

Windows Subsystem for Linux as a theoretical chemist (Part 1)

Windows Subsystem for Linux as a theoretical chemist (Part 1)

- Windows Subsystem for Linux. Why ?
- Bash on Ubuntu On Windows
- The environment
- It's magic ! No it's Linux => Windows syscalls conversion.
- How to install it on your machine
 - prerequisites to install WSL
 - Install the Windows Subsystem for Linux optional feature
 - By command line
 - With PowerShell
 - With DISM
 - With the chocolatey package manager
 - By graphical user interface
 - Download distribution from the store
- Launch WSL
 - From graphical user interface
 - From command line
- Windows Linux interoperability with WSL
 - Windows drives
 - Run executable
 - Share environment variables
 - Change distribution launch behavior
- WSL vs other technologies
- Conclusion
- Useful resources
- WSL on Twitter
- Future posts

Windows Subsystem for Linux. Why ?

In my lab a lot of people use Linux the others use MacOS. It's because a lot of programs used in quantum chemistry was designed for Unix type OS. And consequently to work decently we must have a Unix type OS. We use for example SSH, python, fortran compilers and a lot of quantum chemistry software designed for Unix. For me it's because this programs are console programs and Unix type OS are so better for console usage. There are a lot of great shell like bash compared to Windows and his cmd (Even though I recognize which Microsoft has made a big effort with Windows PowerShell) but I am not convinced by graphical user interfaces present on Linux and there is also a big lack of compatibility with very common GUI programs. I experimented also drivers issues with printers and GPU.

That's how I will schematize things if I must make OS ranking.

Graphical user interface	Command line interface	Compatibility
Windows and MacOS	Linux	Windows
	MacOS	MacOS
Linux	Windows	Linux

So I understand why many of my college chose MacOS. It seems to be the best of two world (I use also MacOS on my office). And it's almost the only way for that (Without dual-boot or virtualization)... Until 2016. Since this year there are also a good way for that included in the core of Windows 10.

Bash on Ubuntu On Windows

At the build 2016 conference Kevin Galo in the head of Windows developer platform, presented a way to run native Linux binaries and Linux distributions directly on Windows 10.

At this point this technology was called Bash on Ubuntu on Windows (You can watch the video of the keynote bellow). And at this time only Ubuntu distribution was available.



02:18:20

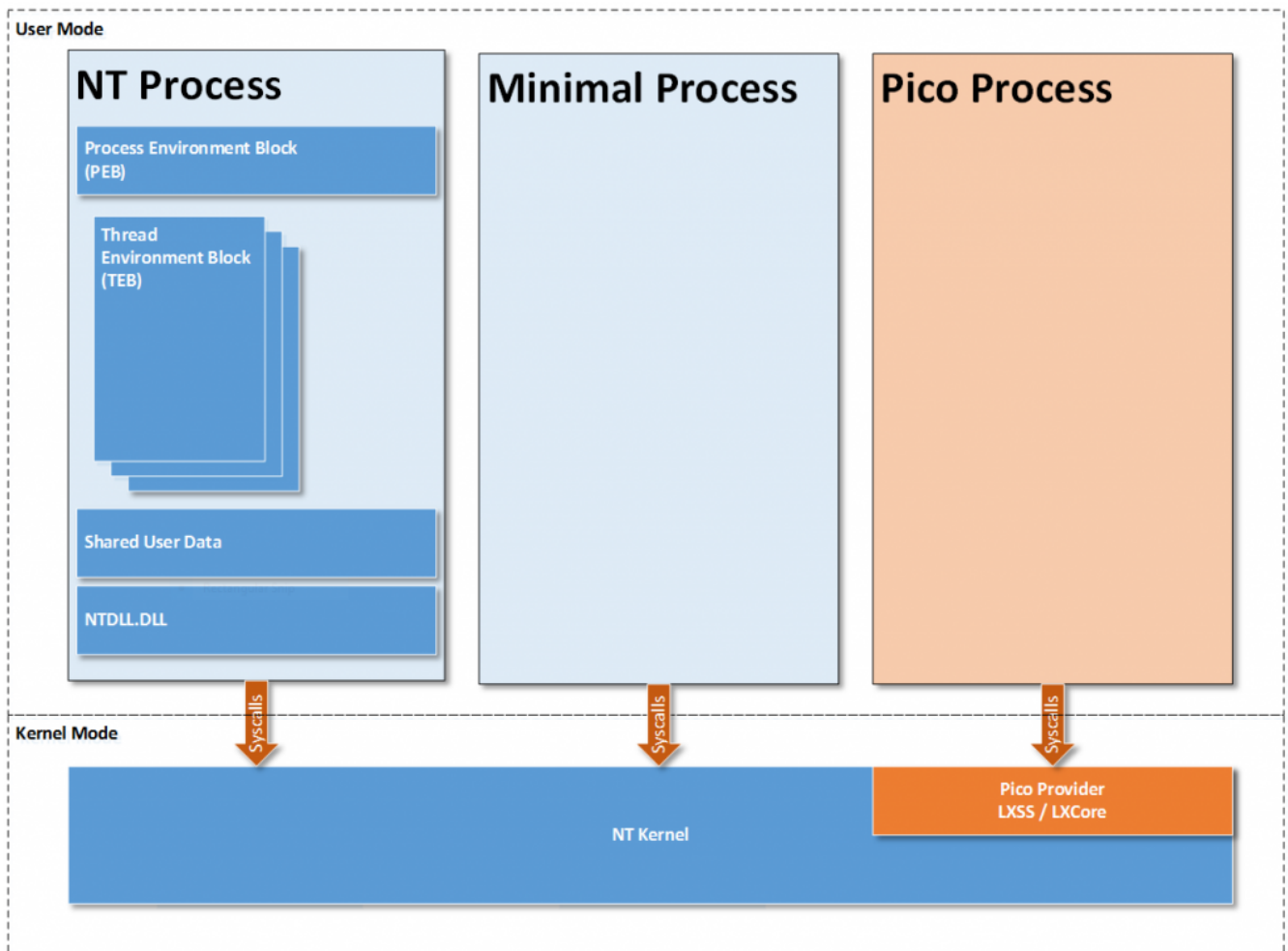
A year later this technology went out of beta with major improvements and also supports of any Linux distributions and the name "Bash on Ubuntu On Windows" was replaced by "Windows Subsystem For Linux" (a.k.a WSL).

The environment

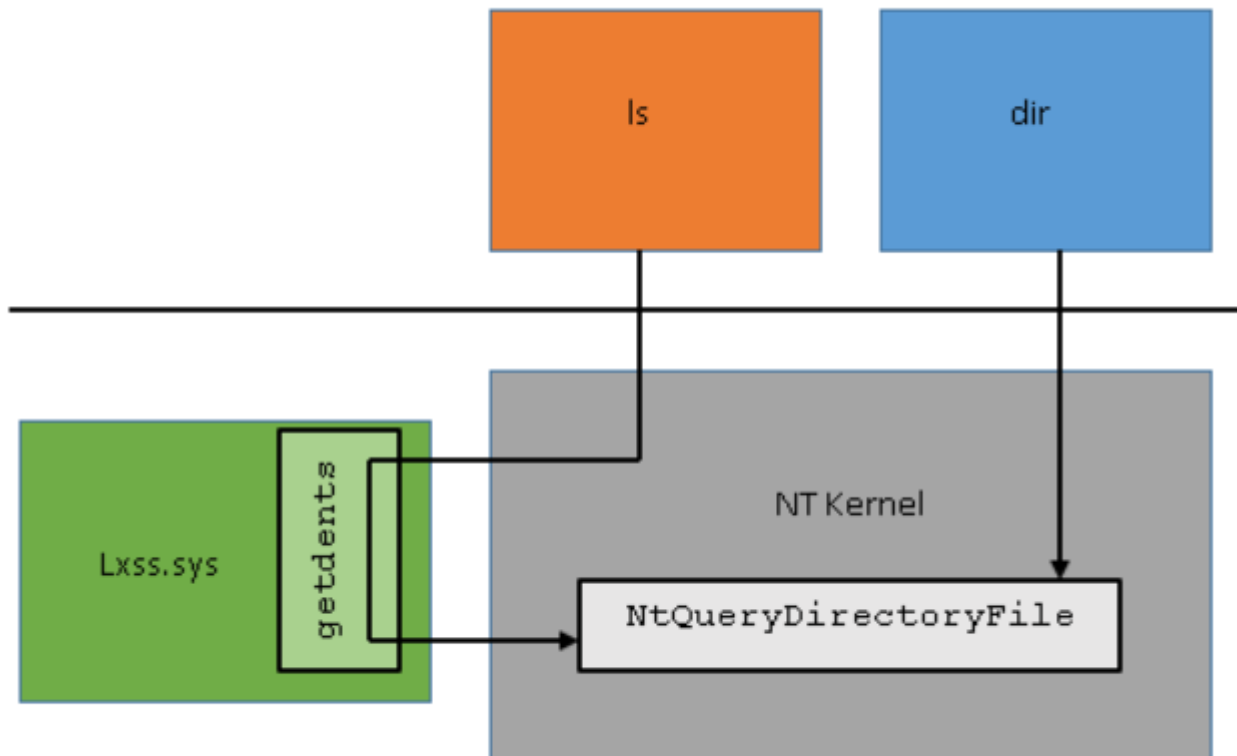
With WSL you can have a Linux command line interface with your favorite distribution and your favorite shell directly inside the Windows console host (or any alternative console available on Windows). Your Windows hard drives are automatically mounted (typically on `/mnt/<driveletter>`) and your Windows programs are also added to the path. And so you can run native Linux programs and Windows programs. inside a true Linux environment and you can interact with your Windows files in it.

It's magic ! No it's Linux => Windows syscalls conversion.

A syscall is the way that a program interact with the kernel to ask it for something (Like open a file, write on a file, ask memory allocation...) Because the binaries are Linux binaries. they attend to interact with Linux Kernel. So how does it work ? This is thanks to the magic of a technology called pico processes. Pico processes are a category of processes handled by Windows where we have a minimal process. i.e. no Windows dll has loaded, also a pico provider which is a kernel-mode driver is placed between the Windows kernel and the pico process. It provide the interface between the kernel and the pico process. So the WSL pico provider (`lxss.sys`) provide all the kernel interface to convert Linux syscalls to Windows kernel equivalent syscalls. For a Linux process inside a Windows pico process the kernel is a true Linux kernel and the provider ensures compatibility.



[Pico process overview](#)



[syscall conversion](#)

WARNING

Because not all syscall are implemented so **There is no guarantee that a specific program works !**

A well know example is the docker daemon.

How to install it on your machine

NOTE

For the installation process I added direct links to Windows 10 settings App pages and Microsoft store pages when its relevant to simplify your installation process. So this links **work only on Windows 10**.

prerequisites to install WSL

- A 64Bits operating system (x86-64 or arm64)
- Windows 10 fall creator update or greater

You can check your system configuration in the [About](#) page of the settings App. We must have the above configuration.

System type	Windows 10 version
64 Bits operating system	1709 or above

If you have a 64Bits OS but an older version of Windows 10 you must update your OS. Even if you have a compatible version, to have a more stable WSL and all the new features it's recommended to use the last version of Windows 10. You can use [Windows Update](#) from the Windows Settings app to update your Windows 10 version for free.

```
{% include hyperlink-button.html content="Check for update now" url="ms-settings:windowsupdate-action"%}
```

Install the Windows Subsystem for Linux optional feature

WSL is distributed as an optional feature of windows 10. You can install it by different and ways. each following section present a way to install WSL. You can choose the way the more convenient way for you. It comes back to the same.

Warning

To install a Windows feature an **administrator account is required**. if you are not the administrator of your PC ask him.

By command line

Generally peoples who want to install WSL are comfortable with command line interfaces. So it's may be the simplest way for you.

With PowerShell

Open a PowerShell console as administrator and type the following command

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

Reboot the PC

[Go to next step](#)

With DISM

Open a PowerShell or cmd console as administrator and type the following command

```
Dism /online /Enable-Feature /FeatureName:Microsoft-Windows-Subsystem-Linux
```

Reboot the PC

[Go to next step](#)

With the [chocolatey](#) package manager

Open a PowerShell or cmd console as administrator and type the following command

```
choco install Microsoft-Windows-Subsystem-Linux --source windowsfeatures
```

Reboot the PC

[Go to next step](#)

By graphical user interface

You can find the Windows Subsystem for Linux as an optional feature on the control panel.

Open [Apps and features](#) from the Windows Settings app and click on **Programs and Features => Turn Windows features on or off**

TIP

To go directly to this window.

Press  + 

Write **optionalfeatures** on the run box and press enter

check **Windows Subsystem for Linux** and click on the ok button.

Follow instructions.

Download distribution from the store

Warning

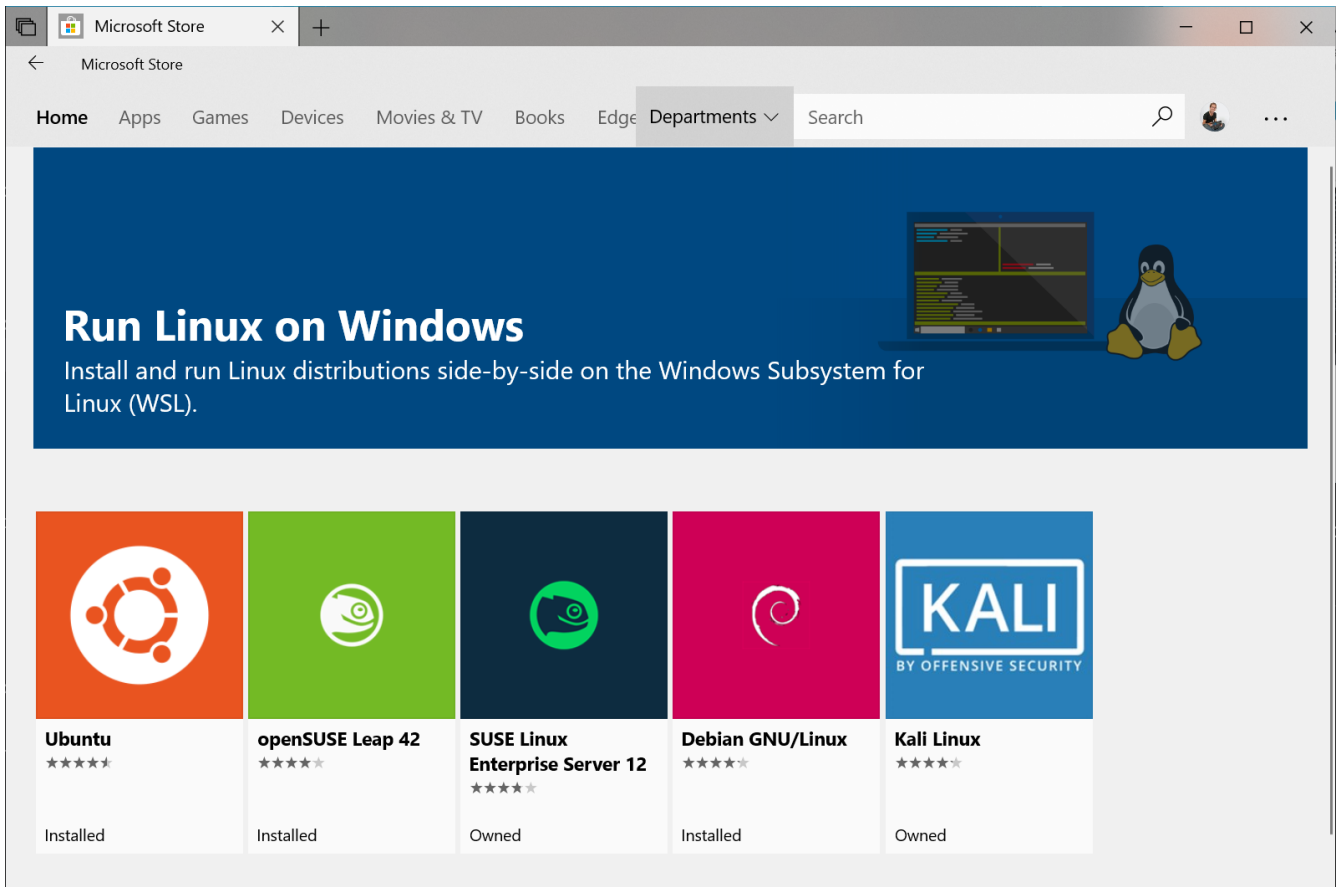
At this point the [Windows Subsystem for Linux optional feature](#) may be installed.

An administrator account is no longer required.

Now go to the [WSL page](#) on the Microsoft store to chose your distribution and download the associated app.

TIP

You can install and use multiple distros at the same time



After the app installation, launch it. The distribution was present as a tab file inside the app and at the first launch it was automatically extracted and the app asks you to create a user account. Set your Linux username and password (It can be different to your Windows user name and passwords.)

After you can play with the Linux distribution you have chosen and you can install and run programs from the Linux world in it exactly like a true Linux OS.

If you are not convinced that this is a real Linux environment you can type the following line of code on the distribution console

```
file /bin/bash
```

you can see

```
/bin/bash: ELF 64-bit [...] for GNU/Linux [...]
```

so bash is in the `Executable and Linkable Format`, the format used by Linux OS and many BSD OS for executable files and it's compiled for the Linux kernel. So bash (and the other binaries of the distribution) are the same Linux binary as true Linux installation.

you can also try

```
uname -a
```

you can see something like this

```
Linux <pc-name> 4.4.0-<windowsbuldversion>-Microsoft #1-Microsoft <windowsbuilddate> x86_64
x86_64 x86_64 GNU/Linux
```

You can see that is not a Linux generic kernel but it's considered as a Linux Microsoft kernel. this is by this way that a Linux program can see which is on WSL and not on a true Linux OS. a Linux program can also get the Windows 10 build number by this mean.

The Linux distributions distributed on the Microsoft Store are published by it's maintainers and are almost the same as the ISO distributed version (But by default without graphical programs and for Kali Linux without pentest programs) we can download all programs supported by a distribution with it packages manager because programs was downloaded on the same repositories. Of course you can also compile programs from sources, or download directly Linux binaries from the web and run it on WSL. It's also possible to compile from source on WSL and distribute the binary on the Web (**But by precaution and may be excess of caution** I recommend to test it on a true linux machine).

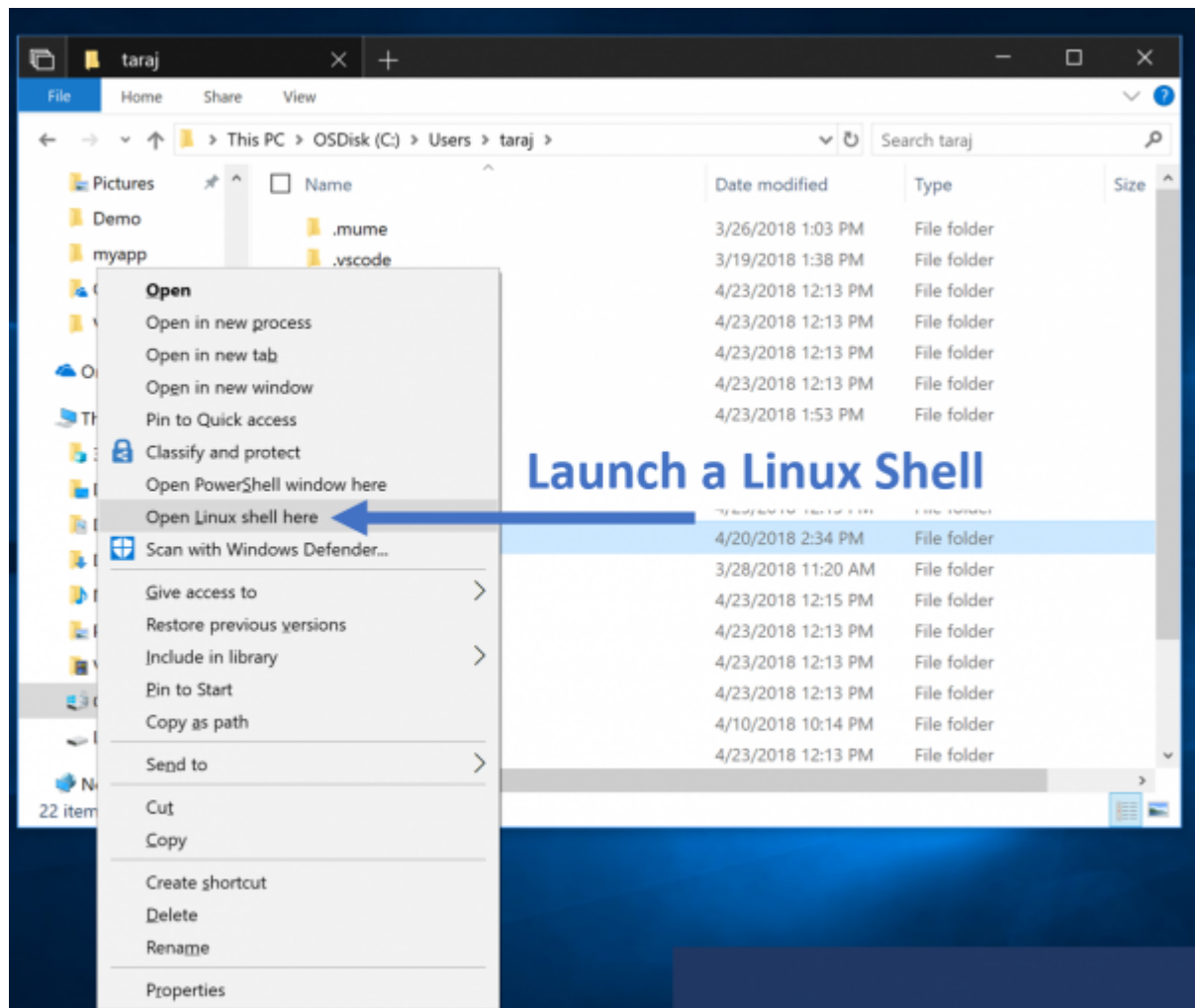
Useful resources

[Official installation guide](#)

Launch WSL

From graphical user interface

- Launch the distro app you installed previously to open a console window with this distro.
- Since October 2018 update it's also possible to launch the default distro from Windows explorer. For that press `shift` and right click. So click on `open a Linux shell here`



(http

[s://devblogs.microsoft.com/commandline/whats-new-for-wsl-in-the-windows-10-october-2018-update/](https://devblogs.microsoft.com/commandline/whats-new-for-wsl-in-the-windows-10-october-2018-update/))

From command line

To launch WSL from windows command line there are multiple possibilities

- Launch a specific distro in the Linux home directory

```
<distroname>
```

or

```
wsl ~ -d <distroname>
```

- Launch the default distro in the current directory

```
wsl
```

- Launch the default distro in the Linux home directory

```
wsl ~
```

- Launch the the default distro with the bash shell (and override the default shell) in the current directory

```
bash
```

- Launch the the default distro with the bash shell (and override the default shell) in the Linux home directory

```
bash ~
```

- Launch a specific distro in the current directory

```
wsl -d <distroname>
```

For more info about how to run WSL and learn how to configure the default distro (If we have multiple distros installed) check the documentation [here](#)

Windows Linux interoperability with WSL

Windows drives

- You have access to your Windows fixed drives by default (typically on /mnt)
- You can mount other Windows drive like USB key with the Linux classical mount command like this

```
sudo mount -t drvfs F: /mnt/f
```

TIP

To mount a Windows drive and set owner and group to your user and your main group run :

```
sudo mount -t drvfs F: /mnt/f -o uid=$(id -u),gid=$(id -g)
```

A lot of option are available

More info [here](#)

Run executable

- You can run Windows software from WSL if there is on the Windows Pat like this (The name of the Windows executable ending with .exe)

```
cmd.exe
```

- You can run Linux command line from Windows console like this

```
wsl git clone https://github.com/username/repo
```

or like this

```
<disroname> run git clone https://github.com/username/repo
```

[See the official guide to invoking WSL](#)

- You can also combine Windows and Linux command like this For example to print Windows version with Linux lolcat (Rainbow colors) in cmd you can type

```
ver | wsl lolcat
```

More information [here](#)

Share environment variables

You can share environment variable between Linux and Windows. more info [here](#).

So with a good configuration you can do this:

```
cd $USERPROFILE
```

And go to your Windows user directory.

And more interop and other WSL features comes twice per year with each new version of Windows 10.

Change distribution launch behavior

You can configure the wsl functionality of a distro in the in the `/etc/wsl.conf` file

More info [here](#)

WSL vs other technologies

An overview of WSL and its alternative I have tested on windows. I have not really use Cygwin in real situations consequently I have no idea of it's performances.

	WSL	Dual-Boot	Virtual machine (VM)	CyGwin
Binaries	Native linux binaries	Native linux binaries	Native linux binaries	Windows Binaries
Windows<=>Linux interop	Very good interop	No interop at all	Network and file sharing possible	Very good interop because it's Windows binaries
Linux binaries recompilation	Not required	Not required	Not required	Required
Performance	Bad file IO performance..	The best performances	Bad performance because hardware has emulated and we can give just a part of the machine to the VM	?
Reboot required	No	Yes	No	No
Official Linux distribution	Yes	Yes	Yes	No
Syscall handler and handling	lxss.sys (a Linux compatibility layer driver in kernel mode). All Syscall are not implemented. No GPU access. (Can be evolved)	The true linux Kernel. So all syscall are handled.	The true linux Kernel. So all syscall are handled.	Cygwin1.dll (A POSIX compatibility layer dynamic library in User mod) and the Windows kernel

Conclusion

Even if a bare-metal Linux install (on the hard disk) is always the best choice for those who want to run only Linux binaries. For those who dislike Linux GUI or have drivers problem on Linux; but want a Linux command line interface or want to run Linux binaries. WSL is probably an excellent choice. So for Windows users WSL is therefore a very good alternative to switch to MacOS which is also a very good choice because even in details its structure differ from Linux, like him, MacOS is a Unix-like OS and consequently the majority of Linux software are MacOS compatible.

Useful resources

[Official documentation](#)

[Learn About Windows Console & Windows Subsystem For Linux \(WSL\)](#)

[WSL in Windows command line blog](#)

Please read the documentation above before ask me questions !! Thanks.

If you think that this blog-post can be improved please post a comment below.

WSL on Twitter

[@craigaloewen](#) Program Manager of WSL

[@richturn_ms](#) Senior Program Manager of the Windows console

[#WSLinux](#)

Future posts

In the first part of this series of post we

Here we seen why we can want to use WSL. How does it work, how to install it, how to use Interop functionality and comparison with other alternatives.

In futures post we will see.

- A tweak on WSL to work with an X server for local Linux programs and remote SSH server too.
- A practical example of running a quantum chemistry software in WSL and a tips to prevent your antivirus from using a large portion of CPU.