**https://www.open-mpi.org/faq/?category=debugging**

[**https://stackoverflow.com/questions/329259/how-do-i-debug-an-mpi-program**](https://stackoverflow.com/questions/329259/how-do-i-debug-an-mpi-program)

**http://www.sci.utah.edu/~tfogal/academic/Fogal-ParallelDebugging.pdf**

**Debug MPI program with Visual Studio Code**

04 Dec 2020

**Introduction**

Debugging an MPI program can be a nightmare. I explain how easily you can debug a parallel MPI program with visual studio (VS) code.

**Video**

**https://www.youtube.com/watch?v=lTt02xuD51w&feature=emb\_logo**

**Superuser password**

To get rid of Linux asking for Superuser password try below command

echo 0| sudo tee /proc/sys/kernel/yama/ptrace\_scope

For more info, see [here](https://github.com/Microsoft/MIEngine/wiki/Troubleshoot-attaching-to-processes-using-GDB).

**Settings**

VS Code creates folder *.vscode* which contains settings files:

**c\_cpp\_properties.json ↓**

{

"configurations": [

{

"name": "Linux",

"includePath": [

"${workspaceFolder}/\*\*",

"/usr/lib/x86\_64-linux-gnu/openmpi/include"

],

"defines": [],

"compilerPath": "mpic++",

"cStandard": "gnu17",

"cppStandard": "gnu++14",

"intelliSenseMode": "gcc-x64"

}

],

"version": 4

}

**launch.json ↓**

{

"version": "0.2.0",

"configurations": [

{

"name": "g++ - Build and debug active file",

"type": "cppdbg",

"request": "attach",

"processId": "${command:pickProcess}",

"program": "${fileDirname}/${fileBasenameNoExtension}",

"MIMode": "gdb",

"setupCommands": [

{

"description": "Enable pretty-printing for gdb",

"text": "-enable-pretty-printing",

"ignoreFailures": true

}

],

"preLaunchTask": "C/C++: g++ build active file",

"miDebuggerPath": "/usr/bin/gdb"

}

]

}

**tasks.json ↓**

{

"tasks": [

{

"type": "shell",

"label": "C/C++: g++ build active file",

"command": "mpic++",

"args": [

"-g",

"${file}",

"-o",

"${fileDirname}/${fileBasenameNoExtension}"

],

"options": {

"cwd": "/usr/bin"

},

"problemMatcher": [

"$gcc"

],

"group": {

"kind": "build",

"isDefault": true

},

"detail": "Generated task by Debugger"

}

],

"version": "2.0.0"

}

**Code**

The content of *hello.cpp* is:

#include <mpi.h>

#include <unistd.h>

using namespace std;

int main(){

{

int i=0;

while (0 == i)

sleep(5);

}

MPI\_Init(NULL, NULL);

int rank,size;

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);

char data = 'A';

if (rank==0) {

data = 'B';

MPI\_Send( &data , 1 , MPI\_INT , 1 , 0 , MPI\_COMM\_WORLD);

}

else if (rank==1){

MPI\_Recv( &data , 1 , MPI\_INT , 0 , 0 , MPI\_COMM\_WORLD , MPI\_STATUS\_IGNORE);

}

return 0;

}

**Related Tags**

[MPI](https://iamsorush.com/tags/mpi) [C++](https://iamsorush.com/tags/c++)

**Latest Posts**

* [Span is a new norm in C++ codes](https://iamsorush.com/posts/cpp-span/)
* [Essential VS Code font and extensions for C++](https://iamsorush.com/posts/vs-code-extensions/)
* [An overview of C++ perfect forwarding](https://iamsorush.com/posts/perfect-forwarding-cpp/)
* [Is C++ static polymorphism useful?](https://iamsorush.com/posts/static-polymorphism-cpp/)
* [From lvalue, prvalue, and xvalue to move semantics in C++](https://iamsorush.com/posts/move-semantics-cpp/)