

Administration de machine

Gaétan Richard
gaetan.richard@info.unicaen.fr

L2 – S4 2011/2012

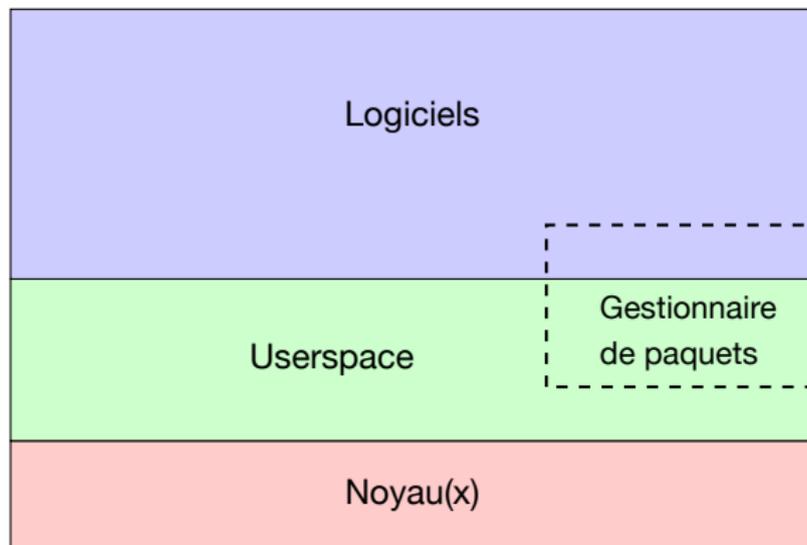
I. Distribution

Définition : Une distribution est un ensemble cohérent de logiciels rassemblés autour d'un noyau pour en faire un système d'exploitation complet.

Ces distributions peuvent prendre des formes très variés :

- ▶ grand public vs. spécialiste ;
- ▶ stable vs. *cutting edge* ;
- ▶ source vs. binaire ;
- ▶ complète vs. de faible taille ;
- ▶ large support vs. spécialisé pour une machine ;
- ▶ support vs. fournit tel quel ;
- ▶ ...

Contenu



« Linux » :

- ▶ **Ubuntu** : dérivé de Debian, grand public ;
- ▶ **Debian** : une valeur sûre, utilisé pour les serveurs ;
- ▶ **Fedora** : dérivée de *Red Hat Linux*, soutenue par Red Hat ;
- ▶ **Gentoo** : distribution source « pour experts » ;
- ▶ **DSLlinux** : pour faire touter Linux sur une DS ;
- ▶ **Damn Small Linux** : un dérivé de Debian fonctionnant sur des configurations réduites (pentium I, 16Mo de RAM) ;
- ▶ ...

Exemples (suite)

BSD :

- ▶ **NetBSD** : tourne sur un très grand nombre d'architectures
- ▶ **freeBSD** : fiable et utilisée pour les serveurs ;
- ▶ **openBSD** : la référence en matière de sécurité ;
- ▶ ...

Si l'ordinateur le supporte, il est possible de lancer l'installation depuis :

- ▶ une disquette ;
- ▶ un lecteur externe ;
- ▶ un lecteur de CD/DVD-ROM ;
- ▶ une clef USB ;
- ▶ le réseau.

Note : ces méthodes se configurent dans le BIOS (ou équivalent). Comme elles donnent un accès total à la machine, il est possible d'en restreindre l'accès par mot de passe.

2. Partitionnement

Nous avons vu dans le cours précédent que le système de fichiers pouvait être composé de « blocs » issus de supports différents.

Ces blocs peuvent être :

- ▶ l'intégralité d'un disque ;
- ▶ une partie d'un disque ;
- ▶ un ensemble de disques.

Chacun de ces blocs peut être écrit dans un format spécifique sur le disque.

Dans une installation, il est courant de diviser le disque en plusieurs partitions différentes. Ceci permet d'avoir des blocs indépendants et d'assurer une meilleure protection des parties sensibles.

On retrouve ainsi souvent :

- ▶ la partition racine ;
- ▶ une partition de type SWAP (qui permet de suppléer la RAM et d'activer l'hibernation) ;
- ▶ une partition utilisateur (/home) ;
- ▶ une partition pour (/var) ;
- ▶ une partition pour (/boot) ;
- ▶ ...

Il existe différents types de format pour le système de fichiers. Les différences portent principalement sur les points suivants :

- ▶ limites (taille et nombre de fichiers) ;
- ▶ présence de journalisation ;
- ▶ possibilité de redimensionner à froid / à chaud ;
- ▶ performances ;
- ▶ ...

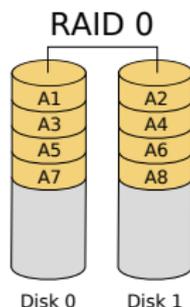
RAID (Redundant Array of Independent Disks) est une méthode pour dispatcher un bloc au travers de plusieurs disques et assurer une certaine redondance :

- ▶ **RAID 0** : deux disques mis en parallèle pour des raisons de performances (pas de redondance) ;
- ▶ **RAID 1** : deux disques contenant les mêmes données ;
- ▶ **RAID 5** : $n + 1$ disques permettant de retrouver l'intégralité des données si 1 est hors-service.

RAID

RAID (Redundant Array of Independent Disks) est une méthode pour dispatcher un bloc au travers de plusieurs disques et assurer une certaine redondance :

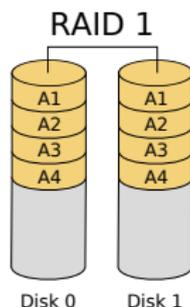
- ▶ **RAID 0** : deux disques mis en parallèle pour des raisons de performances (pas de redondance) ;
- ▶ **RAID 1** : deux disques contenant les mêmes données ;
- ▶ **RAID 5** : $n + 1$ disques permettant de retrouver l'intégralité des données si 1 est hors-service.



RAID

RAID (Redundant Array of Independent Disks) est une méthode pour dispatcher un bloc au travers de plusieurs disques et assurer une certaine redondance :

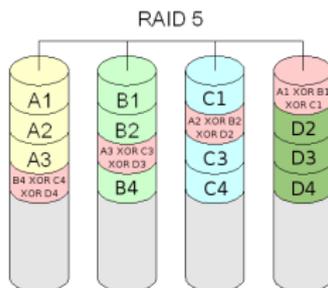
- ▶ **RAID 0** : deux disques mis en parallèle pour des raisons de performances (pas de redondance) ;
- ▶ **RAID 1** : deux disques contenant les mêmes données ;
- ▶ **RAID 5** : $n + 1$ disques permettant de retrouver l'intégralité des données si 1 est hors-service.



RAID

RAID (Redundant Array of Independent Disks) est une méthode pour dispatcher un bloc au travers de plusieurs disques et assurer une certaine redondance :

- ▶ **RAID 0** : deux disques mis en parallèle pour des raisons de performances (pas de redondance) ;
- ▶ **RAID 1** : deux disques contenant les mêmes données ;
- ▶ **RAID 5** : $n + 1$ disques permettant de retrouver l'intégralité des données si 1 est hors-service.



3. Compte administrateur

Il existe un utilisateur privilégié qui possède tous les droits : l'utilisateur **root** dont l'identifiant est **0**.

Pour changer d'utilisateur, il existe la commande **su**.

La commande **sudo** permet d'effectuer certaines (ou toutes les) commandes en tant que super-utilisateur (ou autre). Les autorisations données sont stockés dans le fichiers `sudoers`. Il est également possible de connaître ses propres droits à l'aide de l'option `-l`.

Il existe aussi des comptes utilisateurs et pour les applications qui possèdent des droits restreints.

Important : l'utilisation d'un compte administrateur doit être restreint au minimum pour des raisons de sécurité.

4. Paquet

Dans une distribution, les logiciels sont encapsulés dans des **paquets**.

Ces paquets sont regroupés en un emplacement appelé **dépôt**. Il existe souvent plusieurs dépôts contenant différentes variations d'une même distribution.

Des programmes spécifiques se chargent alors des installations et suppressions : les **gestionnaires de paquet**.

Contenu d'un paquet

Dans un paquet, on trouve généralement :

- ▶ le programme (sous forme d'exécutable, de source ou de lien) ;
- ▶ des patches ;
- ▶ des scripts d'installation ;
- ▶ son numéro de version ;
- ▶ les dépendances ;
- ▶ une description ;
- ▶ ...

Du fait des contraintes sur le logiciel et sa licence, on retrouve ce modèle presque exclusivement dans le monde du **logiciel libre**.

Exemple

Ubuntu : apt-get, aptitude, synaptic, ...

Quelques mots sur l'aspect sécurité.

5. Installation manuelle

- ▶ Récupérer le code source ;
- ▶ configurer l'installation ;
- ▶ compiler ;
- ▶ installer ;
- ▶ configurer l'utilisation ;
- ▶ utiliser ;
- ▶ maintenir à jour.

Récupérer le code source

Release :

- ▶ Correspond à une version stable ;
- ▶ le plus souvent fournit sous la forme d'une archive `.tar.gz` ou `.tar.bz2` ;
- ▶ À décompresser avec la commande **tar**.

Développement :

- ▶ Version en mouvement ;
- ▶ Accès direct au dépôt de travail collaboratif (format `cv`s, `svn`, `mercurial`, `git`, ...);
- ▶ Utiliser le logiciel ad-hoc ;
- ▶ Existe parfois également en « snapshot ».

Dans la plupart des projets, les fichiers et dossiers suivants sont présents :

- ▶ README : un peu de lecture ;
- ▶ INSTALL : documentation d'installation ;
- ▶ ChangeLog : les changements par version.
- ▶ Doc/ : la documentation ;
- ▶ src/ : les sources.

La numérotation des versions est propre à chaque application mais on peut distinguer le schéma général suivant :

major.**minor**-**variante**

- ▶ **major** est le numéro de version majeure, il est incrémenté lors de gros changement introduisant des incompatibilités (0 est souvent utilisé pour le début de développement) ;
- ▶ **minor** est le numéro de version mineur (dans certain cas, un numéro impaire indique une version de développement, 99 désigne souvent la version préparatoire à la version majeure suivante) ;
- ▶ **variante** sert à indiquer le stade de développement
 - ▶ **a1, a2, ...** : versions **alpha** (ajout de fonctionnalités) ;
 - ▶ **b1, b2, ...** : versions **beta** (correction de bugs, test) ;
 - ▶ **RC1, RC2, ...** : versions **Release Candidate** (test avant release) ;
 - ▶ **1, 2, ...** : **patchlevel** (patches)

Configuration de l'installation

le script **configure** permet de vérifier la présence (et la version) des dépendances, de choisir différentes options de compilation. Il génère alors les fichiers (Makefile) nécessaire pour la compilation.

Parmi les options courante de **configure**, on retrouve en particulier :

- ▶ `-help` : affichez la liste des options disponibles ;
- ▶ `-prefix=/chemin/` : installe le logiciel dans le répertoire */chemin/* ;
- ▶ `-with-X / -without-X` : active / désactive la fonctionnalité *X* ;
- ▶ `-enable-X / -disable-X` : comme précédemment.

Note : dans certains cas, ils est nécessaire de générer l'exécutable **configure** à partir du fichier `configure.in` en utilisant le programme **autoconf**.

Compilation

La compilation s'effectue en utilisant la commande **make** qui utilise les fichiers `Makefile`. Il est possible de préciser à `make` un **cible** pour désigner la tâche à effectuer.

Usuellement les tâches reconnues sont les suivantes :

- ▶ **make** : compile le programme ;
- ▶ **make install** : effectue l'installation (nécessite parfois d'être super-utilisateur) ;
- ▶ **make clean** : nettoie les fichiers générés ;
- ▶ **make doc** : génère la documentation.

Note : il existe également des alternatives à `make` utilisés par certains projets : **scons**, **ant**, ...

Configuration du programme

Finalisation : Il reste alors à adapter les fichiers de configuration généraux et par utilisateur. Cette étape est dépendante de l'application mais il est courant de fournir des fichiers commentés avec les options les plus courantes.

6. Maintenance

Une fois un système et des logiciels installés, il reste à assurer que l'ensemble des composants restent à jour, en particulier en appliquant les mises à jour de sécurités.

Fichiers de logs

Afin de surveiller le bon fonctionnement, de nombreuses informations importantes sur les évènements survenus sont stockés dans les fichiers de logs situés dans le répertoire ; `/var/log`.

On y trouve entre autre les fichiers :

- ▶ `wtmp` contient les connections (accessible par la commande **last**) ;
- ▶ `syslog` contient les informations systèmes ;
- ▶ `daemon.log` contient les informations des programmes lancés en mode démon ;
- ▶ `kern.log` contient les évènements du noyau.

Ces fichiers sont gérés par l'intermédiaire de **logrotate** qui permet de ne conserver les logs que sur une durée limitée et de les compresser (pour des raisons de place).