

**Dernier T{D,P}**

---

**Exercice 1.***GNU/Linux*

1. Comment détecter le type d'un fichier ?
2. Comment afficher le contenu d'un fichier ?
3. Comment copier, supprimer, déplacer un fichier, dans un shell ?
4. Même question pour un dossier. .
5. Comment trouver un fichier ou un dossier à partir d'une partie de son nom, dans un dossier ?
6. Comment afficher le contenu d'un fichier suivant un certain motif ?
7. Comment afficher tout le contenu d'un fichier, sauf un motif ?
8. Comment trouver le manuel d'un programme ? Comment rechercher vers l'avant dans ce manuel ? Comment rechercher vers l'arrière ? Comment quitter `man` ?

**Exercice 2.***Coquillage*

1. Qu'est ce qu'un pipe ? Qu'est ce qu'un argument ?
2. Comment rediriger la sortie d'un programme dans un fichier ?
3. Comment envoyer `stderr` dans un fichier, `stdout` dans un autre fichier ?
4. Comment envoyer `stderr` dans `stdout` ?
5. Comment "convertir" un pipe liste d'arguments ?
6. Comment rediriger la sortie d'un programme dans un fichier ?
7. Comment envoyer `stderr` dans un fichier, `stdout` dans un autre fichier ?
8. Comment envoyer `stderr` dans `stdout` ?
9. Comment faire une condition ?
10. Comment faire une boucle `for` qui compte ?
11. Comment faire une boucle `while` ?

**Exercice 3.***Fichiers*

1. Comment ouvrir un fichier en C ?
2. Que représentent "a", "a+", "r+", "w+" ?
3. Comment déplacer le curseur dans un fichier ?
4. Que se passe-t-il si un fichier dont on a un descripteur est supprimé ?

**Exercice 4.***Memories (1995)*

1. Quelles sont les différentes parties d'un programme en mémoire ?
2. Quelle différence entre pile(*stack*) et tas(*heap*) ?

3. Comment allouer de la mémoire sur le tas? Que doit on faire quand on a fini d'utiliser de la mémoire allouée sur le tas?
4. Dans quelle zone est allouée un VLA (Variable Length Array)?
5. Quelle commande utiliser pour trouver la taille limite de la stack?
6. Un programme peut-il utiliser plus de mémoire qu'il n'y a de mémoire vive sur une machine?

#### Exercice 5.

*Fils etc.*

1. Quelle est la différence entre un thread et un processus?
2. Comment créer un thread?
3. Comment récupérer le résultat d'un thread? et d'un processus?
4. Comment créer de la mémoire partagée entre threads? et entre processus?
5. Quelle est la différence entre mutex et variable de condition?

#### Exercice 6.

*Compilation*

1. Comment spécifier à un linker qu'on veut utiliser une bibliothèque? On utilise l'option `-l`, par exemple :  
`gcc truc.cpp -lmath -lgurobi -lmasuperlib`. Il faut que la bibliothèque se trouve dans un dossier par défaut, ou spécifié au linker.
2. Comment spécifier à un linker où trouver une bibliothèque?
3. Quelles options utiliser pour demander à gcc/clang de nous afficher plus d'avertissements?

#### Exercice 7.

*Réseau*

1. Comment sont organisés les protocoles de communication, de manière générale?
2. À quoi sert ARP?
3. À quoi sert un switch (ou commutateur réseau)?
4. Quelle forme a une adresse IPv4? Combien d'adresses IPv4 existe-t-il?
5. Quelle forme a la notation d'un bloc CIDR pour une adresse IPv4?
6. À quoi sert l'adresse IPv4 127.0.0.1?
7. Quelles adresses IPv4 sont réservées pour les réseaux locaux?
8. Quels champs d'un paquet IP connaissez-vous?
9. Quelles sont les différences entre UDP et TCP? Quelles sont leurs utilisations?
10. À quoi sert le protocole DNS?

#### Exercice 8.

*Sockets*

1. Comment créer une socket?
2. Quelles sont les étapes que doit réaliser un client pour se connecter à un serveur, avec une socket?
3. Quelles sont les étapes que doit réaliser un serveur pour permettre aux clients de se connecter à lui, avec une socket?

**Exercice 9.***Routage*

1. À quoi sert une route ?
2. Comment ajouter une route ?
3. Qu'est ce qu'une passerelle ?
4. Comment ajouter une passerelle par défaut ?
5. Quelle commande utiliser pour trouver à quel fournisseur d'accès internet (FAI) appartient l'IP 140.77.3.8 ?