

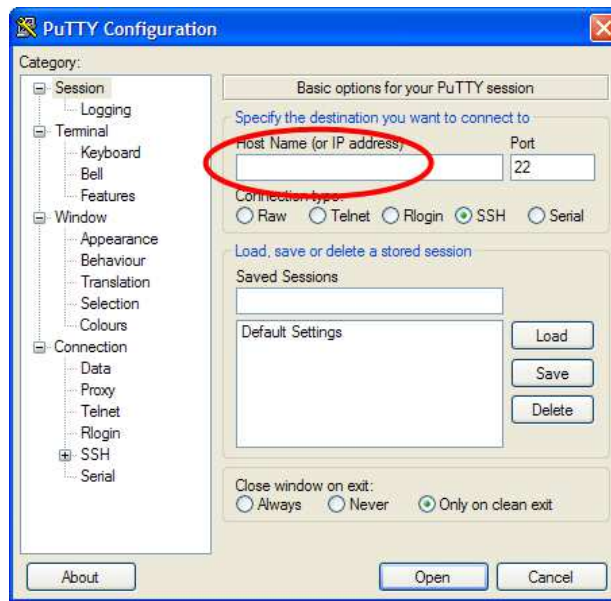
## Lab 1: Introduction to MPI

### 1. Download and install required programs

1.1 Download PuTTY from <http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>, save it on Desktop.

1.2 Download WinSCP from <https://winscp.net/eng/download.php>, install it.

### 2. Use PuTTY to login to cluster.ict.mahidol



#### 2.1 Enter your login and password

2.2 For the first time login, press Enter 3 times until you get the command prompt [yourlogin@cluster ~]\$

```
It doesn't appear that you have set up your ssh key.
This process will make the files:
/home/ekasit/.ssh/id_rsa.pub
/home/ekasit/.ssh/id_rsa
/home/ekasit/.ssh/authorized_keys

Generating public/private rsa key pair.
Enter file in which to save the key (/home/ekasit/.ssh/id_rsa): ↵
Created directory '/home/ekasit/.ssh'.

Enter passphrase (empty for no passphrase): ↵

Enter same passphrase again: ↵
Your identification has been saved in /home/ekasit/.ssh/id_rsa.
Your public key has been saved in /home/ekasit/.ssh/id_rsa.pub.
The key fingerprint is:
57:8b:01:dc:cc:58:29:67:7d:11:d8:af:0a:43:73:5b ekasit@muclcluster.mahidol
[ekasit@muclcluster ~]$
```

2.3 Type `ls` command to list all files in your home directory. For first time login, you should not see any file.

### 3. Create a `helloworld.c` MPI program.

3.1 Use any editor in your machine, e.g notepad, notepad++, editplus, eclipse, or visual C, to create the following program.

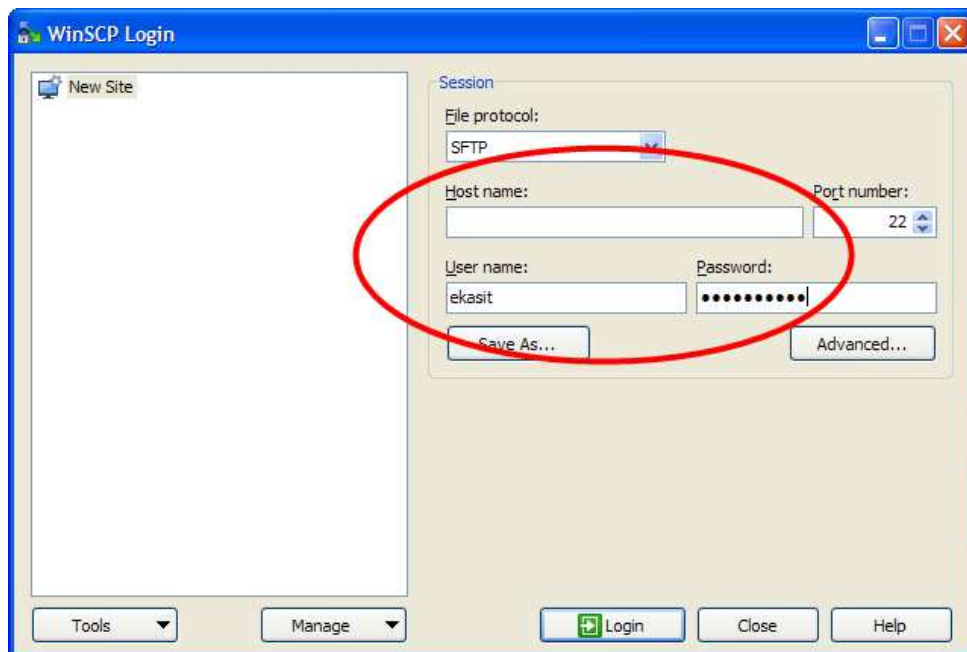
```

#include "mpi.h"
#include <stdio.h>

int main(int argc, char *argv[] )
{
    int rank, size;
    char hostname[MPI_MAX_PROCESSOR_NAME];
    int resultlen;
    MPI_Init( &argc, &argv );
    MPI_Comm_rank( MPI_COMM_WORLD, &rank );
    MPI_Comm_size( MPI_COMM_WORLD, &size );
    MPI_Get_processor_name(hostname,&resultlen);
    printf( "Hello! I'm %d of %d running on %s\n",rank,size,hostname);
    MPI_Finalize();
    return 0;
}

```

3.2 Save the program. Use WinSCP to upload it to your home directory on cluster.ict.mahidol.



3.3 In PuTTY, type `ls` and you must see the `helloworld.c` in home directory.

3.4 Compile the program. Type  
`mpicc -o helloworld helloworld.c`

3.5 If there is an error, correct it in the editor and re-upload it again. If no error, you'll see the program `helloworld` (without `.c` extension) in your home directory.

3.6 Run the program with 2 processes. Type  
`mpirun -np 2 helloworld`

You will see the output. Try running program with 3,4,5 and 6 processes (change the `-np` argument).

4. Create and run the `integer.sum.c` program as follows.

```

#define LEFT 1
#define RIGHT 1000

#include <stdio.h>
#include <mpi.h>

int main(int argc, char *argv[])
{
    int rank, size;
    MPI_Status status;
    int interval;
    int number, start, end, sum, GrandTotal;
    int proc;

    MPI_Init( &argc, &argv );
    MPI_Comm_rank( MPI_COMM_WORLD, &rank );
    MPI_Comm_size( MPI_COMM_WORLD, &size );

    if (rank == 0) {
        GrandTotal = 0;
        for (proc=1; proc<size; proc++) {
            MPI_Recv(&sum,1,MPI_INT,proc,123,MPI_COMM_WORLD,&status);
            GrandTotal=GrandTotal+sum;
        }
        printf("Grand total = %d \n", GrandTotal);
    }
    else {
        interval=(RIGHT-LEFT+1)/(size-1);
        start=(rank-1)*interval+LEFT;
        end=start+interval-1;
        if (rank == (size-1)) { /* for last block */
            end = RIGHT;
        }

        sum=0; /*Sum locally on each proc*/
        for (number=start; number<=end; number++)
            sum = sum+number;

        /*send local sum to Master process*/
        MPI_Send(&sum,1,MPI_INT,0,123,MPI_COMM_WORLD);
    }
    MPI_Finalize();
}

```

5. Modify the `integersum.c` program such that it prints out the starting number, ending number, and local summation computed by each slave process. The following is one possible output.

```

Rank 2: start 251, end 500, local sum 93875
Rank 1: start 1, end 250, local sum 31375
Rank 4: start 751, end 1000, local sum 218875
Grand total = 500500
Rank 3: start 501, end 750, local sum 156375

```