M6 : Administration Système (TP3)

Olivier Dalle – Katell Morin-Allory – Eric Vecchie

---

1 Utilisation et configuration de syslog

Il est possible d'envoyer des messages à syslogd en utilisant une API C (openlog(), syslog(), closelog()). Lors de ce TP nous utiliserons plutôt la commande `logger(1)`.

- Générer le log (**daemon**.**emerg**) en utilisant la commande `logger(1)`.

Nous allons maintenant modifier le fichier de configuration de syslogd (/etc/syslog.conf). Il est possible de réinitialiser syslogd en lui envoyant un SIGHUP ce qui le faire relire le fichier de configuration.

Une autre solution, plus élégante, consiste à utiliser le script système de démarrage de syslog, de la façon suivante :

```
/etc/init.d/syslog restart
```

- Ajouter une entrée pour écrire les logs user dans /var/log/user
- Envoyer un signal **user**.**emerg**. Quelles sont les règles dans syslog.conf qui s'y appliquent ?
- Modifier la règle *.**emerg** pour ne pas traiter les logs **user** (2 façons)
- Modifier la règle concernant /var/log/messages pour qu'elle ignore seulement les logs **mail** de priorité strictement inférieure à **crit**.
- écrire une règle qui place les logs de **news**,**mail** et **user**de niveau supérieur ou égal à **crit** dans /var/log/messages
- Écrire une règle pour que les logs **local0** de **warning** à **alert** soient placés dans /var/log/local0

2 Observation du système

- Installez tripwire
- Lancez une vérification avec tripwire. Des modifications sont détectées ? Lancez tripwire de façon à valider définitivement ces modifications.
- Modifiez l'attribut "Immutable" de quelques fichiers de /sbin, et relancez une vérification avec tripwire. La modification a-t-elle bien été détectée ?

3 Crontab

- Programmer une tâche qui toutes les minutes affichera "Bonjour" sur tous les terminaux.
- Où cron met-il les informations pour savoir quelle tâche exécuter ?
- Programmer la vérification du système de fichiers pour le vendredi soir à 20h00.
- Changer la date et l'heure du système pour verifier que la tâche précédente s'exécute bien (`man date` !).

Pour les questions suivantes, il faudra créer un utilisateur "test" sans aucun droit particulier

- Vérifier que la configuration actuelle permet à n'importe qui d'utiliser cron
- Créer un fichier /etc/cron.allow vide. Quelle conséquence cela a-t-il pour "test" ?
- Supprimer le fichier et créer un fichier vide /etc/cron.deny. Test peut-il programmer des taches ?
- Quel est le comportement si un nom d'utilisateur apparait dans le fichier cron.allow et aussi dans le fichier cron.deny ?
- Un utilisateur indélicat a programmé une tache génante (par exemple un wall toutes les minutes). L'ajouter dans le fichier cron.deny et le supprimer de cron.allow suffit-il à arréter ses taches ?

4 Création de RPM

La création d'un RPM des sources du noyau est un problème un peu trop compliqué pour débuter ! Nous allons donc commencer par fabriquer le paquetage RPM pour un tout petit programme, le fameux "hello world !".

- Placez-vous dans votre homedir. Créez-y un répertoire de nom `hello-1.0`.
- Dans ce répertoire, écrivez un programme C affichant simplement la chaine "Hello World !".
- Ecrivez un Makefile dont la première cible permet de compiler votre programme. Ce Makefile devra aussi comporter une cible "clean" pour faire le ménage.
- Une fois que vous avez testé que votre makefile et votre programme fonctionnent correctement, revenez dans votre homedir, et créez une archive `hello-1.0.tar.gz` contenant le répertoire hello.

- Créez ensuite un répertoire `RPMtest` contenant les sous-répertoires suivants : `BUILD, RPMS, SOURCES, SPECS` et `SRPMS`.
- Dans le répertoire `SPECS` éditez un fichier de nom `hello-1.0.spec` sur le modèle de celui qui vous a été présenté en cours. Attention, n'oubliez pas de préciser le nom du source :
  `Source: hello-1.0.tar.gz`
- Quelques indications :
  - Vous demanderez que la fabrication soit faite dans un répertoire temporaire plutot que dans la véritable arborescence du système :
    `Buildroot: %{_tmppath}/%{name}-root`
  - Pour la construction (à la suite de `%build`), il suffit d'indiquer `make`, qui est la commande qui suffit dans notre cas à compiler le programme
  - Pour l'installation, nous allons demander à ce que le binaire soit installé dans `/bin` :
    ```
    %install
    rm -rf $RPM_BUILD_ROOT
    mkdir -p $RPM_BUILD_ROOT/bin
    install -m 755 hello $RPM_BUILD_ROOT/bin/hello
    ```
  - N'oubliez pas de donner la liste des fichiers que contient notre paquetage (`/bin/hello`) dans la section `%files` !
- Créez dans votre homedir un fichier `.rpmmacros` contenant les déclarations suivantes :
  ```
  %_topdir /root/RPMtest
  %packager Prénom NOM
  ```
- Vérifiez que la construction des paquetages sources et binaires se déroule bien en invoquant la commande suivante :
  `rpm -ba SPECS/hello-1.0.spec`
- Vous pouvez maintenant installer ou retirer à loisir votre paquetage, ou regarder ce qu'il contient en tapant par exemple :
  `rpm -qpil RPMS/hello-1.0-R1.i386.rpm`