

COMPILARE CODICE C

OBIETTIVI

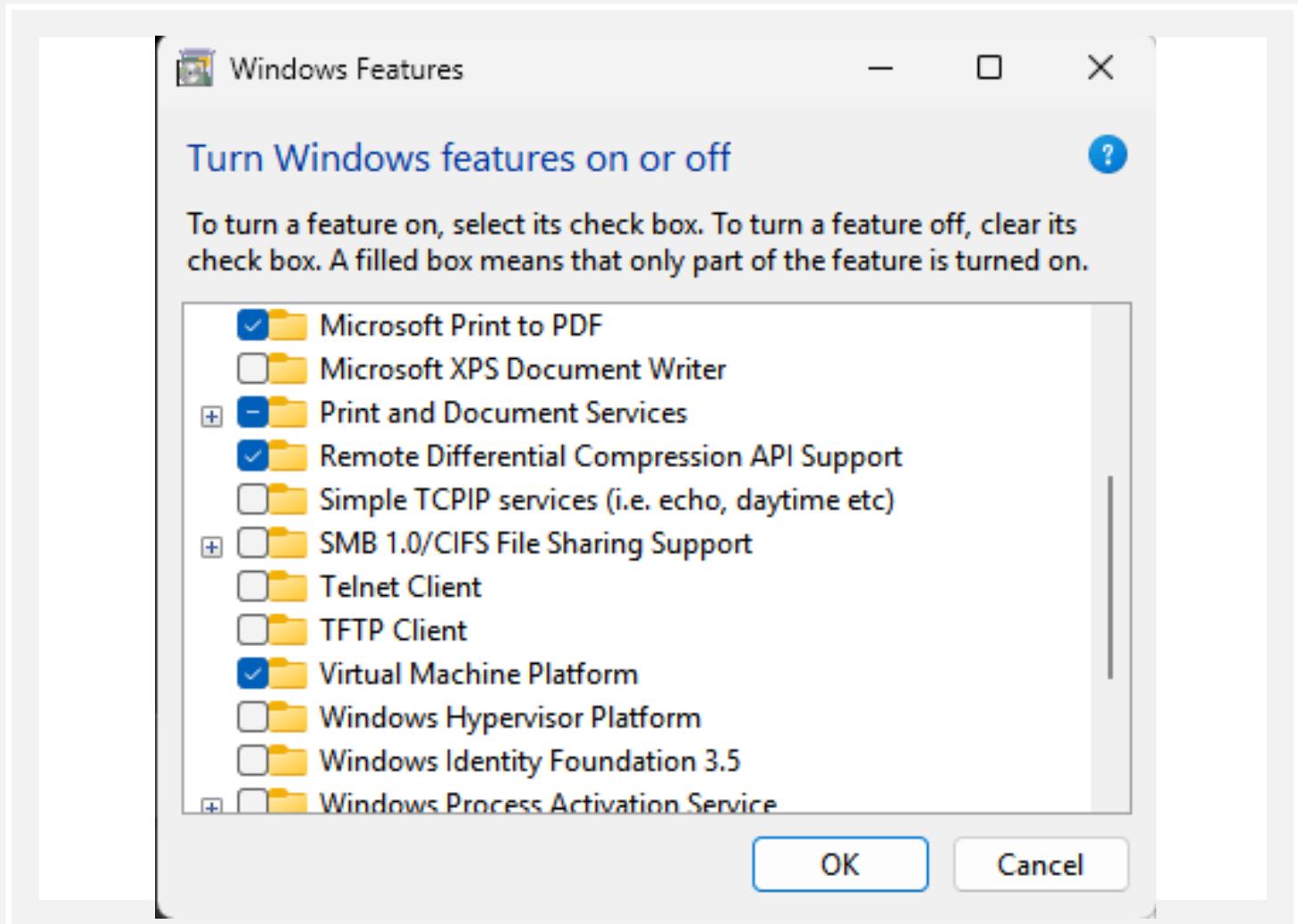
- Installare WSL
- Installare VSCode
- Eseguire codice C/C++

INTRODUZIONE A WSL E LINUX

- Ambiente Linux dentro Windows
- Unisce vantaggi di Windows e Linux
- Funziona come una macchina virtuale ottimizzata

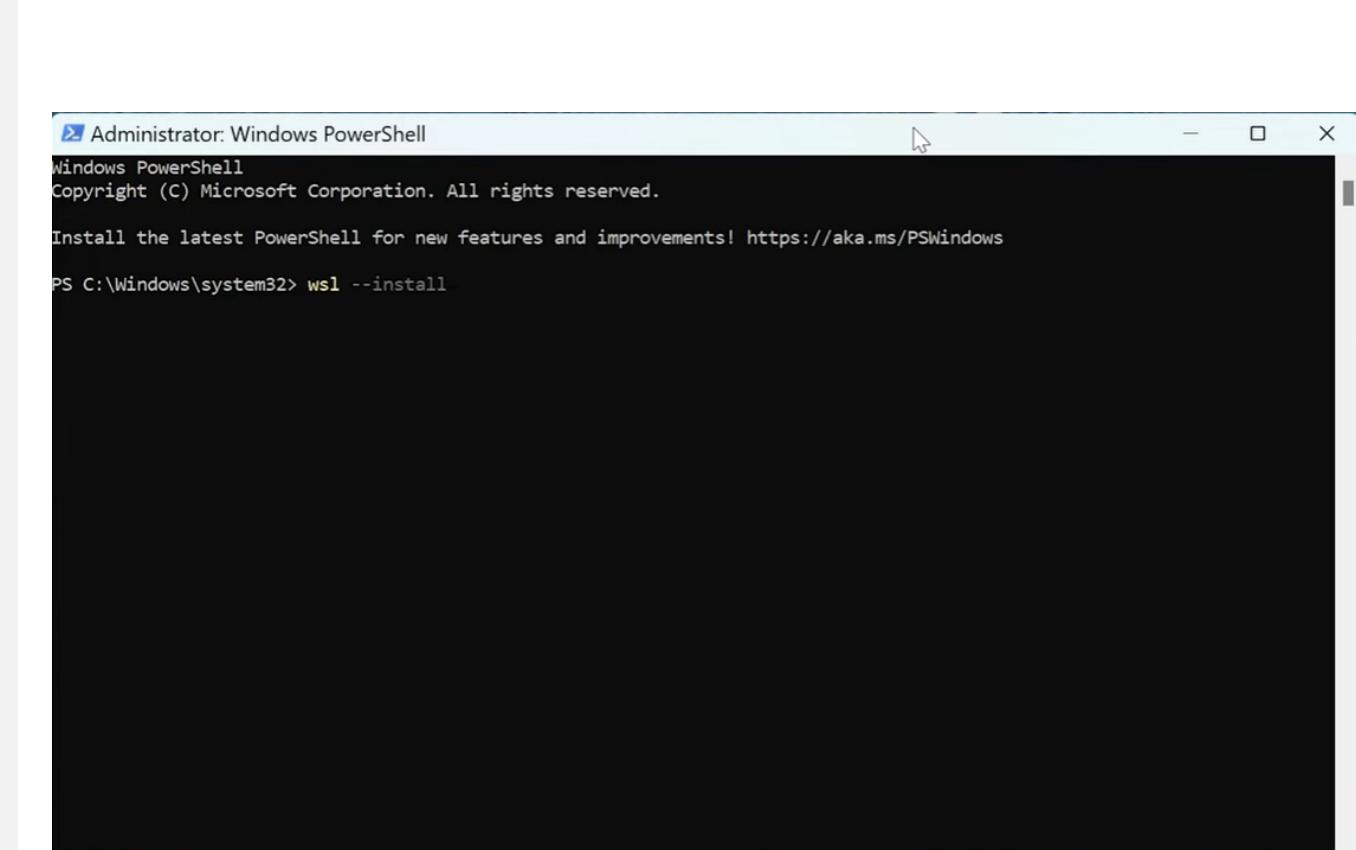
INSTALLARE WSL SU WINDOWS

- Avviare Windows Features
- Spuntare
 - “Virtual Machine Platform”



INSTALLARE WSL SU WINDOWS

- Lanciare Poweshell
(admin)
- Inserire
 - “wsl –install”



A screenshot of an "Administrator: Windows PowerShell" window. The window title bar says "Administrator: Windows PowerShell". The content area shows the following text:
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>
PS C:\Windows\system32> wsl --install

ORGANIZZAZIONE FILESYSTEM

Tramite i seguenti comandi potete organizzare il filesystem

- ‘ls’: visualizza il contenuto della directory locale
- ‘cd’: cambio di directory
 - ‘cd ..’: ritorno alla directory superiore (torno indietro)
- ‘mkdir’: crea una directory

UTILIZZO APT-GET

- Packet manager di Linux
- Prima di procedere con l'installazione del nostro setup occorre aggiornare APT-GET:

```
$ sudo apt-get update
```

SCARICARE GCC E GDB

- All'interno di WSL inseriamo questo comando

```
giacomo@GiacomosPC:~/Coding/Codici_C$ sudo apt-get install build-essential gdb
```

Vogliamo installare un pacchetto

Gestore di pacchetti

Privilegi da amministratore

Pacchetto che contiene:
- Compilatore gcc per C
- Compilatore g++ per C++

STRUMENTI INSTALLATI

- Il comando ha installato le seguenti componenti:
 - - GCC: compilatore codice C
 - - G++: compilatore C++
 - - GDB: debugger
- Check dell'installazione

PYTHON VS C

- Python:
 - Linguaggio interpretato
 - Il codice passa all'interprete e gira sulla Python Virtual Machine
- C:
 - Linguaggio compilato
 - Genera un binario ottimizzato per l'architettura

AVVIARE VS CODE DA WSL

- All'interno di WSL è possibile avviare VSCode con il seguente comando
 - “code .”

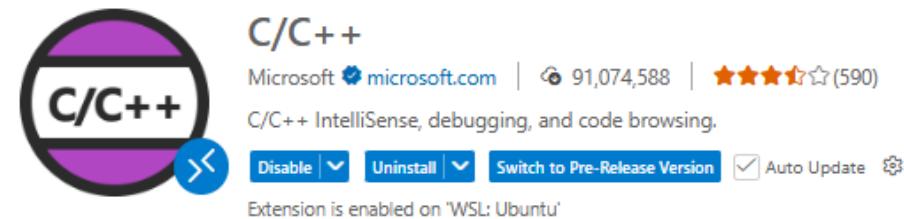


A screenshot of a terminal window titled "giacomo@GiacomosPC: ~/Co". The terminal shows the following command sequence:

```
giacomo@GiacomosPC:~$ cd Coding
giacomo@GiacomosPC:~/Coding$ mkdir Codici_C
giacomo@GiacomosPC:~/Coding$ ls
Codici_C
giacomo@GiacomosPC:~/Coding$ cd Codici_C/
giacomo@GiacomosPC:~/Coding/Codici_C$ code .
```

ESTENSIONE C

- Dentro VSCode, come fatto per Python, andiamo ad installare l'estensione necessaria.



C/C++ for Visual Studio Code

[Repository](#) | [Issues](#) | [Documentation](#) | [Code Samples](#)

Live Share enabled

The C/C++ extension adds language support for C/C++ to Visual Studio Code, including [editing \(IntelliSense\)](#) and [debugging](#) features.

Pre-requisites

C++ is a compiled language meaning your program's source code must be translated (compiled) before it can be run on your computer. VS Code is first and foremost an editor, and relies on command-line tools to do much of the development workflow. The C/C++ extension **does not include a C++ compiler or debugger**. You will need to install these tools or use those already installed on your computer.

- C++ compiler pre-installed
- C++ debugger pre-installed

Here is a list of compilers and architectures per platform officially supported by the extension. These are reflected by the available [IntelliSense modes](#) from the extension's IntelliSense configuration. Note that support for other compilers may be limited.

Platform	Compilers	Architectures
Windows	MSVC, Clang, GCC	x64, x86, arm64, arm
Linux	Clang, GCC	x64, x86, arm64, arm
macOS	Clang, GCC	x64, x86, arm64

For more information about installing the required tools or setting up the extension, please follow the tutorials below.

Overview and tutorials

- [C/C++ extension overview](#)
- [Introductory Videos](#)

PROVIAMO UN CODICE IN C

- Scriviamo un breve codice in C, come nella figura
- Per compilare il codice C invochiamo il compilatore (gcc)
- Andiamo ad eseguire il programma compilato

```
C helloworld.c > ...
1 #include <stdio.h>
2
3 int main() {
4     printf("Hello, world!\n");
5     return 0;
6 }
7
```

```
giacomo@GiacomosPC:~/tutor/codici_C$ gcc helloworld.c -o nome_che_preferite
giacomo@GiacomosPC:~/tutor/codici_C$ ./nome_che_preferite
```