[Système de fichiers]

1. Commande de recherche dans le système de fichiers sous GNU/Linux (description, utilisation et syntaxe)

Nom de la commande	find
Description	Une commande bash permettant de rechercher un ou plusieurs fichiers dans un système de fichier GNU/Linux
Utilisation	find "/home/Rakoto/Documents/CNTEMAD" -iname "exercice"
Syntaxe	find "path..." -iname "fichier" • "path...": indique le repertoire ou find va trouver le fichier. • -iname: l'option de find pour faire une requet a partir du nom du fichier, on utilise -iname au lieu de -name pour ignorer la casse. • "fichier": c'est le nom de fichier que nous voulons rechercher dans "path..."
Autre commande de recherche	grep locate

2. SWAP : Swap est une zone de mémoire virtuelle dont lequel un système utilise la partie du disque dur comme de la mémoire vive.

[Commandes de base]

3. date | tee fic : Cette commande retourne la date et l'heure actuelle au format indiqué et redirige cet résultat vers le fichier "fic" tout en affichant le meme résultat dans le terminal de l'utilisateur.

4. Commande "filtre" (3 exemples)

Définition d'une commande « filtre » : Un filtre est une commande qui lit les données sur l'entrée standard, effectue des traitements sur les lignes reçues et écrit le résultat sur la sortie standard.	
less	**[Syntaxe]** : less /home/Rakoto/CNTEMAD/projet/*.c > myprogram.c **[Explication] :** Cette commande va concaténer tout les codes écrit en C qui se trouve dans le dossier projet vers un seul fichier nommé myprogramm.c
cat	**[Syntaxe]** : cat /home/Rakoto/CNTEMAD/projet/file0.js >> file1.js

	[Explication] : Cette commande se comporte comme less mais a utiliser pour faire une opération sur quelques parties du fichier. Ici, l'utilisation de cat limite la longueur du code JS a concaténer avec celle du fichier file1.js
grep	**[Syntaxe]** : grep "^k" /etc/passwd **[Explication]** Cette commande permet de rechercher/filtrer les lignes commençant par k dans le fichier /etc/passwd

[Utilisateurs et Droits]

5.
SuperUser : On appel super utilisateur quand un utilisateur est dans un compte root et qu'il dispose tous les privilèges nécessaires pour contrôler l'integralitE d'un système GNU/Linux.
Distinction : Alors ce qui le distingue d'un utilisateur normal est le fait de posséder tous les droits sur tout le système de fichier ainsi la permission d'exécuter la commande sudo car il fait partie d'un groupe appelé "sudoers".

6. UID: l'abreviation de "User IDentifier", c'est un nombre permettant d'identifier un utilisateur dans un système multi-utilisateur.

7.

```
-rwsr-xr-x  1     root root    42824  sept. 13  2012    /usr/bin/passwd
```

Le premier champ "rws"	Signifie que le propriétaire root qui est le super utilisateur a le droit de lire, d'écrire et d'exécuter le fichier /usr/bin/passwd.
Le second champ "r-x"	Permet au groupe propriétaire qui est ici le super-utilisateur lui meme de lire et d'exécuter le fichier /usr/bin/passwd
Le dernier champ "r-x"	Réservé a tout les autres utilisateurs qui ont le droit de lire et d'exécuter mais qui ne peuvent pas modifier le fichier /ust/bin/passwd

8. /etc/group et son contenu :

Description *etc/group*	C'est un fichier texte qui définit les groupes auxquels appartiennent les utilisateurs sous le système GNU/Linux.
Contenu	Ce fichier comprend des lignes dans chacune d'elles contiennent 4 champs séparés par le symbole ":" dont :

	• 1er champ contient le **nom du groupe** (ex: admin, sudo, root, ...) • 2e champ contient le **mot de passe du groupe**, généralement le système met la valeur "x" qui signifie vide mais il peut stocker des mots de passe cryptes • 3e champ contient le **GID** (Group Identifier) • 4e champ contient le/les liste(s) de noms de(s) utilisateur(s) qui est/sont **membre(s) du groupe**

[Processus]

9. cron :

Définition	C'est un programme permettant de planifier des taches regulieres et soient de réaliser automatiquement ces taches par le système plutôt que d'avoir a les lancer manuellement en tant qu'utilisateur.
Fonctionnement	cron est un demon, ce qui veut dire qu'il tourne donc en tache de fond du système, il peut êtres arrêté, démarré ou redémarré.
Syntaxe (général)	crontab -e (pour accéder au fichier de configuration de crontab)

10. crontab (/usr/local/versement tous les vendredi a 12h sauf pour le mois d'août) **:**

LEÇON crontab

> crontab -e
m h M j tache>log

> Signification :
m: minutes (0 à 59)
h: heure (0 à 23)
M: mois (1 à 12)
j: jour (0 à 6)
tâche: commande à planifier
log: journalisation (du output)

> Notation :

Pour chaque unité de temps
***** : A chaque unité de temps
2-5 : les unités de temps (2,3,4,5)
***/3** : tous les 3 unités de temps (0,3,6,...)
5,8 : les unités de temps 5 et 8

> Exemple

Afficher espace libre tout les jours a 23h30 :
30 23 * * * df >> /temp/log.txt

Chaque minute :
*/1 * * * * df >> /temp/log.txt