**Project ISPC based Programming Assignment**

In project 2 you are required to write an ISPC code for a given problem and run the code on the ORBIT testbed. The ISPC compiler is already installed in “mariasfirstimage” but you may download ISPC at <http://ispc.github.com/>.

Test run your program codes on ORBIT nodes that have been allocated to you. Running your codes on these nodes give the best results. Follow the steps mentioned below.

Steps to access the GRID or Nodes you have been allocated:

* Login to the PUTTY if you are a Windows user or Terminal if you have LINUX or MAC system.
* For successful login you need your Group Number as Login ID and private key as Password.
* Once you are logged in to the system, you can check the status of orbit machines using the following command:
* omf stat –t all

Here, the status of all the orbit machines will be displayed. However, if you want to know the status of any particular nodes that you have been allocated, you need to type in the following command:

* omf stat –t node”node that you have been allocated without the quotes”
* After this, you need to check whether the nodes are powered ON or OFF. If they are OFF, you need to power ON the nodes :
* omf tell –a on –t all (or) omf tell –a on –t node2-1,node2-2
* Next, we need to load the image which would help and support us in running a code based on the programming languages that it supports :
* omf load –t all –i mariasfirstimage.ndz

For different nodes you need to follow the same procedure as stated above.

If you want to load the image to a particular node, say node1-2 of the sandbox you have logged into, you would have to make a minor change to the above command as stated below:

omf load –t node1-2 –i mariasfirstimage.ndz

The image takes some time to load, so in that time you can open multiple PUTTY windows and repeat the above procedure for different nodes so that you have access to greater number of nodes and test your code over multiple nodes simultaneously.

* Next, you will have to ssh to the particular node where you have loaded the image.
* ssh root@node1-1
* ssh root@node1-2
* ssh root@node1-3
* Next, modify the code that you have prepared in a text editor by adding the required header files and perfrom the following operations :
* Copy the code.
* Type vi filename.c in the PUTTY window and hit ENTER key.
* You will see multiple vertical lines. Press Insert tab and paste the code.
* Once you have pasted the code, hit ESC and then type :wq [colon <space> wq] and hit ENTER.

Here, we make use of vim text editor.

* Compile the code using the following commands:
* gcc filename.c –o filename

At this stage, errors if any are displayed on the screen and you can again type vi filename.c to enter the editor and make the required changes. Once you are done making the changes, you have to recompile the file using the same above command.

* ./filename

This is to run the code you have entered and saved in the above filename.

* Observe the output on the screen as per the code you have entered.

You need to include the header files for ISPC program codes. While trying to ssh to your node, it might show you a message “No Route to Host “. In this situations, you need to close the PUTTY and start it again. Repeat the steps mentioned above till you are able to ssh into the node successfully.

The ispc compiler is in your /root directory, to use it, you need to add the compiler path into your working path:   
$vi ~/.bashrc  
  
Add following command in the first row:   
export PATH=$PATH:/root/ispc-v1.9.1-linux  
  
then enable your .bashrc file:  
$source ~/.bashrc  
  
go the the ispc compiler's example fold:   
cd ~/ispc-v1.9.1-linux/examples/simple

type   
$make  
you will see some errors about the missing clang, so install them by:  
$apt-get install clang-3.5  
$apt-get install clang++-3.5  
  
open your makefile, you'll see it use 'clang' rather than 'clang3.5', so you should link them:  
$sudo ln -s /usr/bin/clang-3.5 /usr/bin/clang  
$sudo ln -s /usr/bin/clang++-3.5 /usr/bin/clang++  
  
then make again, you should see the executable file.