

# Using MPI

## Initial MPI setup

In order to use MPI, some setup is necessary. It appears that some of the scripts used by Version 2 of MPI assumes that the user is using bash as the default shell. Also, our implementation of MPI uses SSH to contact each machine, and therefore requires some setup of security keys. Fortunately, you should only need to perform these steps the first time you use MPI. The steps to follow are:

1. Log into sunsol as a normal user.
2. Be sure that your default shell is bash. You can determine this by:

```
echo $SHELL
```

If your default shell is not bash, then contact the CS system administrator to have it changed.

3. Add `/usr/local/mpi/bin/` to your `$PATH`. Since MPI will need to log into each individual machine, this addition must be added to your startup script (usually `.profile`):

```
export PATH=$PATH:/usr/local/mpi/bin/
```

4. MPI requires a “secret word,” stored in the file `.mpd.conf`. To set up this file:

```
cd $HOME
touch .mpd.conf
chmod 600 .mpd.conf
```

and then use an editor to insert a line like the following into the file:

```
MPD_SECRETWORD=mr45-j9z
```

5. Now, set up keys. Type:

```
ssh-keygen -t rsa
```

This will generate a private/public key pair. The private key will be saved in `~/.ssh/id_rsa` and the public key will be saved in `~/.ssh/id_rsa.pub`

6. Place your public key in your `.ssh/authorized_keys` file, and restrict file access so only you can access the file. All keys listed in that file are allowed access:

```
cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
chmod go-rwx ~/.ssh/authorized_keys
```

You should now be set up to run MPI programs!

### Running an MPI Program

Each time you wish to run MPI, you will need to log in and do the following:

1. Start the mpi system (this may take a short but noticeable amount of time, typically a few seconds):

```
mpdboot -n 7
```

2. Compile and run your program. To test it out, try running the test program `mpi.c`. Do the following:

```
ssh sunsol.cs.uidaho.edu
cp /usr/local/mpi/share/examples_graphics/cpi.c .
mpicc -o cpi cpi.c
mpiexec -np 7 ./cpi
```

You should get no errors during compilation. The output should look similar to the following:

```
earth:~$ mpiexec -n 7 ./cpi
Process 0 on earth
Process 1 on mars
Process 4 on callisto
Process 2 on uranus
Process 5 on saturn
Process 3 on neptune
Process 6 on jupiter
pi is approximately 3.1416009869231249, Error is 0.0000083333333318
wall clock time = 0.305170
earth:~$
```

3. When finished, clean up by typing:

```
mpdallexit
```