

Installation d'OCaml sous Windows avec WSL

Le but de ce tutoriel est de présenter la procédure d'installation d'OCaml via Windows Subsystem for Linux (WSL). WSL est une couche de virtualisation qui simule un noyau et une distribution Linux complète comme une application Windows. On a donc accès à un Terminal permettant d'exécuter et compiler des exécutables au format ELF (Linux), et accès à une distribution Linux (Ubuntu par défaut).

1 Avantages et inconvénients

avantages

- accessible hors ligne
- permet l'utilisation de VSCode avec le support syntaxique pour OCaml et la complétion
- installe aussi un environnement Unix (shell, commandes de bases) permettant de développer « comme sous Linux »

inconvénients

- installation lourde (temps de téléchargement et d'installation, plus que la version native)
- occupation du disque (plus de 2Go entre le système Ubuntu de base, OCaml et VSCode)
- de nombreuses étapes à suivre
- configuration de VSCode particulière

On suppose pour ce tutoriel une version de Windows 10 64 bits, correctement mise à jour. L'installation devrait aussi fonctionner sans problème avec Windows 11 mais n'a pas été testée. Les versions inférieures de Windows ainsi que Windows sous ARM (Windows sur Microsoft Surface et certains ChromeBooks) ne sont pas supportées.

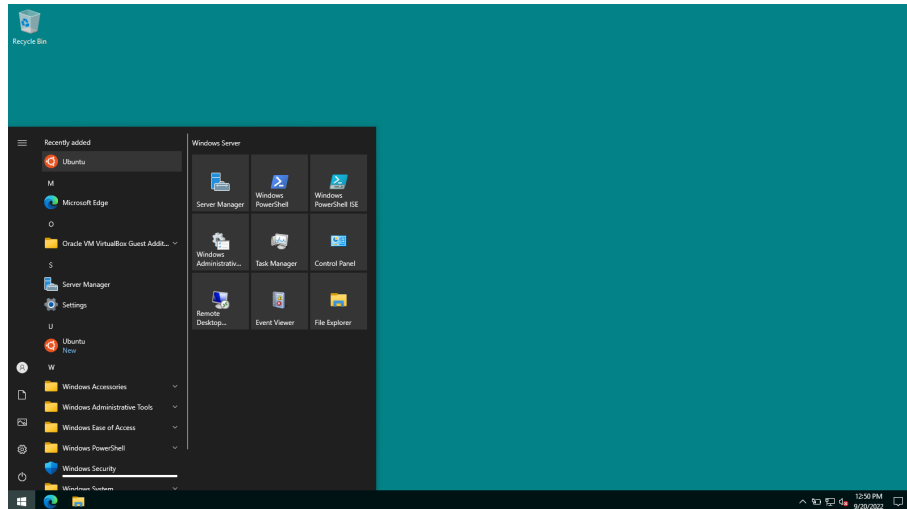
2 Installation de WSL

1. Ouvrir un invite de commande Windows ou PowerShell en mode administrateur. Traditionnellement il se trouve dans le menu « Outils Windows » ou « Windows Système ». Cliquer avec le bouton droit sur l'icône et choisir « Exécuter en tant qu'Administrateur ».

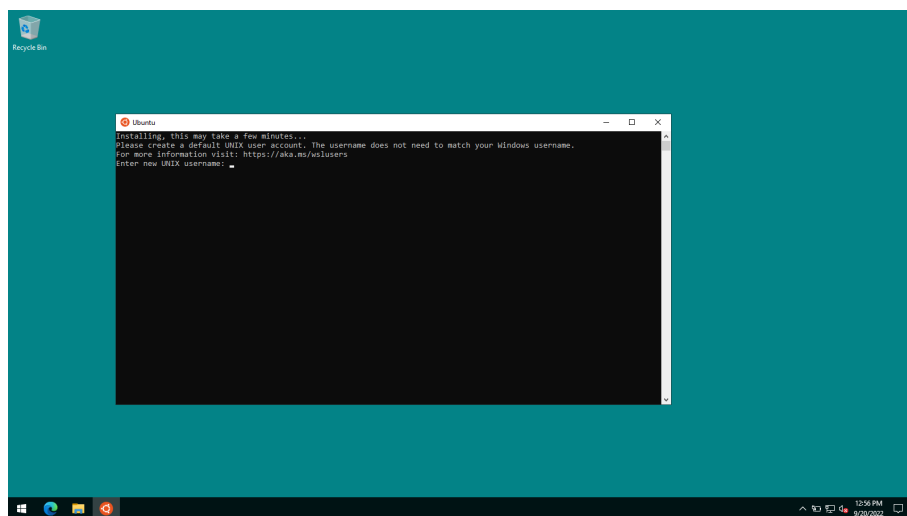
```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.20348.169]
(c) Microsoft Corporation. All rights reserved.
C:\Windows\system32>wsl --install
```

2. L'installation prend un certain temps (activation de fonctionnalité dans Windows, téléchargement). Si cette étape ne fonctionne pas, votre version de Windows est incompatible avec WSL, vous devez choisir une autre approche. Redémarrer la machine lorsque l'installateur le demande.

- Après le redémarrage, l'installation devrait se poursuivre. Une icône « Ubuntu » apparaîtra dans le menu Démarrer :

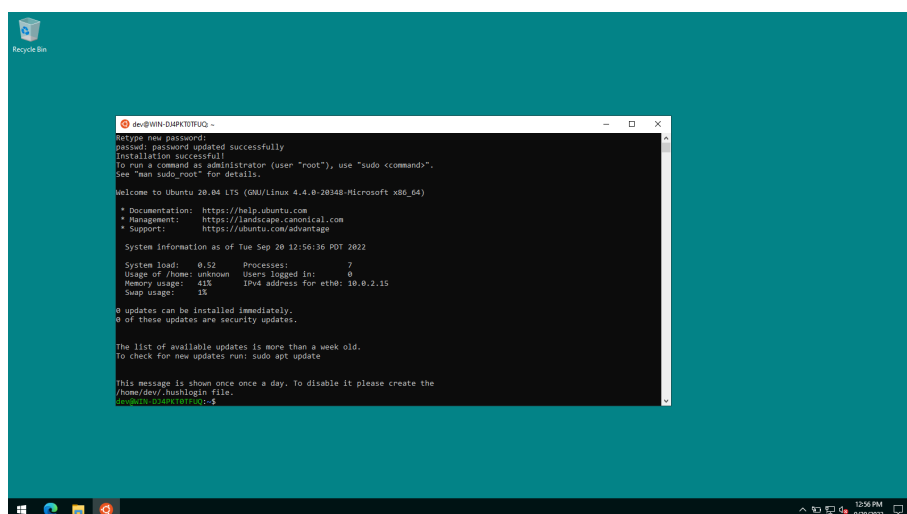


- Lancer le programme « Ubuntu ». Ce dernier démarre la couche de compatibilité lance un terminal Linux (le premier lancement peut être long).



Le processus de configuration vous demande de saisir un nom d'utilisateur et un mot de passe. Nous choisissons dev comme utilisateur dans la suite (et dev comme mot de passe). Cet utilisateur n'est pas accessible depuis l'extérieur (dans un mode normal de fonctionnement), nous avons donc choisi un mot de passe simple.

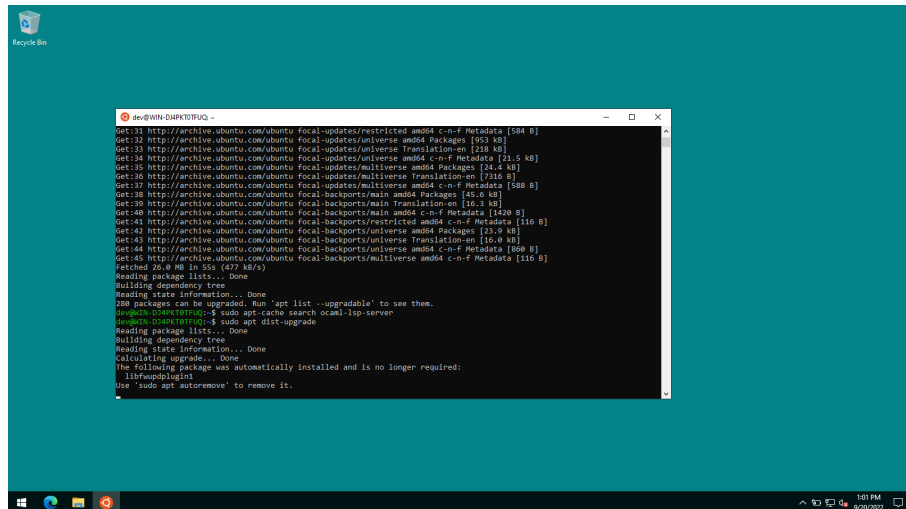
- Après cette initialisation, vous vous retrouvez avec un Terminal Linux (permettant d'accéder à une distribution Ubuntu).



- La première étape consiste à mettre à jour la distribution. Pour cela saisir la commande :

`sudo apt update`

Si un mot de passe vous est demandé, saisir celui défini lors de la création de l'utilisateur dev. La commande télécharge les listes de paquets logiciels disponibles :

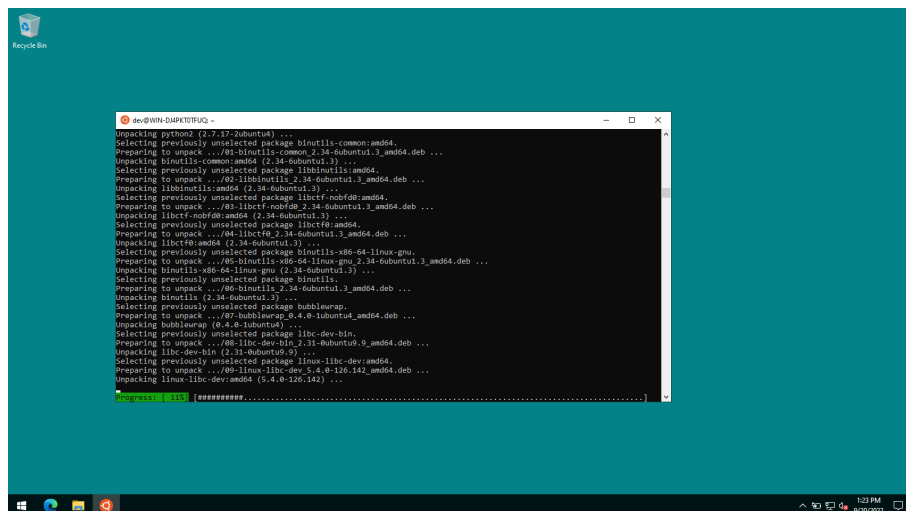


```
dev@WIN-DMPKTRTUQ:~$ sudo apt update
Get:31 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 c-n-f Metadata [584 B]
Get:32 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [293 kB]
Get:33 http://archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [219 kB]
Get:34 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [21.5 kB]
Get:35 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [24.6 kB]
Get:36 http://archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [710 B]
Get:37 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 c-n-f Metadata [580 B]
Get:38 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [45.6 kB]
Get:39 http://archive.ubuntu.com/ubuntu focal-backports/main Translation-en [16.3 kB]
Get:40 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 c-n-f Metadata [1420 B]
Get:41 http://archive.ubuntu.com/ubuntu focal-backports/restricted amd64 c-n-f Metadata [116 B]
Get:42 http://archive.ubuntu.com/ubuntu focal-backports/restricted Translation-en [21.9 kB]
Get:43 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [16.0 kB]
Get:44 http://archive.ubuntu.com/ubuntu focal-backports/universe Translation-en [16.0 kB]
Get:45 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 c-n-f Metadata [560 B]
Get:46 http://archive.ubuntu.com/ubuntu focal-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 26.0 MB in 50s (477 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
288 packages can be upgraded. Run 'apt list --upgradable' to see them.
dev@WIN-DMPKTRTUQ:~$ sudo apt-cache search ocaml.1op.server
dev@WIN-DMPKTRTUQ:~$ sudo apt dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  libsupplicant
Use 'sudo apt autoremove' to remove it.
```

Saisir ensuite la commande :

`sudo apt dist-upgrade`

Cette dernière télécharge et installe les mises à jour :



```
dev@WIN-DMPKTRTUQ:~$ sudo apt dist-upgrade
Unpacking python2 (2.7.17-2ubuntu4) ...
Selecting previously unselected package binutils-common:amd64.
Preparing to unpack .../81-binutils-common_2.34-6ubuntu1.3_amd64.deb ...
Unpacking binutils-common:amd64 (2.34-6ubuntu1.3) ...
Selecting previously unselected package libbinutils:amd64.
Preparing to unpack .../82-libbinutils_2.34-6ubuntu1.3_amd64.deb ...
Unpacking libbinutils:amd64 (2.34-6ubuntu1.3) ...
Selecting previously unselected package libc6:amd64.
Preparing to unpack .../83-libc6_2.34-6ubuntu1.3_amd64.deb ...
Unpacking libc6:amd64 (2.34-6ubuntu1.3) ...
Selecting previously unselected package libc6-dev:amd64.
Preparing to unpack .../84-libc6-dev_2.34-6ubuntu1.3_amd64.deb ...
Unpacking libc6-dev:amd64 (2.34-6ubuntu1.3) ...
Selecting previously unselected package binutils-x86-64-linux-gnu.
Preparing to unpack .../85-binutils-x86-64-linux-gnu_2.34-6ubuntu1.3_amd64.deb ...
Unpacking binutils-x86-64-linux-gnu (2.34-6ubuntu1.3) ...
Selecting previously unselected package binutils.
Preparing to unpack .../86-binutils_2.34-6ubuntu1.3_amd64.deb ...
Unpacking binutils (2.34-6ubuntu1.3) ...
Selecting previously unselected package bubblewrap.
Preparing to unpack .../87-bubblewrap_0.4.0-1ubuntu4_amd64.deb ...
Unpacking bubblewrap (0.4.0-1ubuntu4) ...
Selecting previously unselected package libc-dev-bin.
Preparing to unpack .../88-libc-dev-bin_2.31-6ubuntu9.9_amd64.deb ...
Unpacking libc-dev-bin (2.31-6ubuntu9.9) ...
Selecting previously unselected package linux-libc-dev:amd64.
Preparing to unpack .../89-linux-libc-dev_5.4.0-124.142_amd64.deb ...
Unpacking linux-libc-dev:amd64 (5.4.0-124.142) ...
Progress: [ 11%] [#####.....]
```

Une fois les mises à jour installées, installer le paquet logiciel opam permettant de gérer le compilateur OCaml et les bibliothèques OCaml

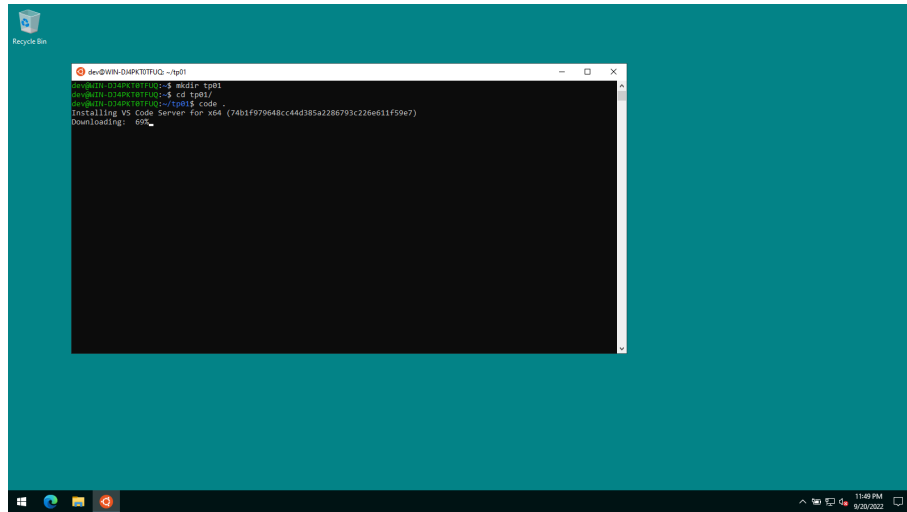
`sudo apt install opam`


3 Installation d'OPAM et extensions nécessaires

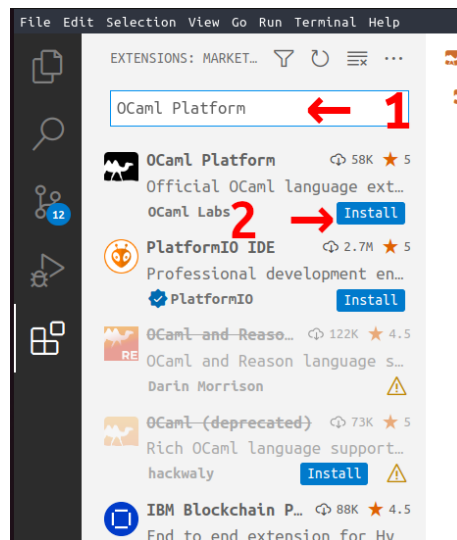
1. Ouvrir le Terminal Linux (programme « Ubuntu »), et saisir la commande :

`opam init`

Attendre le processus d'installation (un peu long) et valider les choix par Y (deux questions posées en fin d'installation)



6. Une fois VSCode lancé, retourner dans le gestionnaire d'extensions (icône ) , puis saisir le texte OCaml Platform (1) et installer le plugin correspondant (2) :



7. VSCode peut alors être utilisé pour éditer du code OCaml.

Il est obligatoire de toujours passer par le Terminal Linux pour démarrer VSCode afin qu'il puisse « voir » les fichiers stockés sous Linux

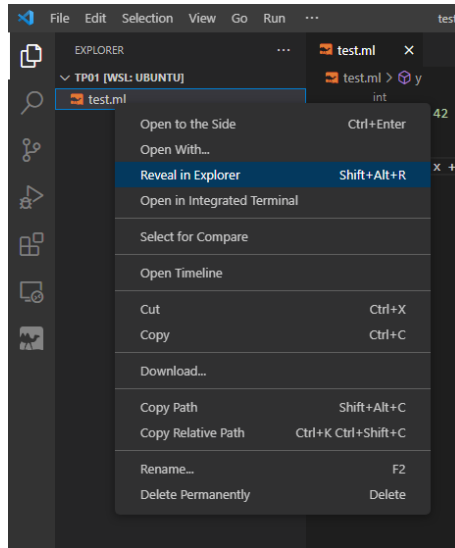
5 Utilisation de VSCode + OCaml

Le plus simple est de travailler exactement comme en TP :

- Ouvrir un Terminal Linux
- Créer un répertoire (ou se déplacer vers le dossier contenant les fichier à éditer)
- Lancer VScode avec la commande « `code .` » (ne pas oublier le point).

Les fichiers créés dans le Terminal Linux sont récupérable à partir de l'explorateur de fichiers Windows :

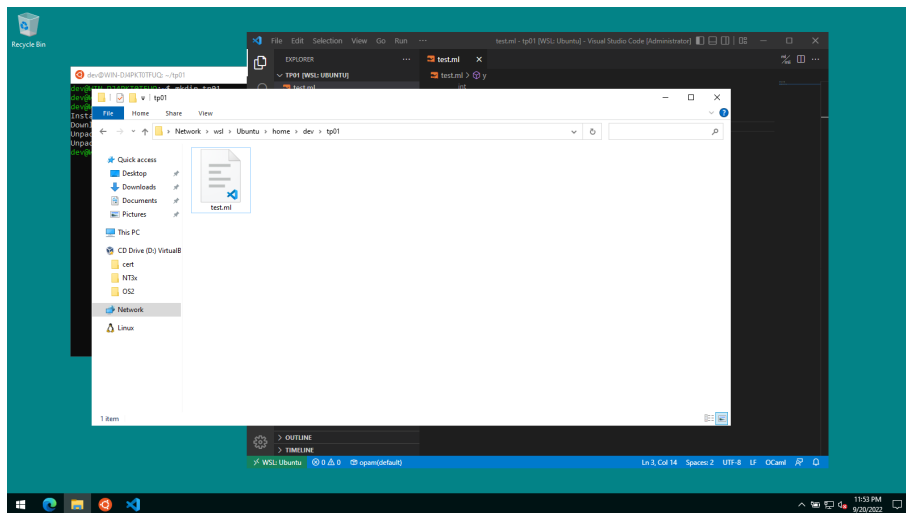
- Depuis VSCode, faire un clic droit sur le fichier puis choisir « Reveal in Explorer »



- Windows associe automatiquement un disque réseau fictif à la partition contenant l'installation Linux. L'URL Réseau de ce dernier est :

\\ws1\Ubuntu

Ce disque se manifeste par une icône de manchot disponible dans la barre de navigation de l'explorateur Windows.



Sur cette partition, le répertoire utilisateur se trouve dans le répertoire home ce qui donne dans notre exemple :

\\ws1\Ubuntu\home\dev