# Laboratoires système

# Laboratoire 5 :

# Introduction aux gestions du traitement de texte et des fichiers

Une image contenant texte, logo, Police, Marque

Description générée automatiquement

## Avant-propos

Dans ce laboratoire, nous allons aborder deux sujets différents mais néanmoins importants :

* Le traitement de texte.
* La gestion des fichiers.

Ils seront traités de façon introductive.

## Objectifs

* Créer les fichiers textes en CLI.
* Ecrire dans ces fichiers.
* Rediriger un contenu vers un fichier.
* Utilisation de VIM et Nano
* Gestion des fichiers et des répertoires.

## Introduction

L’interface en ligne de commande (CLI - Command Line Interface) reste un outil fondamental pour les professionnels de l’informatique. Elle offre une puissance et une flexibilité inégalées, permettant des interactions précises avec le système d'exploitation (OS) et les logiciels. Même sous Windows, des outils comme PowerShell et le Windows Subsystem for Linux (WSL) ont réaffirmé l'importance de la CLI.

### Spécificités des Systèmes

Tout système d'exploitation a ses particularités. Dans le cadre de Linux et Windows, ces différences sont notables, notamment dans la gestion des fichiers et des permissions. Une compréhension approfondie de ces spécificités permet une utilisation efficace et sécurisée du système. Une des différences notables quand on compare les deux systèmes, c’est la notation des répertoires et périphériques.

### Sous Windows :



Nous avons 3 choses remarquables :

* on utilise le « \ » pour séparer les dossiers et sous-dossiers
* on utilise une lettre (ici « C : » pour le disque dur) pour indiquer le périphérique actif
* on a pas de distinction visuelle dans le prompt pour différentier user/administrateur

### Sous Linux :



Nous avons 5 choses à signaler :

* On utilise le « / » pour séparer les dossiers et sous-dossiers



* Le périphérique actif dans la ligne de commande n’est pas différentié du reste
* Il y a une colorisation du prompt du user
* il y a une distinction entre user/administrateur



* Il y a un raccourci (« ~ ») pour le répertoire « home » du user (idem pour le root). C’est le même raccourci pour tous, mais ils pointent tous

sur un dossier différent.

### Organisation des Fichiers et Répertoires

L'organisation des fichiers en répertoires est cruciale. Elle permet non seulement une gestion ordonnée des données, mais aussi une hiérarchisation des droits. Cette hiérarchisation est fondamentale pour contrôler l'accès aux fichiers et sous-dossiers, garantissant ainsi la sécurité des données.

### Distributions Debian : Une Solution Distinguée

Debian, une des distributions Linux les plus populaires, adopte une structure de fichiers et de répertoires bien organisée, conformément à la norme Filesystem Hierarchy Standard (FHS). Cette structuration permet une gestion simplifiée des packages et des dépendances.

### Fichiers et Dossiers Système

Les fichiers et dossiers système sont essentiels au fonctionnement de l'OS. Ils contiennent des configurations, des scripts de démarrage, et des binaires cruciaux. Leur modification ou suppression incorrecte peut entraîner des dysfonctionnements du système.

### Buts et Fonctions

* Binaires: Exécutables nécessaires au fonctionnement du système.
* Configuration: Fichiers de configuration pour personnaliser le système.
* Logs: Traces des événements du système pour le débogage.

### Architecture de Dossiers sous Debian

Debian suit la FHS avec des dossiers clés comme /bin, /etc, /usr, et /var. Par exemple :

* /bin : Binaires essentiels.
* /etc : Fichiers de configuration.
* /home : Répertoires personnels des utilisateurs.
* /lib : Bibliothèques essentielles.
* /opt : Logiciels additionnels.
* /tmp : Fichiers temporaires.
* /usr : Binaires, bibliothèques, documentation, etc., non essentiels.
* /var : Fichiers variables comme les logs.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquementUne représentation “graphique” des dossiers donnera ceci :

Fonctionnalités Spécifiques des Dossiers sous Debian Chaque dossier sous Debian a un but précis, facilitant la localisation des fichiers et la réparation des problèmes.

### La Notion de Lien

Un lien est une référence à un fichier ou un répertoire. Sous Linux, il existe deux types de liens : les liens symboliques (ou soft links) et les liens durs (ou hard links).

## Chemin Absolu vs Chemin Relatif

Cette notion est valable sous Windows, comme sous Linux. Une différence existe cependant entre les deux systèmes au niveau es caractères utilisés.

* Une image contenant texte, Police, capture d’écran, conception

  Description générée automatiquementChemin Absolu : Indique la localisation précise d'un fichier ou répertoire depuis la racine (/).
* Chemin Relatif : Indique la localisation d'un fichier ou répertoire par rapport au répertoire courant.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

## Répertoires . et ..

Cette notion aussi, est commune aux systèmes Windows et Linux.

* . : Référence au répertoire courant.
* .. : Référence au répertoire parent.

Une image contenant texte, Police, capture d’écran, nombre

Description générée automatiquementCes notions fondamentales de la CLI et de l'organisation des fichiers sous Debian fournissent la base nécessaire pour explorer et interagir efficacement avec le système d’exploitation Linux. Le laboratoire suivant vous permettra de mettre en pratique ces concepts essentiels.

## Gestion des fichiers

Nous en arrivons à l’un des objectifs de ce laboratoire : la gestion des fichiers.

Il est très important de pouvoir gérer les fichiers et dossiers en ligne de commande, car cela permet d’inscrire des commandes dans un fichier de script qui permet d’automatiser certaines opérations. De plus, les commande dans un terminal permettent une multitude de variations que l’interface graphique ne permet pas.

### Gestion des répertoires

Les fichiers seront mis dans un répertoire. Le premier répertoire du système, c’est la racine (/). C’est une très mauvaise pratique que d’utiliser la racine pour stocker des fichiers (ou des dossiers).

La bonne pratique, sous Linux comme sous Windows, consiste à organiser des répertoires de façon réfléchie.

Une des premiers choses à faire pour le faire quelque part, c’est d’y aller ! Nous allons donc naviguer dans les dossiers.

Pour changer de répertoire, il y a la commande « cd » (pour change directory).

|  |  |
| --- | --- |
| **Utilisation** | **Action** |
| cd | déplacement vers le répertoire « home » de l’utilisateur courant |
| cd ~ | idem |
| cd / | déplacement vers la racine du système de fichiers |
| cd - | déplacement vers le dossier précédent (pas le parent) |
| cd xxxx | déplacement vers le dossier xxxx qui est un sous-répertoire du répertoire courant (erreur s’il n’existe pas). « xxxx » est un chemin relatif. |
| cd /xxxx | déplacement vers le dossier xxxx qui est à la racine du système de fichier (erreur s’il n’existe pas). « /xxxx » est un chemin absolu. |
| cd ../xxxx | déplacement vers le dossier xxxx qui est au même niveau que le répertoire courant |

Une fois que l’on sait où on est, on a souvent quelque-chose à y faire. Nous allons maintenant créer des dossiers, les supprimer et les déplacer.

|  |  |
| --- | --- |
| **Utilisation** | **action** |
| mkdir | Erreur (il faut donner le nom du dossier à créer) |
| mkdir rep1 | Création du dossier « rep1 » dans le répertoire courant |
| mkdir rep1 rep2 | Création des dossiers « rep1 » et « rep2 » dans le répertoire courant |
| mkdir rep3/rep4 | erreur si le répertoire « rep3 » n’existe pas |
| mkdir rep3/rep4 -p | là, ça passe |
| mkdir /rep1 | création du dossier « rep1 » à la racine du système de fichier … si vous êtes en « root » : question : pourquoi ? |

Où suis-je ? C’est une question que l’on se pose parfois quand on navigue dans la ligne de commande. Outre le fait que, souvent, le répertoire courant est affiché dans le prompt de la ligne de commande, on a parfois besoin de le savoir dans un script (fichier qui rassemble une liste de commandes).

*pwd*

Une action souvent utilisée aussi, c’est lister le contenu d’un fichier.

### Utilisation

*ls*



*ls –l*

Une image contenant texte, Police, capture d’écran, nombre

Description générée automatiquementCette option permet d’afficher plus de détails sur les fichiers. On voit ici, le propriétaire du fichier, le groupe auquel est attaché le fichier, sa date de création et les droits appliqués à ces fichiers.

*ls -la*

Une image contenant texte, capture d’écran, Police, menu

Description générée automatiquement

Cette option permet de lister les fichiers cachés. Pour cacher un fichier ou un répertoire, sous Linux, on fait commencer son nom avec un “.”. Ces fichiers n’apparaissent pas dans les commandes sans l’option qui va bien. De plus, le “.” fait partie du nom du fichier. Il faut donc le mettre quand on veut réaliser une commande sur ce fichier.

Exemple : cat .bashrc

D’autres options très intéressantes sont encore disponibles avec la commande ls. Utilisez l’aide de la commande pour vous familiariser avec les principales.

### Gestion des fichiers

Les fichiers sont d’autres objets accessibles dans un système de fichier. Nous venons de voir les répertoires, voici les fichiers. Sur ces objets, il est possible d'exécuter des commandes. Certaines peuvent s’exécuter sur les deux. (copy, move …)

Copier un fichier, on peut le faire avec des options diverses.

Version avec fichier unique

|  |  |
| --- | --- |
| Utilisation | action |
| cp fichier1 fichier2 | copie le fichier1 dans un nouveau fichier de nom fichier2 |
| cp fichier1 rep1 | copie le fichier fichier1 dans le répertoire rep1 (s’il existe sinon …) |
| cp fichier1 rep1/. | copie le fichier fichier1 dans le répertoire rep1 (s’il existe sinon …) |
| cp fichier1 /home/<user>/rep1 | copie le fichier fichier1 dans le répertoire rep1 (s’il existe sinon …) |
| cp fichier1 /home/<user>/rep1/. | copie le fichier fichier1 dans le répertoire rep1 (s’il existe sinon …) |

Nous avons ici des solutions avec chemin absolus et relatifs.

Versions multiples

|  |  |
| --- | --- |
| cp fichier1 fichier2 fichier3 rep1 | copie les fichiers fichier1, fichier2 et fichier3 dans le répertoire rep1 |
| cp -t rep1 fichier1 fichier2 fichier3 | idem |
| mv fichier1 fichier2 | déplace le fichier1 vers le fichier2 en écrasant un éventuel fichier2 existant et le renomme |
| mv fichier1 rep 1 | Déplace le fichier1 vers le répertoire rep1 |
| rm -r rep1 | efface le dossier rep1 |
| rm fichier1 fichier2 | efface les fichiers fichier1 et fichier2 |

Beaucoup d’autres options sont possibles. Explorez-les pour vous faire une idée.

## Les jokers

Pour les commandes que nous venons de voir, ainsi que pour les autres, nous pouvons utiliser des Jokers qui permettent de traiter plusieurs fichiers sans forcément les nommer tous.

Les premiers sont « *\** » et « *?* ».

« *?* » remplace un caractère

« *\** » remplace 1 ou plusieurs caractères

« *{}* » énumère des caractères Exemples :

Si on a la situation suivante :

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

Si on veut déplacer tous les fichiers contenant un « 4 » comme quatrième caractère :



Ce qui donnera :



Si nous avions fait :



Ceci aurait donné :

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

« \* » remplace 0,1, ou plusieurs caractères. Donc, avec « \*4\* », le « 4 » est pris n’importe où dans le nom du fichier.

Nous avons maintenant la situation suivante :

Une image contenant texte, capture d’écran, affichage, Police

Description générée automatiquement

Pour replacer les fichiers du dossier « lev2 » au dossier « test », on peut utiliser la commande :

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

« ls -R » est là pour montrer le résultat.

Si, en une seule commande, je veux créer les fichiers « fica », « ficb »,

« ficc », « ficd », « fice », « ficf », « fic00 », « fic01 », « fic02 »,

« fic03 », « fic04 », « fic05 », « fic06 », « fic07 », « fic08 », « fic09 »,

Une image contenant texte, capture d’écran, Police

Description générée automatiquement« fic10 », « fic11 », « fic12 », « fic13 », « fic14 », « fic15 », je peux le faire avec la commande :

Notez qu’il existe encore beaucoup d’options dans ce genre (caractères numérique, ponctuation, …).

## Utilisation du résultat d’une commande

Certaines commandes communiquent des informations que l’on peut réutiliser dans d’autres commandes. On peut le faire en passant par des variables, ou en intégrant une instruction dans une autre.

Une image contenant texte, logiciel, Logiciel multimédia, capture d’écran

Description générée automatiquementUne commande souvent utilisée est la commande « *echo* » qui permet d’afficher une information.

… et bienvenue, au passage, dans le monde merveilleux des expressions régulières.

Le caractère « \ » est un caractère dit d’échappement, c’est-à-dire, que l’on va donner une fonction au caractère.

Allez voir sur les ressources internet tous ceux que vous pouvez trouver.

Dans cette commande, vous pouvez utiliser le résultat d’une autre commande. Exemple :

Si, dans nano, on édite le fichier liste.txt comme suit :Une image contenant texte, capture d’écran, affichage, Police

Description générée automatiquement

… et qu’on lance la commande :

Une image contenant texte, Police, capture d’écran, nombre

Description générée automatiquement

… on obtient le résultat donné par la commande ls.

## Exercices

Cet exercice sera évalué à partir d’un fichier qui rassemble les commandes que l’on vous demande d’exécuter. Chaque commande sera sur une ligne séparée. Il n’y a pas besoin de mettre de texte en commentaire dans le fichier.

Ce fichier s’appelle « *monPremierScript.sh* ».

Il vous est demandé, en utilisant l’éditeur « nano » du terminal Linux, de recopier dans l’ordre les commandes qui vont donner les actions suivantes :

(N.B. toutes les actions qui vous sont demandées sont à exécuter en une seule commande)

1. (/1) Créer, en une commande un ensemble de répertoires de « fic00 » à

« fic99 » dans lesquels il y a les dossiers « subfolder00 » à

« subfolder99 »

1. (/1) Effacer tous les dossiers « subfolder55 »
2. (/1) Dans le dossier « fic55/subfolder56 », créer les fichiers

« fichiera » à « fichierz »

1. (/1) Déplacer le dossier « fic11 » dans le dossier « fic12 »
2. (/3) Afficher un message affichant la date courant et l’heure courante au format suivant :

Une image contenant texte, Police, capture d’écran, nombre

Description générée automatiquement(au format heure locale)

1. (/3) Dans chaque dossier « ficxx » contenant un « 0 », créez un fichier « nv\_date » où « date » est la date au format <année sur 2 digits>\_<mois sur 2 digits>\_<jour sur 2 digits>

## Traitement de texte

Qu’est-ce qu’un traitement de texte ? Un logiciel conçu pour créer, éditer et manipuler des documents textes. Par exemple, LibreOffice, LaTeX,… Mais il existe des commandes en CLI (ligne de commande) qui font la même chose.

Les commandes les plus populaires sont « nano » et « vim ». Il existe aussi Emacs, …

Nano est un éditeur de texte utilisé exclusivement sous les OS Linux. Comme les logiciels cités ci-dessus, il permet d’éditer, de créer et de manipuler des fichiers. C’est un éditeur simple à prendre en main.

Vim est un éditeur de texte beaucoup plus puissant que nano. Il est principalement utilisé sous les OS Linux mais il est possible de l’installer sur d’autres OS. Son nom complet est Vi Improved signifiant une amélioration de Vi.

Mode d’utilisation de Nano

Pour utiliser la commande, il n’y a pas de condition. Cela veut dire que l’on peut créer un fichier texte en exécutant la commande ou soit utiliser un fichier déjà créé avec la commande touch.

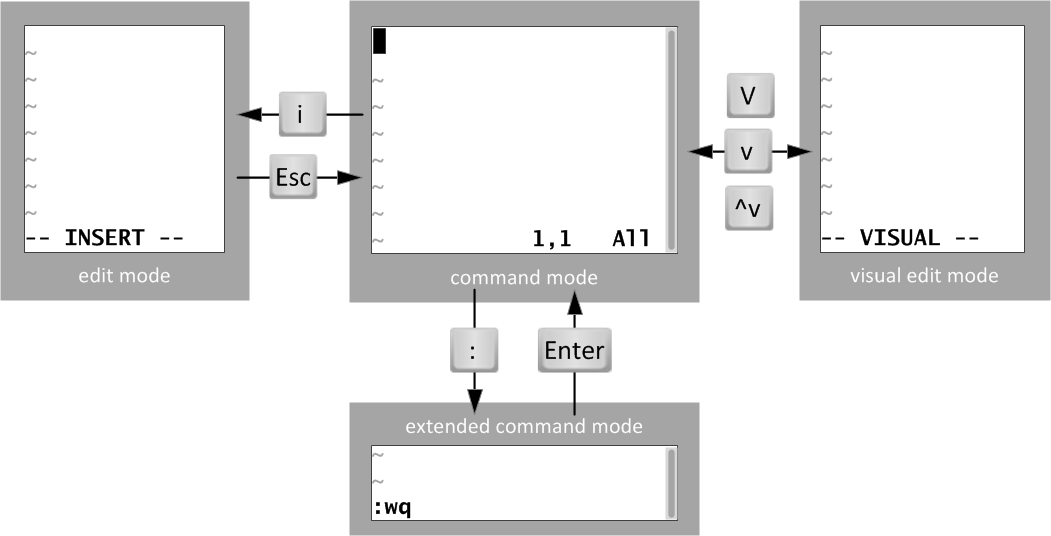
* Pour ouvrir/créer un fichier : il suffit de taper la commande suivi le nom du fichier que l’on veut éditer. Par exemple, pour éditer un fichier nommé test.txt, on fera :



* Interface utilisateur : une fois le fichier ouvert, l’interface nano s’ouvre. Il y a le nom du fichier en haut et son chemin. En bas, il y a un menu affichant les commandes courantes.

Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

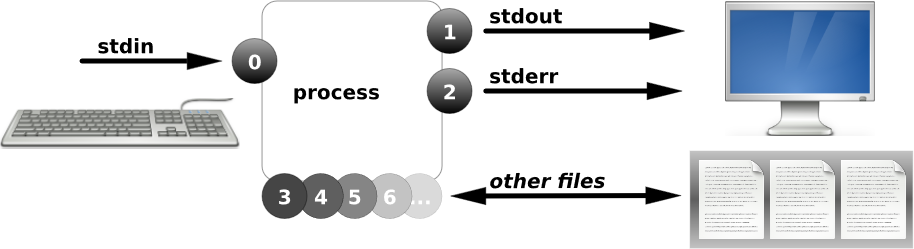
* Édition de texte : Vous pouvez commencer à éditer le texte en utilisant les touches de votre clavier. Pour ajouter ou modifier du texte, utilisez simplement votre clavier pour saisir ou supprimer du contenu.
* Sauvegarde du fichier : Pour sauvegarder vos modifications, appuyez sur "Ctrl + O". Vous serez invité à confirmer le nom du fichier. Appuyez sur "Entrée" pour enregistrer.
* Fermeture du fichier : Pour quitter nano tout en conservant les modifications, appuyez sur "Ctrl + X". Si vous avez apporté des modifications non enregistrées, nano vous demandera de les enregistrer avant de quitter.

### Mode d’utilisation de Vim

Lorsqu'il s'ouvre pour la première fois, Vim démarre en mode commande, utilisé pour la navigation, les couper-coller et autres manipulations de texte. Les autres modes et des fonctionnalités d’édition spécifiques sont accessibles à l’aide d’une seule frappe sur le clavier :

* Une pression sur i ouvre le mode d'insertion, dans lequel tout le texte saisi s'ajoute au contenu du fichier. Appuyez sur Échap pour retourner au mode de commande.
* Une pression sur v ouvre le mode visuel, dans lequel il est possible de sélectionner des caractères multiples pour manipuler le texte. Utilisez V pour sélectionner plusieurs lignes, et Ctrl+v pour sélectionner un bloc. Le même raccourci est utilisé pour passer en mode visuel (v, V ou Ctrl+v) et pour en sortir.
* La touche : ouvre le mode de commande étendu pour effectuer des tâches telles que l'écriture et l'enregistrement d'un fichier ou pour quitter l'éditeur Vim.
* La combinaison de :w pour écrire (enregistrer) le fichier et rester en mode de commande afin de continuer l’édition.
* La combinaison :wq pour écrire (enregistrer) le fichier et quitter l'éditeur Vim.
* La combinaison :q! pour quitter l'éditeur Vim en annulant toutes les modifications apportées au fichier depuis la dernière écriture.

Vim offre des combinaisons de touches efficaces et coordonnées pour effectuer des tâches d'édition évoluées. Alors qu'il est considéré comme très utile une fois maîtrisé, l'éditeur Vim peut désorienter les nouveaux utilisateurs. Le flux de travail suivant présente les touches de base que chaque utilisateur de Vim doit connaître pour accomplir n'importe quelle tâche d'édition.

Les redirections

Un programme d'exécution, ou processus, doit lire l'entrée depuis un emplacement et écrire la sortie sur l'écran ou dans des fichiers. Une commande exécutée depuis l'invite de shell lit habituellement son entrée depuis le clavier et envoie sa sortie vers sa fenêtre de terminal.

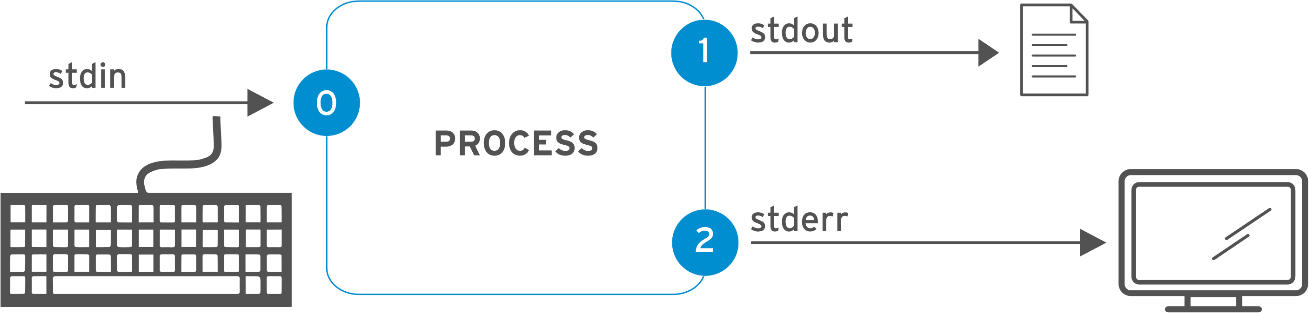
Un processus utilise des canaux numérotés appelés descripteurs de fichiers pour obtenir une entrée et envoyer une sortie. Tous les processus disposent d'au moins trois descripteurs de fichiers pour commencer. L'Entrée standard (canal 0) lit l'entrée depuis le clavier. La Sortie standard (canal 1) envoie la sortie normale vers le terminal.

L'Erreur standard (canal 2) envoie les messages d'erreur vers le terminal. Si un programme ouvre des connexions séparées avec d'autres fichiers, il peut utiliser des descripteurs de fichiers portant des numéros plus élevés.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numéro | Nom du canal | Description | Connexion par défaut | Utilisation |
| 0 | stdin | Entrée standard | Clavier | lecture seule |
| 1 | stdout | Sortie standard | Terminal | écriture seule |
| 2 | stderr | Erreur standard | Terminal | écriture seule |
| 3+ | nom de fichier | Autres fichiers | aucune | lecture et/ou écriture |

La redirection d'E/S remplace les destinations de canal par défaut par les noms de fichiers qui représentent des fichiers de sortie ou des périphériques. En utilisant la redirection, les sorties de processus et les messages d'erreur normalement envoyés vers la fenêtre de terminal peuvent être capturés sous forme de contenu de fichier, envoyés vers un périphérique ou supprimés.

La redirection de stdout empêche l'affichage de la sortie du processus sur le terminal. Comme indiqué dans le tableau suivant, la redirection de *stdout* uniquementn'empêche pas l'affichage des messages d'erreur stderr sur le terminal. Si le fichier n'existe pas, il sera créé. Si le fichier existe et que la redirection ne s'ajoute pas au fichier, le contenu du fichier sera écrasé. Le fichier spécial /dev/null élimine discrètement la sortie de canal redirigée vers ce fichier et est toujours un fichier vide.



### Opérateurs principaux de redirection de sortie

Commande > fichier

Redirection du résultat de la commande (stdout) dans un fichier tout en écrasant le contenu du fichier (si contenu existant)

Commande >> fichier

Redirection du résultat de la commande (stdout) dans un fichier en l’ajoutant au reste du fichier (si contenu existant)

Exemple de redirection de sortie

* Enregistrer un horodatage pour consultation ultérieure.

Une image contenant texte, Police, capture d’écran, blanc

Description générée automatiquement

* Copier les 100 dernières lignes d’un fichier log vers un autre fichier.



### Exercices

Chaque exercice vaut pour 2 points.

1. Créez un fichier vim\_exercice1 avec la commande vim. Annotez dedans « Mais, vous savez, moi je ne crois pas qu’il y ait de bonne ou de mauvaise situation. – Otis »

Indice : vim nom\_fichier

1. Créez un fichier nano\_exercice2 avec la commande nano et éditez-le en inscrivant vos nom, prénom, groupe et section l’un en dessous des autres. Sauvegardez vos modifications.
2. Listez les 2 premières lignes de votre fichier et redirigez-les dans un fichier nommé result\_exercice3.
   1. Faites un cat result\_exercice3 pour afficher le contenu du fichier.
3. Listez les répertoires de la racine et redirigez cette liste dans un fichier

## liste\_repertoire\_racine4.

* 1. La commande ls pour vous aider.

1. Listez les disques sur votre VM et redirigez cela dans un fichier disk\_ex5.
   1. La commande su - pour vous connecter en root.
   2. La commande fdisk -l peut vous aider.