

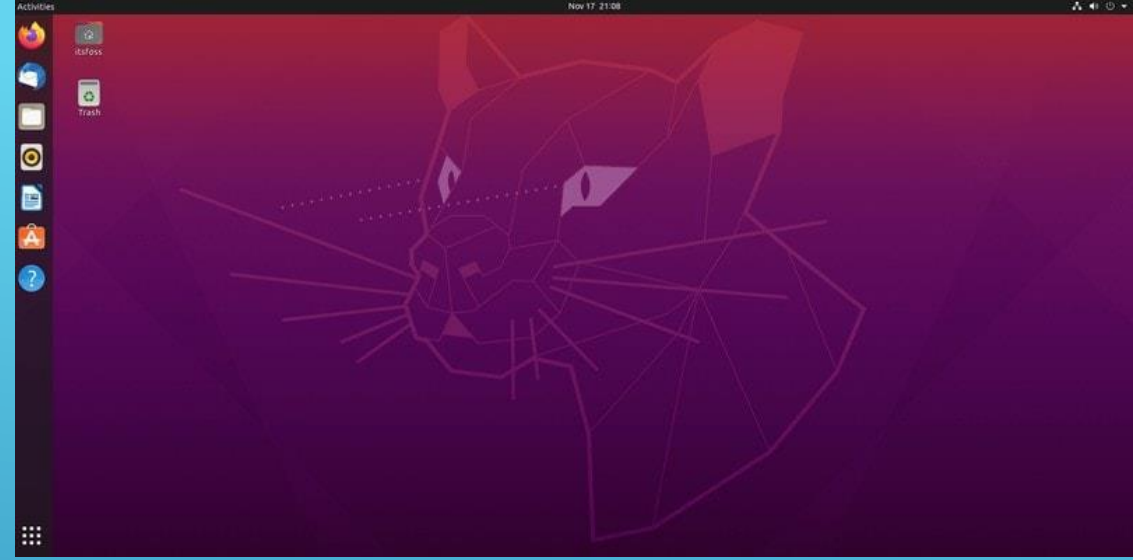
TO WRITE AND IMPLEMENT MSSR SOFTWARE:

- 1) A REAL-TIME OPERATING SYSTEM (RTOS) THAT IS SUITABLE FOR REAL-TIME SOFTWARE DEVELOPMENT.
 - 2) A DEVELOPMENT ENVIRONMENT FOR C++ PROGRAMMING LANGUAGE.
 - 3) A RADAR HARDWARE PLATFORM THAT SUPPORTS THE MSSR SYSTEM.
 - 4) A SET OF REQUIREMENTS AND SPECIFICATIONS FOR THE MSSR SYSTEM.
- **Real-time operating system (RTOS):** Some popular RTOSs that are suitable for real-time software development include **Ubuntu**, **NI Linux Real-Time**, etc. These RTOSs provide a deterministic response to external events and aim to minimize the response time guarantee.
 - **Development environment for C++ programming language:** We can use any Integrated Development Environment (IDE) that supports C++ programming language. Some popular ones include **Visual Studio Code**, **Eclipse**, and **Code::Blocks**. These IDEs provide a rich set of features such as syntax highlighting, code completion, debugging, and profiling tools.

LINUX DISTRIBUTIONS:

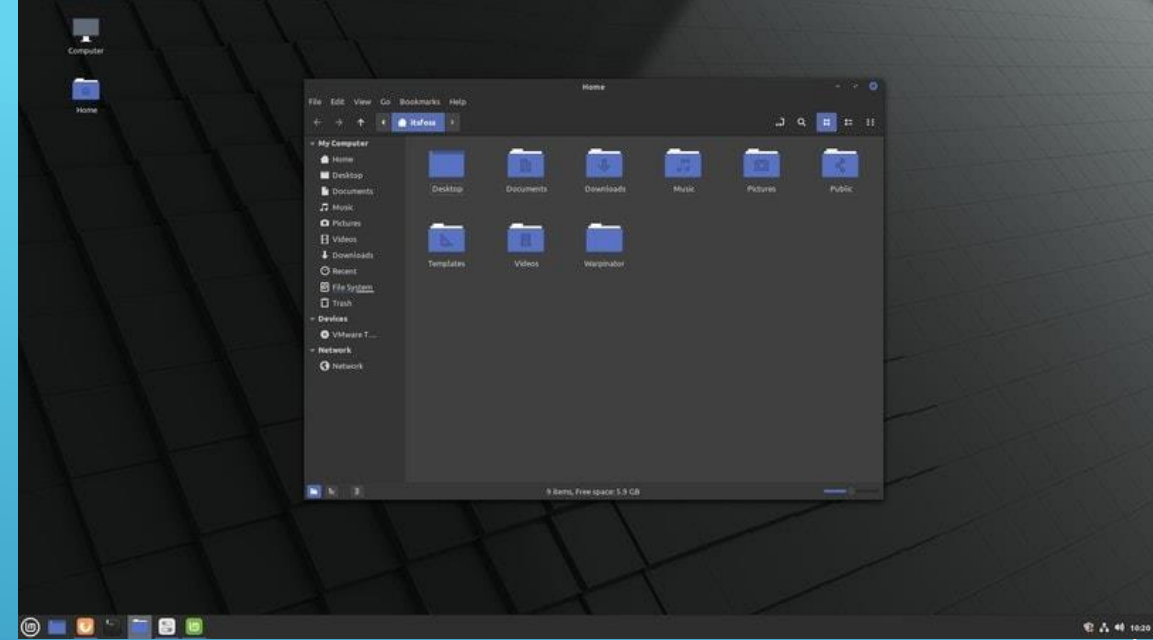
1- Ubuntu

- Free and open-source: Ubuntu is free to use and open-source.
- Easy to use: Ubuntu has a simple and easy-to-use interface that makes it easy to get started with.
- Large community: Ubuntu has a large community of users and developers who can help you if you run into any issues.
- Large number of software packages available: Ubuntu has a large number of software packages available that you can install with just a few clicks.
- Customizable: Ubuntu is highly customizable and you can customize it to suit your needs.
- Privacy and security: Ubuntu provides better privacy and security than Windows.
- User-friendly: Ubuntu is user-friendly with a high degree of customization.
- Minimal hardware requirements: Ubuntu has minimal hardware or system requirements.



2- Linux Mint

- Easy to use: Linux Mint has a simple and easy-to-use interface that makes it easy to get started with.
- Stable: Linux Mint is known for its stability and reliability.
- Large community: Linux Mint has a large community of users and developers who can help you if you run into any issues.
- Large number of software packages available: Linux Mint has a large number of software packages available that you can install with just a few clicks.
- Customizable: Linux Mint is highly customizable and you can customize it to suit your needs.
- Free: Linux Mint is free of cost and open source.
- Easy to use: Linux Mint is very comfortable and easy to use but at the same time, it is also powerful and configurable.
- Stable and reliable: Linux Mint is very stable and reliable.
- Low hardware requirements: Linux Mint doesn't require powerful hardware.
- Community-driven: Linux Mint is community-driven where users are encouraged to send feedback to the project.



3- Pop!_OS

- Easy to use: Pop!_OS has a simple and easy-to-use interface that makes it easy to get started with.
- Built-in support for NVIDIA graphics cards: Pop!_OS has built-in support for NVIDIA graphics cards which can make it easier to get your graphics card up and running.
- Large number of software packages available: Pop!_OS has a large number of software packages available that you can install with just a few clicks.
- Free and open-source: Ubuntu Server is free to use and open-source.
- Stable: Ubuntu Server is known for its stability and reliability.
- Large community: Ubuntu Server has a large community of users and developers who can help you if you run into any issues.
- Customizable: Ubuntu Server is highly customizable and you can customize it to suit your needs.
- Large number of software packages available: Ubuntu Server has a large number of software packages available that you can install with just a few clicks.



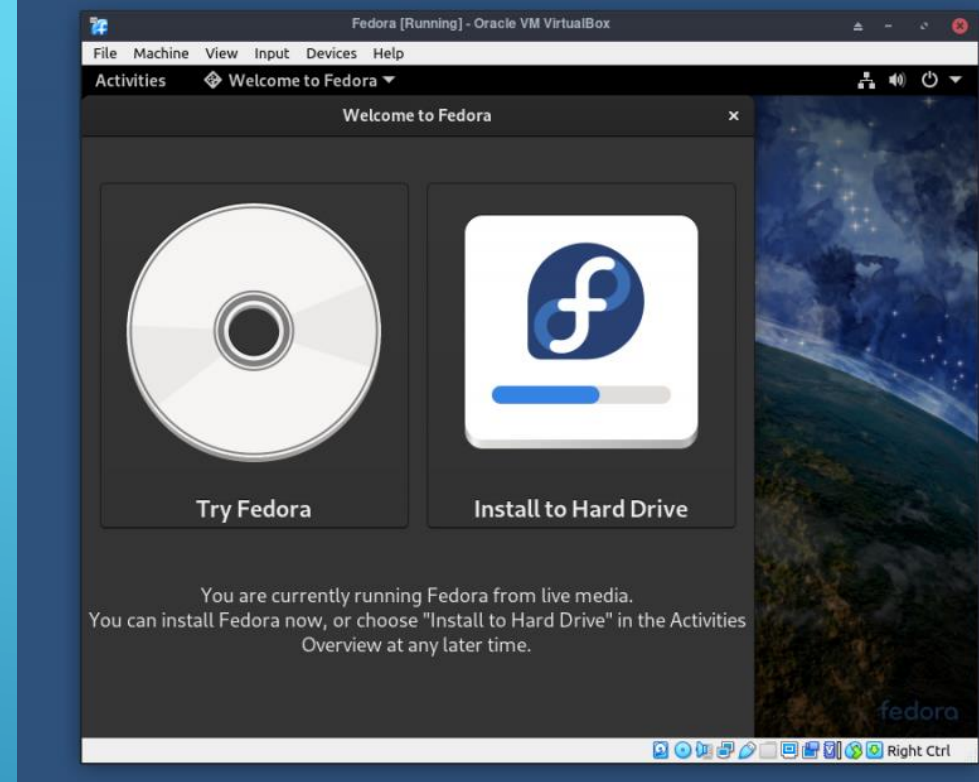
4- Debian

- **Stability and Security:** Debian has a comprehensive policy manual that ensures that all packages are stable and secure.
- **A Balance Between Cutting Edge and Stability:** Debian offers a balance between cutting-edge software and stability.
- **The Largest Number of Installed Packages:** Debian has the largest number of installed packages among all Linux distributions.
- **Easy Transitions Between Technologies:** Debian makes it easy to transition between different technologies.
- **Multiple Hardware Architectures:** Debian supports multiple hardware architectures.
- **A Choice of the Degree of Freedom:** Debian offers users a choice of the degree of freedom they want.
- **A Comprehensive Installer:** Debian has a comprehensive installer that makes it easy to install the operating system.



5- Fedora

- Reliability and Stability: Fedora is a very reliable and stable operating system.
- Enhanced Security: Fedora enhances security in its operating system.
- Graphical Tools: Fedora offers many graphical tools.
- Automatic Updates: Fedora updates automatically.
- Support for Many File Formats: Fedora supports many file formats.
- Education Software: Fedora offers many education software.
- Large Community: Fedora has a large community.
- Unique Security Features: Fedora provides unique security features.



To run a C++ code on Linux, we can use any Integrated Development Environment (IDE) that supports C++ programming language such as: **Visual Studio Code, Eclipse, and Code::Blocks**. Alternatively, we can compile and run the code in the Linux terminal using the following steps:

- 1- Open the terminal window.
- 2- Navigate to the directory where your C++ code is saved.
- 3- Compile the code using the following command: `g++ -o output_file_name input_file_name.cpp`
- 4- Run the code using the following command: `./output_file_name`

we may encounter some conflicts when running your code on Linux if your code uses **Windows-specific libraries** or **functions** that are not available on Linux.

In this case, we may need to modify our code to make it compatible with Linux.

There are many Linux distributions that are appropriate for C++ programming such as Ubuntu, Fedora, etc. However, choosing a Linux distribution depends on our personal preference and requirements.

There may be several conflicts that arise when running C++ code on Linux such as differences in file **paths, libraries, and dependencies**. To avoid these conflicts, we can use a cross-platform build system such as **CMake** or **GNU Autotools**. These tools can help you generate platform-independent build files that can be used on both Windows and Linux.

INTEGRATED DEVELOPMENT ENVIRONMENT & COMPILERS:

An integrated development environment (IDE) is a software application that helps programmers develop software code efficiently. It increases developer productivity by combining capabilities such as software editing, building, testing, and packaging in an easy-to-use application.

GCC: It is a free and open-source compiler that supports many programming languages including C++, C, Objective-C, Fortran, Ada, and others. It is the default compiler for most Linux distributions and is known for its stability and compatibility with older code. It supports many optimization options that can improve the performance of your code.

Clang: It is a free and open-source compiler that supports C++, C, Objective-C, and OpenCL. It is known for its fast compile times and better error messages. Clang is part of the LLVM project which is a collection of modular and reusable compiler and toolchain technologies.

Both **GCC** and **Clang** support C++ standard on Linux. If we wrote our code in Windows with C++, we can use either compiler to compile our code on Linux. However, there might be some **differences** between the **two compilers** that can affect the behavior of our code. It is recommended to test our code with both compilers to make sure it works correctly.

IDES:

Code::Blocks: It is a free and open-source IDE that supports multiple compilers including GCC and Clang. It has a simple and easy-to-use interface and supports real-time debugging.

Eclipse: It is a free and open-source IDE that supports multiple programming languages including C++. It has a powerful code editor with syntax highlighting, code completion, and refactoring tools.

NetBeans: It is a free and open-source IDE that supports multiple programming languages including C++. It has a simple and easy-to-use interface with powerful code editing features.

Qt Creator: It is a free and open-source IDE that is designed for developing applications using the Qt framework. It supports C++ development with real-time debugging.

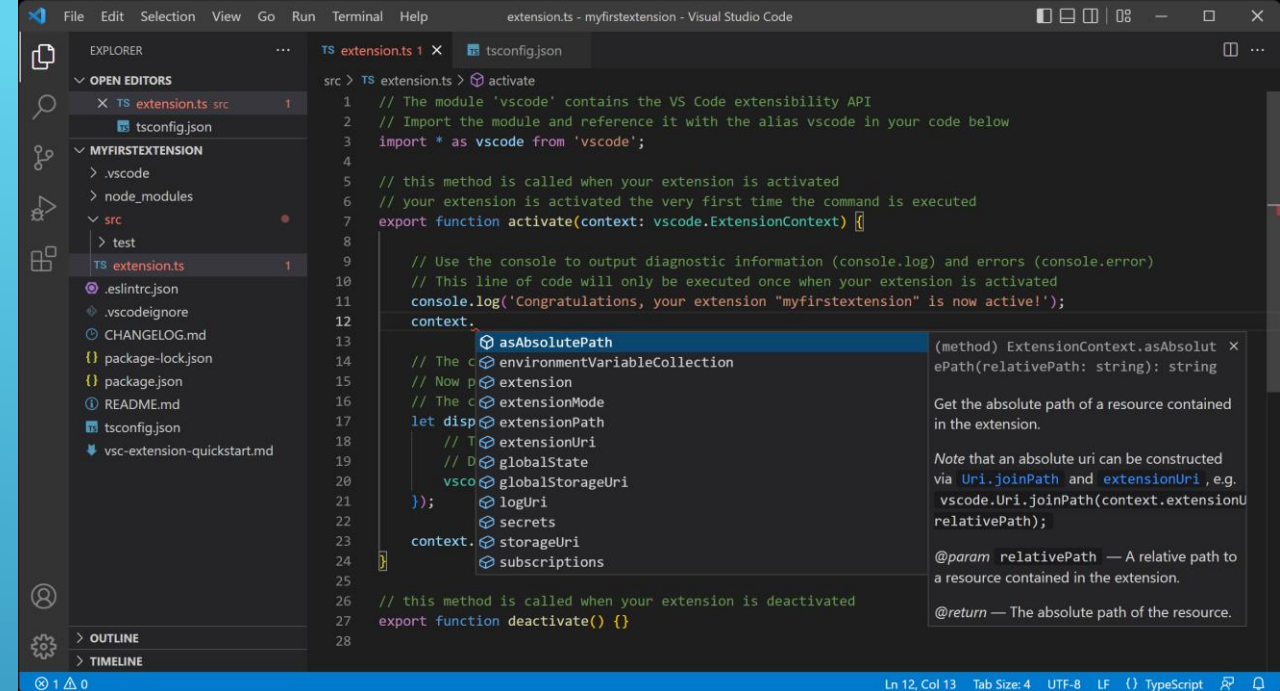
Visual Studio Code: is a free and open-source code editor that supports multiple programming languages including C++. It has a simple and easy-to-use interface with powerful code editing features. It does not have built-in support for real-time debugging or refactoring tools.

1- VS Code

It is a free and open-source code editor that supports multiple programming languages including C++. It has a simple and easy-to-use interface with powerful code editing features. It has a large number of extensions available that can enhance its functionality. However, it is not a full-fledged IDE like the other two.

Some advantages of using **Visual Studio Code**:

- **Cross-platform support:** Visual Studio Code is available on Windows, macOS, and Linux.
- **Lightweight:** Visual Studio Code is a lightweight code editor that is fast and responsive.
- **Powerful code editing features:** Visual Studio Code has powerful code editing features such as syntax highlighting, code completion, and refactoring tools.
- **Large number of extensions:** Visual Studio Code has a large number of extensions available that can enhance its functionality.
- **Integrated terminal:** Visual Studio Code has an integrated terminal that allows you to run commands directly from the editor.

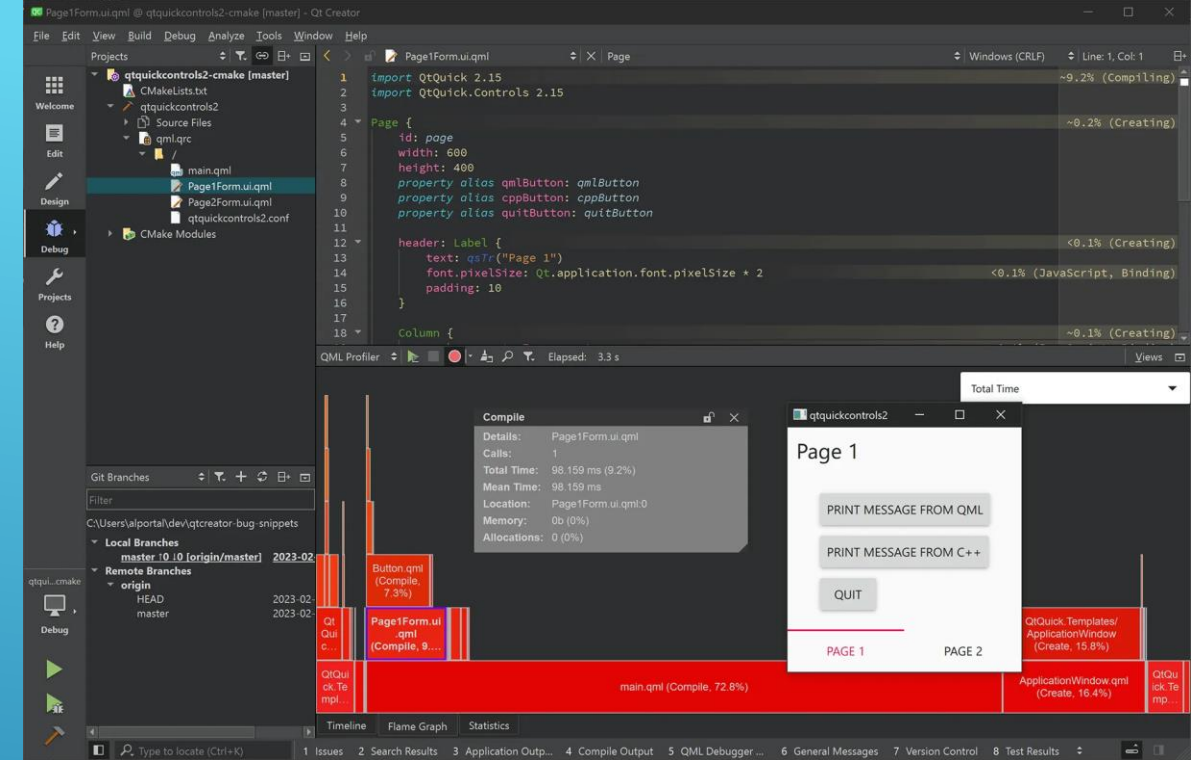


2- Qt Creator

It is a free and open-source IDE that is designed for developing applications using the Qt framework. It supports C++ development with real-time debugging. It has a simple and easy-to-use interface with powerful code editing features. It has built-in support for the Qt framework which can make development easier if you are using the Qt framework.

Some advantages of using **Qt Creator**:

- **Designed for developing applications using the Qt framework:** Qt Creator is designed specifically for developing applications using the Qt framework. It has built-in support for the Qt framework which can make development easier if you are using the Qt framework.
- **Cross-platform support:** Qt Creator is available on Windows, macOS, and Linux.
- **Real-time debugging:** Qt Creator supports real-time debugging which can help you find and fix bugs more quickly.
- **Powerful code editing features:** Qt Creator has powerful code editing features such as syntax highlighting, code completion, and refactoring tools.
- **Simple and easy-to-use interface:** Qt Creator has a simple and easy-to-use interface that makes it easy to get started with.

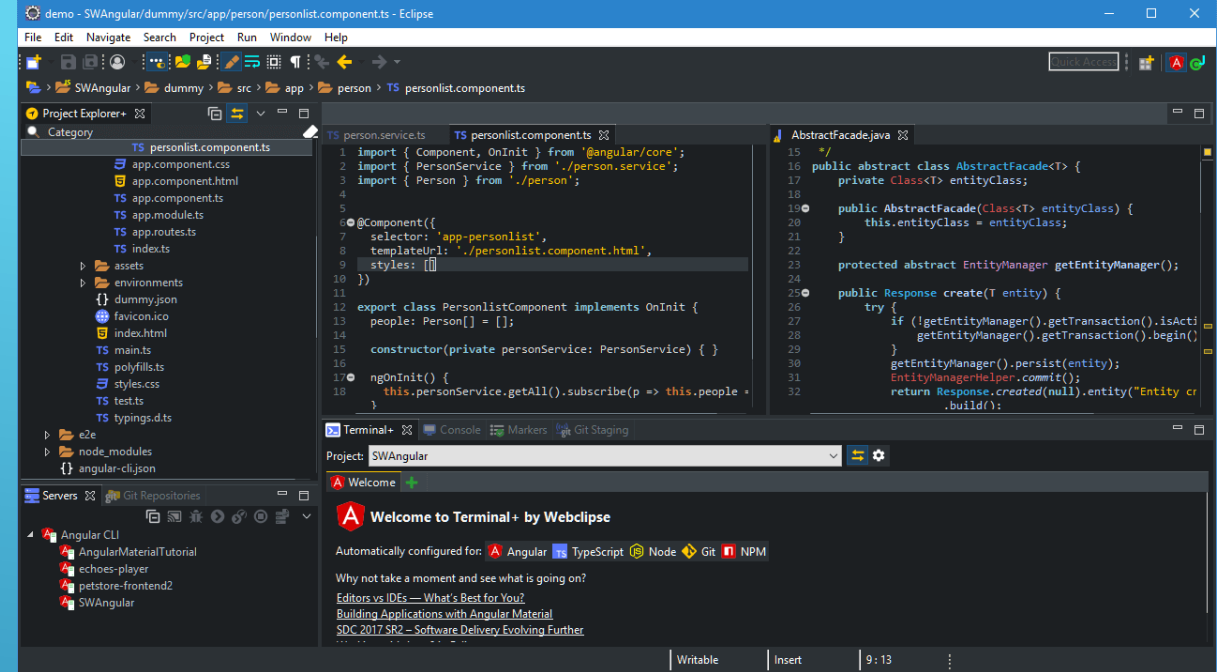


3- Eclipse

It is a free and open-source IDE that supports multiple programming languages including C++. It has a powerful code editor with syntax highlighting, code completion, and refactoring tools. It has built-in support for real-time debugging and profiling. It has a large number of plugins available that can enhance its functionality.

Advantages:

- **Cross-platform support:** Eclipse is available on Windows, macOS, and Linux.
- **Powerful code editing features:** Eclipse has a powerful code editor with syntax highlighting, code completion, and refactoring tools.
- **Built-in support for real-time debugging and profiling:** Eclipse has built-in support for real-time debugging and profiling which can help you find and fix bugs more quickly.
- **Large number of plugins available:** Eclipse has a large number of plugins available that can enhance its functionality.



All of these IDEs are suitable for C++ development on Linux. However, each one has its own strengths and weaknesses. For example, Code::Blocks is known for its simplicity and ease of use while Eclipse is known for its powerful code editing features. NetBeans is known for its simplicity and ease of use while Qt Creator is designed specifically for developing applications using the Qt framework.

PARALLEL & MULTI THREADING & GPU PROGRAMMING:

Parallel programming is a broad concept. It can describe many types of processes running on the same machine or on different machines. Multithreading specifically refers to the concurrent execution of more than one sequential set (thread) of instructions.

To implement a project with C++ in Linux that is important to be real-time, we can use the following tools:

CUDA: It is a parallel computing platform and programming model developed by NVIDIA for general computing on GPUs.

OpenMP: It is an API that supports multi-platform shared memory multiprocessing programming in C, C++, and Fortran.

OpenCL: It is a framework for writing programs that execute across heterogeneous platforms consisting of central processing units (CPUs), graphics processing units (GPUs), digital signal processors (DSPs), field-programmable gate arrays (FPGAs) and other processors or hardware accelerators.

Requirements:

1. A Linux distribution that supports real-time computing and multi-threading.
2. A C++ compiler. We can use GNU Compiler Collection (GCC) or Clang.
3. A CUDA-enabled GPU if we plan to use CUDA.