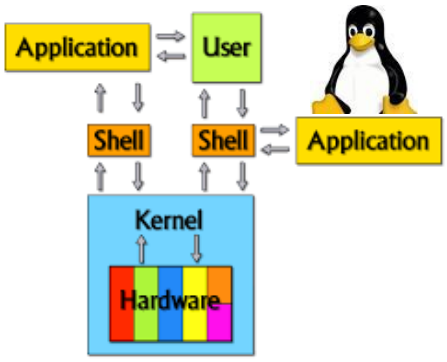
Système d'exploitation
Shell Unix - BASH

Tutoriel - Shell Unix - BASH



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.

SIGLES UTILISÉS

: Important	: Information	: Éditeur de code (voir prof)

RESSOURCES DOCUMENTAIRES, LOGICIELS ET MATÉRIELS

PC, Raspberry Pi, réseau Wi-Fi, terminal (SSH, SCP), éditeur de code, navigateur web, serveur NAS_SIN.

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

Table des matières

1. Mise en situation 3

2. Étude de la problématique 4

2.1 Connexion au Raspberry Pi 4

2.2 Découverte de Bash 4

2.3 Utilisateurs et groupes 4

2.4 Arborescence de fichiers et chemins 4

2.5 Permissions (droits) et propriétés des fichiers et des répertoires 6

2.5.1 Changement absolu des droits d'accès à un fichier 6

2.5.2 Changement absolu des droits d'accès à un répertoire 7

2.6 Redirections 7

2.7 Script Shell 8

3. Synthèse 8

3.1 Préparation 8

3.2 Transfert de la page web sur le Raspberry Pi 8

Annexes 10

- Le contenu des répertoires (extrait) 10
- Les commandes BASH (extrait) 10

Lexique 10

Table des illustrations

Figure 1 : Linux + Raspberry Pi 3
Figure 2 : Tux 3
Figure 3 : Position du Shell 3
Figure 4 : La ligne de commande 3
Figure 5 : Fichiers spéciaux 7
Figure 6 : log.txt 7
Figure 7 : Organisation du bureau pour le transfert des fichiers 8
Figure 8 : Résultats du transfert des fichiers 8
Figure 9 : Page web finale 9

1. Mise en situation

On souhaite **installer** un site web sur un nano-ordinateur monocarte Raspberry Pi connecté à un réseau local afin de superviser des grandeurs physiques. Le système d'exploitation installé sur ce Raspberry Pi, **Raspberry Pi OS Lite** (anciennement Raspbian Lite), **ne dispose pas d'un environnement de bureau !**

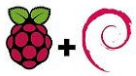
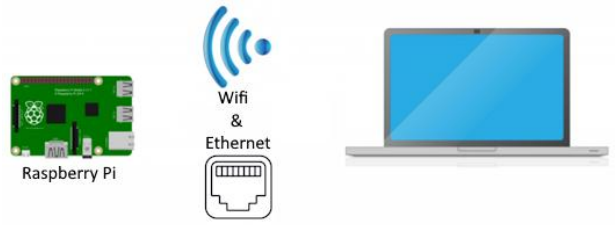


Figure 1 : Linux + Raspberry Pi



Raspberry Pi OS

Raspberry Pi OS (anciennement nommé **Raspbian**) est un système d'exploitation libre et gratuit, basé sur Debian, optimisé pour fonctionner sur les différents Raspberry Pi. Il s'agit d'une modification de Debian spécifiquement adaptée pour les systèmes sur une puce de type ARM dotée d'une unité de calcul en virgule flottante (FPU).

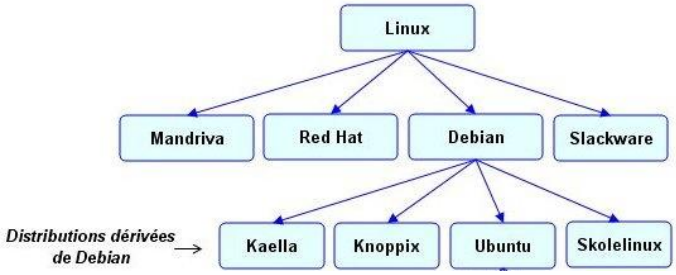


Figure 2 : Tux

POSIX

Tux est un manchot, **mascotte officielle du noyau Linux**. Dessiné par **Larry Ewing** en 1996, son usage est libre et se retrouve dans de très nombreux projets et logotypes liés à Linux. Wikipédia

Debian GNU / Linux est une distribution Linux, compatible avec la norme **POSIX**, composée de logiciels libres et open source.



POSIX est un ensemble de standards définissant aussi bien les fonctions des bibliothèques que doit offrir le système d'exploitation (lire, écrire dans un fichier, accès réseau, etc.) que les programmes de base permettant d'utiliser le système. La plupart des systèmes d'exploitation modernes (à l'exception de Windows) sont compatibles avec la norme POSIX.

Les actions nécessaires à l'installation du site se feront donc à l'aide du shell **Bash**, l'**interpréteur** en **ligne de commande,** disponible dans **Raspberry Pi OS Lite**.

Ligne de commande

*« Une **interface système** ou **coque logicielle** (**shell** en anglais) est une couche logicielle qui fournit l'**interface utilisateur** d'un système d'exploitation. Elle correspond à la couche la plus externe de ce dernier. L'interface système est utilisée comme diminutif de l'**interface utilisateur du système d'exploitation**. »* Wikipédia



Figure 3 : Position du Shell dans le S.E.

Le shell peut prendre deux formes :

- Une forme graphique, généralement un environnement de bureau (KDE).
- **Une forme textuelle, l'interpréteur en ligne de commande tel que Bash** (acronyme de ***Bourne-Again Shell***).

Bash est le **shell Unix** du projet **GNU**. Wikipédia

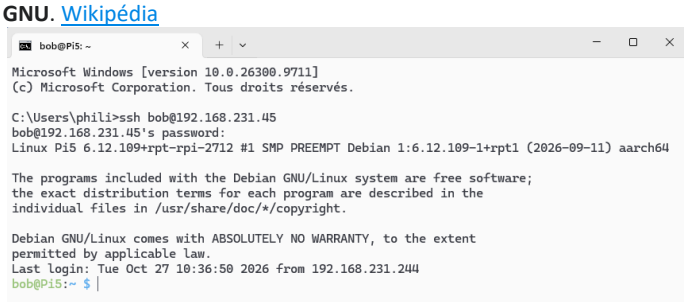


Figure 4 : La ligne de commande

2. Étude de la problématique

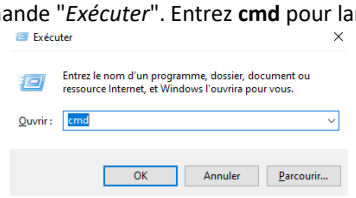
2.1 Connexion au Raspberry Pi

Le contrôle du Raspberry Pi à partir du PC se fera à l'aide de **SSH** en **ligne de commande**.

***Secure Shell** (**SSH**) est à la fois un **programme informatique** et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite, tous les segments TCP sont authentifiés et chiffrés. Il devient donc impossible d'utiliser un sniffer pour voir ce que fait l'utilisateur. Wikipédia*

Connectez-vous à votre compte sur le Raspberry Pi en suivant la procédure suivante :

1. **Touches Windows + R** pour ouvrir la commande "*Exécuter*". Entrez **cmd** pour lancer le **terminal** Windows.



2. Dans l'invite de commande de Windows, entrez :
   ssh *votre_identifiant_de_connexion***@***ip_du_Raspberry* puis votre*_mot_de_passe*

***SSH ouvre une invite de commande Linux sur le Raspberry Pi** (exemple : bob@Pi5:~ $).*
Les commandes que vous allez entrer ne s'exécuteront plus sur le PC, mais sur le Raspberry Pi !

2.2 Découverte de Bash

L'invite de commande Linux

- ***Organisation***

  *nom_utilisateur***@***nom_hôte:répertoire_de_travail prompt*

- ***Exemple***

  bob@Pi5:~ $
  | | | |__ prompt : **$** ⇔ normal (non-administrateur), **#** ⇔ super utilisateur (root)
  | | |____ répertoire de travail : **~** ⇔ répertoire personnel
  | |____ nom de l'hôte
  |____ nom de l'utilisateur

Activité 1 - Complétez la ligne de commande à laquelle vous êtes connecté sur le **DR1.**

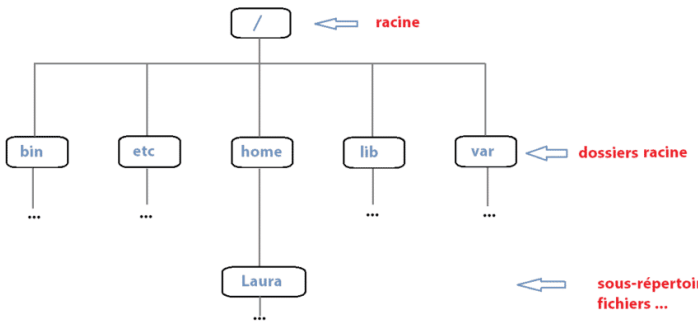
2.3 Utilisateurs et groupes

Les systèmes POSIX sont **multi-utilisateurs**. À l'identifiant utilisateur correspond un **identifiant numérique** unique nommé **UID** (**U**ser **ID**entifier). En interne le système d'exploitation utilise exclusivement l'**UID**.
Les utilisateurs peuvent être réunis en *groupes*. Un utilisateur appartient à un ***groupe principal*** et à des ***groupes secondaires***. L'identifiant numérique du groupe principal est nommé **GID** (**G**roup **ID**entifier).

Activité 2
Entrez la commande **id** dans l'invite de commande. **Identifiez** l'**UID**, le **GID** et les **groupes secondaires** du compte. **Répondez** sur le **DR1.**

2.4 Arborescence de fichiers et chemins

L'ordinateur stocke des fichiers sur son disque dur. Ces fichiers peuvent être **lus**, **modifiés**, **renommés**, **copiés** ou **supprimés**. *Dans les systèmes **POSIX**, les fichiers sont aussi organisés. Certains **fichiers spéciaux**, appelés **répertoires** (directory) ou **dossiers** ont pour seul but de contenir d'autres fichiers et répertoires.*
*Il existe un **répertoire racine** c'est-à-dire un répertoire contenant tous les autres. Il est identifié par « **/** ». L'ensemble des fichiers et répertoires forment une **arborescence**. Votre répertoire est au même niveau que celui de Laura dans l'arborescence ci-dessous.*



Pour Linux tout est fichier !

***Note** : Le mot "dossier" est apparu avec les interfaces graphiques types Macintosh et Windows, dans lesquelles les répertoires sont représentés par des dossiers.*

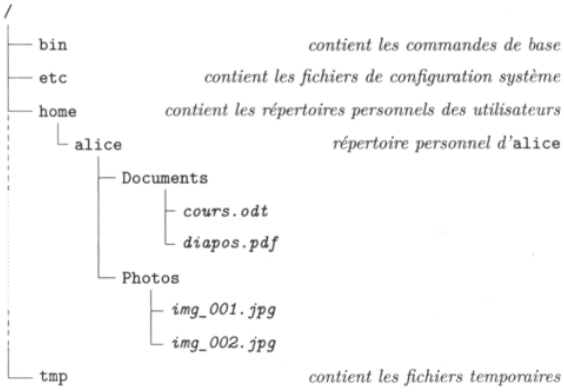
pwd

La commande **pwd** (**p**rint **w**orking **d**irectory) permet de connaître le nom du **répertoire courant** et renvoie le **chemin absolu** permettant d'y accéder.

Activité 3 - Entrez pwd dans l'invite de commande et relevez l'information renvoyée par l'interpréteur. **Complétez** le **DR2.**

Chemin absolu, chemin relatif

- *Un chemin est une suite de noms séparés par des « / ». Si le chemin **commence par « / »**, il est dit **absolu**. Il montre l'ensemble des répertoires à traverser, depuis la racine jusqu'au fichier ou répertoire.*



Exemple d'arborescence obtenue avec la commande **tree**

Exemple : dans l'arborescence ci-dessus, le chemin absolu permettant d'accéder au fichier img_001.jpg est :
/home/alice/Photos/img_001.jpg.

- *Un chemin **ne** commençant **pas** par / est un chemin **relatif**. Il situe un fichier ou un répertoire par rapport au répertoire courant.*

Exemples
- Si le répertoire courant est alice, la commande **cd** *Documents* permet de se déplacer de alice dans Documents
- Si le répertoire courant est Documents, la commande **cd** ***../****Photos* permet de se déplacer de *Documents* dans *alice* puis de *alice* dans *Photos*.

Remarque
« **..** » identifie le répertoire **parent.**
« **.** » identifie le répertoire **courant.**

mkdir

La commande **mkdir** (**m**a**k**e **dir**ectory) crée un répertoire.

Activité 4
Créez un répertoire avec le nom **« *test* »** dans votre répertoire personnel. **Complétez** le **DR2** avec la commande que vous avez entrée**.**

cd

La commande **cd** (**c**hange **d**irectory) est utilisée pour changer de répertoire.

Activité 5
À l'aide de la commande **cd, déplacez-vous** dans *« **test** »* puis dans « ***home*** » à partir de test en utilisant des **chemins relatifs**. Déplacez-vous dans « **test** » à partir de « home » en utilisant un chemin absolu. **Complétez** le **DR2** avec les commandes que vous avez entrées**.** ***Remarques** : si nécessaire, la commande **tree** vous aidera à écrire les chemins. **cd** seul renvoie vers le répertoire personnel ~.*

tree

La commande **tree** affiche l'arborescence du **répertoire courant**.

Activité 6 - Affichez l'arborescence du répertoire « ***home*** ». **Complétez** les blocs de l'arborescence sur le **DR2** avec la ou les commande(s) que vous avez entrée(s).

*Dans les systèmes **POSIX**, l'**extension** d'un nom de fichier, c'est-à-dire les caractères placés après le caractère « **.** », situé le plus à droite, n'ont **pas de signification particulière.** Par exemple, un fichier peut être exécutable sans que son nom se termine par .exe. L'utilisation de l'extension pour déterminer le type de fichier est une caractéristique du système Windows (hérité de MS-DOS).*

touch

La commande **touch** *nomfichier.extension* crée un fichier dans le répertoire courant.

Activité 7 - Créez un fichier ***fichnom_utilisateur.txt*** dans le répertoire « ***test »***. **Complétez** le **DR2** avec la commande que vous avez entrée.

nano
cat

La commande **nano** ouvre un **éditeur de texte**. **cat** affiche le contenu d'un ou plusieurs fichiers.

Activité 8
Ouvrez le fichier *fichnom_utilisateur.txt* avec la commande : **nano -l** *fichnom_utilisateur.txt* et entrez le texte :
"*Ce fichier est utilisé par <votreprénom votrenom> dans les tests du TP Bash*".
Quittez nano avec **Ctrl+X**, **O** et la touche **Entrée**. **Vérifiez** le contenu du fichier avec **cat**. **Complétez** le **DR2**.

2.5 Permissions (droits) et propriétés des fichiers et des répertoires

La commande **ls -l** (**l**i**s**t) affiche le contenu détaillé d'un répertoire.

\s

Activité 9 – **Affichez** le contenu détaillé du répertoire « ***test*** ». **Complétez** le **DR3** avec la commande que vous avez entrée et le contenu détaillé de ce répertoire.

*Les systèmes **POSIX** étant **multi-utilisateurs**, ils doivent permettre à chaque utilisateur de différencier ses fichiers de ceux des autres. À cette fin, le système d'exploitation associe à chaque fichier l'**UID** de son propriétaire et le **GID** de son groupe principal. Le système permet aussi de définir des **permissions** pour le **propriétaire**, le **groupe principal** et les **autres utilisateurs**.*

L'option **-l** de la commande **ls** que vous avez utilisée produit un affichage détaillé des fichiers. Vous avez obtenu un affichage proche de celui donné dans l'exemple ci-dessous.

Exemple : *Ligne de commande ->* bob@Pi5:~/test $ **ls -l**

Résultat (sortie de la commande) -> **total 4**

-rw-r--r-- 1 bob sin 58 26 oct. 10:45 fichbob.txt

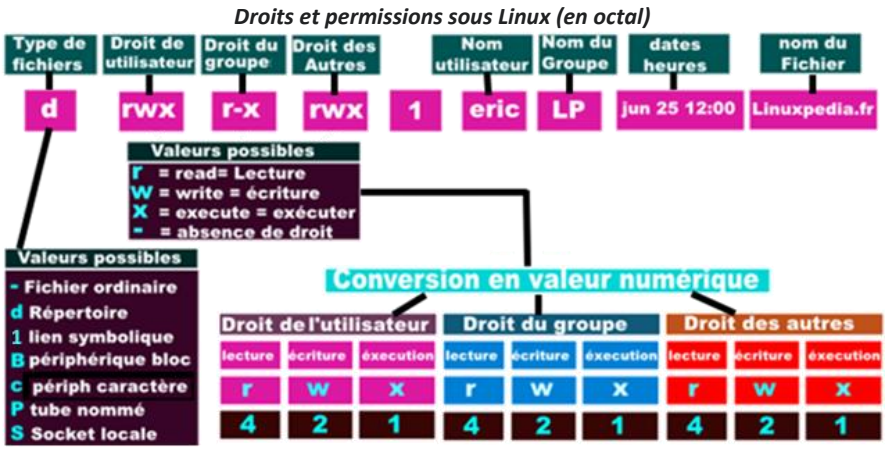
Dans cet exemple, la sortie de la commande indique :

- **-rw-r--r-- :** le **type de fichier** et les **droits accordés** (ils s'interprètent avec le tableau du paragraphe 2.5.1)
  On a ici : ***premier*** *tiret* « - » signifie que *fichbob.txt* est un fichier ordinaire, **rw-** précise que son propriétaire (bob) dispose des droits de lecture (**r**) et d'écriture (**w**), les deux **r--** précisent que le groupe principal (*sin*) et les *autres* utilisateurs ont seulement un droit de lecture.
- **1 : nombre de liens** vers le fichier, **bob :** nom du propriétaire, **sin** : nom du groupe principal auquel appartient bob
- **58 : taille du fichier en octets**
- **26 oct. 10:45 :** date et heure de la dernière modification
- **fichbob.txt** : nom du fichier
- **total 4** : la **taille occupée** par les fichiers dans le répertoire est égale à : 4 blocs de 1024 octets (n'a pas d'intérêt ici)

2.5.1 Changement <u>absolu</u> des droits d'accès à un fichier

On spécifie de nouveaux droits, **indépendamment des droits existants** à l'aide de la commande **chmod**.

Changement <u>absolu</u> des droits d'accès à un fichier



chmod *valeur* nomfichier **modifie les permissions** sur le fichier *nomfichier*. *valeur* est calculée en utilisant le tableau ci-dessus.

chmod

<u>Exemple</u> : *valeur* correspondant aux droits sur le fichier *fichnom_utilisateur.txt vu précédemment*

Droits de l'**utilisateur** : **r w -** ⇒ **4 + 2** + 0 = **6**

Droits du **groupe** principal : **r- -** ⇒ **4** + 0 + 0 = **4** => *valeur* = **644** (droits donnés par le système d'exploitation lors de la création du fichier)

Droits des **autres** : **r - -** ⇒ **4** + 0 + 0 = **4**

Pour que ce fichier devienne exécutable uniquement pour l'utilisateur (**rwx)**, on écrit : **chmod 744** *fichnom_utilisateur.txt*

Préalable à l'activité 10 : par défaut, votre **répertoire personnel** est inaccessible au groupe et aux autres. Pour le rendre accessible en lecture au groupe principal et aux autres, entrez la commande suivante :

chmod 755 /home/repertoire_personnel (Explication dans le §2.5.2)

Activité 10

1. À l'aide du tableau ci-dessus, **identifiez** le **type** de votre **<u>fichier</u>** et les **droits d'accès**. **Complétez** le **DR3.**

2. Modifiez ces droits pour que **votre fichier** soit **accessible en lecture et en écriture au groupe principal**.

<u>ATTENTION</u> : le propriétaire et les autres doivent conserver leurs droits sur le fichier.

Vérifiez avec ls -l. **Notez** le résultat sur le **DR3.**

3. Demandez à votre voisin d'écrire dans votre fichier sur son PC**.** A-t-il réussi ? Si non, pourquoi ? **Répondez sur le DR3.**

La commande chmod 755 repertoire_personnel a attribué les droits de lecture et d'exécution au groupe principal et aux autres sur votre répertoire. **Nous allons détailler leur effet.**

2.5.2 Changement absolu des droits d'accès à un répertoire



Attention : les droits r, w et x n'ont pas la même signification selon qu'ils s'appliquent à un fichier ou à un répertoire.

Droit	Nom	Sur un fichier	Sur un RÉPERTOIRE
r	Lecture	Lire le contenu du fichier (cat, less…)	Lister le contenu du répertoire (ls). Sans r, ls renvoie « Permission denied » et tree « error opening dir ».
w	Écriture	Modifier le contenu du fichier (nano, echo >…)	Créer, supprimer ou renommer des fichiers DANS ce répertoire (touch, rm, mv). Ne concerne pas le contenu des fichiers eux-mêmes.
x	Exécution	Lancer le fichier comme programme (./script.sh)	Traverser / entrer dans le répertoire (cd). INDISPENSABLE pour accéder aux fichiers qu'il contient, même avec r.

Règle pratique pour les répertoires — r et x vont généralement ensemble : r sans x permet de lister les noms de fichiers mais pas d'y accéder ; x sans r permet d'entrer si l'on connaît le chemin exact, mais pas de lister le contenu.

chmod

Activité 11
1. Modifiez les droits sur **votre répertoire** pour en interdire l'accès au groupe et aux autres
ATTENTION : la commande doit commencer par : chmod 7-- /
Vérifiez le résultat de la commande et **complétez le DR3**



Les droits des répertoires sont modifiés à l'aide du **tableau du 2.5.1** en tenant compte de la signification accordée à r, w et x au **paragraphe 2.5.2**.

2.6 Redirections

Les systèmes **POSIX** proposent à chaque programme trois fichiers spéciaux permettant de faire des entrées-sorties. L'***entrée standard*** ou **stdin** (**st**andar**d in**put) est un fichier « virtuel » dans lequel un programme peut lire, mais pas écrire. Pour les programmes lancés dans un terminal, l'entrée standard est reliée au **clavier**. La ***sortie standard*** ou **stdout** (**st**andar**d ou**tput) correspond à un fichier relié à l'**affichage** dans le terminal. Le troisième fichier est la ***sortie d'erreur*** ou **stderr** (**st**andar**d err**or).

Les opérateurs >, >>, 2> et |

Redirections

- L'opérateur **> redirige** la sortie standard d'un programme **vers un fichier**
  [ATTENTION : si le fichier existe, il est écrasé.]
  Exemple : ls -l > listerep.txt

  L'opérateur **>> redirige** à la fin du fichier pour ne pas écraser son contenu
  Exemple : ls -l >> listerep.txt

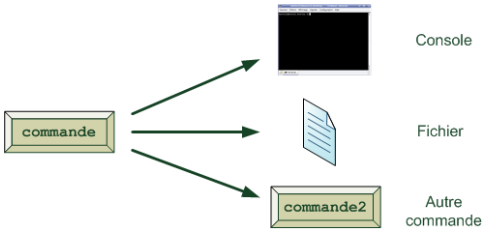


Figure 5 : Fichiers spéciaux

- L'opérateur **2>** redirige les messages d'erreur
  Exemple : ls -z **2>** *erreur.txt (z n'existant pas, erreur.txt contiendra : « ls : option invalide – 'z' »)*

- L'opérateur **| (pipe) relie** la sortie d'une commande à l'entrée de la suivante.
  Exemple : lscpu **|** *grep "MHz" # Affiche les fréquences du CPU*

Sauvegarde des commandes vues précédemment dans un fichier
À partir du répertoire personnel, on souhaite enregistrer dans un fichier ***log.txt*** *(à créer dans le répertoire test),* **le résultat des commandes** : ***pwd, ls, tree, cd, ls -l*** entrées dans cet ordre au clavier. Chaque résultat doit être précédé d'un numéro d'ordre et du nom de la commande. Ce numéro et le nom de la commande seront entrés avec la commande **echo**.

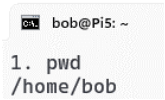


Figure 6 : log.txt

echo

Exemple : *echo 1. pwd* **>>** *test/log.txt* suivi de
pwd >> *test/log.txt* produisent les deux premières lignes de la figure 6 ci-contre.

Activité 12 – Complétez le **DR4 avec la liste** des commandes nécessaires à la création du résultat attendu**.** Ce texte sera réutilisé dans l'activité 13a.
Remarque : choisissez l'opérateur de redirection à utiliser parmi ceux présentés ci-dessus (**>**, **>>**, **2>** ou **|**).

2.7 Script Shell

« 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. »

Premier script

Vous allez automatiser la sauvegarde des commandes ***pwd, ls, tree, cd, ls -l*** *dans log.txt à l'aide d'un script. En cas d'erreur dans log.txt, il suffira de corriger le script et de le relancer.*

Activité 13a - Renseignez le document réponse 5 (**DR5**).

Deuxième script

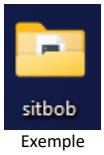
Vous allez mesurer et afficher la température CPU du Raspberry Pi.

Activité 13b - Renseignez le document réponse 5 (**DR5**).

3. Synthèse

Objectif : superviser la température d'un processeur.

3.1 Préparation



Exemple

Téléchargez le répertoire « **Domotique** » situé sur le serveur **NAS_SIN** dans *Répertoire_du_Cours/3_Systèmes d'exploitation/,* placez-le sur le bureau du PC et **renommez-le** en **sitx** (le **x** de **sitx** est à remplacer par **les trois premières lettres de votre login)**.

Activité 14 - **Ouvrez** le répertoire sitx dans l'éditeur de code.
1. Complétez les balises <link>, <footer> et <script> dans ***index.html*** et supprimez la ligne d'avertissement.
2. Dans la ligne « FICHIER= » du fichier ***tempweb.sh*****, adaptez** sitx au nom de votre répertoire.
Complétez le DR6.

3.2 Transfert de la page web sur le Raspberry Pi

Transfert de répertoires avec SSH

*Une des nombreuses utilisations possibles de SSH est **SCP**, pour **S**ecure **C**oPy. Il s'agit d'un protocole permettant de transférer des fichiers d'un ordinateur à un autre en s'appuyant sur le protocole **SSH** qui **chiffrera les données transférées**.*

1. Ouvrez une autre invite de commande. Organisez votre écran comme sur la copie ci-dessous. À **droite**, la **ligne de commande Linux** servira à vérifier les transferts sur le Raspberry Pi, à **gauche** la **ligne de commande Windows** sera utilisée pour effectuer le transfert.

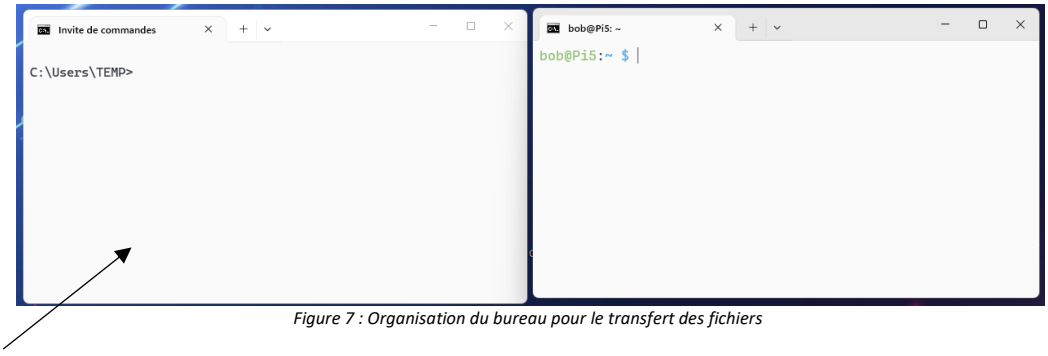


Figure 7 : Organisation du bureau pour le transfert des fichiers

2. Entrez une ligne de commande en suivant l'exemple ci-dessous pour transférer le répertoire sitx sur le Raspberry Pi.
C:\Users\eleve>scp **-r** *Chemin_vers_le_répertoire_à_télécharger\sitx* votrelogin@192.168.x.x:/var/www/html/

3. Faites de sitx votre répertoire courant et affichez son arborescence. *Exemple de résultat attendu*

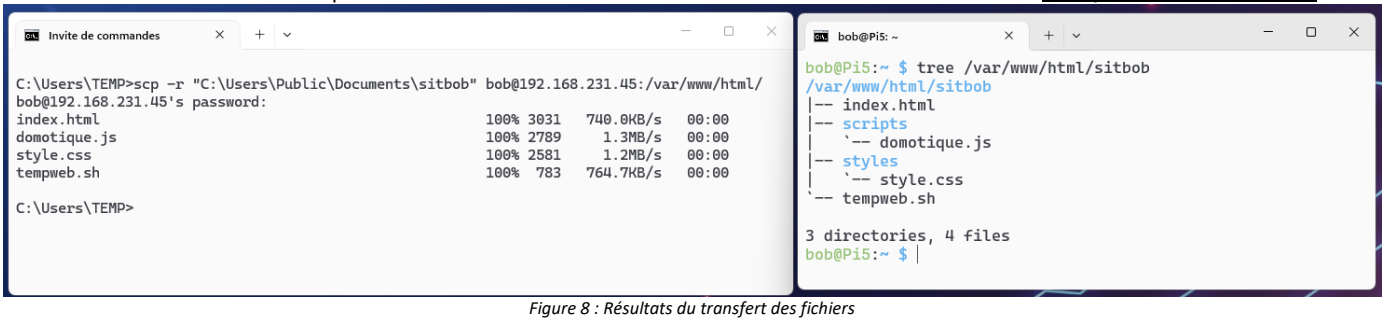


Figure 8 : Résultats du transfert des fichiers

4. Accordez les droits de lecture et d'exécution au répertoire sitx à l'aide de la commande : **chmod -R 755 /var/www/html/sitx**.

5. Affichez la page ci-dessous en entrant l'url : **192.168.x.x/sitx** dans un navigateur.



Activité 15

1. **Déplacez** *tempweb.sh* de sitx vers **votre répertoire personnel** sur le Raspberry Pi. Complétez le **DR6**.
2. **Vérifiez la présence** du droit d'exécution sur *tempweb.sh.*
3. **Testez** le script avec la commande **: ./tempweb.sh. Ctrl+C** pour arrêter.

Appel prof *Pour faire vérifier le fonctionnement de la page*

Exemple de résultat attendu

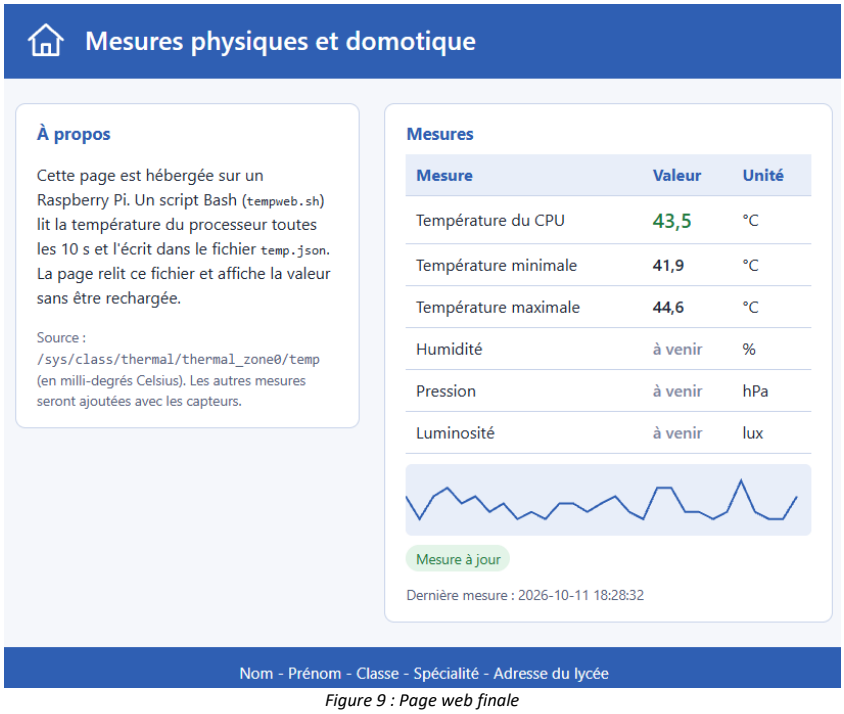


Figure 9 : Page web finale

Pour aller plus loin…

La commande **nohup ./tempweb.sh > /dev/null 2>&1 &** crée un processus en tâche de fond. Il s'affiche sous la forme suivante **[numéro de tâche] PID**.

Exemple

```
bob@Pi5:~ $ nohup ./tempweb.sh > /dev/null 2>&1 &
[1] 479735
```

Pour arrêter ce processus, entrez : **kill PID** ou **pkill -f** *tempweb.sh* dans la console.

Annexes

- **Le contenu des répertoires (extrait)**

Répertoire	Contenu / Description
/	**Le répertoire racine.**
/boot	noyau et fichiers de démarrage de **Linux.**
/bin	exécutables et binaires essentiels (bash, ls, cp, etc.).
/dev	fichiers de périphériques. Linux traite chaque périphérique comme un fichier spécifique.
/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)**

Source : https://edurl.fr/pivZhiW3

Répertoires et fichiers

Nom	Description de la commande	Signification	Exemples d'utilisation
***cat** fichier*	*Affiche le contenu d'un fichier.*	***c**onc**at**enate*	***cat** fichier.txt*
***cd** chemin/nomrépertoire*	*Changement de répertoire.*	***c**hange **d**irectory*	***cd** /var/www* ***cd** .. → répertoire parent* ***cd** ou **cd ~** → compte utilisateur*
***chmod** valeur fichier (rep)*	*Change les droits sur un fichier ou un répertoire.*	***ch**ange **mode***	***chmod** 770 fichier.txt*
***cp** source… destination* ***cp -r** source/ destination/*	*Copie de fichiers ou de répertoire.*	***c**o**p**y*	***cp** fichier.txt chemin/*
***ls** répertoire*	*Affiche le contenu d'un répertoire, le répertoire courant si aucun argument.*	***l**i**s**t*	***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.*	***m**a**k**e **dir**ectory*	***mkdir** site*
***mv** source… destination*	*Déplace ou renomme des fichiers.*	***m**o**v**e*	***mv** fichier1.txt … fichiern.txt chemin/* ***mv** fichier.txt copie.txt*
nano	*Éditeur de texte.*		***nano** fichier.txt*
pwd	*Affiche le nom du répertoire courant.*	***p**rint **w**orking **d**irectory*	
***rm** fichier…* ***rm -r** répertoire*	*Supprime définitivement des fichiers ou des répertoires. -r : récursif*	***rm**ove*	***rm** fichier.txt* ***rm** -r nomrépertoire*
***touch** fichier*	*Crée un fichier.*		***touch** fichier.txt*
tree	*Affiche l'arborescence d'un répertoire.*		

Divers

Nom	Description de la commande	Exemples d'utilisation
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*

Lexique

BASH : "**B**ourne **A**gain **SH**ell". Type de shell utilisé dans Debian (la distribution Linux sur laquelle est basé Raspberry Pi OS).
COMMAND LINE : ligne spécifique sur laquelle on travaille dans l'environnement textuel.
Console : terminal physique avec un clavier. Les consoles étaient des terminaux connectés à un ordinateur central.
Console virtuelle : version virtuelle d'une console physique. Dans Linux, on accède à plusieurs consoles virtuelles en utilisant CTRL + ALT + les touches de fonction.
SHELL : interpréteur de ligne de commande. Entoure le noyau du système d'exploitation. On accède au noyau en passant par le Shell (coquille). Le Shell interprète les commandes textuelles et les transforme en code compréhensible par le noyau.
Terminal : programme utilisé pour accéder à l'invite de commande à partir du bureau.