

# Installation of OpenMPI

## *Parallel and Distributed Programming*



Granada's University



LSI Departement

By:  
Daniel Guerrero Martínez  
Sergio Rodríguez Lumley

## Index of content

Installation of OpenMPI.....	3
Quick install for Ubuntu (Debian, Linux/Unix).....	3
Manual install for any linux distribution.....	4
Some common errors.....	5



## Installation of OpenMPI

Open MPI is an open source MPI-2 implementation that is developed and maintained by a consortium of academic, research, and industry partners. Open MPI is therefore able to combine the expertise, technologies, and resources from all across the High Performance Computing community in order to build the best MPI library available.

You can download the last version from it's official page:

<http://www.open-mpi.org/>

### Prerequisites:

- It is necessary to have installed a C/C++ compiler **before** installing OpenMPI. Installing this on Ubuntu by package manager will automatically resolve this problem.
- It is necessary to have the “make” utility for manual installation.

### Quick install for Ubuntu (Debian, Linux/Unix)

Here we will explain how to install it only for Ubuntu and similar OS. If you don't use it or you are not sure, please consider the next step, the manual installation. Necessary packages are:

**openmpi-bin**: Parallel code executor program (mpirun).

Installs: *openmpi-common libopenmpi1.3*

**openssh-client, openssh-server**: Communicating programs (control and presentation routines) between processes.

**libopenmpi-dbg**: Debug information generator for MPI.

**libopenmpi-dev**: Necessary to develop parallel programs based on MPI (mpicc command...).

Quick command:

```
sudo apt-get install openmpi-bin openmpi-common openssh-client openssh-server libopenmpi1.3  
libopenmpi-dbg libopenmpi-dev
```

*Please note that if you are using Ubuntu, it will automatically install a C/C++ compiler, and it will check for compatibilities and installed components.*

## Manual install for any linux distribution

Start downloading the last version of OpenMPI from it's official page:

<http://www.open-mpi.org/software/ompi>

Here you'll be able to download a version on *tar.gz*, *tar.bz2* or on *rpm*. If your system supports *rpm* this is the package you should download, if this is the case, you'll only have to install it without any of the following steps (usually double-clicking).

If your system does not support *rpm*, follow the next steps:

1. Decompress the downloaded file (should be called something like *openmpi-x.x.x.tar.xxx*, changing *x.x.x* for the version downloaded):

```
tar -xvf openmpi-*
```

2. Go into the new folder created from the decompress.

```
cd openmpi-*
```

3. Configure the installation file (making use of the superuser privileges of your system) and start preparing a cup of coffee, because this task usually takes between 5 and 10 minutes.

*It is necessary to add on the prefix the installation directory we want to use for OpenMPI. The normal thing to do would be to select the next directory `"/home/<user>/.openmpi"`.*

```
./configure --prefix="/home/$USER/.openmpi"
```

4. Now is the time for the hard work, install it. For it, we'll make use of the "make" tool. This is a good moment for the coffee, this should take between 10 and 15 minutes, so let it work.

```
make  
sudo make install
```

5. All that is left to do is to include the path to our path environment the path `"installation_directory/bin"` and to the library environment variable `"installation_directory/lib/"`. If your system uses bash you'll have to use the command `export`.

```
export PATH="$PATH:/home/$USER/.openmpi/bin"  
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/home/$USER/.openmpi/lib/"
```

1. If you want the exportation to be effective for the next sessions and terminals, you'll have to write the exports in the environment variable's file. By default it should be `"/home/<user>/.bashrc"` for *bash* users.

```
echo export PATH="$PATH:/home/$USER/.openmpi/bin" >> /home/$USER/.bashrc  
echo export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/home/$USER/.openmpi/lib/"  
>> /home/$USER/.bashrc
```

If everything went Ok, when you run the command `mpirun` it'll show the “how to use” message, the same for `mpicc`.

## Some common errors

- *“I'm on an Ubuntu modern system or compatible and when I run the quick command to install it says it doesn't find openmpi-lib”*

If you are sure your OS should have this package, the error is surely because you don't have your repositories updated. A way of doing this is, if you are using “apt” (Advanced Packaging Tool), running the next command:

```
sudo apt-get update
```

- *“On step 3, configure gave me an error.”*

If the error it gave is something like the next:

```
*** C++ compiler and preprocessor
checking for g++... no
checking for c++... no
checking for gpp... no
checking for aCC... no
checking for CC... no
checking for cxx... no
checking for cc++... no
checking for cl.exe... no
checking for FCC... no
checking for KCC... no
checking for RCC... no
checking for xlc_r... no
checking for xlc... no
checking whether we are using the GNU C++ compiler... no
checking whether g++ accepts -g... no
checking dependency style of g++... none
checking how to run the C++ preprocessor... /lib/cpp
configure: error: in `/home/.../Descargas/openmpi-1.4.1':
configure: error: C++ preprocessor "/lib/cpp" fails sanity check
See `config.log' for more details.
```

*Figura 1. Error de compilador.*

This is because you don't have installed any compiler. It's necessary to install all the compilers we want to use **before** installing OpenMPI. To use it with C and C++, we strongly recommend GNU gcc/g++ compiler version 4.4 or superior.

- *“When I run the command `MPIRUN` or `MPICC` an error tells me that no file was found”*

Please make sure the environmental variables configuration (Step 5 on manual installation). In case you made tha automatic installation, we recommend to restart the computer, if this doesn't work, you can always do step 5 manually. By default the paths are “/usr/include/openmpi/” for `PATH` and “/usr/lib/openmpi/lib” for `LD_LIBRARY_PATH`).