

# Guide to Attaching the VSCode Debugger to xv6-riscv

## Steps

1. Modify the file `.gdbinit.tmpl-riscv`:

- Remove the following lines:

---

```
target remote 127.0.0.1:1234
symbol-file kernel/kernel
```

---

2. Install `riscv64-elf-gdb`:

- On macOS, use the following command to install it via Homebrew:

---

```
brew install riscv-gnu-toolchain
```

---

- On Ubuntu, you will also need to install it via [Homebrew](#), as the binary for `riscv64-elf-gdb` is not available in `apt`. First, install Homebrew on Ubuntu if not already installed: Then, install `riscv64-elf-gdb`:

---

```
brew install riscv64-elf-gdb
```

---

- If you cannot install `riscv64-elf-gdb` via Homebrew, you can compile it from source. The source and detailed instructions for compilation are available in the official [repository](#).

3. Run the following command in the terminal to start QEMU in GDB mode:

---

```
make qemu-gdb
```

---

- Note: The GDB port differs between Ubuntu and macOS:
  - On Ubuntu, the default port is 26000.
  - On macOS, the default port is 25501.
- If these ports are not available, the port might change. Check the terminal output to see the port assigned.

4. Create a `.vscode` directory in your workspace and add a `launch.json` file with the following content:

---

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Debug",
      "type": "cppdbg",
      "request": "launch",
      "program": "${workspaceFolder}/kernel/kernel", // kernel land program
      // "program": "${workspaceFolder}/user/_wc", // user land program
      "cwd": "${workspaceFolder}",
      "miDebuggerPath": "path_to_riscv64-elf-gdb",
      "miDebuggerServerAddress": "127.0.0.1:<port>",
      "MIMode": "gdb",
      "stopAtEntry": true,
      "setupCommands": [
        {
```

---

```
        "description": "Enable pretty-printing for gdb",
        "text": "-enable-pretty-printing",
        "ignoreFailures": false
    }
  ]
}
}
```

---

Replace `<port>` with the port assigned by `make qemu-gdb`.

## 5. Debugging User Programs

- If you want to debug a user program (e.g., `wc`), update the `program` key in `launch.json` to point to the compiled user program. For example:

---

```
"program": "${workspaceFolder}/user/_wc"
```

---

## 6. Start debugging in VS Code:

- Press F5 or go to Run -> Start Debugging in the GUI.