

## TP 6 — Administration des systèmes

Le TP doit être fait sur l'image `mageia4` qui doit être restaurée en début de séance.

### Exercice 1 — Tâches à exécution différée

La commande `at` permet de programmer des tâches à exécuter une seule fois à un instant donné.

- Q. 1.1 Les tâches sont lancées par le service `atd`. Démarrer ce service si nécessaire.
- Q. 1.2 Créer un fichier `/tmp/fic` vide. Programmer la suppression du contenu du répertoire `/tmp` dans 1 minute. Vérifier après 1 minute que `/tmp/fic` a bien été supprimé.
- Q. 1.3 Programmer le redémarrage de l'ordinateur (commande `reboot`) à 1 heure du matin.
- Q. 1.4 Afficher la liste des tâches programmées (commande `atq`) et annuler le redémarrage.

### Exercice 2 — Tâches périodiques

La commande `crontab` permet de créer des tâches qui seront exécutées périodiquement par le service `crond`. Les possibilités d'utilisation les plus courantes sont :

- `crontab -e` ⇒ pour éditer les tâches
- `crontab -l` ⇒ pour voir la liste des tâches
- `crontab -r` ⇒ pour supprimer toutes les tâches

L'éditeur par défaut lancé par `cron` pour saisir des règles est `vi`. On peut le modifier via la variable d'environnement `EDITOR` qui doit contenir le chemin complet de l'éditeur à utiliser. Par exemple :

```
$ export EDITOR=/usr/bin/emacs
$ crontab -e # on ouvre emacs plutôt que vi
```

- Q. 2.1 Il faut que les services `crond` et `postfix` soient démarrés pour pouvoir exécuter les tâches `cron` et envoyer les mails écrits par `crond`. Démarrer ces services si nécessaire.
- Q. 2.2 Modifier l'éditeur pour un éditeur de votre choix (`emacs`, `gedit`, ...) puis éditer les règles `cron` et créer une règle ajoutant la date courante à la fin du fichier `/tmp/la_date` toutes les minutes. Utiliser pour cela la commande `date` suivie d'une redirection.
- Q. 2.3 Éditer les règles `cron` et affecter à la variable `MAILTO` votre adresse email pour recevoir à cette adresse le résultat des tâches exécutées par `crond`. Toujours en utilisant la commande `date`, créer une règle affichant "salut, la date est <date>." toutes les 5 minutes pendant la séance de TP (et pendant la séance uniquement).
- Q. 2.4 Supprimer le contenu de la table `cron` après avoir vérifié que les tâches programmées sont bien exécutées.

### Exercice 3 — Analyse des fichiers des utilisateurs et groupes

Les descriptions des fichiers et des commandes relatives aux groupes et utilisateurs sont données dans le cours 4 et dans le memo des commandes.

- Q. 3.1 Consulter le fichier `/etc/passwd` pour trouver les informations ci-dessous :
  - UID de l'utilisateur `root`
  - GID du groupe primaire de l'utilisateur `root`
  - shell de l'utilisateur `root`
  - répertoire personnel de l'utilisateur `root`
- Q. 3.2 Quel est le nom du groupe primaire de l'utilisateur `root` ?
- Q. 3.3 Quels sont les utilisateurs ayant un mot de passe crypté ?

### Exercice 4 — Modification des groupes et utilisateurs

- Q. 4.1 Créer un utilisateur `bilou` d'UID 1234 ayant pour shell `/bin/bash`. Consulter le manuel pour trouver les options adéquates.
- Q. 4.2 Consulter les fichiers `/etc/passwd`, `/etc/shadow` et `/etc/group`. Que constate-t-on après la création de la question précédente ? Quel est le répertoire personnel du nouvel utilisateur ? Quels sont les droits associés à ce répertoire ?
- Q. 4.3 Affecter un mot de passe au nouvel utilisateur. Quel modification a été apportée au fichier `/etc/shadow` ?

- Q. 4.4** Vérifier que le compte a bien été créé avec le mot de passe associé en ouvrant un terminal puis en ouvrant dans celui-ci une session sous l'identité `bilou`. Fermer ensuite le terminal.
- Q. 4.5** Créer un groupe TP-RT.
- Q. 4.6** Consulter le fichier `/etc/group`. Que constate-t-on ?
- Q. 4.7** Peut-on supprimer le groupe primaire de `bilou` avec la commande `groupdel` ? Pourquoi ?
- Q. 4.8** Ajouter le groupe TP-RT en groupe secondaire de l'utilisateur `bilou` en éditant le fichier `/etc/group`. Vérifier avec la commande `id` que l'ajout s'est bien passé.

### Exercice 5 — Le fichier `.bashrc`

Le fichier `.bashrc` à la racine du répertoire personnel d'un utilisateur est un script lancé à l'ouverture d'une session `bash` (c'est-à-dire quand on ouvre un terminal). On l'utilise généralement pour modifier certaines variables d'environnement ou pour afficher des informations pour l'utilisateur.

- Q. 5.1** Modifier le fichier `.bashrc` de votre répertoire personnel afin que :
- Le message "Bienvenue <nom-utilisateur>, tu es sur la machine <nom-machine>" soit affiché à l'ouverture de la session. Utiliser pour cela les variables d'environnement `$USER` et `$HOSTNAME` qui contiennent le nom de l'utilisateur connecté et le nom de sa machine.
  - La date soit affichée.
  - L'`umask` soit tel qu'à la création d'un nouveau fichier (ou répertoire) :
    - aucun droit ne vous soit retiré
    - le droit en écriture soit retiré à tous les utilisateurs du groupe
    - tous les droits soient retirés pour les autres utilisateurs
- Q. 5.2** Tester en ouvrant un nouveau terminal, puis en créant un fichier et un répertoire et en vérifiant les droits qui leur sont associés.
- Q. 5.3** La variable `PS1` contient une chaîne de caractères qui correspond au *prompt*. Le prompt est le message d'invite affiché en début de ligne par l'interpréteur de commandes. Modifier votre fichier `.bashrc` pour que ce prompt soit de la forme suivante :

```
<login> sur <machine> dans le répertoire <chemin-complet-du-repertoire-courant>  
<heure> $
```

Ce qui pourra par exemple donner :

```
etudiant sur q20305 dans le répertoire ~/TP7  
10:44 $
```

La variable `PS1` peut contenir des méta-caractères de la forme `'\X'` qui seront remplacés à l'affichage du prompt par une valeur donnée (le login de l'utilisateur connecté, par exemple). Rechercher sur Internet les méta-caractères adéquats.