Rapport TP1
Administration Système

Année Universitaire 2024-2025

Master Professionnel (1ère année)

École Supérieure Polytechnique

Département Génie Informatique

Ousseynou Seck

Sous la direction de Dr Mandicou Ba

Université Cheikh Anta Diop

Juin 2025

Contents

1 Introduction

Ce rapport présente les travaux pratiques réalisés dans le cadre du TP1 d'Administration Système pour l'année universitaire 2024-2025. L'objectif est de mettre en pratique des notions fondamentales telles que la journalisation, la segmentation des privilèges, la création de documentation, l'automatisation et la planification de tâches. Les exercices couvrent l'utilisation des outils `script`, `sudo`, `troff`, `expect` et `crontab`.

2 Exercice 1 : Journalisation avec la commande script

2.1 Utilité de l'option -a

La commande `script -a` permet d'enregistrer les actions effectuées dans une session de terminal dans un fichier, généralement nommé `typescript`. L'option `-a` (append) ajoute les nouvelles entrées à la fin du fichier existant au lieu de l'écraser.

2.2 Suite de commandes de test

Les commandes suivantes ont été exécutées sous `script -a` :

- `date` : Affiche la date et l'heure actuelles.
- `id` : Affiche les informations sur l'utilisateur courant (UID, GID, groupes).
- `uname -a` : Fournit des informations détaillées sur le système (nom du noyau, version, etc.).
- `uptime` : Indique la durée de fonctionnement du système.

2.3 Visualisation du résultat

Pour quitter la session `script`, la commande `exit` est utilisée. Le contenu du fichier `typescript` est visualisé avec `cat typescript` ou `less typescript`, affichant toutes les commandes exécutées et leurs sorties.

2.4 Solution pour exécution automatique

Pour automatiser l'exécution de `script`, on peut modifier le fichier `.bashrc` de l'utilisateur pour inclure :

```
script -a ~/logs/session_$(date +%F_%H-%M-%S).log
```

Cela crée un fichier de log unique à chaque ouverture de session avec un horodatage.

3 Exercice 2 : Utilisation de la commande sudo

3.1 Création du compte Pierre

Un nouvel utilisateur `pierre` a été créé avec :

```
sudo adduser pierre
```

Le mot de passe a été modifié avec :

```
sudo passwd pierre
```

3.2 Configuration des privilèges de Pierre

Pour permettre à `pierre` de créer des comptes utilisateurs, le fichier `/etc/sudoers` a été modifié avec `visudo` pour ajouter :

```
pierre ALL=(ALL) /usr/sbin/adduser, /usr/sbin/passwd
```

Cela restreint `pierre` à exécuter uniquement `adduser` et `passwd` avec `sudo`.

3.3 Création de l'utilisateur Titeuf

En tant que `pierre`, l'utilisateur `titeuf` a été créé avec :

```
sudo adduser titeuf
```

3.4 Affichage de l'identité de Titeuf

L'identité de `titeuf` a été vérifiée avec :

```
id titeuf
```

Sortie : `uid=1001(titeuf) gid=1001(titeuf) groups=1001(titeuf)`.

3.5 Configuration plus efficace

Une configuration plus efficace consiste à créer un groupe `admins` et à accorder des privilèges au groupe. Création du groupe :

```
sudo groupadd admins
```

Ajout de `pierre` au groupe :

```
sudo usermod -aG admins pierre
```

Modification de `/etc/sudoers` :

```
%admins ALL=(ALL) /usr/sbin/adduser, /usr/sbin/passwd
```

Cela permet une gestion centralisée des privilèges pour plusieurs utilisateurs.

3.6 Test de la configuration

Un nouvel utilisateur a été ajouté au groupe `admins`, et la création d'un utilisateur a été testée avec succès, confirmant le bon fonctionnement de la configuration.

4 Exercice 3 : Création de documentation avec Troff

4.1 Fichier backup.1

Un fichier `backup.1` a été créé avec `troff` pour documenter le script `backup.sh`. Le contenu est le suivant :

```
.TH BACKUP 1 "Juin 2025" "Version 1.0" "Manuel Utilisateur"
.SH NOM
backup.sh \- Script pour sauvegarder des répertoires
.SH SYNOPSIS
.B backup.sh
.I source_directory destination_directory
.SH DESCRIPTION
Le script \fBbackup.sh\fR effectue une sauvegarde d'un répertoire source vers un répe
.IP \(bu 4
Vérification de l'existence des répertoires source et destination.
.IP \(bu 4
Exécution de la sauvegarde avec \fBrsync -av\fR.
.IP \(bu 4
Journalisation de l'opération dans un fichier de log.
.SH OPTIONS
.TP
.I source_directory
Répertoire contenant les fichiers à sauvegarder.
.TP
.I destination_directory
Répertoire où les fichiers seront sauvegardés.
.SH EXEMPLE
.B backup.sh /home/user/documents /mnt/backup
.SH AUTEUR
Ousseynou Seck, étudiant en administration système (L3GLSI).
```

4.2 Test de la page de manuel

La page a été testée avec :

```
man ./backup.1
```

Elle s'affiche correctement avec toutes les sections.

4.3 Macros Troff utilisées

- `.TH` : Définit l'en-tête de la page de manuel (titre, section, date, version).
- `.SH` : Crée une section (par exemple, NOM, SYNOPSIS).
- `.B` : Affiche le texte en gras (utilisé pour les commandes).
- `.I` : Affiche le texte en italique (pour les arguments).
- `.IP` : Crée un paragraphe indenté avec une puce.
- `.TP` : Définit une paire étiquette/description pour les options.

5 Exercice 4 : Automatisation avec Expect

5.1 Script automate$_s$sh.exp

Le script suivant automatise une connexion SSH et la mise à jour du système :

```
#!/usr/bin/expect
set timeout 20
spawn ssh user@remote_host
expect "password:"
send "mypassword\r"
expect "$ "
send "sudo apt update && sudo apt upgrade -y\r"
expect "password for user:"
send "mypassword\r"
expect "$ "
send "exit\r"
expect eof
```

- `spawn` : Lance une commande (ici, `ssh`).
- `expect` : Attend un motif spécifique dans la sortie (par exemple, `password:`).
- `send` : Envoie une chaîne de caractères (par exemple, le mot de passe).

5.2 Alternatives à Expect

- **SSH Keys** : Utiliser des clés SSH pour une authentification sans mot de passe.
- **Ansible** : Outil d'automatisation pour gérer des tâches sur plusieurs serveurs.
- **Bash avec sshpass** : Utiliser `sshpass` pour fournir le mot de passe en ligne de commande.

5.3 Planification avec Crontab

Pour exécuter `automate`$_s$*sh.exptouslesjours*3*h*00 :

```
0 3 * * * /usr/bin/expect /path/to/automate_ssh.exp
```

Ajouté via `crontab -e`.

Pour exécuter `backup.sh` toutes les semaines (par exemple, le lundi à 2h00) :

```
0 2 * * 1 /path/to/backup.sh /home/user/documents /mnt/backup
```

6 Conclusion

Ce TP1 a permis de consolider les compétences en administration système à travers des exercices pratiques. Les notions de journalisation, gestion des privilèges, documentation, automatisation et planification ont été appliquées avec succès, renforçant la compréhension des outils fondamentaux pour un administrateur système.