



# Connexion à d'autres ordinateurs et au réseau de l'ENSEEIH

## Objectifs

- se connecter à distance et lancer des applications sur d'autres ordinateurs
- trouver les principales caractéristiques d'un ordinateur

## 1 Connexion à un autre ordinateur

### 1.1 Indépendance du point de connexion

Il est possible de se connecter à distance (*remote* en anglais) à un autre ordinateur.

Le système de gestion des utilisateurs n'étant pas local, vous pouvez vous connecter depuis votre ordinateur à **n'importe quel ordinateur du deuxième étage** : vous retrouvez alors votre compte comme si vous étiez connecté « physiquement » sur l'ordinateur (il n'y a en fait quasiment aucune différence entre ces deux types de connexion) et vous pouvez lancer des commandes normalement.

#### 1.1.1 Les commandes de connexion à distance

Il existe plusieurs commandes pour effectuer cette connexion :

- **telnet** : c'est un interpréteur de commandes en soi<sup>1</sup>, mais il est souvent utilisé pour effectuer une connexion simple ;
- **ssh** (**S**ecure **S**ession **H**ost) est l'une des plus utilisées.

**Syntaxe** Leur syntaxe de base est la suivante : `<commande> <nomOrdinateur>` où *commande* désigne l'une des commandes. Le nom d'un ordinateur est indiqué sur le dessus ou sur la façade de l'unité centrale.

#### 1.1.2 Sécurité des commandes

L'une des caractéristiques importantes de **ssh** est que **ssh** effectue des connexions sécurisées, c'est-à-dire que les transmissions sont cryptées. De plus, **ssh** demande toujours l'authentification, alors que ce n'est pas toujours le cas pour les autres commandes.

**ssh** permet de garantir :

- La confidentialité : le chiffrement des paquets permet de garantir celle-ci. Les anciens services tels que telnet, rlogin... envoyaient les données en clair.

---

1. Le SHELL est un interpréteur de commandes.

- L'UTILISATION DE `ssh` EST FORTEMENT ENCOURAGÉE.

### 1.2.1 Syntaxe

### 1.2.2 Stockage des clés cryptées d'authentification

**Travail à effectuer :**

- Depuis un terminal de votre ordinateur, connectez-vous, avec `ssh`, sur un ordinateur voisin, et vérifiez ce qui vous est demandé,
- Depuis un autre terminal de votre ordinateur, exécutez la commande `ssh-copy-id nomOrdinateur`, avec `nomOrdinateur` désignant un deuxième ordinateur voisin,
- Connectez-vous sur cet ordinateur voisin avec la commande `ssh`, et observez ce qui se passe,
- Connectez-vous sur un autre ordinateur voisin via la commande `ssh`, et notez si un mot de passe vous est demandé.
- Visualisez la clé qui est stockée dans :  
`~/.ssh/authorized_keys`

En réalité, lorsque vous vous connectez sur l'une des machines voisines avec `ssh`, la clé d'authentification de l'utilisateur qui est utilisée n'est pas propre à cette machine, car par le biais de l'auto-montage de votre espace depuis le serveur, l'authentification se fait sur le serveur. Ce qui vous permet de vous connecter sur différentes machines avec la même clé.

### 1.2.3 Exécution d'une commande sur un autre ordinateur

La commande `ssh` permet aussi de lancer l'exécution d'une commande sur un autre ordinateur. Tapez, par exemple, la commande `ssh nomOrdinateur ps` et observez ce qui se passe.

### 1.2.4 Déconnexion

Pour terminer la connexion sécurisée et donc revenir sur votre ordinateur de départ la commande est `exit`

## 1.3 Autres types de connexion à distance

À noter qu'il existe des possibilités de transfert de fichiers avec :

- `ftp` (**F**ile **T**ransfer **P**rotocol) et sa version sécurisée `sftp` (**S**ecure **F**TP) qui fonctionnent comme des interpréteurs de commandes ;
- `scp` (**S**ecure **C**opy) qui a une syntaxe analogue (bien qu'un peu plus complexe) que `cp` et qui permet d'effectuer des copies de façon directe.

N.B. : `scp` utilise `ssh`.

## 1.4 Qui est sur « mon » ordinateur ?

Un point important : plusieurs utilisateurs peuvent être connectés **sur le même ordinateur en même temps** (personne assise devant l'ordinateur, `ssh`...).

Une fois connecté sur un ordinateur, vous avez alors la possibilité de voir qui est connecté sur cet ordinateur : pour cela vous disposez de trois commandes : `w`, `who` et `finger` (sans aucun paramètre).



**Travail à effectuer :** Essayez ces trois commandes sur l'ordinateur qui est devant vous et sur les 3 ordinateurs distants sur lesquels vous vous êtes connecté plus haut.

## 1.5 Trouver les autres utilisateurs

Vous pouvez également avoir une vue d'ensemble du réseau en utilisant la commande **rusers**. Si vous voyez un même login plusieurs fois, cela signifie que cet utilisateur a plusieurs connexions ( $\simeq$  terminaux en général) sur cet ordinateur. Habituellement la commande **rusers** met du temps à se terminer ; si vous voulez l'arrêter, tapez **Ctrl-C**.

La commande **finger** présentée ci-dessus, permet aussi de faire la correspondance entre login et nom réel.

Mais il se peut que ces deux commandes ne soient pas installées sur les ordinateurs de certaines salles de tp.

## 1.6 L'affichage à distance d'applications graphiques

Lorsque vous vous connectez sur un ordinateur distant, vous pouvez lancer n'importe quelle application comme si vous étiez sur votre propre ordinateur... à un détail près : l'affichage graphique.

Par défaut, l'affichage graphique se fait sur l'ordinateur sur lequel vous avez lancé l'application, qui — si vous vous êtes connecté à distance — n'est pas le vôtre. Autrement dit, l'application aura bien été lancée mais vous ne verrez rien !<sup>2</sup>

Car les flux de données graphiques ne sont pas, par défaut, automatiquement transmis à travers la connexion à l'ordinateur distant. Il faut explicitement l'autoriser.

### Autoriser le flux de données graphiques avec **ssh** :

Dans le cas où l'option n'est pas activée, l'autorisation de flux graphique se fait en ajoutant l'option **-X** à la commande **ssh** :

> **ssh -X nomOrdinateur**



**Travail à effectuer :** Exécutez **gvim** sur un autre ordinateur que le vôtre.

## 2 Mieux connaître son ordinateur

### 2.1 Les caractéristiques physiques

#### 2.1.1 Caractéristiques du processeur

Sous Linux, la commande **cat /proc/cpuinfo** permet d'afficher les principales caractéristiques physiques du processeur. Essayez cette commande et vérifiez les points suivants :

- le nombre de coeurs (cores)
- la fréquence
- la taille de la mémoire cache
- la largeur des adresses physiques et virtuelles

---

<sup>2</sup>. Nous parlons ici d'affichage de fenêtres graphiques. L'affichage dit « en mode texte » (dans le terminal) n'est pas affecté.

- la présence d'un fpu (floating point unit)

### 2.1.2 Caractéristiques du disque dur

Sous Linux, la commande `df -h` permet d'afficher les principales caractéristiques de l'espace disque :

- la taille
- l'espace utilisé
- l'espace restant

### 2.1.3 Caractéristiques de la mémoire

Sous Linux, la commande `free -m` permet d'afficher la taille de la RAM en Méga-octets et son taux d'utilisation.

## 2.2 Les caractéristiques de connexion

### 2.2.1 L'adresse MAC

L'adresse MAC (Media Access Control), parfois nommée adresse physique est un identifiant physique stocké dans une carte réseau ou une interface réseau similaire.

L'adresse MAC est utilisée pour identifier l'ordinateur (ou les appareils connectés) lors d'échanges sur les réseaux.

Elle peut être codée sur différentes formes, dont la plus utilisée est la MAC-48 ou EUI-48, qui se présente sous la forme d'une séquence de 48 bits affichés en hexa-décimal (4 bits forment un chiffre de 0 à 9 et de a à f). Un autre format EUI-68 a été défini pour étendre l'utilisation de l'adresse MAC.

Sous Linux, la commande `ifconfig` permet de trouver l'adresse MAC (HWaddr).

### 2.2.2 L'adresse IP

Chaque ordinateur connecté à un réseau ou à internet dispose d'une adresse IP (pour Internet Protocol), qui est à la base du système d'acheminement des données sur internet.

Il existe des adresses IP de version 4 (sur 32 bits, soit 4 octets) et de version 6 (sur 128 bits, soit 16 octets). La version 4 est actuellement la plus utilisée : elle est généralement représentée en notation décimale avec quatre nombres compris entre 0 et 255, séparés par des points. Pour l'ENSEEIH, les adresses IP sont, pour la plupart des machines de TP, de la forme : 147.127.133.xxx

Sous Linux, la commande `ifconfig` permet de trouver l'adresse IP (voir deuxième ligne ci-dessous).

```
eth1      Link encap:Ethernet  HWaddr 80:c1:6e:fa:f5:32
          inet adr:147.127.133.108  Bcast:147.127.143.255  Masque:255.255.240.0
```

### 2.2.3 Correspondance entre nom et adresse IP

Exécutez la commande `ping` avec comme argument le nom d'un ordinateur de la salle où vous vous trouvez. Observez ce qui se passe.

Déduisez l'adresse IP de cette machine; adresse IP qui peut être utilisée comme argument des commandes `ssh`, `ping`, etc.

Pour désigner une machine sur le réseau, l'adresse IP n'est pas facile à retenir ou à manipuler, et l'on préfère utiliser un nom. Il existe donc un service (appelé DNS : Domain Name Service) qui permet d'établir la correspondance entre nom et adresse IP.

## 3 Accès aux ordinateurs de l'ENSEEIH T depuis l'extérieur

Si vous pouvez accéder depuis l'extérieur à différentes ressources (moodle, intranet, etc.) avec, seulement, votre login et mot de passe, vous ne pouvez accéder à vos données, ou lancer des travaux qu'à travers une connexion sécurisée.

Le VPN ou RPV (Réseau Privé Virtuel) est un dispositif logiciel qui permet de créer un tunnel entre un ordinateur et un réseau (ou serveur). L'ordinateur connecté au VPN se trouve ainsi sur le même réseau local (virtuel), et dispose des accès à différentes ressources protégées.

Cela constitue, pour vous un moyen d'accéder à vos données, et de travailler sur vos travaux et projets depuis chez vous. Pour installer ce dispositif sur votre ordinateur personnel, allez sur le lien suivant : <http://dsi.inp-toulouse.fr/fr/mon-compte/mobilite-nomadisme-wi-fi-vpn.html> et suivez les instructions.

## 4 Test d'auto-évaluation

Lorsque l'on veut exécuter une séquence de commandes dans un ordre précis ou de façon répétitive, on peut les mettre dans un fichier appelé script.

Et pour exécuter cette séquence de commandes, il suffit de lancer ce fichier exécutable.

Par exemple, pour les 3 ordinateurs sur lesquels vous vous êtes connecté, plus haut, via `ssh`, vous pouvez mémoriser les adresses IP par la séquence de commandes :

```
> ssh nomOrdinateur1 ifconfig > fichier_tmp
> ssh nomOrdinateur2 ifconfig >> fichier_tmp
> ssh nomOrdinateur3 ifconfig >> fichier_tmp
> grep Bcast fichier_tmp > fichier_ip
```

ou mettre cette séquence dans un fichier et l'exécuter. Testez cette séquence de commandes en la mettant dans un fichier script, et en l'enrichissant par l'ajout du nom de l'ordinateur avant la ligne contenant son adresse IP.

Afin de vérifier, vous mêmes, si vous avez bien compris les mécanismes principaux, et retenu les principales commandes, réalisez les actions suivantes :

- lister les noms des utilisateurs connectés à une machine voisine,
- lister le nombre de processus appartenant à un utilisateur donné sur une machine voisine,
- lister les utilisateurs connectés à un ordinateur voisin avec ssh,
- lister les noms des utilisateurs connectés à un ensemble d'ordinateurs désignés par leur nom ou leur adresse IP.
- vérifier si un utilisateur est connecté sur un ensemble d'ordinateurs voisins.

Si la liste des ordinateurs testés est longue, il devient fastidieux de faire ce test en énumérant les traitements effectués pour chaque ordinateur. Pour faciliter l'écriture de scripts, il existe, comme pour les langages de programmation, des structures de contrôle qui permettent de faire des boucles, de tester des conditions, etc.

Par exemple, dans un script, vous pouvez faire :

```
for m in nomOrdinateur1 nomOrdinateur2 ...
do
    ssh $m ....
done
```

Ainsi, les traitements à l'intérieur de la boucle `for` sont exécutés avec toutes les valeurs de la variable `$m` (parmi la liste énumérée).

Encore plus, la liste des différentes valeurs de `$m` dans la structure `for` peut être obtenue par le résultat d'une commande :

```
for m in `ls`      #m prendra tous les noms produits par ls
for m in `more fichierliste`  #m prendra tous les noms obtenus dans fichierliste
```

Un script peut avoir des arguments. Ils sont accessibles dans le code du script avec `$1` pour le premier, `$2` pour le deuxième, etc.

Effectuez de nouveau les actions demandées plus haut en utilisant la structure `for` dans votre script.

Dernier point : que faire si votre session sur l'ordinateur devant vous se trouve bloquée, et que vous n'avez plus la main ?

## 5 CQFAR (Ce Qu'il Faut Avoir Retenu)

- savoir se connecter à distance sur un autre ordinateur (`ssh`),
- savoir trouver ses propres processus actifs ou bloqués sur d'autres ordinateurs, et les arrêter,
- savoir retrouver les principales caractéristiques d'un ordinateur : cpu, RAM, disque, adresse MAC, adresse IP.