

Travailler sur sa machine

Comment travailler efficacement depuis chez soi/à distance.

- [MonTP : Accéder à sa session INSA depuis chez soi](#)
- [Transférer des fichiers entre son PC et sa session INSA](#)
- [WSL: Linux dans Windows](#)
- [Obtenir des licences de logiciels](#)
- [Accéder à ses mails](#)
- [Accéder à sa session en SSH](#)
- [Session VSCode distante par SSH](#)
- [Installer C sur son PC](#)

MonTP : Accéder à sa session INSA depuis chez soi

Le CSN a mis en place une plateforme de bureau distant permettant à tout étudiant d'accéder à sa session Windows ou Linux de l'INSA. La démarche demande d'abord d'installer le VPN avant de pouvoir accéder à l'accès distant.

Cela vous permet d'accéder aux logiciels (parfois payants et très onéreux) installés sur ces sessions ainsi que de travailler sur des fichiers qui sont restés en salle de TP ?

VPN

Il faut d'abord avoir configuré et être connecté au VPN, car pour des raisons de sécurité, ce service n'est pas accessible depuis un réseau extérieur à l'INSA. Le VPN va vous donner accès au réseau de l'INSA, et donc vous autoriser à utiliser l'outil mis à disposition. **Avant de procéder, configurez et connectez-vous au VPN en suivant ce tutoriel.**

Une analogie pour bien faire la différence entre le VPN et le bureau à distance : vous ne pouvez pas utiliser un ordinateur de l'INSA sans être dans la salle de TP. Imaginer se connecter au VPN est comme entrer dans la salle de TP, et se connecter au bureau distant est comme allumer l'ordinateur qui est dans la salle. Si vous n'êtes pas dans la salle, il sera compliqué d'allumer le PC.

Accès à vos sessions INSA avec le bureau distant

Avec le VPN activé, ouvrez votre navigateur et allez à cette adresse : <https://montp.insa-toulouse.fr>. Si la fenêtre ne charge pas, vérifiez que votre VPN est activé.

Entrez votre login INSA et votre mot de passe INSA.

Vous pourrez ensuite choisir une connexion à un Bureau à distance Linux ou un Bureau à distance Windows. Il suffit de cliquer sur celui de votre choix et c'est parti !

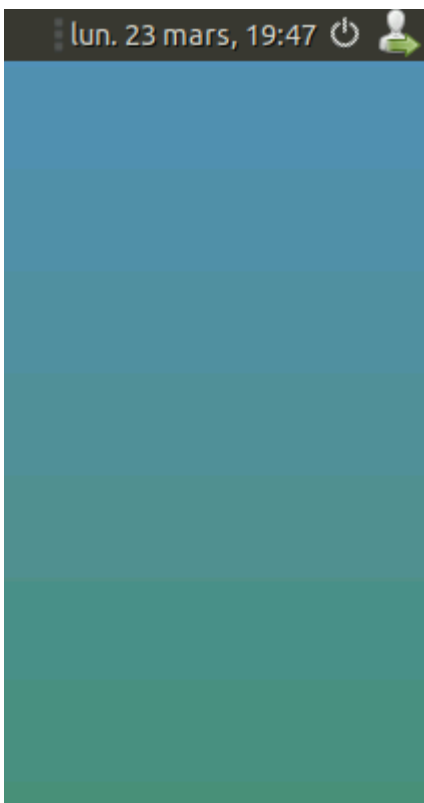
En cas de problème, visitez la documentation du CSN : <http://csn.insa-toulouse.fr/fr/services/services-a-la-pedagogie/salles-tps-virtuelles.html>

Déconnexion

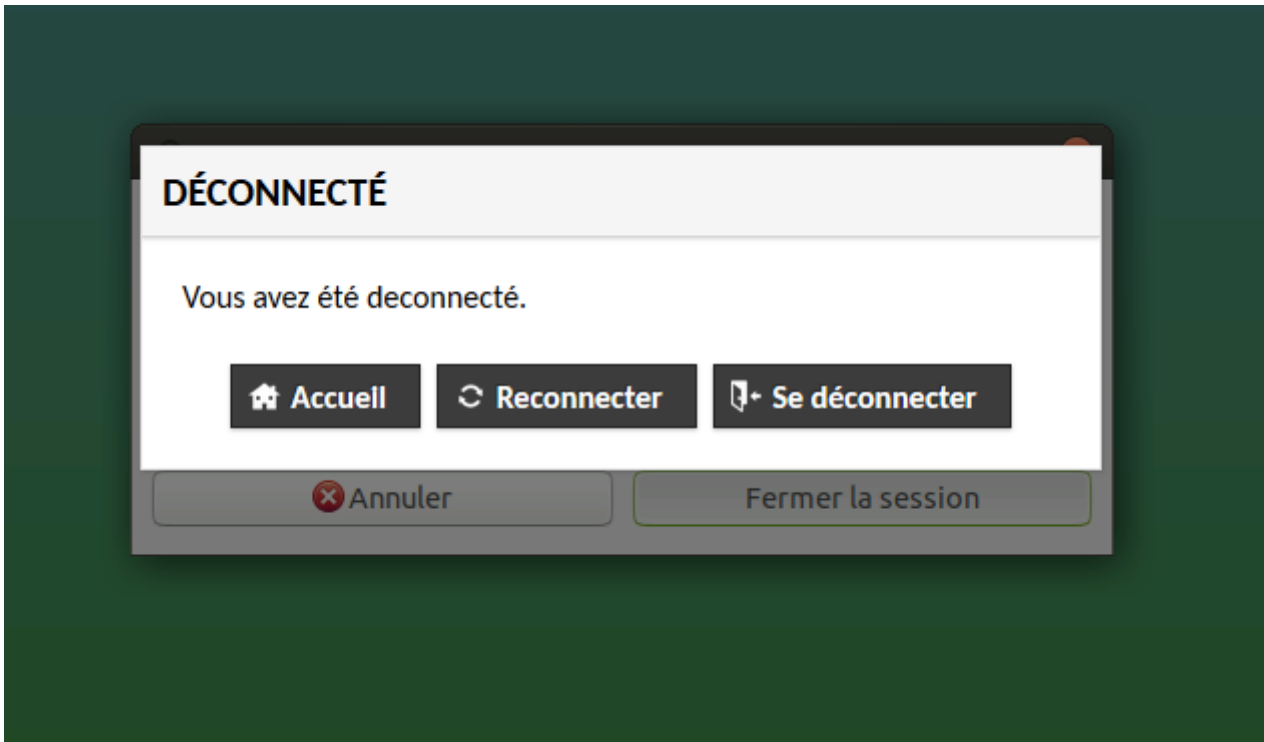
Merci de vous déconnecter une fois votre travail terminé. Les ressources sont limitées en ces temps difficiles et vous déconnectez permet de libérer les ressources associées à votre session pour que d'autres étudiants puissent profiter, eux aussi, de ce système.

Déconnecter de votre bureau Linux

Cliquer sur l'icône de déconnexion située dans le coin supérieur droit de votre écran.

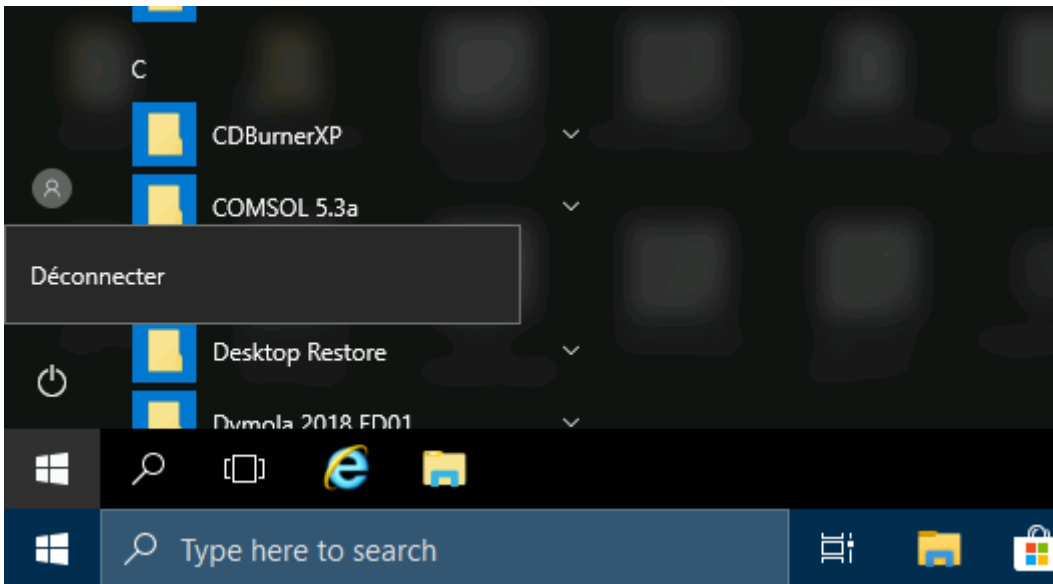


Un pop-up s'ouvrira, et il vous faut cliquer sur "Fermer la Session". Un autre pop-up, comme le montre l'image ci-dessous apparaît, il vous faut cliquer sur **Accueil** pour revenir à la page d'accueil.

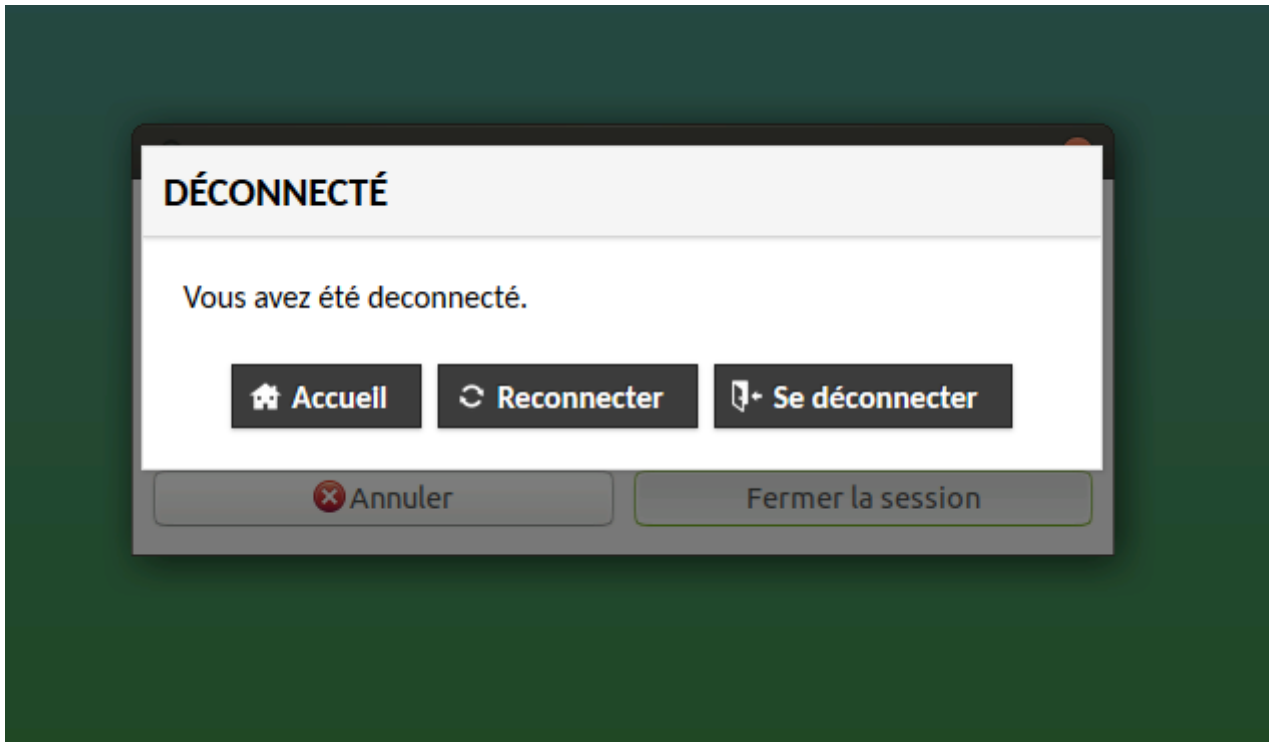


Déconnecter de votre bureau Windows

Cliquer sur l'icône Windows situé dans le coin inférieur gauche de votre écran. Cliquer sur "Marche/Arrêt" puis Déconnecter comme le montre l'image ci-dessous.



Un autre pop-up, comme le montre l'image ci-dessous apparaît, il vous faut cliquer sur **Accueil** pour revenir à la page d'accueil.



Contact

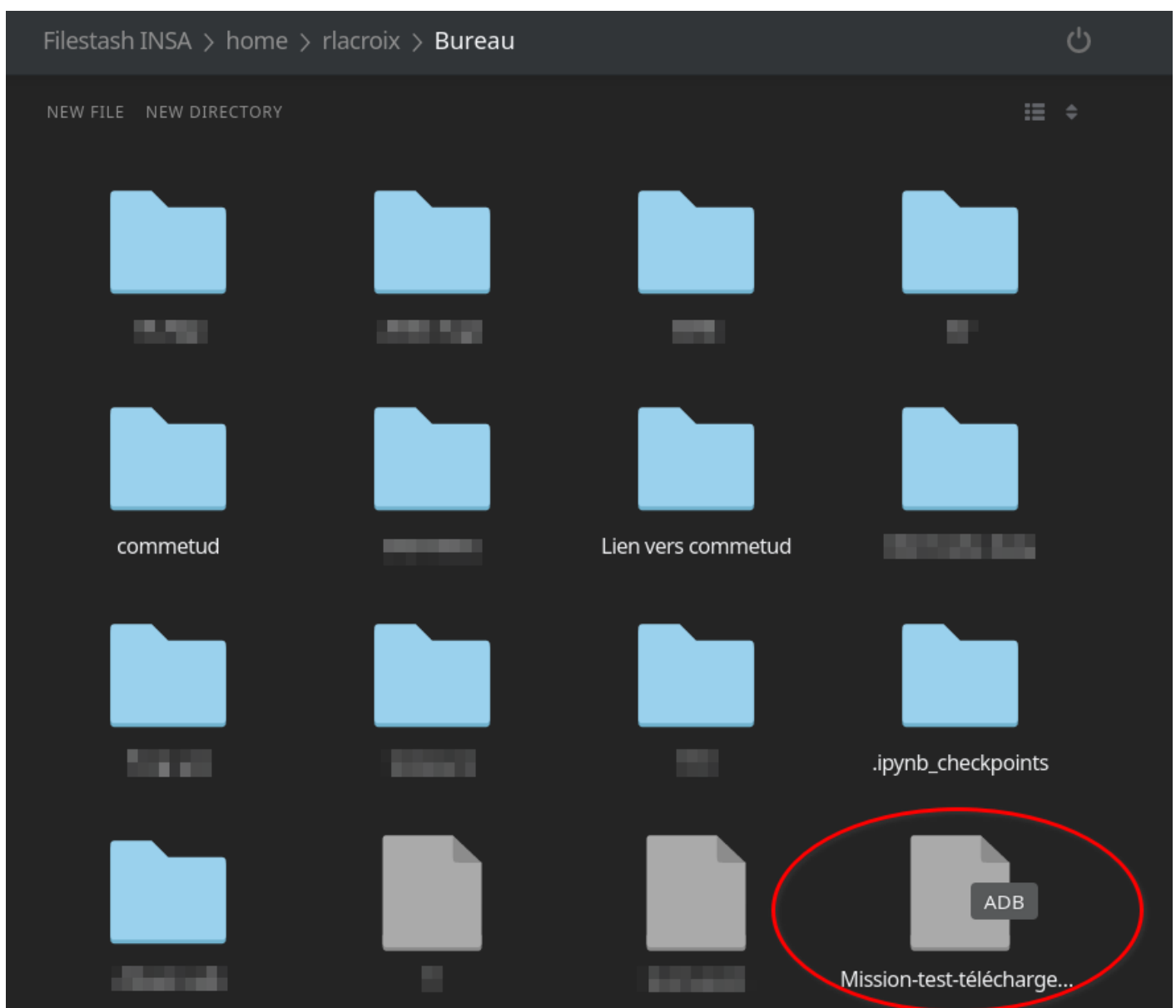
Si une erreur persiste, contacter le club'Info à l'adresse e-mail suivante : club.info@amicale-insat.fr

Transférer des fichiers entre son PC et sa session INSA

L'INSA de Toulouse permet à chaque étudiant d'accéder à ses fichiers créés lors d'une session sur l'un des ordinateurs de l'INSA (pour récupérer ses TPs par exemple !).

Depuis peu, le CSN a mis en place un WebFTP sur lequel vous pouvez voir, modifier et télécharger les fichiers qui se trouvent sur votre session INSA. Il suffit d'aller à l'adresse

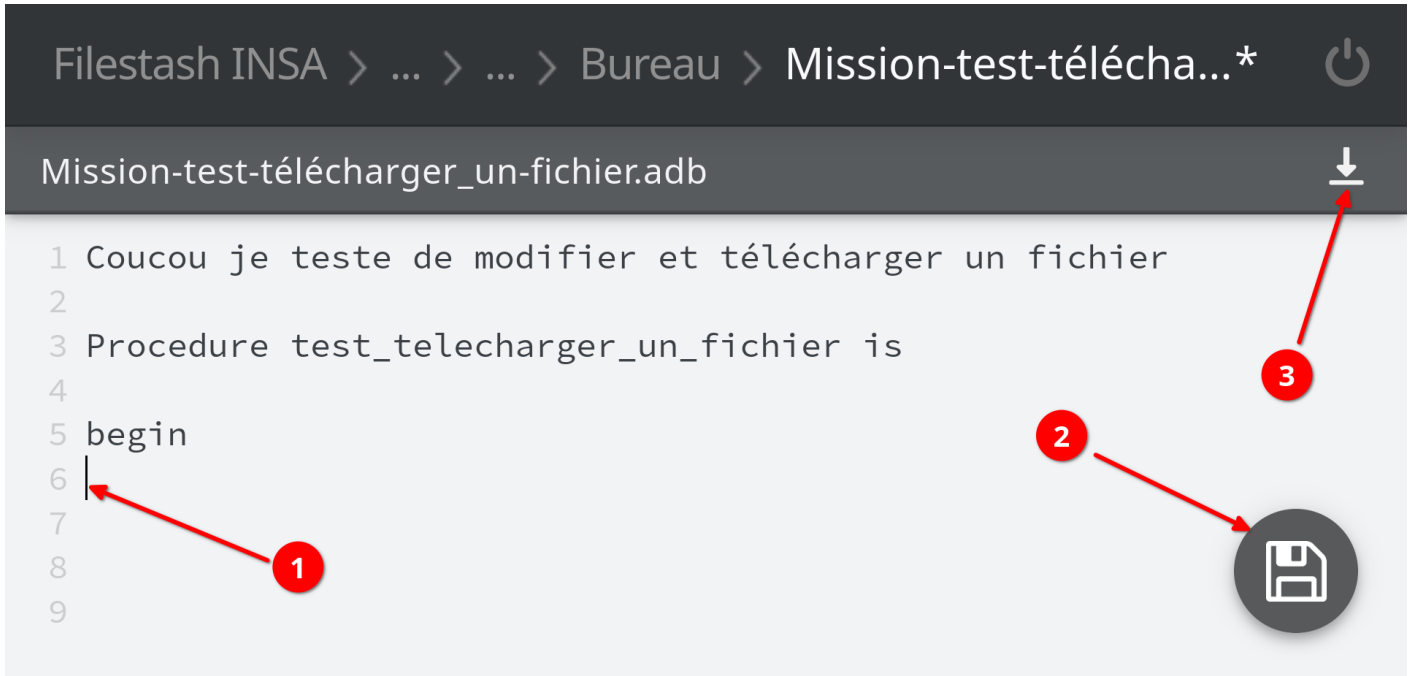
<https://mesfichiers.insa-toulouse.fr>.



Le dossier *Bureau* correspond à votre Bureau Linux. Les fichiers situés dans votre Bureau Windows se trouvent dans le dossier *Windows* puis *Bureau*.

Pour uploader un fichier, il suffit de faire glisser ce fichier sur le dossier dans lequel vous le voulez.

Vous pouvez également éditer un fichier en cliquant dessus (1) et en sauvegardant (2). Vous pouvez enfin télécharger le fichier en cliquant sur l'icône en haut (3)



Méthode de PGM

Le site *mes fichiers* peut être pénible pour transférer un dossier complet ou simplement plusieurs fichiers. Pour cela une solution est d'utiliser le SFTP :

- Connectez-vous au VPN
- Sur votre machine locale ouvrez votre meilleur terminal (Windows : Super+x puis *Windows Powershell*, Linux : Ctrl+Alt+T ou rechercher "Terminal")
- Tapez `sftp login@srv-ens-calcul.insa-toulouse.fr`
- Si on vous demande la confiance, tapez `yes`
- Entrez votre mot de passe
- Et voilà, vous êtes connecté ! Voici les commandes qui vous seront utiles :
 - `ls` pour lister les fichiers et répertoires
 - `cd` pour changer de répertoire (`cd ..` pour remonter)
 - `get -r file .` pour récupérer un fichier
 - `get -r * .` pour récupérer tous les fichiers d'un répertoire
 - `put -r file .` pour mettre un fichier en ligne

Le . sert à spécifier où vous voulez envoyer le fichier sur votre machine. Ici "." permet de dire "l'endroit où était votre terminal avant de lancer la commande sftp" (probablement votre home directory)

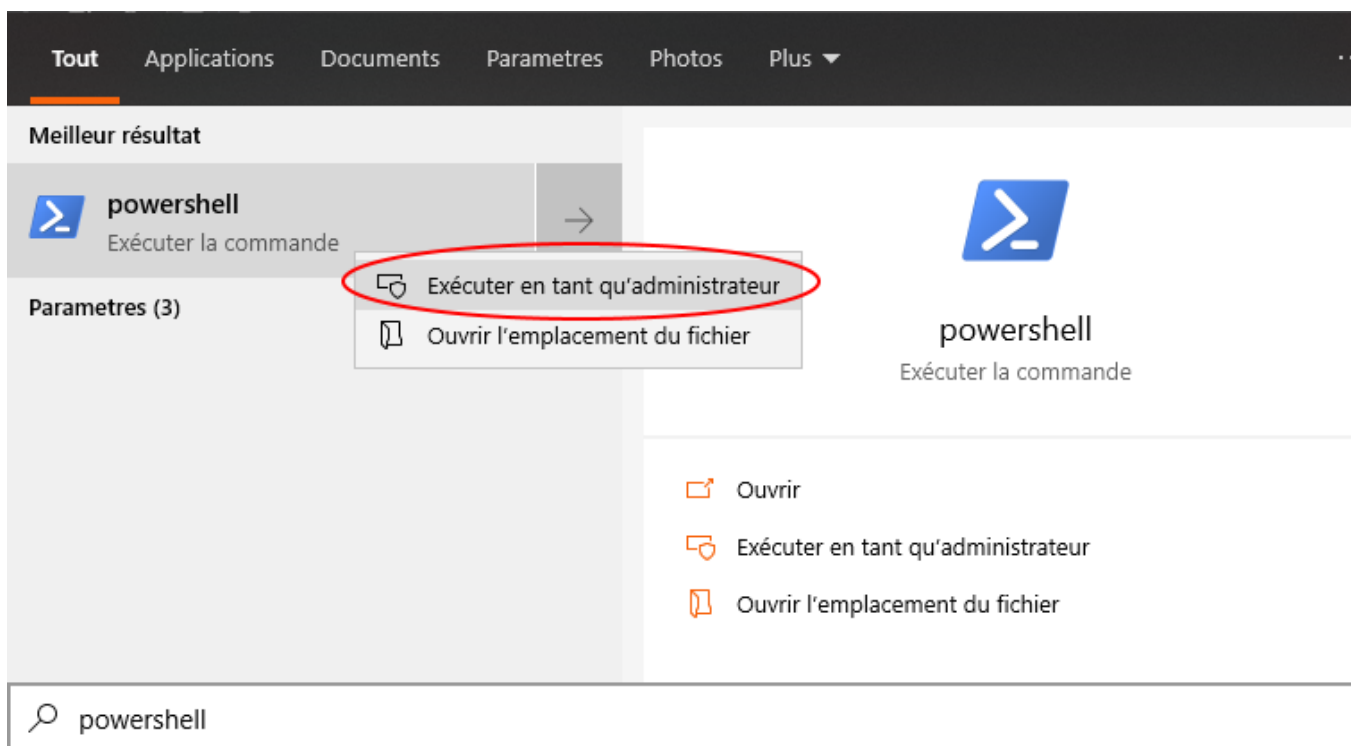
WSL: Linux dans Windows

Il peut être très utile d'avoir accès à un terminal Linux, mais mettre en place l'environnement, même avec une machine virtuelle, est parfois compliqué. Une solution proposée par Microsoft, s'appelle Windows Subsystem for Linux (WSL). Elle permet d'avoir un terminal Linux, parmi un choix de distributions répandues, notamment Ubuntu.

Cette solution n'est disponible que sur Windows 10 et 11.

Préparation

1. Ouvrez le menu Démarrer, cherchez l'outil "powershell".
2. À l'aide d'un clic droit, exécutez-le en tant qu'administrateur.



3. Exécutez la commande suivante :

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
```

```
Administrateur : Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
```

4. Redémarrez votre ordinateur.

Installation

1. Ouvrez le menu Démarrer, cherchez l'outil "Terminal", et ouvrez le.
2. Exécutez la commande suivante pour lister les distributions Linux que vous pouvez installer:

```
wsl --list --online
```

Au moment de l'écriture de ce tutoriel, cette commande retourne ces informations:

```
The following is a list of valid distributions that can be installed.
Install using 'wsl.exe --install <Distro>'.

NAME
AlmaLinux-8
AlmaLinux-9
AlmaLinux-Kitten-10
AlmaLinux-10
Debian
FedoraLinux-42
SUSE-Linux-Enterprise-15-SP6          SUSE      Linux      Enterprise
SUSE-Linux-Enterprise-15-SP7          SUSE      Linux      Enterprise
Ubuntu
Ubuntu-24.04
archlinux
kali-linux
openSUSE-Tumbleweed
openSUSE-Leap-15.6
```

```
Ubuntu-18.04
Ubuntu-20.04
Ubuntu-22.04
OracleLinux_7_9
OracleLinux_8_7
OracleLinux_9_1      Oracle Linux 9.1
```

3. Choisissez la distribution que vous souhaitez installer.

Si vous n'y connaissez pas grand chose, mieux vaut installer `Ubuntu`.
Pour le reste de ce tutoriel, nous allons utiliser cette dernière.

4. Installez votre WSL en exécutant la commande suivante:

```
wsl --install -d Ubuntu
```

Si vous souhaitez installer une autre distribution, vous devez remplacer Ubuntu par le nom de votre distribution.

Vous pouvez définir un nom à votre WSL en rajoutant l'argument `--name`. Ceci peut être utile si vous planifiez d'utiliser plusieurs wsl sur votre ordinateur.

Par exemple, si je veux une installation Ubuntu nommée INSA, la commande à exécuter sera la suivante:

```
wsl --install -d Ubuntu --name INSA
```

5. Au bout de quelques secondes (ou minutes selon votre connection), il vous sera demandé de donner quelques informations:

- Un nom pour votre compte Unix dans le WSL

Ce nom doit commencer par une lettre MINUSCULES et peut seulement contenir des lettres MINUSCULES, des chiffres, des underscore (`_`) et des tirets (`-`).

- Un mot de passe pour ce compte (il vous sera demandé de le taper 2 fois)

Celui-ci ne sera pas affiché dans le terminal

```
Provisioning the new WSL instance INSA
This might take a while...
Create a default Unix user account: tuto
New password:
Retype new password:
passwd: password updated successfully
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.6.87.2-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Thu Jun 26 16:56:10 CEST 2025

System load:  0.0          Processes:            31
Usage of /:   0.1% of 1006.85GB  Users logged in:    0
Memory usage: 5%          IPv4 address for eth0: 172.21.196.233
Swap usage:   0%

This message is shown once a day. To disable it please create the
/home/tuto/.hushlogin file.
tuto@picasso2005-Main:/mnt/c/Users/Utilisateur$ |
```

6. Une fois la commande finie, vous êtes dans votre WSL.

Afin d'éviter les futurs problèmes avec `apt` (la commande permettant d'installer tout et n'importe quoi sous Ubuntu), exécutez dans votre WSL la commande suivante:

```
sudo apt update
```

Il peut vous être demandé votre mot de passe du WSL

Commandes utiles de `apt`

- Pour rechercher le nom d'un paquet `apt search <NOM>` (remplacez `<NOM>` par votre recherche)
- Pour installer un paquet `sudo apt install <NOM>` (remplacez `<NOM>` par le nom de votre paquet)
- Pour mettre à jour votre WSL `sudo apt update && sudo apt upgrade`

7. Pour sortir de votre WSL, exécutez la commande `exit`

Comment accéder à votre WSL

Pour rentrer dans votre WSL, il vous suffit d'exécuter la commande `wsl` sans arguments.

Si vous possédez plusieurs installations de WSL, cette méthode vous fera rentrer dans votre distribution par défaut. Supposons que dans mes WSL, j'en ai un nommé `INSA`, qui n'est pas la distribution par défaut, je dois utiliser `wsl -d INSA`

Vous pouvez également accéder à vos dossiers présent dans votre WSL depuis l'explorateur de fichiers Windows.

Bonnes pratiques

Vous pouvez accéder à vos fichiers Windows depuis ce terminal. Il vous suffit d'aller dans le répertoire `/mnt`, dans lequel vous retrouverez vos disques, notamment le disque `C:`, qui aura pour chemin `/mnt/c` dans votre terminal.

Il est même **fortement recommandé** de travailler sur votre disque Windows, plutôt que dans votre répertoire personnel de Linux, sauf si vous affectionnez éditer vos fichiers dans un terminal.

Si vous utilisez **git**, il vaut mieux le faire **depuis Windows**. Des problèmes relatifs aux droits peuvent survenir si vous l'utilisez depuis Linux (c'est lié au système de fichier NTFS utilisé par Windows).

Pour aller plus loin

Vous pouvez voir la liste des WSL installés sur votre machine avec la commande `wsl -l -v`.

Vous pouvez également consulter la documentation officielle pour plus d'informations sur WSL:

- [Version anglaise](#)
- [Version française \(en partie traduite par IA\)](#)

Obtenir des licences de logiciels

[Pour cela, rendez vous ici.](#)

Accéder à ses mails

[Tout est expliqué ici.](#)



Accéder à sa session en SSH

Pour travailler vos cours d'UNIX, pour accéder à vos fichiers sans avoir une interface graphique encombrante ou simplement par commodité, vous pouvez vous SSH sur votre session INSA. Pour cela, rien de plus simple :

- Connectez-vous au VPN
- Sur votre machine locale ouvrez votre meilleur terminal (Windows : Win+x puis *Terminal*, Linux : Ctrl+Alt+T ou rechercher "Terminal")
- Tapez `ssh login@srv-ens-calcul.insa-toulouse.fr`
Remplacez `login` par votre login INSA
- Si on vous demande la confiance, tapez `yes`
- Entrez votre mot de passe
- Et voilà, vous êtes connecté ! Tapez `exit` pour vous déconnecter.

Session VSCode distante par SSH

Tu souhaites finir un TP mais en tu en as marre de travailler sur la salle de TP virtuelle ? Utilise une session distante VSCode ! Suis les étapes ci-dessous pour ouvrir une session par SSH.

- Télécharge VSCode sur ta machine
- Connecte-toi au VPN (cf. [page VPN](#))
- Ouvre VSCode
- Configure l'hôte SSH
 - Clique en bas à gauche sur le logo 
 - Sélectionne l'option SSH
 - Ajoute un nouvel hôte SSH `toto@srv-ens-calcul.insa-toulouse.fr`
Où `toto` est ton login INSA
 - Choisis l'emplacement du fichier de config SSH à utiliser
Si tu ne sais pas lequel choisir sélectionne le premier proposé
- Connecte-toi à l'hôte distant
 - Clique en bas à gauche sur le logo 
 - Choisis l'option de connexion SSH
 - Sélectionne l'hôte que tu viens de créer : `srv-ens-calcul.insa-toulouse.fr`
 - Entre ton mot de passe INSA
- Ouvre ton TP
 - Ouvre le dossier sur lequel tu veux travailler
 - Entre ton mot de passe INSA

- Si tu as besoin d'un terminal ouvre-le dans VSCode

Vous pouvez également vous connecter à un WSL si vous en avez un d'installé sur votre machine. Il vous suffit de choisir l'option WSL à la place de l'option SSH. Pensez à installer le nécessaire pour compiler votre langage si vous utilisez cette méthode !

Sous Ubuntu / Debian:

- Pour faire du C

```
sudo apt install gcc
```

- Pour faire de l'ADA

```
sudo apt install gnat
```

Installer C sur son PC

Installation de VSCode (5 min)

Quel que soit votre système d'exploitation, je vous recommande d'utiliser Visual Studio Code (VSCode) pour développer en C. Un tuto pour l'installer, et installer des extensions, est disponible sur [cette page WikiEtud](#).

De plus, pour faire du C, il vous faut installer au minimum les extensions VSCode suivante (vous pouvez en installer d'autre en plus selon vos préférences) :

- `C/C++` (par Microsoft)
- `C/C++ Extension Pack` (par Microsoft)
- `C/C++ Themes` (par Microsoft)

Installation de C sur Windows (15 min)

Je vous recommande spécialement d'installer VSCode si vous êtes sur Windows. En effet, VSCode permet de se connecter directement sur un WSL, ce qui rendra l'expérience plus agréable.

La méthode vous est expliquée dans le titre précédent.

Il est difficile d'installer `gcc` sur Windows de manière native (la plus part des méthodes sont trop vieilles et / ou dépréciées). De ce fait, nous allons utiliser un WSL (Windows Subsystem Linux) pour nous simplifier la tâche.

1. Installation de WSL (10-15 min)

Si vous avez déjà installé WSL, vous pouvez sauter cette étape.

Un tutoriel pour installer WSL est disponible sur [cette page WikiEtud](#). Afin de simplifier le point suivant, je supposerais que vous avez installé Ubuntu ou Debian.

Si vous utilisez une autre distribution que Ubuntu ou Debian, il vous faudra adapter les commandes à votre distribution (voir la partie sur l'installation de C sur Linux).

2. Installation de `gcc`, `gdb` et `make` (2 min)

Afin d'installer ces 3 paquets qui vous seront nécessaires pour les TD et TP de C, suivez les points suivants :

1. Ouvrez un Terminal (recherchez Terminal dans la barre de recherche Windows)
2. Connectez-vous à votre WSL
 - Si vous voulez vous connecter à votre WSL par défaut, rentrez la commande `wsl`
 - Si vous voulez vous connecter à un autre WSL, rentrez la commande `wsl -d NomDeVotreWSL` (en remplaçant évidemment le nom)
3. Une fois dans votre WSL, rentrez ces commandes :

```
sudo apt update
sudo apt install gcc gdb make

# Cette commande permet de mettre à jour tous les paquets de votre WSL
# Vous pouvez la sauter, même si je la recommande
sudo apt upgrade
```

Pour maintenant utiliser ces commandes comme en salle de TP, vous devrez d'abord vous connecter à votre WSL, et rentrer les commandes dans le bon dossier depuis votre celui-ci.

3. Connectez vous à votre WSL dans VSCode

La méthode pour se connecter à un WSL est similaire à celle pour connecter VSCode à un ordinateur distant. Pour vous connecter à WSL, vous pouvez donc suivre [le tutoriel présent sur cette page WikiEtud](#), en prenant en compte le paragraphe en bas de page.

Installation de C sur Mac (5 min)

Pour faciliter l'installation de nos paquets, nous allons utiliser le gestionnaire de paquets `homebrew`, non présent par défaut sur Mac.

1. Installation de Homebrew (2 min)

1. Ouvrez un terminal
2. Copiez-collez la commande suivante pour installer Homebrew

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

2. Installation de `gcc`, `gdb` et `make` (2 min)

Toujours dans votre terminal (réouvrez en un si vous l'avez fermé), rentrez les commandes suivantes:

```
brew install gcc  
brew install gdb  
brew install make
```

Vous pouvez à présent utiliser ces commandes comme en cours.

Installation de C sur Linux (2 min)

Vous devez seulement rentrer quelques commandes dans votre terminal. Selon votre distribution, celles-ci peuvent légèrement changer.

Si vous ne savez pas quelle distribution vous avez, celle-ci se retrouve avec la commande suivante:

```
cat /etc/os-release
```

 (attention à l'espace après le égal)

Vous retrouverez également la version de cette dernière si besoin.

Si votre distribution n'apparaît pas dans la liste, elle a de grandes chances d'être basé sur une distribution présente dans celle-ci. Vous pouvez utiliser les mêmes commandes.

Debian / Ubuntu / Mint

```
sudo apt update  
sudo apt install gcc gdb make
```

RedHat / CentOS (Version ≤ 7)

```
sudo yum install gcc gdb make
```

Fedora / CentOS (Version ≥ 8)

```
sudo dnf install gcc gdb make
```

Arch / Manjaro

Que faites-vous ici ? Vous vous êtes perdus ?

```
sudo pacman -S gcc gdb make
```

Gentoo

```
# Si les commandes suivantes sont erronées, merci de les corriger
```

```
# Je n'ai jamais utilisé emerge auparavant :(
```

```
sudo emerge -a sys-devel/gcc
```

```
sudo emerge -a dev-debug/gdb
```

```
sudo emerge -a dev-build/make
```

Si vous n'arrivez pas à installer ces outils, vous pouvez venir sur le [discord du Club Info](#) pour demander de l'aide.