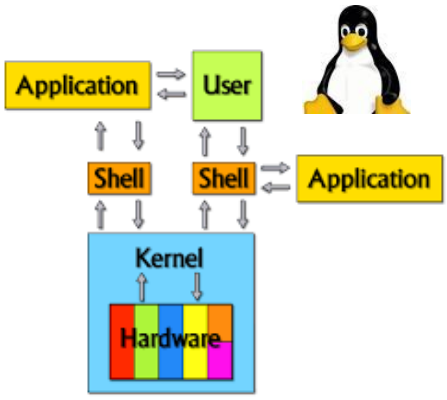
Système d'exploitation

Shell Unix - BASH

Document réponse **Date** _____

Nom : _____ **Classe** : _____

Prénom : _____ **Groupe** : _____



Mots-clés : multi-utilisateur, identifiant de connexion, mot de passe, session, compte, groupe, SSH/SCP, permissions, redirections, script, UID, GID, répertoire, chemin absolu et relatif.

Compétences
- Utilisation du terminal et des commandes de base
- Gérer les droits et permissions d'accès aux fichiers

Objectif

Superviser la température d'un processeur et installer un projet de site web sur un serveur ne disposant pas d'une interface graphique.

CONDITIONS DE DÉROULEMENT DE L'ACTIVITÉ

Phases de travail	Objectifs	Activités
1 - Mise en situation	Découvrir l'environnement de travail.	Lecture de la mise en situation.
2 - Étude de la problématique	Se familiariser avec l'interpréteur de commandes Linux et les commandes de base.	Connexion à une machine en SSH. Utilisateurs et groupes. Création et manipulation de répertoires, de fichiers. Gestion des droits. Redirections. Écriture de premiers scripts shell.
3 - Synthèse	Superviser la température d'un processeur. Installer un site web sur un serveur distant avec la ligne de commande Linux.	Complétion de la page web (HTML). Transfert de fichiers avec SCP. Gestion des droits. Lancement d'un script en tâche de fond. Test dans un navigateur.

Pour aller plus loin
Initiez-vous à Linux
Transférer des fichiers sur la Raspberry Pi sans FTP avec SSH.
QCM Linux

Appréciation

_____

_____

_____

DR1 - Connexion au Raspberry Pi, utilisateurs et groupes

Activité 1

_____**@**_____:**~ $**
| | | |__ **prompt** : **$** ⇔ normal (non administrateur), **#** ⇔super utilisateur (root)
| | | ___ répertoire de travail : **~** ⇔ personnel
| |_____ nom de l'**hôte**
|_____ nom de l'utilisateur (identifiant de connexion textuel)

Activité 2

UID = _____ **GID** =_____ **groupes secondaires :** _____

DR2 - Arborescence de fichiers et chemins

Activité 3

*nom_utilisateur**@**nom_hôte:~ $* _____

_____

Activité 4

*nom_utilisateur**@**nom_hôte:~ $* _____

Activité 5

Chemins relatifs

- *Du répertoire personnel -> test*

*nom_utilisateur**@**nom_hôte:~ $* _____

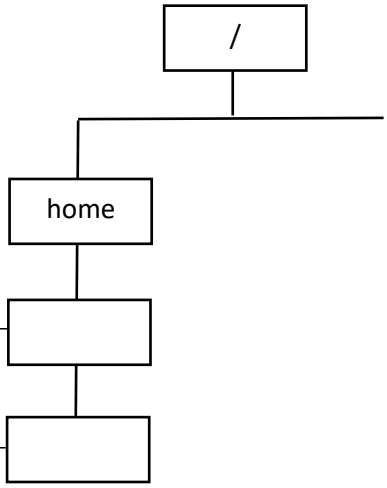
- *Du répertoire test -> home*

*nom_utilisateur**@**nom_hôte:~/test $* _____

Chemin absolu

- Du répertoire home -> test

*nom_utilisateur**@**nom_hôte:/home $* _____



Activité 6

*nom_utilisateur**@**nom_hôte:~ $* _____

Activité 7

*nom_utilisateur**@**nom_hôte:~ $* _____

Activité 8

*nom_utilisateur**@**nom_hôte:~/test $* _____

Texte :_____

DR3 - Permissions (droits) et propriétés (des fichiers et des répertoires)

Activité 9

*nom_utilisateur**@**nom_hôte:~/test $* _____

_____

Activité 10

1. Type : _____

Droits actuels et ***valeur*** correspondante_____

2. *nom_utilisateur***@***nom_hôte:~/test $* _____

3. Vérification de l'accès en écriture par votre voisin : _____

Activité 11

1. *nom_utilisateur***@***nom_hôte:~ $* **chmod 7**_____

DR4 – Redirections

Activité 12

```
#!/bin/bash
echo "Effet des commandes pwd, ls, tree, etc." > test/log.txt
echo 1. pwd >> test/log.txt
pwd >> test/log.txt
```

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

```
Effet des commandes pwd, ls, tree, etc.
1. pwd
/home/bob
2. ls
essai.sh
test
3. tree
.
|-- essai.sh
`-- test
    |-- fichbob.txt
    `-- log.txt

2 directories, 3 files
4. cd test
5. ls -l
total 8
-rw-rw-r-- 1 bob sin  79 26 oct.  12:02 fichbob.txt
-rw-r--r-- 1 bob sin 190 26 oct.  17:05 log.txt
```

Résultat attendu

_____

Annexe 1

Autre méthode pour gérer les permissions sur les fichiers (<u>changement relatif</u>)

Cette méthode permet de spécifier les droits à ajouter ou à supprimer relativement aux droits existants. La commande **chmod** permet de modifier les permissions sur les fichiers avec la syntaxe :

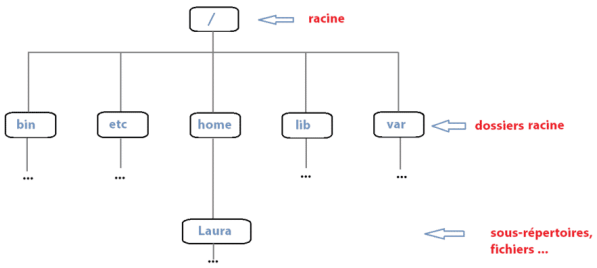
chmod $c_1m_1p_1$, …, $c_nm_np_n$ chemin/fichier

Les $\mathbf{c_i}$ sont les **cibles** et peuvent valoir :	Les $\mathbf{m_i}$ sont les **modifications**	Les $\mathbf{p_i}$ sont les **permissions**
u : propriétaire (user)	**+** : ajouter	**r** : lire (read)
g : groupe (group)	**-** : retirer	**w** : écrire (write)
o : autres (other)		**x** : exécuter (execute)
a : tous (all)		

<u>Exemple</u> : pour ajouter les droits en écriture à toutes les personnes du groupe et retirer les droits en lecture aux autres sur le fichier img_001.jpg situé dans le répertoire Photos, on écrira :

chmod g+w,o-r Photos/img_001.jpg

Annexe 2



Pour Linux tout est fichier !

Le contenu des répertoires (extrait)

Répertoire	Description
/	**Le répertoire racine.**
/bin	exécutables et binaires essentiels (bash, ls, cp, etc.).
/etc	fichiers de configuration du système.
/home	**Chaque utilisateur dispose d'un répertoire personnel dans /home. C'est le seul endroit où il peut écrire des fichiers sans autorisation étendue.**
/lib	fichiers de bibliothèques partagées utilisées par les programmes du système principal.
/var	**On y trouve le contenu web (répertoire www), mais aussi les logs (journaux).**

Les commandes BASH (extrait)

Répertoires et fichiers

Source : https://edurl.fr/pivZhiW3

Nom	Description	Exemples
***cat** fichier*	*Affiche le contenu d'un fichier*	***cat** fichier.txt*
***cd** chemin/répertoire*	*Changement de répertoire*	***cd** /var/www* ***cd** ../ → répertoire parent* ***cd** ou **cd ~** → compte utilisateur*
***chmod** valeur fichier (rep)*	*Change les droits sur un fichier ou un répertoire*	***chmod** 770 fichier.txt*
***cp** source… destination* ***cp -r** source/ destination/*	*Copie de fichiers ou de répertoire*	***cp** fichier.txt chemin/*
***ls** répertoire*	*Affiche le contenu d'un répertoire, le répertoire courant si aucun argument*	***ls** nomrépertoire* ***ls** -l nomrep → avec détails* *-h nomrep → affiche Mo ou Ko (human readable)* *-a nomrep → affiche tous les fichiers (all)*
***less** fichier*	*Affiche le contenu d'un fichier (défilement avec les touches flèche et page)*	*less fichier.txt*
***mkdir** répertoire*	*Crée un répertoire*	***mkdir** site*
***mv** source… destination*	*Déplace ou renomme des fichiers*	***mv** fichier.txt chemin/* ***mv** fichier.txt copie.txt*
nano	*Éditeur de texte*	***nano** fichier.txt*
pwd	*Affiche le nom du répertoire courant.*	
***rm** fichier…* ***rm -r** répertoire*	*Supprime définitivement des fichiers ou des répertoires*	***rm** fichier.txt* ***rm** -r nomrépertoire*
***touch** fichier*	*Crée un fichier*	***touch** fichier.txt*
tree	*Affiche l'arborescence d'un répertoire*	

Divers

Nom	Description	Exemples
clear	*Efface la console*	
exit	*Sortie de la console*	
echo *chaîne*	*Affiche la chaîne de caractères sur le canal de sortie standard.*	***echo** "Bonjour"*
history	*Affiche l'historique des commandes entrées dans le terminal.*	
id	*Affiche l'identifiant utilisateur UID et le groupe GID.*	
***man** nomcommande*	*Manuel (aide en ligne)*	***man** mkdir*

Pour aller plus loin

Initiez-vous à Linux https://edurl.fr/JApAye3P

Transférer des fichiers sur la Raspberry Pi sans FTP avec SSH. https://edurl.fr/4s1IQW0l

QCM Linux https://edurl.fr/nb40xPSZ

Nom : _____

Prénom : _____

DR5 – Scripts Shell

Scripts Shell et programmation

« Un script Shell permet d'automatiser une série d'opérations. Il se présente sous la forme d'un fichier contenant une ou plusieurs commandes qui seront exécutées de manière séquentielle. Un script Shell s'écrit en langage Shell. Le gros avantage de ce langage est d'être totalement intégré à Linux. Il n'y a rien à installer, rien à compiler. »
https://edurl.fr/ZPghKq6O



Activité 13a

1. Créez un fichier **log.txt** dans le répertoire test

nom_utilisateur@nom_hôte:~ $ _____

2. Créez un fichier **essai.sh** dans votre répertoire personnel

nom_utilisateur@nom_hôte:~ $ _____

```
.
|-- essai.sh
`-- test
    |-- fichbob.txt
    `-- log.txt
```

3. Ouvrez essai.sh avec nano

nom_utilisateur@nom_hôte:~ $ _____

4. Saisissez le code de l'activité 12 dans essai.sh
Remarque : **#!** est appelé le **sha-bang** (ou **shebang**). Il précise que nous utilisons **bash**.

5. Donnez les droits d'exécution à l'utilisateur sur le fichier essai.sh en conservant les droits initiaux (méthode relative, voir « Annexe 1 »)

nom_utilisateur@nom_hôte:~ $ _____

6. Exécutez le script ***essai.sh*** avec la commande : **./essai.sh**

7. Visualisez le contenu de ***log.txt*** sans utiliser nano.
Il doit ressembler à l'encadré ci-contre.

nom_utilisateur@nom_hôte:~ $ _____

```
Effet des commandes pwd, ls, tree, etc.
1. pwd
/home/bob
2. ls
essai.sh
test
3. tree
.
|-- essai.sh
`-- test
    |-- fichbob.txt
    `-- log.txt

2 directories, 3 files
4. cd test
5. ls -l
total 8
-rw-rw-r-- 1 bob sin  79 26 oct.  12:02 fichbob.txt
-rw-r--r-- 1 bob sin 190 26 oct.  17:05 log.txt
```

Résultat attendu

Activité 13b

On donne le code suivant :

```
#!/bin/bash
# Surveillance de la température CPU du Raspberry Pi

# Lire la température dans le fichier système
temperature=$(cat /sys/class/thermal/thermal_zone0/temp)

# Traiter – Conversion de la mesure en dixièmes de °C
dixiemes=$((temperature / 100))

# Afficher la température
echo "Temperature CPU : $((dixiemes / 10)).$((dixiemes % 10)) °C"
```

$() est utilisée pour exécuter des commandes.

Exemple

```
current_date=$(date)
echo "La date actuelle est : $current_date"
```

- date est exécuté dans **$(...)**, et sa sortie (la date actuelle) est assignée à la variable current_date.
- Ensuite, la commande echo affiche cette variable.

1. Créez un fichier ***tempcpu.sh*** avec **nano** dans votre répertoire personnel. Complétez-le avec le code ci-dessus.

2. Accordez le droit d'exécution et testez ***tempcpu.sh***

Exemple de résultat attendu : `Temperature CPU : 43.3 °C`

On souhaite afficher la température du CPU dans la console toutes les 10s.

3. Faites une copie de ***tempcpu.sh*** et nommez-la ***tempcpu2.sh***.

nom_utilisateur@nom_hôte:~ $ _____

4. Utilisez les extraits de code suivants pour modifier ***tempcpu2.sh***

- `$(date '+%Y-%m-%d %H:%M:%S')  # récupération de l'heure actuelle`

- ```
  while true;
  do                    # boucle infinie
  # Code exécuté dans la boucle
  done
  ```

- `sleep 10              # temporisation de 10s`

5. Testez ***tempcpu2.sh***

Exemple de résultat attendu :

```
2025-03-05 16:09:04 - Temperature CPU : 43.8 °C
2025-03-05 16:09:14 - Temperature CPU : 45.2 °C
2025-03-05 16:09:24 - Temperature CPU : 42.3 °C
…
```

Ctrl+C pour arrêter le script

Appel prof *Pour faire vérifier le fonctionnement du script* ☐ validé ☐ partiel ☐ non validé

6. *Quelle modification doit-on apporter au programme pour qu'il mémorise la température dans un fichier* ***logtempcpu.txt*** *?*

_____

DR6 - SYNTHÈSE

Activité 14

<link rel="stylesheet" href="_____">

<script src="_____"></script>

FICHIER=/var/www/html/sit_____/temp.json

Activité 15

*nom_utilisateur***@***nom_hôte:~ $* _____

Appel prof *Pour faire vérifier le fonctionnement de la page* ☐ validé ☐ partiel ☐ non validé
```