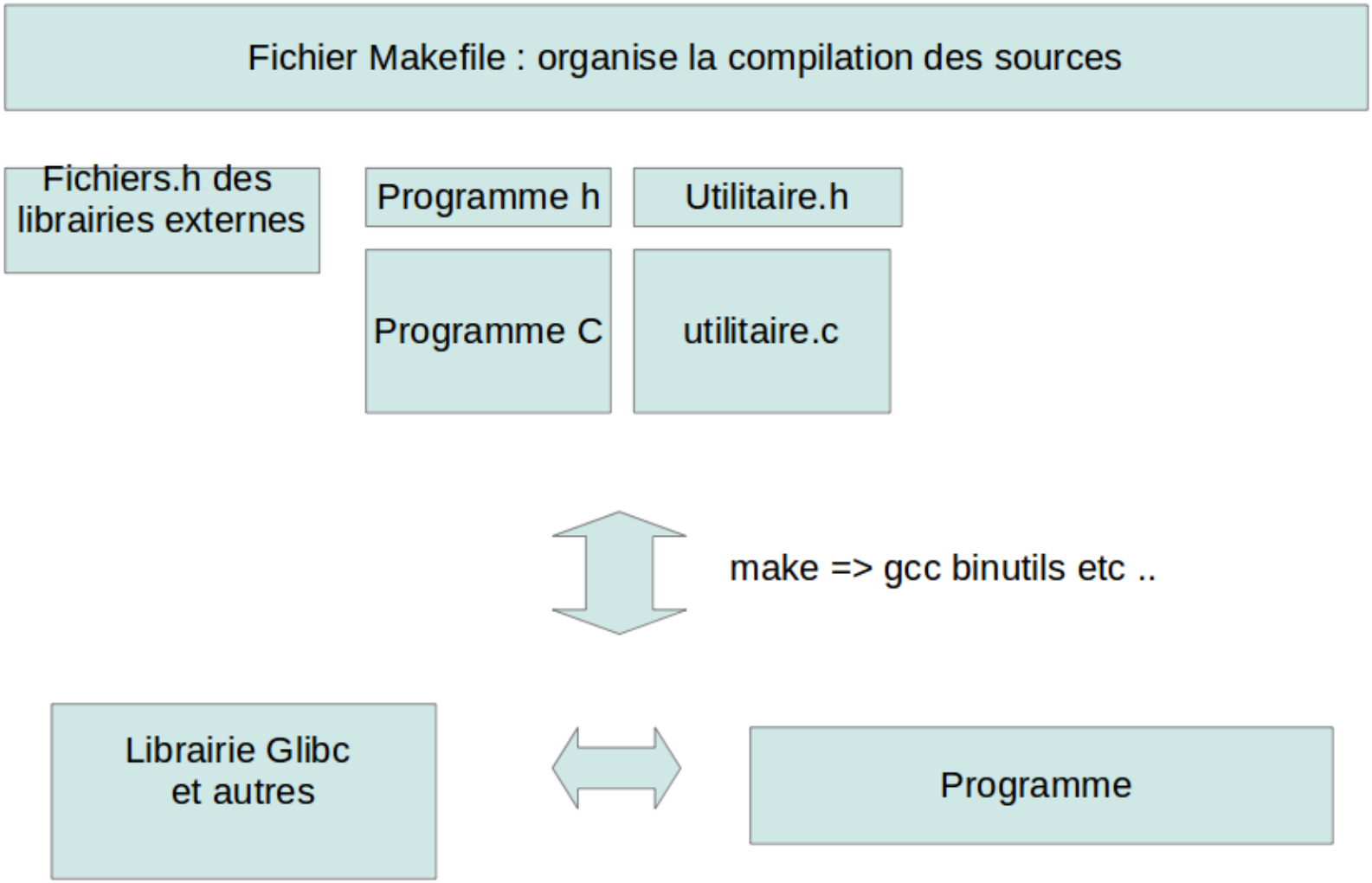
```
- Git

git clone adresse
git pull
```

Morphologies des programmes

- les programmes interprétés (Basic, Perl, Python, ShellSript)
  - le source analysé à la volée est transformé en code microprocesseur
  - l'interpréteur doit être présent sur l'équipement
  - interprétation est lente ne peut convenir à des tâches critiques
  - C'est très portable
- les programmes bytes compilé (Java)
  - le source compilé en «presque assembleur»
  - le résultat est transformé en code microprocesseur à la volée
  - C'est très portable et plus rapide
- les programmes compilés (C, C++)
  - nécessitent une transformation du code microprocesseur
  - c'est le plus rapide
  - il faut recompiler sur chaque processeur

Rappel mécanisme de compilation



Les librairies

Le principe des librairies

- Quand on code un projet on utilise parfois des librairies pour lesquelles on dispose:
  - des fichiers binaires rangés en principe dans /usr/lib avec une extension de fichier .so ou .a
  - de fichiers includes appelés header rangés en principe dans /usr/include
  - de documentations pour comprendre comment utiliser la librairie
- si ces librairies sont open source, on peut disposer des sources et les modifier mais ce n'est généralement pas nécessaire
- Dans son programme on inclut le header et quelque part on appelle la fonction voulue de la librairie

```
#include <malibrairiemp3.h>
fi=decodeMp3ToWav(file);
```

Édition de lien

- La compilation d'un programme produit un fichier .o

```
gcc -c exemple.o exemple.c
```

- Imaginons que ce programme ait besoin d'une librairie malibrairie est rangée dans /usr/lib/libmalibrairie.a
- l'éditeur de lien ld rassemble la librairie et exemple.o pour en faire un exécutable exemple

```
ld -L/usr/lib -lmalibrairie exemple.o -o exemple
```

- l'option -static force l'édition de lien à ne prendre que des librairies statiques.
- On peut aussi appeler l'édition de lien directement par gcc ou g++

Librairies dynamiques

- L'utilisation des librairies statiques produit un fichier binaire plus gros
  - si 1000 programmes utilisent la librairie malibrairie.a on aura 1000 fois le code de la librairie inclus dans des programmes
  - cela provoque une perte d'espace disque
- on peut préférer utiliser des librairies dynamiques d'extension so
  - dans ce cas, il faut être sûr que la librairie est bien installée sur la machine cible.
  - (rappelez vous des dll not found, ces dll sont des librairies dynamiques sous windows)
- Ainsi le développeur choisit le bon compromis entre les deux librairies :
  - un code compact mais il faut s'assurer que les librairies sont présentes
  - ou un code plus lourd qui englobe toutes les librairies et alourdit le système.

Utilisation des librairies

- Pour voir les librairies utilisées par un programme :

```
ldd programme
```

- Pour lancer un programme en indiquant un chemin temporaire pour une librairie :

```
LD_LIBRARY_PATH=/chemin/vers/la/librairie  programme
```

- Pour installer définitivement une librairie dans un emplacement déclarer le chemin dans un fichier conf présent dans /etc/ld.so.conf.d et ne pas oublier de lancer la commande

```
ldconfig
```

Les Makefiles

- Ce sont des fichiers nommés **M**akefile (avec un M majuscule)
- Ils permettent de lancement de la compilation et de l'installation des programmes
- Ils agissent en cascade de répertoire en répertoire
- Ils contiennent des variables parfois en grand nombre
- À la fin de ces fichiers on trouve des étiquettes de la forme :
  - all : help : clean : distclean : install : uninstall :
  - ces noms d'étiquettes sont des conventions qui peuvent varier
- `make clean` : efface tous les fichiers compilés
- `make all` : effectue toutes les compilations
- `sudo make install` : installe les logiciels une fois ceux-ci compilés

Configure

- Parfois vous disposez des sources d'un logiciel et vous ne verrez pas le fichier Makefile mais uniquement un fichier configure.
- Cette commande configure doit être lancé par la syntaxe `./configure` éventuellement suivie d'options (./configure –help)
- Elle va vérifier que les composants nécessaires à l'élaboration du fichier Makefile sont présents sur votre machine de développement et si tous les composants sont présents, il va fabriquer le Makefile
- Mais attention les fichiers configure ne contiennent que les dépendances que le développeur à pensé y inscrire.
- L'option - `-prefix` permet d'indiquer où le logiciel sera installé (par défaut /usr/local/)
- Une fois que la commande ./configure ne donnera plus de message d'erreur, le fichier Makefile sera créé, on pourra alors lancer la commande

```
make
```

Problèmes de compilation

- Souvent on obtient des erreurs de compilation de programmes très éprouvés et réputés fonctionner.
- Ces erreurs peuvent être dues :
  - À votre version de Linux plus récente que celle qui a servi à mettre au point le programme et qui introduit de nouvelles instructions incompatibles
  - À la version de Linux qui utilise d'autres librairies que la version Linux qui a servi à l'élaboration du logiciel.
  - À des fichiers h ou des librairies que le compilateur s'attend à avoir sur votre système et qui ne sont pas présents (il faut alors les installer). Souvent on les trouve dans des packages nom-dev

Installation d'une librairie sous Debian

- Pour compiler un programme qui a besoin de la librairie libtruc on devra
  - installer la librairie libtruc par une commande :
    - `apt-get install libtruc` (sur les souches Debian)
  - installer les fichiers includes :
    - `apt-get install libtruc-dev` (sur les souches Debian)
- Sous les systèmes à paquetage rpm, les paquetage include sont nommés non pas dev mais devel
- Il peut arriver que les paquetages sous Debian se nomment devel s'ils proviennent d'un rpm converti en deb par l'utilitaire `alien`
- Exercice : installer Extreme tux racer à partir des sources

Aide par freenode

Rubriques

- Présentation
- Usage

Présentation

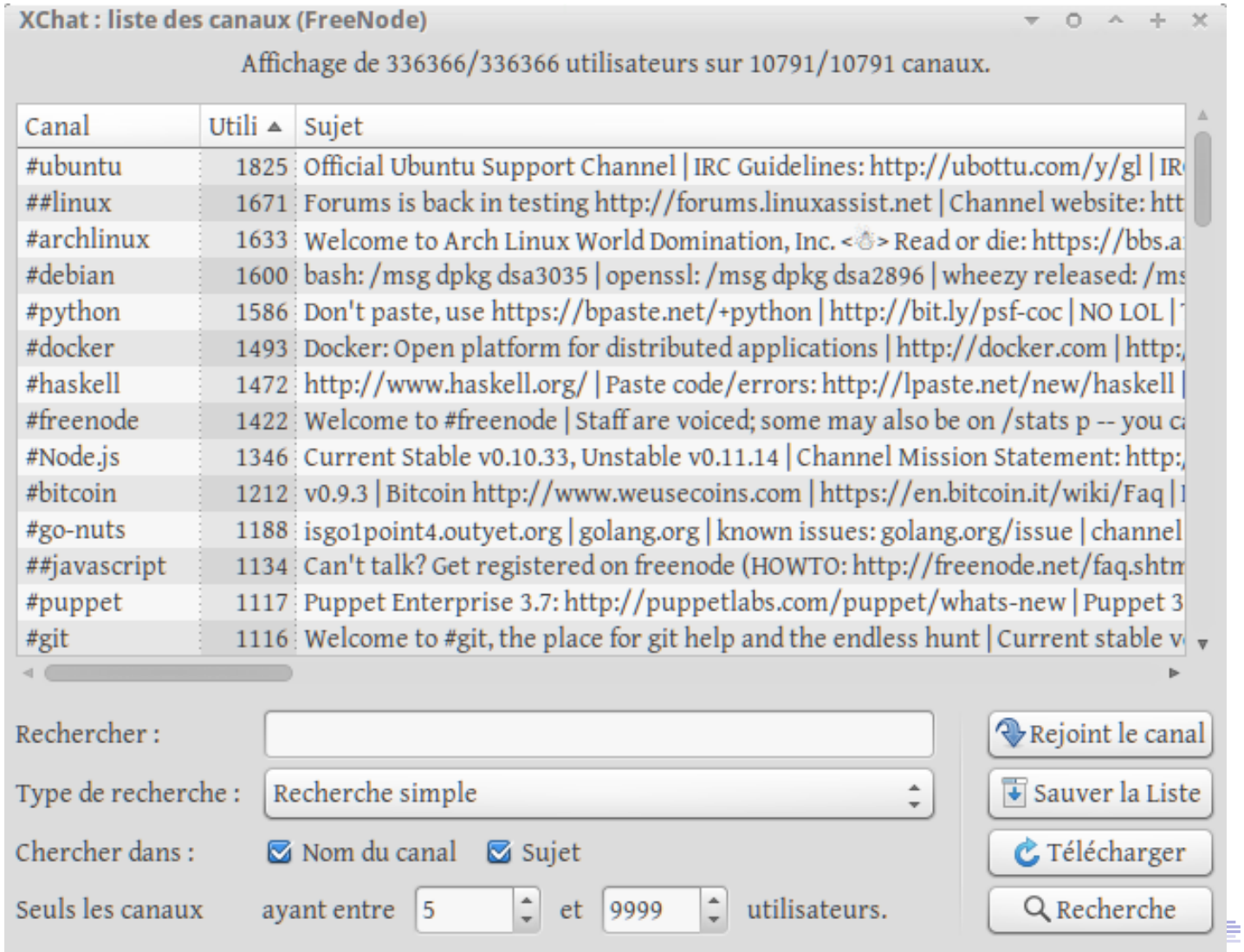
L'aide par IRC

- IRC est un protocole de chat qui est utilisé pour l'entraide dans les développements des logiciels libres
- Les réseaux IRC techniques Open Source se trouvent principalement sur Freenode pour la partie généraliste, mais on trouve également des serveurs IRC consacrés au langage Perl, à la distribution Debian (OTFC), aux Bitcoins
- L'utilisation d'IRC vient à remplacer les forums qui ont tendance à ne plus s'adresser qu'à des utilisateurs très novices.
- Nous allons apprendre à nous servir d'IRC et à respecter les usages en pratique sur ce type de réseau

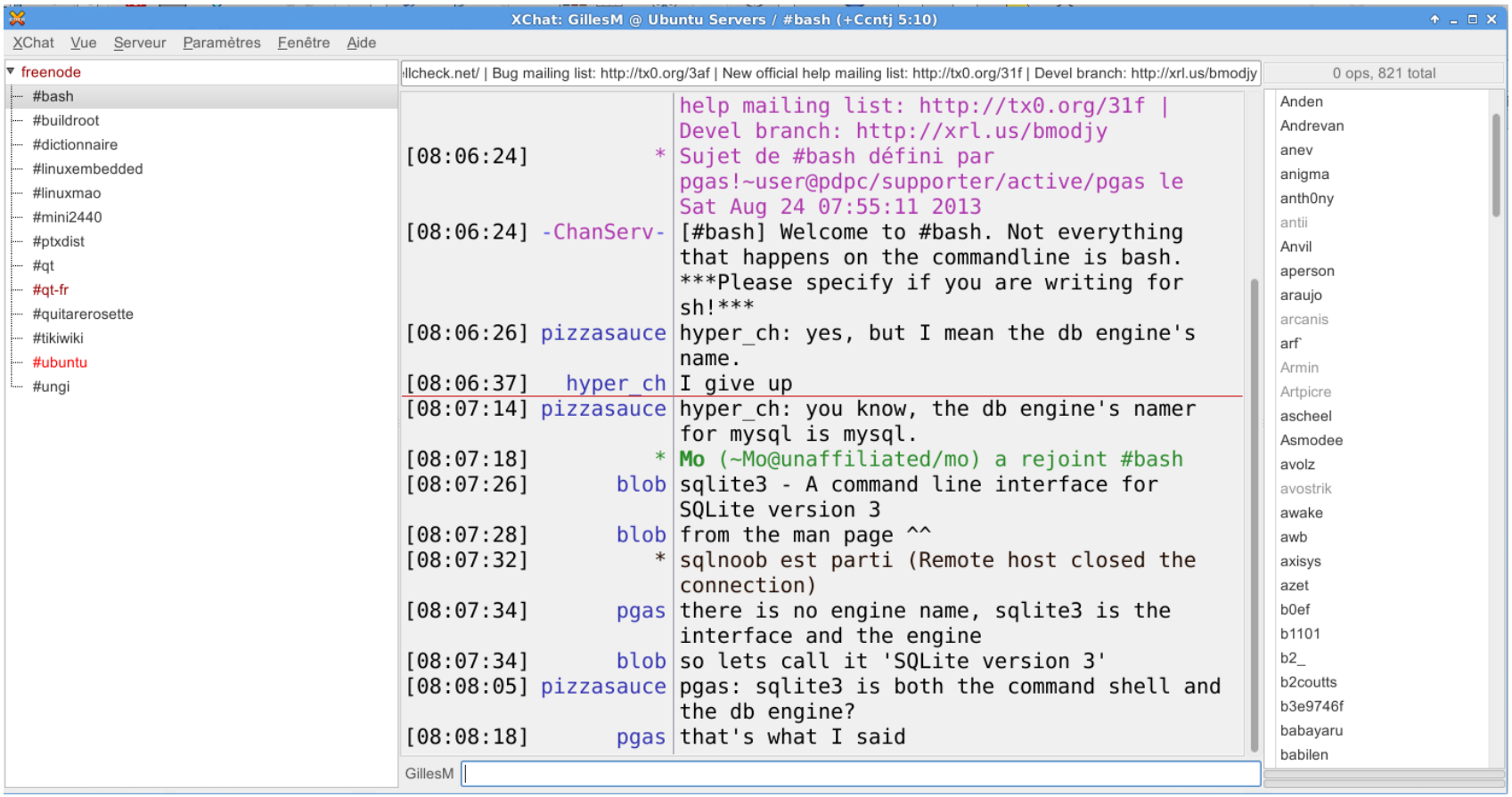
IRC : Présentation du réseau Freenode

- **Freenode :** réseau de type IRC
  - 300 000 utilisateurs
  - plus de 10 000 canaux répartis par logiciel
- **Serveur :** irc.freenode.net
- **Port :** 8001
- Protection des mots de passe par mécanisme NICKSERV
- Utilisé comme canal d'aide par beaucoup de logiciels Open Sources.
- Passerelle Web à http://webchat.freenode.net

Liste des canaux



Exemple de canal



Usage

Logiciels IRC

- Logiciels spécialisés
  - On se connecte à Freenode par un logiciel spécialisé IRC (port 8001) : Linux(XChat, Quassel, Hexchat, Konversation, KVirc, Pidgin) Windows (mirc)
  - Ces logiciels, gardent la trace des précédentes discussions permettant de retrouver des informations contenues dans d'anciennes discussions, ils affichent la liste des canaux de discussions.
- On trouve pour certains réseaux notamment freenode une interface web http://webchat.freenode.net
  - qui nécessite de connaître le canal sur le quel on veut se connecter car la passserelle n'en fournit pas la liste.
  - c'est une interface qui passe par les ports 80 ce qui peut s'avérer pratique si on ne peut utiliser qu'un port http à cause d'un Firewall.

Liste des commandes

- On peut entrer des commandes IRC via la zone de saisie de texte en commançant la ligne par un / en première colonne.
- Voici quelques commandes :
  - **/help :** donne la liste des commandes
  - **/help commande:** aide sur la commande
  - **/msg utilisateur:** envoi d'un message à un utilisateur dans le salon
  - **/query utilisateur message :** envoi d'un message à un utilisateur en privé

Freenode : Nickserv

- Nickserv un service permettant de réserver son pseudo sur Freenode

```
/msg NickServ REGISTER password youremail@example.com
```

- Ensuite pour se connecter :

```
/msg NickServ IDENTIFY monnick password
```

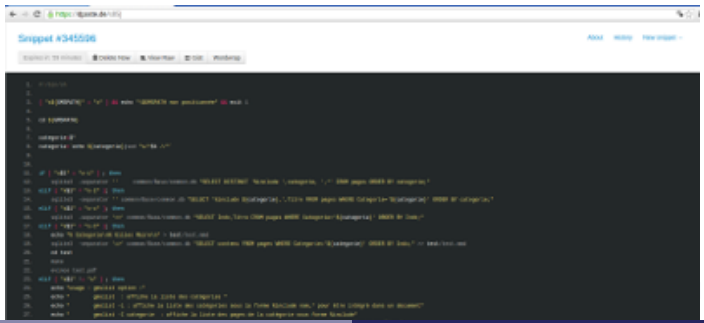
- La connexion automatique se fait dans l'option de connexion du logiciel IRC

Pastebin

- Pastebin est un service Web permettant de copier du code pour ne pas polluer les fenêtres des salons
- Il est directement accessible depuis certains environnements de développement.
- Dans tous les cas, vous copiez les sources qui apparaîtront formatées dans le service pastebin ; une adresse URL vous est communiquée que vous indiquez dans IRC.
- Si vous accédez à pastebin par le web l'adresse est au choix : dpaste.org, fpaste.org ou pastebin.ca
- Le résultat à communiquer vous sera renvoyé par exemple

`http://pastebin.ca/3PYeZEgL`

- Les sites pastebin vous propose de conserver la trace pendant une durée de temps paramétrable. Pour des raisons de sécurité, ne laissez pas traîner à vie des fichiers de configuration qui pourraient finir sur google.



Imagebin

- Pour communiquer une copie d'écran explicitant un problème ou une solution vous utilisez imagebin à l'adresse : http://imagebin.org
- Sur le même principe que pastebin vous récupérez l'adresse de votre image que vous communiquez sur IRC
- Il est efficace de mettre des pastilles indiquant les zones que vous voulez souligner.(voir l'application gimp http://www.gillesmaire.com/tiki-index.php?page=draw-numbers)
- Dans l'exemple suivant, on montre aux développeurs de l'application Ardour un problème avec quelques pastilles numérotées via imagebin.org