LINUX

What is the windows Subsystem for Linux?

The Windows Subsystem for Linux (WSL) is a compatibility layer developed by Microsoft that allows you to run a full Linux distribution natively on Windows. It enables users to run Linux command-line tools, utilities, and applications alongside their Windows applications, without the need for a virtual machine or dual-boot setup.

Key Features of WSL:

Linux Kernel Integration: WSL enables the running of Linux distributions like Ubuntu, Debian, Fedora, and more directly on Windows, using a compatible Linux kernel.

Full Linux Environment: You can install and run a full Linux distribution (e.g., Ubuntu or Kali Linux) with access to Linux tools and utilities such as bash, ssh, apt, and other package managers.

Interoperability: WSL allows for seamless integration between Windows and Linux. You can access Windows files from within the Linux environment and vice versa. This means you can open and edit Windows files from within a Linux terminal and call Windows applications from Linux.

No Need for Dual Boot or Virtual Machines: WSL avoids the complexity of dual-boot setups or virtual machines. It runs directly on Windows, making it lightweight and efficient.

Windows 10/11 and Server Support: WSL is available on both Windows 10 (build 16215 and later) and Windows Server 2019 and newer. It is also fully supported in Windows 11.

Key Versions of WSL:

WSL 1: The original version of WSL, which translates Linux system calls into Windows system calls. It was a more lightweight and efficient solution but didn't offer full Linux kernel capabilities.

WSL 2: A newer version of WSL that includes a full Linux kernel running in a lightweight virtual machine (VM). WSL 2 provides better compatibility, performance, and support for Docker and other Linux-specific tools. It allows you to run Linux programs and services just like on a native Linux machine.

Example Use Cases of WSL:

Development Environment: Developers can use WSL to run Linux-based tools like Git, Node.js, Python, and databases (like MySQL or PostgreSQL) without needing to leave Windows.

Running Command-Line Tools: You can run common Linux utilities such as grep, sed, awk, vim, nano, curl, etc., directly from the Windows terminal.

Running Docker on Windows: WSL 2 provides native support for Docker, enabling users to run Docker containers on Windows with a full Linux kernel.

Cross-platform Compatibility: Developers working on cross-platform applications can now easily test their software in both Windows and Linux environments without needing a separate Linux machine.

File system access in WSL

In Windows Subsystem for Linux (WSL), you can access both the Windows and Linux file systems, which enables seamless integration between your Linux environment and the native Windows environment. Here's how file system access works in WSL:

Accessing Windows Files from WSL

WSL allows you to access your Windows file system from within the Linux environment. Windows drives, such as C: and D:, are mounted under the /mnt directory in WSL. This allows you to read, write, and modify files on your Windows file system as if they were part of the Linux file system.

Example:

C Drive:

In WSL, the C: drive is accessible under /mnt/c/.

Path: /mnt/c/Users/YourUsername/Documents

To access files on your C: drive, you would navigate to /mnt/c in your WSL terminal.

For example, to list files in your C: drive's "Documents" folder:

ls /mnt/c/Users/YourUsername/Documents

D Drive (if you have one):

Similarly, the D: drive is accessible under /mnt/d/.

Path: /mnt/d/

To list files in the D: drive:

ls /mnt/d/

Accessing Linux Files from Windows

You can access files stored in the Linux file system from Windows by navigating to the WSL instance's mount point.

Example:

Accessing Linux Files in Windows:

In WSL 2, the Linux file system is stored in a virtualized file system, and the default location is:

\\wsl$\{DistroName}\

For example, if you're using Ubuntu in WSL, you would access the file system through the following path:

\\wsl$\Ubuntu\

You can use File Explorer to navigate to the Linux file system. Just open File Explorer and type \\wsl$\Ubuntu in the address bar.

Accessing Specific Files from Windows:

From Windows, you can open any file from WSL by navigating to the corresponding Linux path. For example, to open a file in ~/Documents/ from Windows, type:

\\wsl$\Ubuntu\home\yourusername\Documents

File System Considerations in WSL

File System Performance:

Accessing Linux files from Windows: Accessing files on the Linux file system (i.e., files stored under /home, /etc, etc.) is slower when accessed from Windows because it's done via a network connection.

Accessing Windows files from Linux: Accessing files on the Windows file system (e.g., /mnt/c, /mnt/d) is faster and more efficient within WSL, as it's using native file system calls.

Line Endings:

Windows and Linux handle line endings differently. Linux uses LF (\n), while Windows uses CRLF (\r\n). WSL automatically handles these differences when you read/write text files, but be mindful of them when transferring files between systems.

Permissions:

Linux has a more granular file permission system (read, write, execute for user, group, and others), while Windows has NTFS permissions.

When accessing Windows files from WSL, WSL tries to map Windows file permissions to Linux-style permissions, but this may not always be perfect. Conversely, when accessing Linux files from Windows, WSL provides files with default read and write permissions.

Symlinks:

Linux symlinks in WSL work as expected within the Linux environment. However, Windows doesn't support symbolic links to the same extent as Linux, so symlinks in Linux may not behave correctly when accessed from Windows.

Common Commands for File System Navigation in WSL

Access Windows files from WSL:

cd /mnt/c/Users/YourUsername/Desktop # Access Desktop in C: drive

Access Linux files from Windows (using File Explorer):

Open \\wsl$\Ubuntu\home\yourusername\ to browse Linux files.

View WSL file system mount points (using PowerShell or Command Prompt in Windows).

wsl --list --verbose

Example Use Cases

Working with Windows Files in WSL: If you're a developer using tools like vim or nano inside WSL, and you want to edit files stored in the C: drive, you would use:

vim /mnt/c/Users/YourUsername/Documents/myfile.txt

Accessing WSL Files from Windows Applications: If you have a Windows application like VS Code, you can directly open a Linux file from WSL by specifying the path:

code \\wsl$\Ubuntu\home\yourusername\project\file.py