<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HW 1 Overview: Getting Started Using the Student Cluster</td>
</tr>
<tr>
<td></td>
<td>- HW 1.1: Course Tools &amp; Accounts</td>
</tr>
<tr>
<td></td>
<td>- HW 1.2: Unix operations</td>
</tr>
<tr>
<td></td>
<td>- HW 1.3: Install Pacheco Test Code Library</td>
</tr>
<tr>
<td></td>
<td>- HW 1.4: Running Serial Programs on Student Cluster</td>
</tr>
<tr>
<td></td>
<td>- HW 1.5: Timing Serial Programs</td>
</tr>
<tr>
<td>2</td>
<td>General Instructions</td>
</tr>
</tbody>
</table>
Due: 01/29/15

This homework involves demonstrating that you can log onto the student cluster. Tasks:

- access blackboard information for this class
- join Google Group and mailing list
- obtain a user account and logon on the student cluster
- create the correct homework directory structure
- perform simple unix operations to get you familiar with the system
- install the Pacheco demo codes into your directories
- compile, run, test, time a serial C program (histogram)
HW 1.1.a: Course Tools

**Description:** setting up and using required course tools.

- Access your Blackboard account
- Respond to the Google Group invitation.
- **What to ”turn in”:** image of BB access which will also appear in the ”last accessed” column; image of Google Group welcome email (keep these small (1/2 page is fine)).
HW 1.1.b: Accessing the student cluster

**Description:** obtain a user account, and logon using ssh.

- Once you are confirmed in the class, we will create an account using your last name (or using first name initials + last name in the case of duplicate last names).
- This will usually be done after the second day of class.
- **What to turn in:** some image or evidence that you logged on: such as a listing of your directory on tuckoo
HW 1.1.c: Getting cluster information:

- \[\text{gidget:~}]\text{ mthomas}\%\text{ nslookup rohan.sdsu.edu}\n  \text{Server: 10.0.1.1}\n  \text{Address: 10.0.1.1#53}\n
  \text{Non-authoritative answer:}\n  \text{Name: rohan.sdsu.edu}\n  \text{Address: 130.191.3.100}\n
- The cluster is on the internal SDSU campus network with no external login allowed. You can access the cluster from any on campus machine, including the ROHAN Academic Computing system, rohan.sdsu.edu.

- You create the account using your WebPortal account. See:
  - Home Page: \text{http://www-rohan.sdsu.edu/}\n  - Create Rohan Account:
    \text{http://www-rohan.sdsu.edu/raccts.shtml}\n
HW 1.1.d: Remote SSH login

- Launch SSH terminal on your computer
- SSH onto rohan:
  \%
  ssh rohan\textit{UserName}@rohan.sdsu.edu
- SSH onto tuckoo:
  \%
  ssh tuckoo\textit{UserName}@tuckoo.sdsu.edu
- locate the class homework source code directory:
  /
  COMP605
HW 1.2.a.: Test these Unix operations:
- `cat /etc/motd`
  Note 1: try these from your home directory
- `whoami, date, uname -a`
- `cd ~, pwd`
- `ls`, optional arguments `[-al, -R]`
- `mkdir`
- `chmod`, test arguments such as `[-R]`

Create, compile and run a serial "Hello user" program in C (where `user` is your username).

Compiler commands: use the specialized parallel library compiler commands
- C code: `mpicc -o myprogram myprogram.c`
- C code: `mpif90 -o myprogram myprogram.f90`
- where is the command installed?
  `%locate mpicc`

What to turn in: evidence that you completed this:
  images, text file which contains the output captured, session output.
Once you have account information, you will create a homework directory where I will look for all assignment material. You are free to create other directories for development and testing (e.g. dev), but these directories must only contain specific material for the assignment.

- log onto student cluster: tuckoo.sdsu.edu (130.191.127.136)
- create a homework directory call hw
- create a sub directory called hw1
- use unix command `chmod` to set the directory accesses so only you and the instructor can read/access the codes

**What to turn in:** evidence that you completed this: an image, listing, etc.
HW 1.3.a: Install the Pacheco demo codes

Description:

- tar files are located in /COMP605/pacheco_examples
- you may want to read the Unix man pages or Web pages to learn about the commands for `tar` and `gzip`
- you will want to locate the histogram code in the IPP codebase (ch2)
- What to turn in: evidence that you completed this: directory listing.
HW 1.4.a: Compile & Run Serial Pacheco Histogram Code

- run the code for different variables:
  
  usage:  `./histogram
  < bin_count >< min_meas >< max_meas >< data_count >`

- use the following test cases (8 combinations):
  - `bin_count = [5, 20]`
  - `min_meas = 1`
  - `max_meas = [500, 3000]`
  - `data_count = [50, 500]`

- What to turn in: evidence that you completed this: screen image, or text copy of output.
HW 1.4.b: Running the histogram program
HW 1.5: Timing the Histogram code

- Time how long the code takes to run as a function of Problem Size:
  \[ \text{ProbSize} = \text{data\_count} \]
  Wallclock Time: \( T_{\text{wall}} \)

- use the following test cases:
  - \( \text{bin\_count} = 10 \)
  - \( \text{min\_meas} = 1 \)
  - \( \text{max\_meas} = 50000 \)
  - Vary \( \text{data\_count} = 10^n \), where \( n = 0, 1, 2, N_{\text{max}} \)
  - What is \( N_{\text{max}} \)? Explain the limit.

- Modify how the code prints out the results: you don’t need to
  ”plot” the histogram

- You only need to printout the bins, ranges, and the count (not all
  the X’s).

- Plot your test results using excel, Matlab, by hand: \( T_{\text{wall}} \) vs \( \text{ProbSize} \)
Suggestions for timing:
- Use C/Fortran internal timer
- You can find example timer code in /COMP605/tools/code_timer.c
- Avoid using the unix `date` function, it is not very sensitive
HW 1.5: Data Analysis and Presentation

A table of your test data and a plot of the results. Include labels.

<table>
<thead>
<tr>
<th>ProbSize</th>
<th>$T_{wall}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>10</td>
<td>0.2</td>
</tr>
<tr>
<td>100</td>
<td>0.5</td>
</tr>
<tr>
<td>1000</td>
<td>0.6</td>
</tr>
<tr>
<td>10000</td>
<td>0.9</td>
</tr>
</tbody>
</table>

![Graph showing $T_{wall}$ vs ProbSize]
General Instructions

Put homework into a directory:

```
HOME/ <your_username>/hw/hw1
```

- include the source code(s), compiled binaries
- see sections above for what to include
- Write a simple report (this can be TEXT, Word, PDF Doc).
- See each section for what to turn in.
- Turn in hard copy (condensed/minimal number of pages) at start of class.

Once the submission timeline has closed

DO NOT CHANGE THE FILE TIMESTAMPS!
HW directory listing example

```
[/mthomas@tuckoo hw_dir_exj3 ls -R

./username:
  total 20
  drwx------ 5 mthomas mthomas 4096 Aug 28 12:23 ..
  drwx------ 3 mthomas mthomas 4096 Aug 28 12:23 ...
  drwx------ 2 mthomas mthomas 4096 Aug 28 12:23 dev
  drwx------ 5 mthomas mthomas 4096 Aug 28 12:23 hw
  drwx------ 2 mthomas mthomas 4096 Aug 28 12:23 misc

./username/dev:
  total 8
  drwx------ 2 mthomas mthomas 4096 Aug 28 12:23 ..
  drwx------ 5 mthomas mthomas 4096 Aug 28 12:23 ...

./username/hw:
  total 20
  drwx------ 5 mthomas mthomas 4096 Aug 28 12:23 ..
  drwx------ 5 mthomas mthomas 4096 Aug 28 12:23 ...
  drwx------ 2 mthomas mthomas 4096 Aug 28 12:24 hw1
  drwx------ 2 mthomas mthomas 4096 Aug 28 12:24 hw2
  drwx------ 2 mthomas mthomas 4096 Aug 28 12:23 hw3

./username/hw/hw1:
  total 8
  drwx------ 2 mthomas mthomas 4096 Aug 28 12:24 ..
  drwx------ 5 mthomas mthomas 4096 Aug 28 12:23 ...
  -rw-rw-r-- 1 mthomas mthomas 0 Aug 28 12:24 data1.dat
  -rw-rw-r-- 1 mthomas mthomas 0 Aug 28 12:24 file1.c
  -rw-rw-r-- 1 mthomas mthomas 0 Aug 28 12:24 file2.c

./username/hw/hw2:
  total 8
  drwx------ 2 mthomas mthomas 4096 Aug 28 12:23 ..
  drwx------ 5 mthomas mthomas 4096 Aug 28 12:23 ...

./username/hw/hw3:
  total 8
  drwx------ 2 mthomas mthomas 4096 Aug 28 12:23 ..
  drwx------ 5 mthomas mthomas 4096 Aug 28 12:23 ...

./username/misc:
  total 8
  drwx------ 2 mthomas mthomas 4096 Aug 28 12:23 ..
  drwx------ 5 mthomas mthomas 4096 Aug 28 12:23 ..
```