MO101

Installation de Visual Studio Code

François Pessaux

2023-2024 v2.1.1

ENSTA
IP PARIS

Table des matières

1 Introduction

Ce document décrit les différentes étapes d'installation de l'environnement de développement `Visual Studio Code` et son utilisation via `ssh` pour bénéficier d'un accès (presque) transparent à votre compte informatique Linux ENSTA et aux outils logiciels qui vous seront nécessaires pour différents cours de votre cursus.

Le BYOD (« Bring Your Own Device ») est un mode d'organisation consistant à faire en sorte que les utilisateurs d'un système d'information puissent y accéder avec leurs équipements personnels (ordinateurs, smartphones, tablettes, etc.). Cette organisation s'applique dans le cadre de l'école puisque vous utiliserez principalement votre ordinateur personnel pour réaliser vos travaux.

En dépit de l'hétérogénéité de vos ordinateurs (systèmes d'exploitation, versions de ces derniers, logiciels disponibles, etc.) il est nécessaire que vous puissiez manipuler des fichiers qui seront stockés de manière pérenne sur un serveur de l'école. De plus, il est nécessaire que vous ayez toutes et tous un environnement de travail identique quel que soit le type de votre ordinateur. Par exemple, en fonction des systèmes d'exploitation certains logiciels existent ou non, ou bien ont des comportements différents (en particulier dès que l'on aborde le domaine de la programmation). Il faut donc éliminer toutes ces sources de problèmes potentiels avec un unique environnement de référence. Celui que l'école privilégie est basé sur le système d'exploitation Linux qui héberge votre compte informatique ENSTA. Un serveur (ordinateur distant) a été mis en place, avec les outils nécessaires (commandes, logiciels, bibliothèques, etc.) pour la majorité des cours d'informatique de première année et c'est sur lui que vous pourrez travailler depuis votre ordinateur personnel. Cela vous évite également de devoir installer, sur votre machine, la majorité des outils logiciels qui vous seront nécessaires durant les enseignements.

Pour rendre cette interaction avec le serveur aussi transparente que possible lors des travaux pratiques consistant à développer des programmes, nous allons utiliser un éditeur de texte nommé Visual Studio Code (que nous abrégerons désormais en VSCode) qui embarque de nombreuses extensions, combiné avec une extension du système d'exploitation qui permet de travailler sur des fichiers se trouvant sur une machine distante.

Les élèves disposant d'un ordinateur fonctionnant sous Linux ont déjà naturellement un environnement adéquat. Pour autant, ils n'ont pas d'accès (presque) transparent à leur compte informatique ENSTA, et donc à la possibilité de stocker des fichiers simplement depuis leur machine sur le serveur distant (pérenne, sauvegardé et administré professionnellement par la Direction des Systèmes d'Information – DSI). Aussi, ce document reste pertinent pour eux.

Quant aux élèves disposant d'une machine sous macOS, la situation est presque identique à celle ci-dessus, si ce n'est qu'il existe de subtiles différences entre certains logiciels, certaines commandes entre macOS et Linux. La majorité du temps on ne s'en aperçoit pas, mais parfois ça devient visible (voire gênant).

Si vous détectez des erreurs, des typos, je vous serai très reconnaissant de m'en faire part afin de pouvoir rendre ce document de meilleure facture pour les générations suivantes. Bonne lecture...

2 Connexion au réseau `eduroam`

Afin de disposer d'un accès à un réseau sécurisé et disponible dans tous les établissements d'enseignement supérieur et de recherche, le réseau **`eduroam`** est disponible pour tout élève de l'école. Il nécessite toutefois une configuration minime pour être opérationnel.

Connectez vous au réseau WIFI **`eduspot`** ou bien **`VISITORS`** en utilisant votre identifiant et mot de passe ENSTA (i.e. Cascad). Rendez-vous ensuite sur la page :

`https://markdown.data-ensta.fr/s/connexion-eduroam`

qui contient un tutoriel fort bien rédigé (par les élèves de l'association DaTA de l'école) sur la procédure à suivre en fonction de votre système d'exploitation.

Une fois **`eduroam`** configuré, vous pourrez l'utiliser à la place des deux cités ci-dessus (et ayant servi à accéder à Internet pour configurer **`eduroam`**. Cette connexion WIFI présente de nombreux avantages :

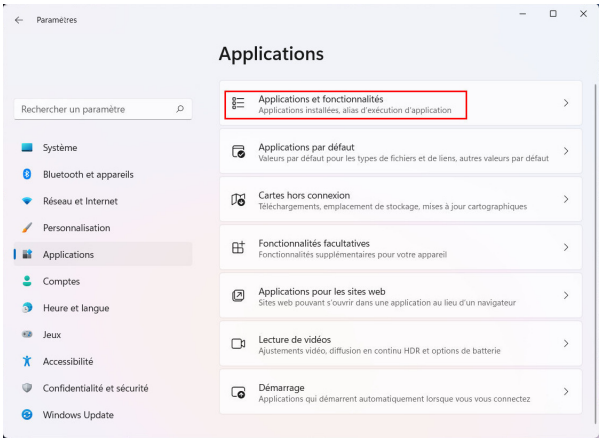
- connexion sécurisée,
- accès dans de nombreux établissements,
- pas de déconnexion au bout d'un certain temps,
- obtention d'une adresse réseau « ENSTA » autorisant l'accès à certains services uniquement accessibles « en interne ».

3 Activation de `ssh` sous Windows

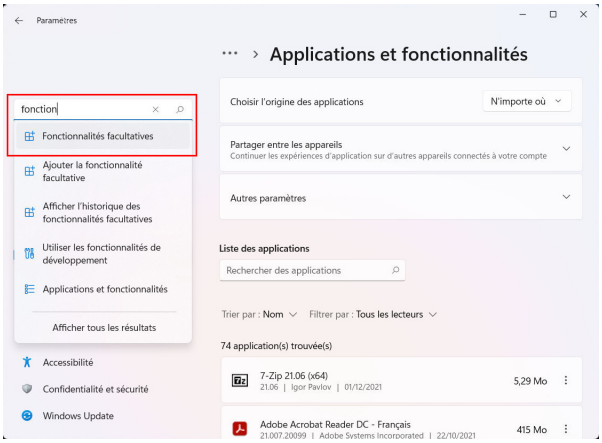
Afin d'établir une connexion sécurisée entre votre machine et le serveur distant, un outil nommé **`ssh`** est nécessaire. Sous Linux et macOS, il est nativement actif. Sous Windows 10 et 11, il faut l'activer explicitement.

3.1 Windows 10 et 11

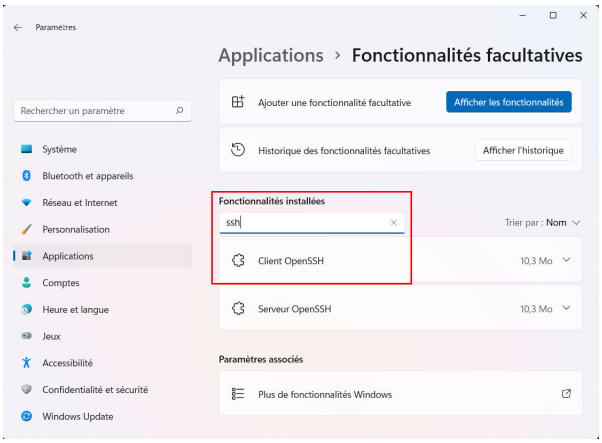
Le composant **`OpenSSH`** client peut être activé à l'aide des *Paramètres Windows* sur les appareils Windows 10 et 11. Ouvrez *Paramètres* et sélectionnez *Applications*.



Dans l'espace de recherche situé à gauche, recherchez puis sélectionnez l'entrée nommée *Fonctionnalités facultatives*.



Regardez dans la liste afin de voir si `OpenSSH` est déjà installé. Si tel n'est pas le cas, sélectionnez *Ajouter une fonctionnalité* en haut de la page, puis recherchez *OpenSSH Client* et cliquez sur *Installer*,

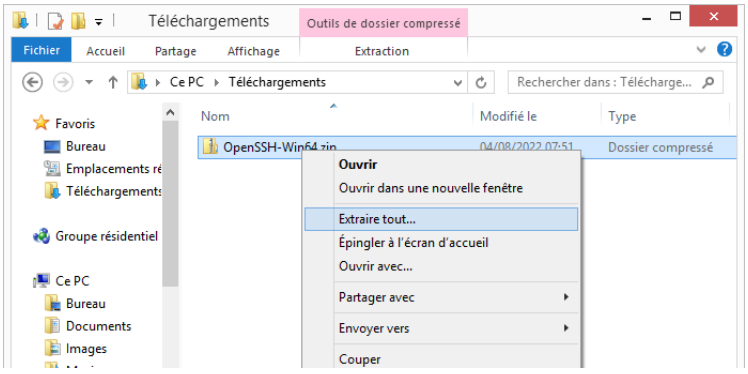


3.2 Windows 8

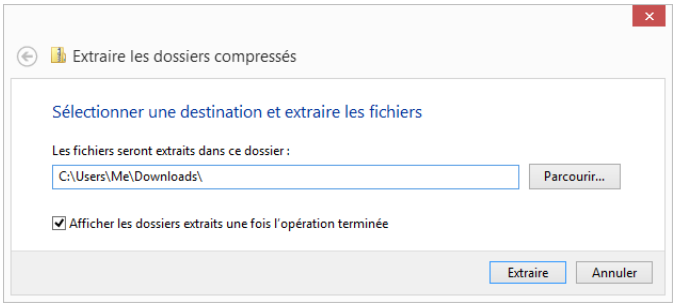
`OpenSSH` n'étant pas nativement présent dans le système d'exploitation, il doit être téléchargé préalablement à son utilisation. Pour ce faire, rendez-vous à l'URL :

`https://github.com/PowerShell/Win32-OpenSSH/releases/`

et sélectionnez l'archive `OpenSSH-Win64.zip`. Effectuez un clic droit sur le fichier `.zip` téléchargé et sélectionnez *Extraire tout*.

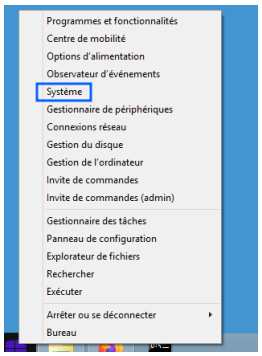


La fenêtre de dialogue vous demande où enregistrer le contenu décompressé en vous proposant un chemin prédéfini dont le dernier répertoire est le nom de l'archive. Supprimez ce dernier pour éviter la création de ce répertoire, le contenu de l'archive étant déjà groupé dans un répertoire.

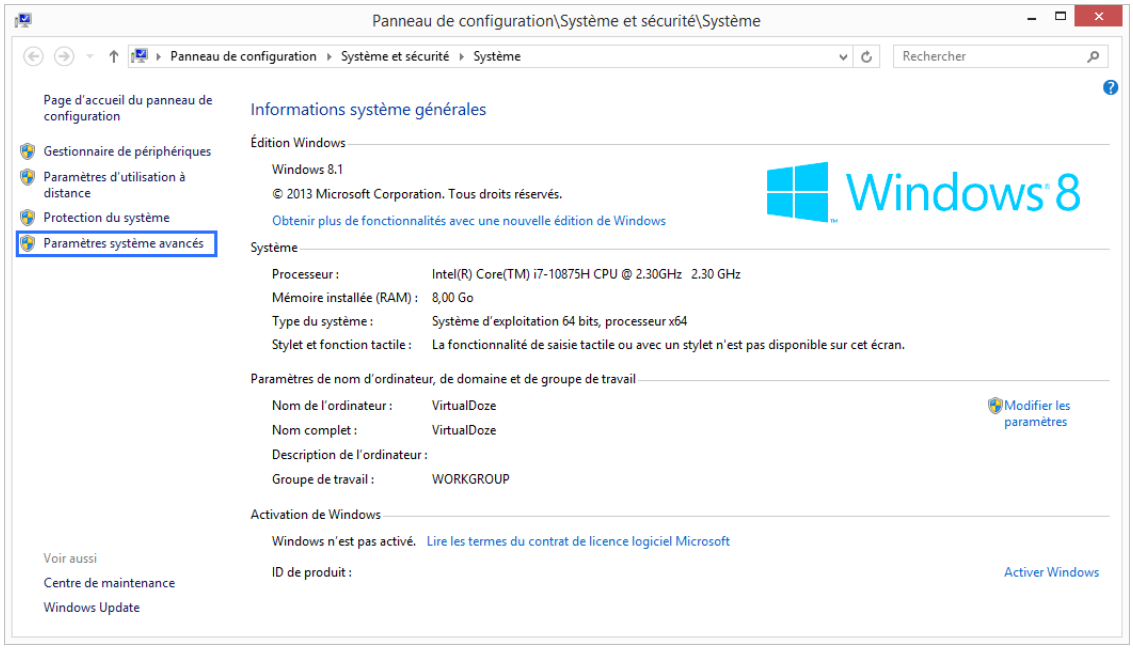


Copiez ensuite le répertoire `OpenSSH-Win64` (attention, pas l'archive `.zip` qui a parfois le même nom que le répertoire si les extensions de fichiers sont masquées) dans le répertoire `C:\Programmes`.

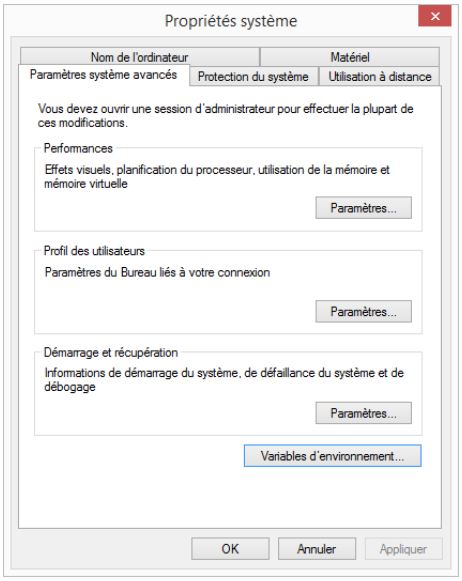
Pour que la commande `ssh` soit accessible, il faut faire en sorte que ce nouveau répertoire soit connu du système dans liste des répertoires de recherche des commandes. Pressez le raccourci clavier Win + x qui fait apparaître un menu contextuel. Sélectionnez l'entrée Système.



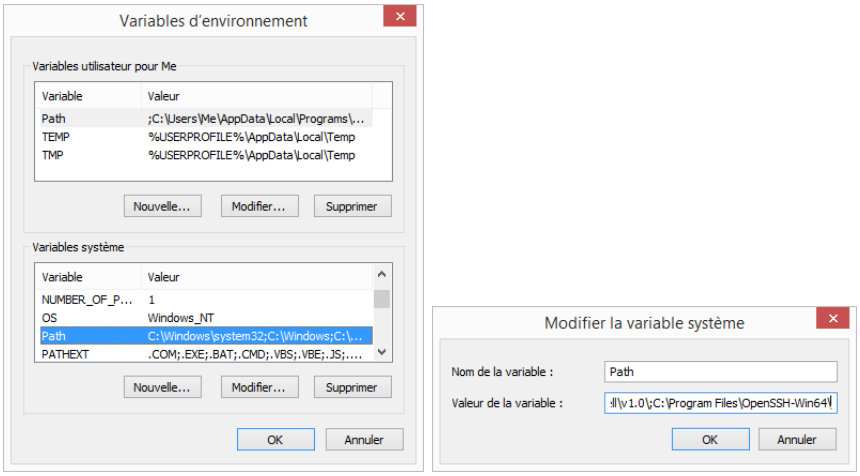
Dans la fenêtre qui s'ouvre, sélectionnez *Paramètres systèmes avancés*.



Dans la nouvelle fenêtre, sélectionnez le bouton *Variables d'environnement* tout en bas.



Une fenêtre s'ouvre alors dans laquelle vous devez faire défiler la liste des *Variables système* (en bas) jusqu'à trouver la variable nommée *Path*. Sélectionnez-la et cliquez sur le bouton *Modifier*.



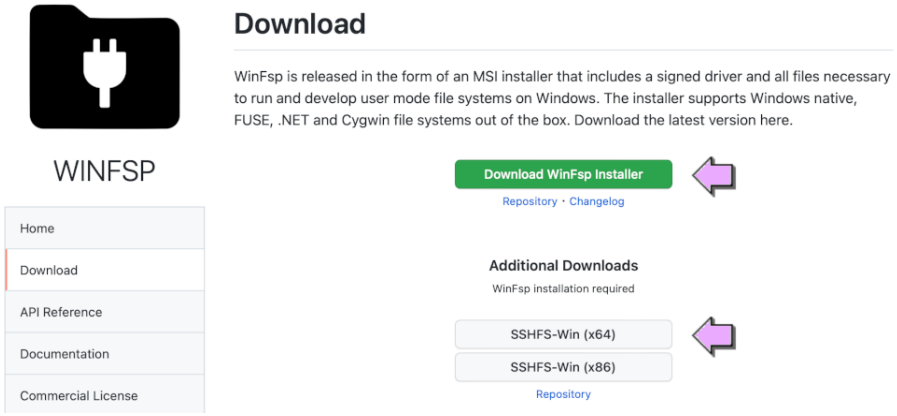
Dans la zone de saisie *Valeur de la variable*, ajoutez tout à la fin la chaîne :
`;C:\Program Files\OpenSSH-Win64\`
Cliquez sur les boutons *Ok* des différentes fenêtres pour valider les modifications.

4 Installation de `sshfs`

Afin de pouvoir accéder de manière transparente aux fichiers se trouvant sur la machine distante depuis votre propre ordinateur, il est nécessaire d'installer deux outils systèmes permettant de créer un « système de fichiers virtuel » qui fera apparaître votre répertoire distant comme s'il faisait partie des répertoires de votre machine.

4.1 Sous Windows

Rendez-vous sur la page `https://winfsp.dev/rel/` pour y télécharger les deux installateurs de `WinFsp` et `SSHFS-Win (x64)`.



puis procédez à l'installation en double-cliquant sur les fichiers téléchargés en commençant par `WinFsp`.

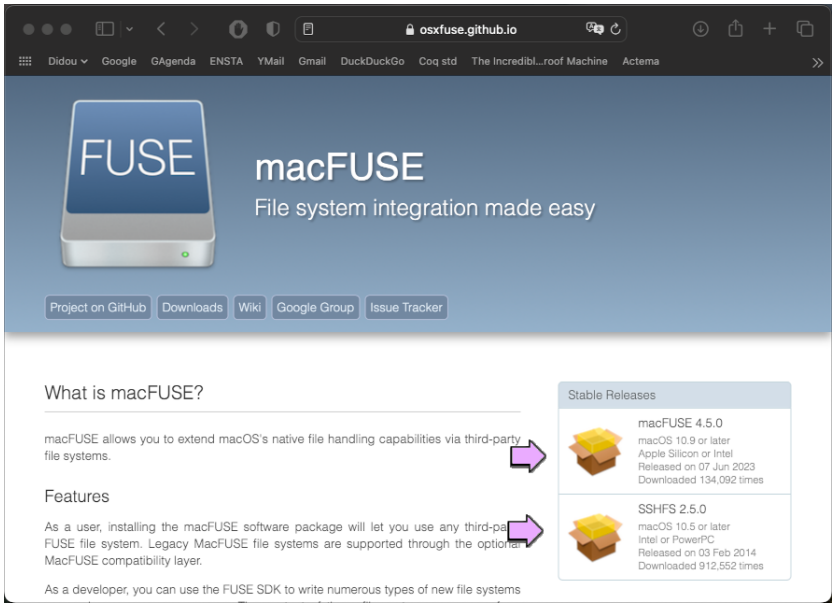
4.2 Sous Linux

Un seul paquetage est nécessaire sous Linux et peut être installé en tapant la commande suivante dans un terminal :

```
sudo apt install sshfs
```
qui vous demandera votre mot de passe.

4.3 Sous macOS

Rendez-vous sur la page `https://osxfuse.github.io` pour y télécharger les deux installateurs de `macFUSE` (4.5.0 ou plus récent) et `SSHFS` (2.5.0 ou plus récente).



puis procédez à l'installation en double-cliquant sur les fichiers téléchargés en commençant par `macFUSE`.

5 Montage du répertoire distant

Afin que votre répertoire se trouvant sur le serveur distant soit visible (on dit aussi « monté ») sur votre machine, il convient de faire une manipulation. Celle-ci sera à faire **à chaque début** de votre session de travail et rendra accessible votre répertoire jusqu'à ce que vous vous **déconnectiez** de votre machine ou que vous **l'éteigniez**. Le serveur distant qui vous permettra d'accéder à vos fichiers se nomme `salle.ensta.fr`.

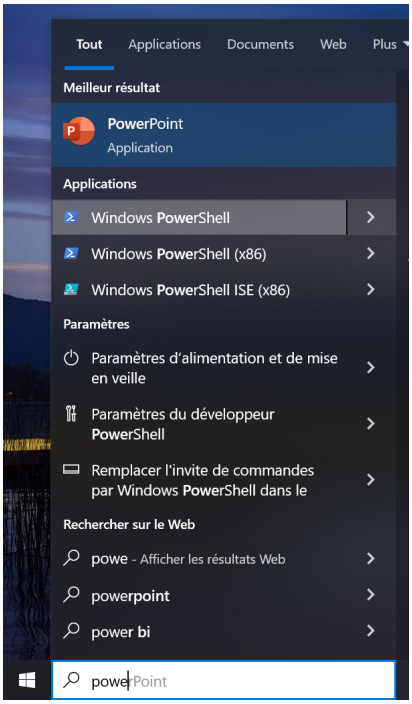
Il est possible d'automatiser cette manipulation à chaque fois que vous vous connectez à votre ordinateur mais nous le déconseillons car à chaque fois que vous seriez en dehors de l'école, le serveur `salle.ensta.fr` n'étant pas joignable, vous auriez un message d'erreur quelque peu dérangeant.

5.1 Sous Windows

Il existe deux manières de « monter » un répertoire distant : une utilisant la ligne de commande, l'autre utilisant l'interface graphique de Windows.

5.1.1 Par la ligne de commande

Comme de toute façon vous aurez besoin d'utiliser un interprète de ligne de commande, autant le voir dès maintenant... L'interprète de ligne de commande le plus évolué disponible par défaut sous Windows est le `PowerShell`. Dans la barre de recherche, commencez à taper « *powershell* » et vous verrez apparaître le programme recherché nommé « PowerShell » (pas celui suffixé par `(x86)`). Cliquez dessus pour le lancer.



Une fenêtre s'ouvre alors dans laquelle vous pouvez entrer des commandes textuelles pour interagir avec votre machine. Entrez la commande suivante suivie par un retour à la ligne.

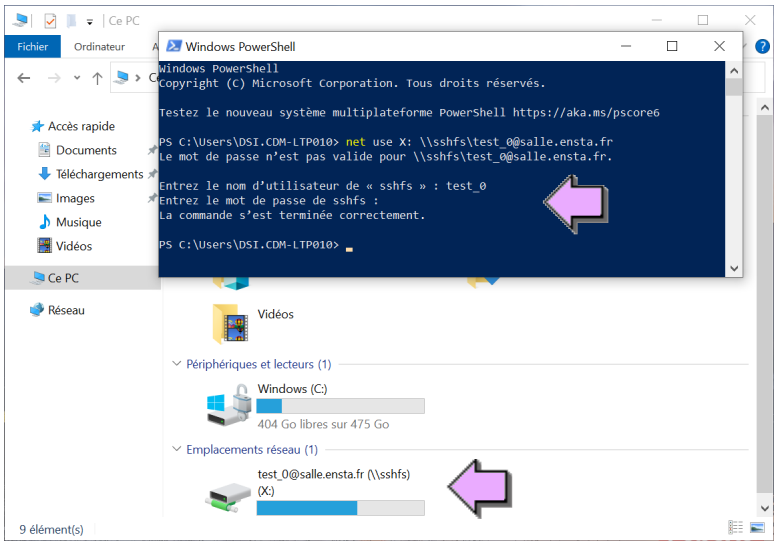
```
net use X: \\sshfs\votre identifiant@salle.ensta.fr
```

La séquence `X:` est le nom que vous souhaitez donner au « nouveau disque » qui représentera votre répertoire distant. Vous pouvez le nommer avec **la** lettre (une seule) que vous souhaitez tant que cela n'entre pas en conflit avec les noms de disques que vous avez déjà sur votre machine (généralement au moins `C:` et parfois `D:`).

La séquence *votre identifiant* fait référence à celui qui vous sert à vous connecter à Cascad, à votre mail ENSTA et plus généralement à tous les services ENSTA. Il correspond généralement plus ou moins à votre nom, éventuellement agrémenté de caractères supplémentaires pour désambiguïser les homonymes.

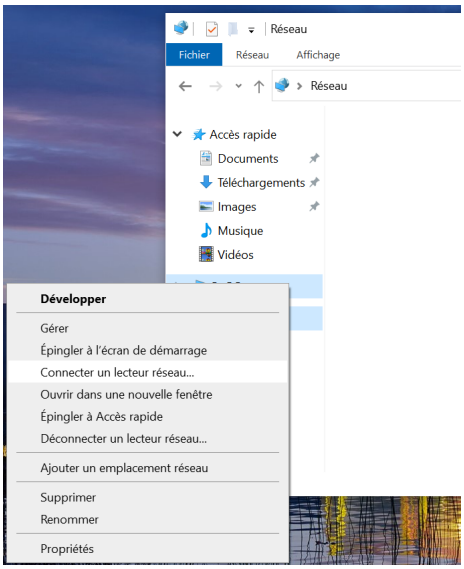
La séquence `salle.ensta.fr` représente le nom du serveur qui vous permettra **d'accéder à vos fichiers distants**. Il est différent de celui que vous utiliserez pour programmer et faire tourner vos programmes dans les cours d'informatique. Donc, oui, vous utiliserez **deux** serveurs différents en même temps.

Votre identifiant vous est alors (re) demandé puis votre mot de passe (ENSTA). Notez que lorsque vous entrez votre mot de passe, celui-ci ne s'affiche pas pour des raisons évidentes de sécurité (des yeux indiscrets regardent peut-être votre écran en même temps). Une fois ces deux informations entrées, l'exécution doit se terminer par un succès et vous devez voir apparaître un « nouveau disque » (nommé `X:` ici) dans votre explorateur de fichiers.

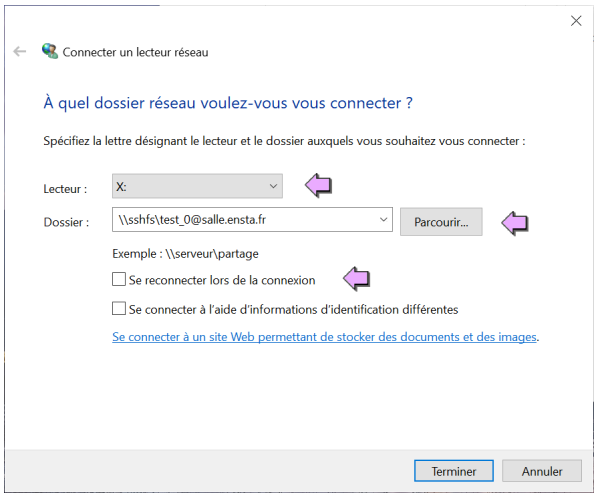


5.1.2 Par l'interface graphique

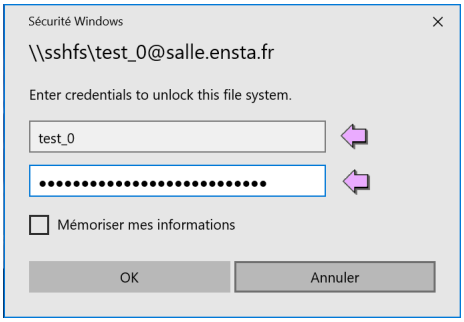
Dans un explorateur de fichier, effectuez un clic droit au-dessus de l'item `Cet ordinateur` et sélectionnez l'entrée Connecter un lecteur réseau ... du menu contextuel.



Renseignez les champs de la fenêtre apparue en indiquant le nom que vous souhaitez pour le « nouveau disque » dans le champ `Lecteur` et le nom du serveur cible (`\\sshfs\`*`votre identifiant`*`@salle.ensta.fr`) dans le champs `Dossier`, en accord avec ce qui a été expliqué dans la section précédente. Il vous est conseillé de décocher la case `Se reconnecter lors de la connexion` afin d'éviter une erreur dans le cas où le serveur ne serait pas accessible, par exemple si vous êtes en dehors de l'école (cf. section précédente).



Une fenêtre apparaît alors vous invitant à entrer votre identifiant et votre mot de passe (ENSTA). Une fois ces informations renseignées, le « nouveau disque » apparaîtra dans votre explorateur de fichiers.



5.2 Sous Linux et macOS

Avant la toute première utilisation, vous devez créer un répertoire vide qui sera celui où les fichiers distants apparaîtront. À chaque fois que vous connecterez le système de fichiers distant, ce répertoire se verra peuplé par les fichiers accessibles depuis le serveur `salle.ensta.fr`.

Nous supposons dans cette documentation que vous l'avez créé dans le répertoire `Documents` de votre espace personnel et que vous lui avez donné le nom `remote-files`.

Nous avons désormais besoin d'interagir avec un interprète de ligne de commande, que l'on appelle un « terminal ». Dans la barre de recherche, commencez à taper « *terminal* » et vous verrez apparaître le programme recherché. Sous macOS la barre de recherche s'active avec la combinaison de touches ⌘ + Space, sous Linux c'est généralement avec la touche ☰.

Dans le terminal désormais ouvert, invoquez la commande suivie d'un retour à la ligne :

```
sshfs votre identifiant@salle.ensta.fr: ~/Documents/remote-files
```

Le chemin `~/Documents/remote-files` référence le répertoire vide que vous avez créé à l'étape précédente.

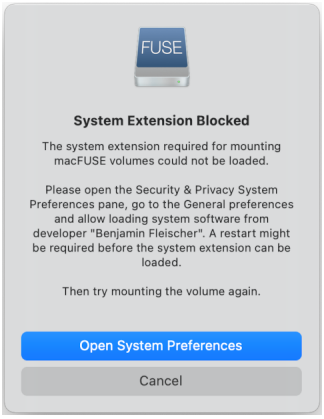
La séquence *votre identifiant* fait référence à celui qui vous sert à vous connecter à Cascad, à votre mail ENSTA et plus généralement à tous les services ENSTA. Il correspond généralement plus ou moins à votre nom, éventuellement agrémenté de caractères supplémentaires pour désambiguïser les homonymes.

La séquence `salle.ensta.fr` représente le nom du serveur qui vous permettra **d'accéder à vos fichiers distants**. Il est différent de celui que vous utiliserez pour programmer et faire tourner vos programmes dans les cours d'informatique. Donc, oui, vous utiliserez **deux** serveurs différents en même temps.

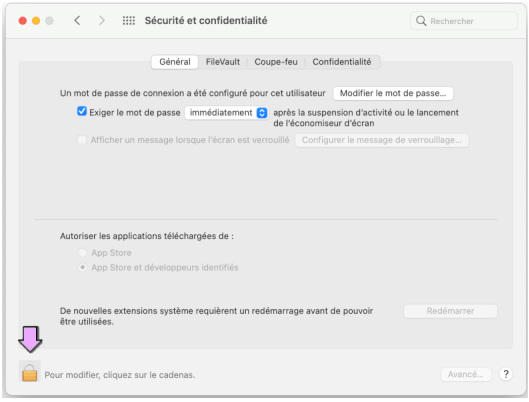
Votre mot de passe (ENSTA) vous est alors demandé pour que la commande puisse être exécutée. Notez que lorsque vous entrez votre mot de passe, celui-ci ne s'affiche pas pour des raisons évidentes de sécurité (des yeux indiscrets regardent peut-être votre écran en même temps).

5.2.1 Attention : macOS uniquement

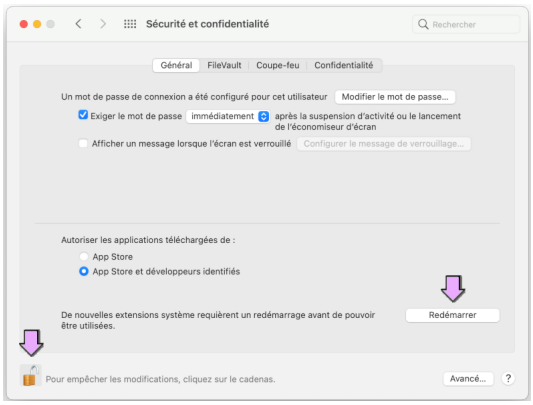
La toute première fois que vous tenterez de « monter » votre répertoire distant, le système vous avertira qu'il a bloqué l'extension :



Vous devrez l'autoriser en vous rendant dans le panneau de préférences système, dans la section `Sécurité et confidentialité` :



Par défaut les modifications sont interdites, comme indiqué par le petit symbole de cadenas fermé. Il vous faudra donc cliquer sur ce symbole pour autoriser les modifications. Le mot de passe de votre session (locale, pas ENSTA) vous sera demandé.



Cela aura pour effet d'autoriser à activer le bouton `Redémarrer` sur lequel vous devrez cliquer. Votre ordinateur va donc redémarrer et les prochaines utilisations de la commande précédemment présentée ne poseront plus d'alerte de sécurité.

5.3 Conclusion

Une fois le système de fichiers « monté », vous pouvez naviguer dans le disque ou le répertoire qui lui correspond avec n'importe quel outil pour accéder à ses fichiers. Ces fichiers sont en fait ceux situés sur le serveur distant : il ne sont pas **physiquement** sur votre machine. Si vous déconnectez le disque / répertoire alors ils disparaîtront de votre machine, mais seront **toujours présents** sur le serveur distant et vous les retrouverez lors de votre prochaine connexion.

6 Installation de Visual Studio Code

Nous allons continuer par l'installation de l'éditeur VSCode qui vous servira à éditer des fichiers texte de différentes natures. Téléchargez VSCode selon votre système d'exploitation à l'URL :

`https://code.visualstudio.com/`

puis installez-le en double-cliquant sur le fichier téléchargé et suivant les instructions. Notez que sous Linux Ubuntu, il est possible d'installer ce logiciel directement depuis le gestionnaire de paquets `Snap` en recherchant le logiciel nommé `code` : de cette façon, il sera automatiquement mis à jour (ce qui est préférable).

Un certain nombre d'extensions de VSCode vous seront utiles afin de rendre son utilisation plus agréable et plus productive :

- L'extension *C/C++ expansion pack* :
  `https://marketplace.visualstudio.com/items?itemName=ms-vscode.cpptools-extension-pack`
  ou ctrl+p puis :
  `ext install ms-vscode.cpptools-extension-pack`
- L'extension *langue française* :
  `https://marketplace.visualstudio.com/items?itemName=MS-CEINTL.vscode-language-pack-fr`
  ou ctrl+p puis :
  `ext install MS-CEINTL.vscode-language-pack-fr`

Notez que pour que le changement de langue soit appliqué, il est nécessaire de quitter puis relancer VSCode.

7 Utilisation de Visual Studio Code

L'intérêt de VSCode est qu'il offre la possibilité d'éditer des fichiers texte de différentes natures en fournissant des fonctionnalités spécifiques selon le type de texte. Cela peut consister en une mise en couleur

des mots-clés, des propositions pour continuer une « phrase » (principalement lors de l'édition de code source de programmes), la possibilité de sauter à la définition d'une entité (fonction, variable) utilisée, etc.

Entre autres dans les cours de programmation, vous serez amené à écrire des codes source puis à les « compiler » pour les transformer en programmes exécutables, puis à les exécuter. Cela nécessitera l'invocation de commandes dans un terminal (interprète de ligne de commande). VSCode permet d'intégrer au sein de sa fenêtre un tel terminal. Cela évite de jongler entre plusieurs fenêtres.

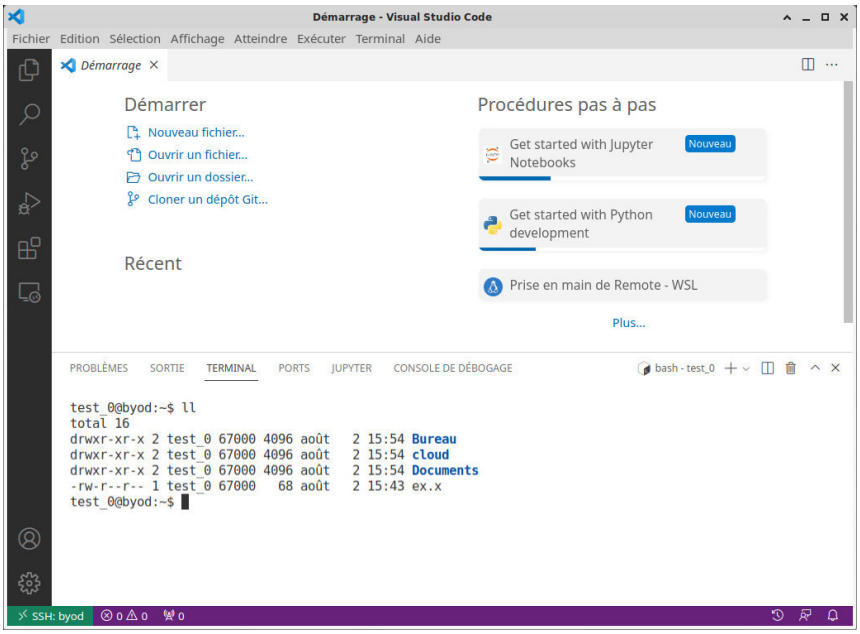
Vous pouvez ouvrir un terminal via le menu Terminal > Nouveau terminal. La fenêtre de VSCode se coupe en deux. Le terminal qui vient d'être créé tourne sur **votre** machine : il permet d'invoquer les commandes disponibles sur **votre** machine. Les outils nécessaires pour programmer se trouvent sur une autre machine : le serveur `info1.ensta.fr`. Il faut donc connecter notre terminal à ce serveur pour pouvoir les utiliser.

Pour vous connecter à `info1.ensta.fr` vous devez invoquer, dans le terminal courant, la commande suivante :

```
ssh -X votre identifiant@info1.ensta.fr
```

qui vous demande alors d'entrer votre mot de passe ENSTA. Une fois la commande exécutée, vous êtes désormais connecté sur le serveur distant. Vous pouvez donc entrer des commandes dans la partie « terminal » et ces commandes seront exécutée **sur la machine distante**. Par exemple, vous pouvez lister le contenu de votre répertoire de base en invoquant la commande :

```
ls -l
```



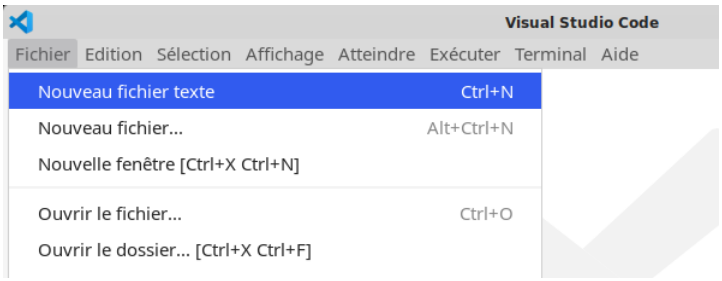
Vous remarquerez que le contenu de ce répertoire est exactement le même que celui que vous avez « monté » dans les sections précédentes et auquel vous pouvez accéder par l'explorateur de fichier de votre ordinateur.

L'intérêt d'avoir « monté » ce répertoire sur **votre** machine est de pouvoir accéder aux fichiers de manière transparente avec les outils installés sur **votre** machine, auxquels vous êtes habitués.

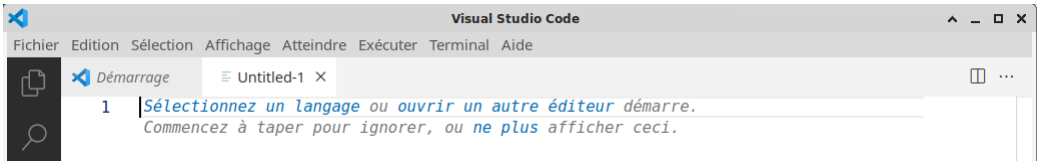
L'intérêt d'avoir ouvert un terminal connecté au serveur est de pouvoir utiliser les outils disponibles sur ce dernier (et non forcément disponibles sur votre machine) et de pouvoir faire tourner vos futurs programmes sur ce dernier.

Le lien entre les deux est que tout le monde à accès aux mêmes fichiers physiques, indépendamment de leur localisation effective.

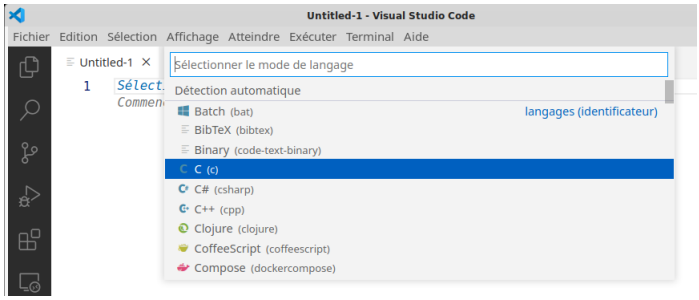
Pour créer un nouveau fichier texte (que ce soit le code source d'un futur programme ou d'un document LaTeX), utilisez le menu Fichier ⟩ Nouveau fichier texte ou le raccourci clavier ctrl + n.



Il est possible de sélectionner initialement un langage pour ce fichier. L'intérêt de spécifier le langage (i.e. le type de fichier texte) dès le début est d'activer automatiquement les extensions gérant ce type (mise en couleur des mots-clés, indentation, complétion, pré-remplissage, etc.). Si aucun langage n'est spécifié, il sera deviné au moment de la première sauvegarde du fichier, en fonction de l'extension de son nom (`.c` → langage C, `.html` → fichier Web, etc.).

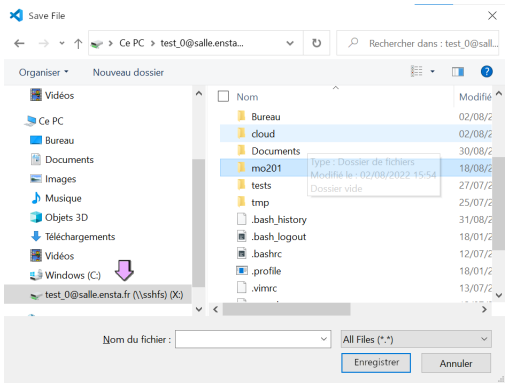


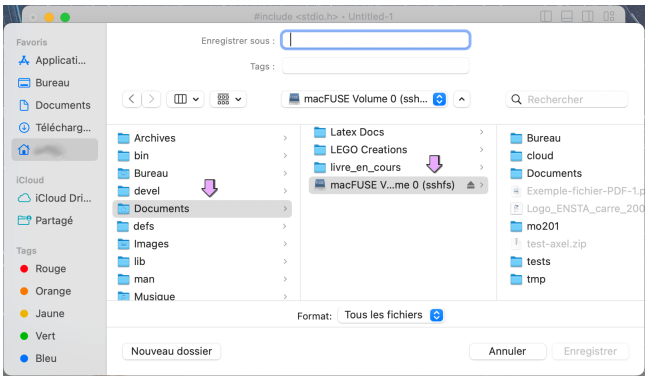
Pour l'occasion, nous choisissons le type « langage C », qui sera le langage de programmation que vous pratiquerez à l'ENSTA en première année.



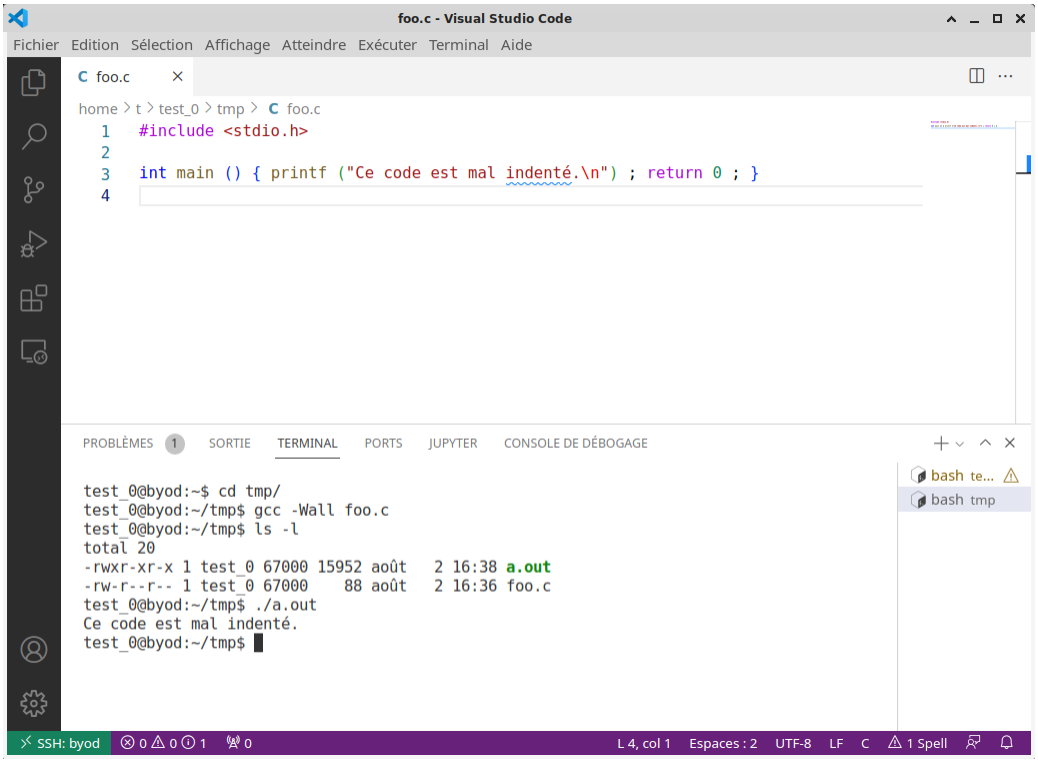
Éditez ensuite votre texte. Vous verrez alors qu'en fonction du type de fichier, VSCode met en couleur ses différents constituants en fonction de leur classe. De même il est capable de générer à la volée des squelettes de constructions du langage de programmation (si le fichier est un code source).

Pour enregistrer un fichier utilisez le menu Fichier ⟩ Enregistrer ou le raccourci clavier ctrl + s. Il est sage et judicieux d'enregistrer les modifications au fur et à mesure de l'édition, ce qui évite de tout perdre en cas de plantage de la machine. Pour que ce fichier soit stocké sur le serveur distant, sauvegardez-le sur le disque / répertoire que vous avez « monté » précédemment.

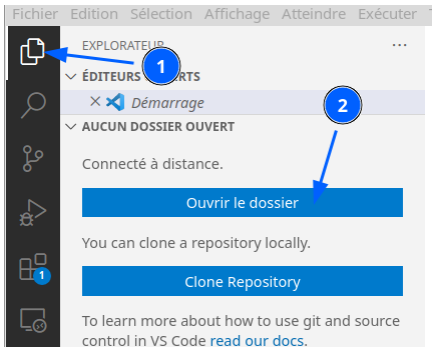




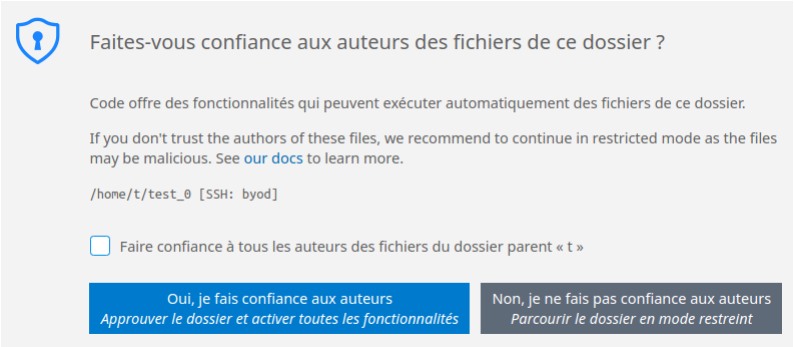
Comme expliqué précédemment, en utilisant la partie « terminal » est possible d'agir directement comme si votre machine était sous Linux, en manipulant les fichiers stockés sur le serveur distant, au moyen des commandes et logiciels installés dessus. Il est ainsi possible de compiler le petit programme écrit ci-dessus (sauvé sous le nom `foo.c`) et de l'exécuter.



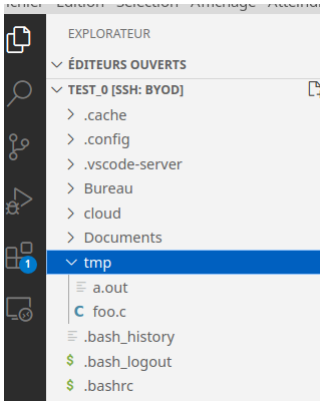
VSCode dispose d'un explorateur de fichiers intégré, permettant d'interagir graphiquement avec le contenu des répertoires (situés sur la machine locale ou sur le serveur distant puisque l'on a fait « apparaître » ses fichiers sur notre machine locale). Pour l'ouvrir, il suffit de cliquer que le bouton en forme de feuilles dans la barre en haut à gauche, puis sur le bouton *Ouvrir le dossier* de la fenêtre qui apparaît.



Un avertissement vous demande si vous faites confiance en le contenu de ce répertoire distant, ce que vous allez bien entendu confirmer (puisque ce sont vos fichiers).



Vous pouvez alors naviguer parmi les fichiers, les déplacer, les supprimer, les dupliquer, etc.



7.1 Utiliser un terminal séparé de VSCode

Sur des écrans de petite taille, couper la fenêtre de VSCode en deux pour faire cohabiter le texte édité et le terminal peut être inconfortable car il n'y a plus assez de surface pour voir suffisamment de texte dans chacune des deux parties.

Il est possible de ne pas ouvrir de terminal dans VSCode afin de garder toute la place disponible pour l'affichage du texte édité et d'utiliser un terminal externe pour se connecter au serveur `info1.ensta.fr`. On a alors deux fenêtre séparées, chacune totalement dédiée à afficher ce qui la concerne. Certes, on perd l'avantage précédemment avancé de ne pas avoir à jongler entre deux fenêtres, mais au moins elles sont désormais de taille raisonnable. Faites votre choix...

Sous Windows, lancez un `PowerShell`, sous Linux ou macOS lancez un `Terminal` et invoquez dans cette fenêtre la commande déjà vue pour se connecter au serveur distant :

```
ssh -X votre identifiant@info1.ensta.fr
```

8 Optionnel : applications graphiques

Actuellement, VSCode ne permet de lancer des commandes et logiciels depuis sa partie « terminal » que tant que ces derniers s'affichent dans le terminal. Bien sûr, il en est de même si vous utilisez un terminal externe à VSCode. Si un programme nécessite l'ouverture d'une fenêtre graphique, l'exécution échouera.

8.1 Installation d'un serveur X11 – Utilisation

Pour pouvoir lancer des applications « graphiques » depuis une machine Linux distante sous Windows ou macOS, il est nécessaire d'installer un logiciel permettant d'interpréter les affichages graphiques (*serveur X11*).

8.1.1 Sous Windows

Commencez par télécharger et installer le serveur X11 `vcxsrv` depuis l'URL ci-dessous (à l'heure où ce document est écrit, la version est 64.1.20.14.0). L'installation se lance en double-cliquant sur le fichier téléchargé.

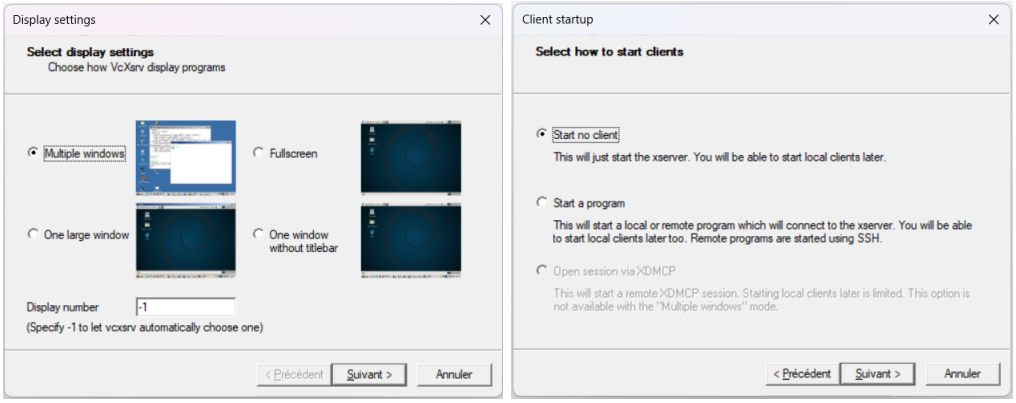
```
https://sourceforge.net/projects/vcxsrv/
```

Dès lors, lorsque vous aurez besoin, lors d'une session de travail, de lancer des programmes nécessitant un retour graphique (par exemple, un programme que vous aurez écrit et qui dessine dans une fenêtre), il faudra effectuer les actions suivantes :

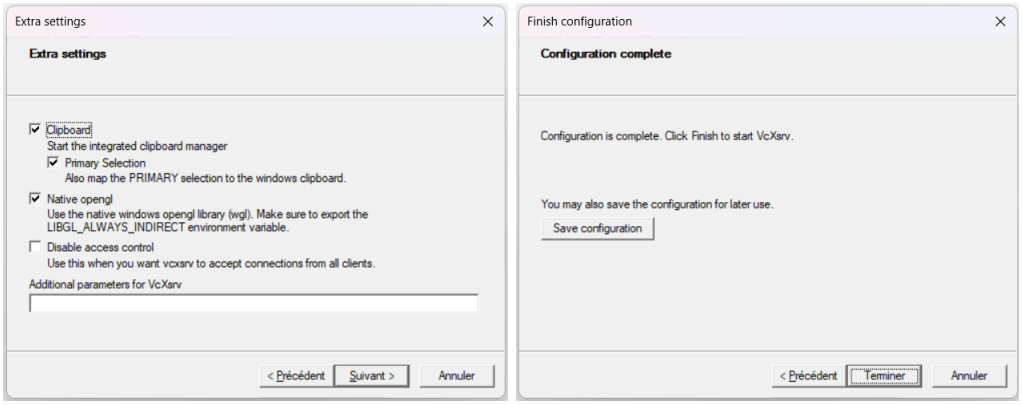
1. Lancer le serveur X11 `vcxsrv`.
2. Déterminer l'adresse réseau de votre machine.
3. Lancer VSCode et se connecter à la machine distante.
4. Dans le terminal de VSCode connecté à la machine distante, spécifier où effectuer l'affichage.
5. Après, vous pourrez travailler…

Lancer le serveur X11 Le programme à lancer sous Windows est `XLaunch` (et non `vcxsrv`). Les deux premiers écrans sont à valider sans changement.

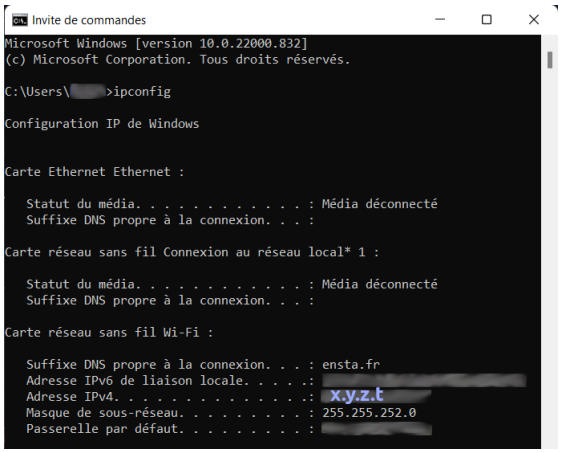
Notez que sur le premier, le champ *Display number* est par défaut à -1, ce qui permet au serveur X11 de choisir automatiquement le numéro du *DISPLAY*. Si vous le changez, vous devrez adapter (dans la dernière section) la valeur décrivant où effectuer l'affichage du côté de la machine distante. Pour moins de complications, laissez-le à -1, ce qui fera que le *DISPLAY* utilisé sera celui numéroté 0.



Sur le troisième écran, *normalement* aucune modification n'est nécessaire. Si d'aventure vous rencontriez des erreurs d'affichage, tentez de cocher la case *Disable access control*. Puis continuez pour terminer le lancement du serveur X11.



Déterminer l'adresse réseau Lancez un `PowerShell` (procédure vue précédemment) et entrez la commande `ipconfig`. Dans le terminal s'affiche un certain nombre d'informations dont l'adresse réseau (*Adresse IPv4*) de la machine. Mémorisez cette adresse (qui a la forme de 4 nombres séparés par des points, remplacés dans l'image par x, y, z, t).

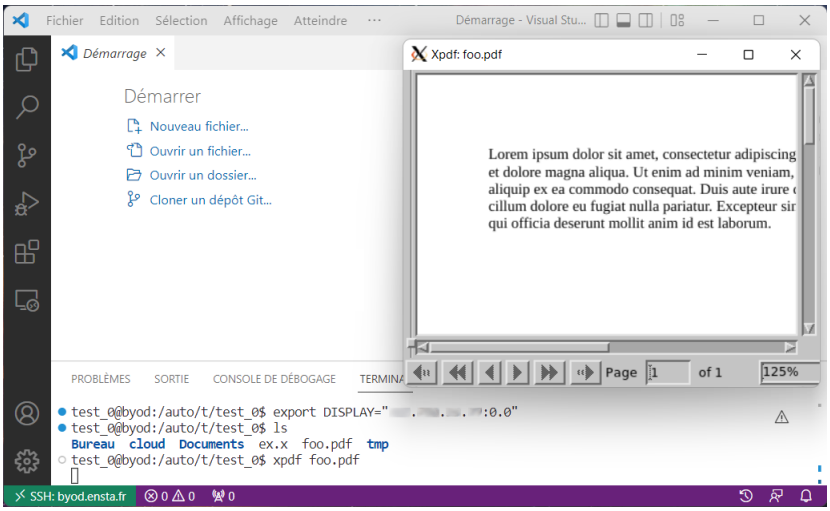


Cette adresse est susceptible de changer d'une session d'ouverture de votre machine à une autre si vous êtes connecté au WIFI. De même, ce ne sera pas la même entre une connexion par WIFI et une connexion par câble Ethernet.

Spécifier où effectuer l'affichage Après s'être connecté à la machine distante en ouvrant un terminal (dans VSCode ou séparément) avec la commande `ssh -X...`, entrez la commande suivante :

`export DISPLAY="`$x.y.z.t$`:0.0"`

où $x.y.z.t$ est l'adresse réseau identifiée ci-dessus.



Vous pourrez alors utiliser les programmes utilisant les fenêtres tournant sur la machine distante (exemple ici, ouverture d'un PDF). La configuration est conservée durant toute l'existence du terminal. Si vous le fermez, il faudra ressaisir la commande `export` ci-dessus. Comme indiqué précédemment, si vous n'avez pas changé de mode de connexion ni redémarré votre machine, l'adresse réseau n'aura pas changé.

Note Nous avons indiqué, lors du lancement de `vcxsrv`, que la valeur *Display number* sur le premier écran était par défaut à -1, ce qui permettant d'utiliser le *DISPLAY* 0 par défaut. Si vous avez décidé de la changer par, disons, 42, alors la commande `export` à invoquer devra prendre la forme :

`export DISPLAY="`$x.y.z.t$`:42.0"`

8.1.2 Sous macOS

Téléchargez puis installez le serveur X11 `xquartz` depuis l'URL ci-dessous. L'installation se lance en double-cliquant sur le fichier téléchargé.

`https://www.xquartz.org/`

Malheureusement, à l'issue de cette installation, deux cas peuvent se présenter.

- Soit tout fonctionne sans aucune autre configuration (cas vécu à l'époque de la rédaction initiale de ce document). Dans ce cas, aucune autre configuration n'est nécessaire, le serveur X11 se lancera automatiquement dès qu'une fenêtre devra être ouverte.
- Soit ça ne fonctionne pas et il faudra utiliser une autre méthode de connexion.

Pour savoir si la seule installation de `xquartz` a suffit, il suffit de se connecter à la machine `info1.ensta.fr` en ouvrant un terminal (sous VSCode ou séparément) en utilisant la commande `ssh -X...` et de tenter de lancer une application graphique (par exemple `xpdf`). Si l'application ouvre bien sa fenêtre, vous êtes dans le cas chanceux. Dans le cas contraire, lisez la suite.

Si ça n'a pas marché En attendant de trouver la raison à ce comportement et une solution simple à utiliser, il vous est proposé de vous connecter (juste pour lancer les applications graphiques) en utilisant le terminal de `xquartz` au lieu de celui du Mac ou celui intégré à VSCode. Dit autrement, vous continuerez à utiliser VSCode pour éditer vos fichiers, mais vous utiliserez **forcément** un terminal « externe » pour taper vos commande *shell* et en particulier lancer les applications (ou programmes compilés que vous aurez écrits) graphiques.

Une seule fois pour toutes, vous devrez autoriser certaines extensions graphiques à être utilisées lors des connexions entre votre machine et la machine distante au travers du serveur x11 `xquartz`. Ouvrez un terminal Mac et invoquez la commande :

`quartz-wm -help`

qui va vous afficher quelques lignes dont seule la dernière nous intéressera (valeur de `default`) :

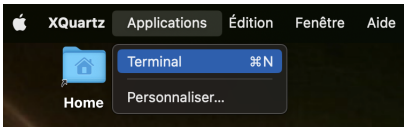
```
$ quartz-wm --help
usage: quartz-wm OPTIONS
Aqua window manager for X11.

--version                 Print the version string
--prefs-domain <domain>   Change the domain used for reading preferences
                          (default: org.xquartz.X11)
```

Dans ce même terminal, invoquez ensuite la commande :

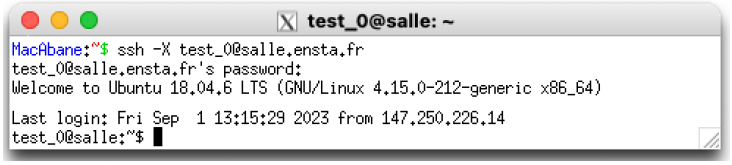
`defaults write org.xquartz.X11 enable_iglx -bool true`

en vous assurant que la chaîne `org.xquartz.X11` est bien la même que celle mentionnée dans l'affichage de la commande précédente, `quartz-wm -help`.

Désormais, à chaque session de travail, pour vous connecter à la machine `info1.ensta.fr` et pouvoir lancer des applications graphiques, vous devrez utiliser le terminal de `xquartz`. Il faudra donc lancer l'application `xquartz` (par la recherche rapide ou bien dans le répertoire du *Finder* Applications › Utilitaires › XQuartz). Une fois l'application lancée, il faudra ouvrir un terminal en allant dans son menu Applications › Terminal.



Un terminal va donc apparaître dans lequel vous pourrez vous connecter en spécifiant la machine `info1.ensta.fr`.



Depuis ce terminal vous pouvez désormais invoquer toute application nécessitant l'affichage de fenêtres graphiques.