# Crib Sheet: NEXTGenIO MPI Exercises

## 1 Logging on

**Mac / Linux** Open a command-line terminal and use your username and password to access the EPCC gateway system using a secure shell: `ssh -XY username@hydra-vpn.epcc.ed.ac.uk` Then, log in to the main system: `ssh -XY nextgenio-login2`

**Windows** You should install MobaXterm from `https://mobaxterm.mobatek.net/`. After starting MobaXterm, click on "Sessions -> New Session" from the top bar and then select "SSH". You should enter the EPCC Gateway address `hydra-vpn.epcc.ed.ac.uk` as the "Remote host". Once on the gateway, type: `ssh -XY nextgenio-login2` at the command line.

## 2 Obtaining source code

The source code is stored on github alongside the slides and other documentation. See the "Course materials" link from the main MPI course page on the ARCHER2 website.

To copy a file directly to NEXTGenIO (rather than clicking a link and downloading to your laptop, or cloning the repository), issue "wget" on NEXTGenIO, e.g. for `MPP-templates.tar` from May 2020:

```
wget https://github.com/EPCCed/archer2-MPI-2020-05-14/raw/master/exercises/MPP-templates.tar
```

Note that, due to peculiarities with github, this is **not** the link you get if you "copy link location" from your browser on github. To use `wget`, you must replace `blob` in the URL with `raw`.

Now unpack the tar file: `tar -xvf MPP-templates.tar` and change directory into it: `cd MPP-templates`

## 3 Compiling code

You can compile the C, C++ and Fortran codes directly:

```
mpicc  -o hello hello.c
mpicxx -o hello hello.cc
mpif90 -o hello hello.f90
```

or using the supplied Makefiles

```
make -f Makefile_c
make -f Makefile_cc
make -f Makefile_f90
```

By default, the MPI compiler wrappers use the Intel compilers and the Intel MPI library.

## 4 Running interactively on the login node

You can run parallel jobs interactively from the command line, e.g.:

```
mpirun -n 4 ./hello
```

# 5 Running in batch on the compute nodes

Running on the login node you are sharing resources with other users. This is fine for development work, but if you want accurate timings for benchmarking you must run on the compute nodes using the SLURM batch system.

The SLURM template file is called `hello.job` and is set up to run an MPI program called `hello` on 4 processes. You will need to edit the file if you want to change this.

You can submit to the default queue in the batch system using: `sbatch hello.job`

The batch system will respond with a unique ID for your job: `Submitted batch job XXXXX`.

When the job has finished, the output will appear in a file called `slurm-XXXXX.out`

You can use `squeue` to see running jobs (`squeue -u $USER` will show only your jobs) and `scancel` to cancel a job.

To change the number of processes you run on, edit the batch file and change the value of the `-n` argument to `mpirun`. To run on more than 48 processes (i.e. more than one NEXTGenIO node) you will also need to change the line `#SBATCH --nodes=1`. For example, for 2 nodes (a maximum of 96 MPI processes) use `#SBATCH --nodes=2`

Regardless of how many processes you are running on, you should always specify `-ppn 48`: you only need to change the values of `-n` and `nodes`.