

Entorno de desarrollo en c++

Por Ariel Parra.

1. terminales

Un emulador de terminal permite al usuario acceder todas las aplicaciones del sistema operativo con interfaces de text, en ingles command-line interfaces (CLI) o text user interface (TUI) applications.

En nuestro caso es donde correremos el compilador y ejecutaremos nuestros algoritmos.

CMD/PowerShell: son las terminales ya incluidas con Windows.

PowerShell v7: es la nueva version opensource de powershell.

git bash: es una capa de compatibilidad de la terminal de linux en Windows.

MSYS2:es una colección de herramientas y bibliotecas para Windows.

WSL: es un ambiente completo de compatibilidad de linux para Windows.

Windows terminal: es una terminal con pestañas para Windows.

Alacritty: es una terminal portable y configurable que usa aceleración de hardware.

Termux: es una terminal con un ambiente de linux sin root para Android.

ISH: es un emulador del sistema operativo alpine linux para dispositivos iOS.

2. Package managers



Un gestor de paquetes es una colección de herramientas de software que automatizan el proceso de instalación, actualización, configuración y eliminación de programas en una computadora. En nuestro caso nos puede ayudar a instalar el IDE y el compilador.

Windows: scoop, winget, MSYS2 (pacman).

MacOSX: homebrew.

Linux: apt, pacman, etc.

iOS: apk (ish).

Android: pkg (termux).

3. compiladores

un compilador es un programa que traduce código de un lenguaje de programación (C++) a lenguaje de bajo nivel (código máquina) para poder ejecutarse.

g++: es el compilador de c++ del proyecto GNU, este es de código abierto y es portable y ampliamente conocido, por lo que es el que usaremos en el curso.

Para windows hay dos opciones principales, MIIngw64 (con MSYS2,scoop,etc.) o g++ bajo linux con WSL, los nombres que suele tener son: gcc, g++, build-base, build-essential ó base-devel.

Clang/LLVM: Es compilador por defecto en MacOS, es de código abierto, altamente optimizado, dice ser compatible con MSV y gcc aunque este tiene toda una infraestructura distinta.

MSVC C++: es el compilador propietario de Microsoft, por lo que ocupas una licencia para usarlo, esta echo específicamente para su IDE Vstudio y enfocado solamente para Windows.

oneAPI DPC++/C++: es el compilador propietario de Intel, por lo que ocupas una licencia para usarlo y esta enfocado en sus procesadores de servidor Xeon.

AOCC: es el compilador propietario de AMD, por lo que ocupas una licencia para usarlo y esta enfocado en sus procesadores "zen".

4. IDEs

Integrated development environment o en español Entorno de desarrollo integrado. Contienen indicadores visuales de código erróneo, debug, sintaxis, cambios, entre otras muchas funciones que no tendrías en un bloc de notas convencional.



Es un IDE de código abierto, con extensibilidad de plugins, es el estandar de IDEs para las competencias, ya que ya incluye el compilador en Windows en la version codeblocks-mingw, a la vez incluye funciones tales como:

- Destacado sintaxis
- Finalización de Código
- Navegador de clase
- Indentado automatico
- Debugger integrado



Es un IDE de código abierto, extendible, integrado con git y una terminal, es desarrollado por Microsoft bajo la licencia MIT con Electron, el cual es un framework para desarrollar aplicaciones de escritorio con HTML, CSS y JavaScript por esto mismo, hay una versión en línea <https://vscode.dev/>. La mayor desventaja es que puede haber muchos problemas en la instalación del compilador y depurador en Windows.

extensiones esenciales: C/C++, code runner, error lens.

temas: nord, Catppuccin, GruvBox, Monokai, etc.

Instalar el compilador en VScode de Windows

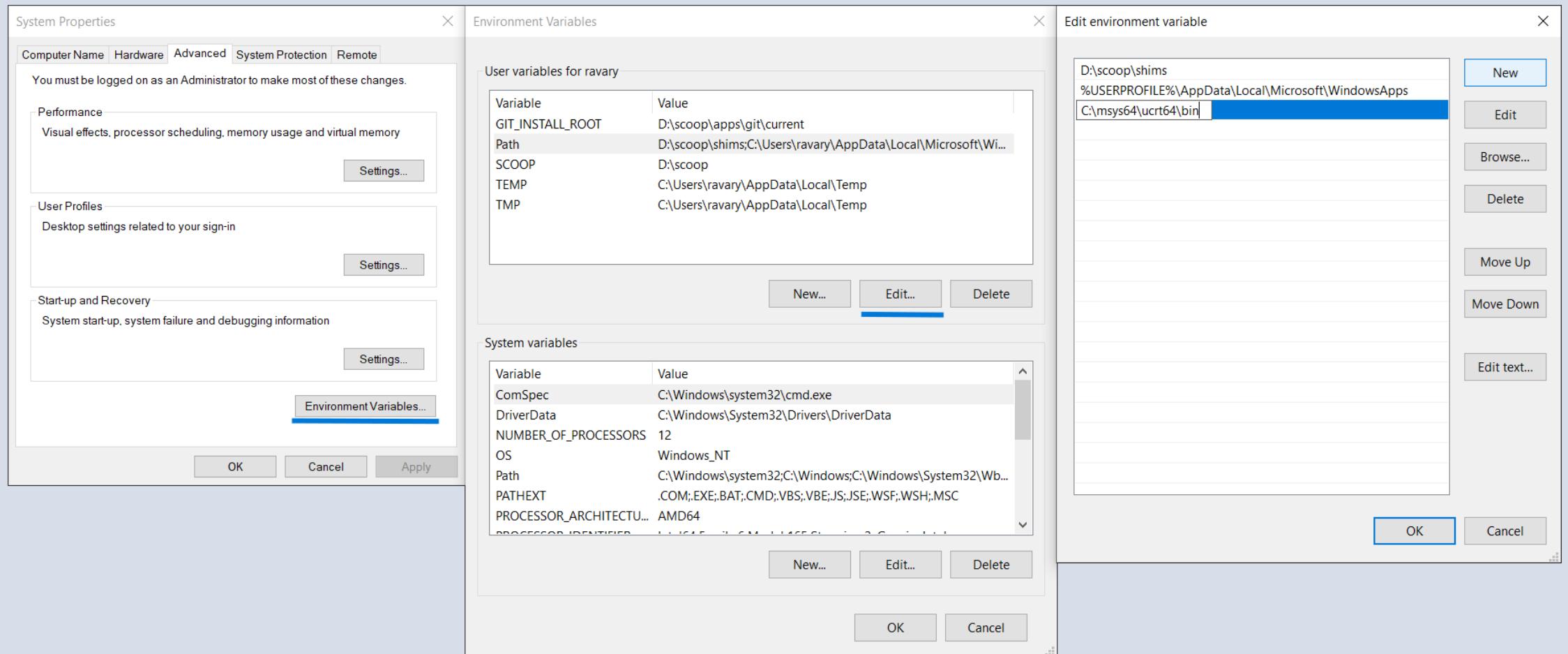


Seguiremos esta [guia oficial de VScode](#).

- Primero descargamos MSYS2 e instalamos dando varios Enters (los errores suelen ser por la mala conexión a internet).
- Ya instalado, abrimos la terminal de MSYS y pegamos este comando para descargar las herramientas necesarias y le damos Enter a cualquier mensaje.

```
pacman -S --needed base-devel mingw-w64-ucrt-x86_64-toolchain
```

- Despues buscamos alguna de estas palabras clave: "variables, path, variables del sistema" en el buscador, abres la aplicación con icono de una computadora con una palomita, allí abriríremos las variables y editaremos el path agregando la dirección: "C:\msys64\ucrt64\bin".



- Tenemos que tener la extensión del compilador C/C++ en vscode
- Al final damos click a la configuración de la extensión y allí buscamos la opción que diga "compiler path", ahí nos manda a settings.json, el cual editamos de la siguiente manera:

```
{  
    "C_Cpp.default.compilerPath": "C:/msys64/ucrt64/bin",  
    "C_Cpp.clang_format_path": "C:/msys64/ucrt64/bin",  
    "code-runner.executorMap": {  
        "cpp": "cd $dir && g++ $fileName -o $fileNameWithoutExt -g && $dir$fileNameWithoutExt"  
    }  
}
```

- Con esto ya podremos compilar y ejecutar el código dentro de VScode dándole click al símbolo de reproducir en la parte superior derecha del editor o con la extensión code runner con las teclas ctrl + alt + N.

Instalar el compilador en VScode de MacOS



- Primero instalamos el gestor de paquetes brew abriendo una terminal y ejecutando el comando, para despues dar varios enters:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

- despues en instalamos el compilador gcc

```
brew analytics off  
brew install gcc  
sudo rm /usr/local/bin/g++  
sudo ln -s $(ls /usr/local/bin/g++-*) /usr/local/bin/g++
```

Integrar Codeforces con VScode

- Primero en VScode agregamos la extension: [Competitive Programming Helper](#)
- Para firefox: Agrega las extensiones [competitive companion](#) y [CPH Submit](#)
- Para navegadores basados en chromium (google chrome, Microsoft Edge, Opera, Brave, etc.): Agrega las extensiones [competitive companion](#) y [CPH Submit](#)

Para usarlas, ten tu sesion de codeforces activa y tambien ten abierto VScode, cuando abras un problema en codeforces haz click en icono de un circulo verde con un +, cambiate a vscode donde veras unas opciones donde eligiras cpp, y listo podras pobrar y publicar tu codigo desde vscode.

VSodium



VSCodium es una distribución binaria hecha por la comunidad y licenciada de manera libre del editor VS Code, este surge ya que vscode aunque tenga un código fuente público, los binarios que distribuyen tienen telemetría activada por defecto.

La diferencia principal con vscode es la tienda [open-vsx](#) en lugar de la de microsoft y la remote window.

Este no tiene las extensiones recomendadas, por lo que si no se está en linux con la aplicación de vscodium-bin-marketplace no recomiendo su uso para programación competitiva.

5. Git y Github



Git

GitHub

Git es una herramienta de código abierto sobre el control de versiones distribuido mientras que GitHub es una plataforma y servicio basado en la nube de Microsoft para el desarrollo de software y el control de versiones utilizando Git, básicamente un servidor donde almacenas tu código y lo controlas con Git.

Para conectar git con github:

- Primero crearemos una cuenta de github <https://github.com/signup>
- Despues creamos un repositorio (proyecto) en <https://github.com/new>
- Luego descargamos git con algun pacakge manager o desde la pagina oficial <https://git-scm.com/downloads>
- si no esta actualizado Windows a la ultima version comprobamos que esta instalado el cliente de OpenSSH.
- En una terminal creamos nuestro usuario local de git:

```
git config --global user.name "Usuario"  
git config --global user.email "correo@ejemplo.com"
```

- y comprobamos nuestros datos con :

```
git config --list
```

- En la misma terminal creamos una llave pública de ssh:

```
ssh-keygen -t ed25519 -c "correo@ejemplo.com"
```

- Copiamos los datos del archivo generado en C:\Users\"Usuario"\.ssh\id_ed25519.pub ó /home/"usuario"/.ssh/id_ed25519.pub (en windows tiene icono de Microsoft Publisher y se puede abrir con bloc de notas) y lo conectamos con nuestra cuenta de github atravez de <https://github.com/settings/ssh/new>, dandole un nombre (el de tu computadora) y pegando la los datos copiados (llave).

- Y por ultimo descargamos nuestro repositorio, en una terminal navega hacia tu carpeta de destino, yo recomiendo la carpeta de Documentos (ojo en Windows la carpeta suele estar bajo C:\Users\"Usuario"\One Drive\Documentos)

```
git clone git@github.com:"Usuario de GitHub"/"Proyecto".git
```

Una vez ya tengamos conectado nuestra cuenta y proyecto, los tres comandos principales de git que usaremos (ya estan integrados en VScode) son:

```
git commit -m "mensaje del commit" # cambios locales a "pushear"  
git push #pusheas los cambios junto con el commit  
git pull #descargas los cambios actuales
```

Extras

Paginas de documentacion de c++: microsoft, cplusplus, w3school ,geeksforgeeks, cppreference, tabla ascii, cppbyexample.

Traductores de Ingles: libretranslate, simplyTranslate.

Operadores de busqueda: Joshua Hardwick, fireship, Busqueda Avanzada por Google.

Multiples buscadores(Metasearch): searxng, Ariel Parra searxng, BigSearch extension.

Teclear velozmente: monkey type, typeclub, TypeRacer, openvim, vimhero.

Para programar más rápido recomiendo usar una distribución de teclado americano (US), o el US-International para seguir usando acentos y caracteres especiales

Compilar y ejecutar en una terminal de Linux:

```
[usuario@computadora ~]$ cd "Ubicacion/del/Proyecto"  
[usuario@computadora "Ubicacion"]$ ls  
archivo1.cpp archivo2.cpp 677A.cpp  
[usuario@computadora "Ubicacion"]$ g++ 677A.cpp  
[usuario@computadora "Ubicacion"]$ ./a.out  
3 7  
4 5 14  
4[usuario@computadora "Ubicacion"]$ rm a.out
```

Compilar y ejecutar en una terminal de Windows:

```
C:\Users\"Usuario"> cd "Ubicacion\del\Proyecto"
C:\Users\"Usuario"\\"Ubicaion"> dir
15/08/2023  09:29 a. m.          639 archivo1.cpp
15/08/2023  09:29 a. m.          461 archivo2.cpp
15/08/2023  09:29 a. m.         612 677A.cpp
C:\Users\"Usuario"\\"Ubicaion"> g++ 677A.cpp
C:\Users\"Usuario"\\"Ubicaion"> .\a.exe
3 7
4 5 14
4
C:\Users\"Usuario"\\"Ubicaion"> del .\a.exe
```

Referencias

- <https://code.visualstudio.com/docs/cpp/config-mingw>
- <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent?platform=windows>
- <https://en.wikipedia.org/wiki/GitHub>
- https://en.wikipedia.org/wiki/Integrated_development_environment
- [https://en.wikipedia.org/wiki/List_of_compilers#C++_compilers](https://en.wikipedia.org/wiki/List_of_compilers#C%2B%2B_compilers)
- https://en.wikipedia.org/wiki/Package_manager
- https://en.wikipedia.org/wiki/Terminal_emulator
- https://en.wikipedia.org/wiki/Visual_Studio_Code
- <https://stackoverflow.com/questions/771756/what-is-the-difference-between-cygwin-and-mingw/771774#771774>
- <https://vscodium.com>
- https://wiki.termux.com/wiki/Main_Page