SHPC4001 Setup Guide

To carry out the assignments and exercises for this unit, you will need access to a computer running a 'Unix-like' operating system with the following software installed:

- Git
- GNU Fortran
- MPI (either OpenMPI or MPICH)
- Python 3

You must also set up the UWA VPN to access the remote-access QUISA Workstation.

The computers in the workshop computer lab (B48) are configured with the above software. You are also encouraged to set up your computer for this unit.

If the contents of this document are utterly unfamiliar to you - that's OK! We will provide more context during the first workshop (and much more over the semester).

Access to the QUISA Workstation

Follow the university's instructions to set up the UWA VPN on your computer. Please get in touch with the IT Service Desk if you have any issues setting up the VPN.

1. With the UWA VPN active open the terminal (macOS) or Powershell (Windows).
2. Log in to the workstation using the below command with your student ID in place of 'student_ID':

```
ssh student_ID@130.95.156.96
```

Enter your Pheme password when prompted. Once connected, you will see a message: "Welcome to Ubuntu 18.04.6 LTS. You are connected to the QUISA workstation."

3. Close the connection with the command:

```
exit
```

Software Installation

'Unix-like' refers to a family of operating systems based on the original Unix operating system developed at Bell Labs in the late 1960s. Such operating systems include macOS, Linux and FreeBSD. Unix-like operating systems are ubiquitous in scientific computing. All of the world's supercomputers run Linux.

Note: Our computers become unique with use, so **a fullproof setup guide is not possible**. We can assist with minor troubleshooting, but it is ultimately up to you to ensure you can access the tools you need. Pay attention to the output of commands and particular attention to any error messages. Solutions can often be found by entering an error message's keywords (or even full text) into a search engine.

Windows 10 and 11 Users

Windows is not a Unix-like operating system. However, Windows 10+ users can run Linux tools using the 'Windows Subsystem for Linux' (WSL).

Installing the WSL

1. Open the Windows Powershell as administrator and enter the command:

```
wsl --install -d Ubuntu-22.04
```

This will download and install a 'command-line' version of Ubuntu-22.04 (a Linux distribution).

2. From the start menu, open Ubuntu 22.04 LTS. Create a username and password as directed.

3. Linux distributions have online software repositories. You must perform a system update to install software from the Ubuntu repository. Enter the commands,

   ```
   sudo apt update && sudo apt upgrade
   ```

and press ‘y’ to continue when prompted.

4. To install Git, GNU Fortran and MPI, enter the command:

   ```
   sudo apt install git gfortran gcc mpi-default-dev
   ```

Using the WSL

With the WSL, you can use Linux tools in the Windows Powershell or Command Prompt using the `wsl` command. For example, the `ls` command lists files in the current directory in Linux. Using the WSL, this becomes:

```
wsl ls
```

For more information, see:

- Install Linux on Windows with WSL
- Working across Windows and Linux file systems

MacOS Users

macOS is a Unix-like operating system. The instructions below outline software installation using the ‘Homebrew’ package manager. Package managers automate the installation of development tools.

1. Navigate to the ‘Homebrew’ homepage and copy the Homebrew installation command.
2. Open the macOS terminal. Paste and enter the copied installation command.
3. Enter the below command to install Git and MPI:

   ```
   brew install python git open-mpi
   ```

Testing Your Setup

Open the terminal (Powershell on Windows) and enter the following commands to download, compile and run some test code.

Note: If you are on Windows with the WSL installed, prefix these commands with `wsl`.

1. ‘Clone’ (download) the SHPC4001_test_repository using Git:

   ```
   git clone https://github.com/Edric-Matwiejew/SHPC4001_test_repository.git
   ```

2. Change from the default (‘home’) terminal folder to the ‘SHPC4001_test_repository’ folder using the ‘change directory’ command:

   ```
   cd SHPC4001_test_repository
   ```

   Note: To see the contents of this directory, use the ‘list’ command

   ```
   ls
   ```

3. Run the test script:

   ```
   bash test.sh
   ```

   The expected output is:

```
Hello world!
Hello from thread:      0
Hello from thread:      1
Hello from thread:      3
Hello from thread:      2
Hello from rank:        0
Hello from rank:        1
```

Note: The number order will be different on your machine.)