# CSC 101 Spring 2010 – Lab #8 Report Gradesheet

Name ______________________________ WFU Username __________________
Lab Section:    A    B    C    D
Partner’s Name (if you had one): ______________________________________

<table>
<thead>
<tr>
<th>Topic</th>
<th>Points</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-lab questions</td>
<td>20 total - 5 at 4 points each</td>
<td>Recorded on pre-lab.</td>
</tr>
<tr>
<td>Lab report questions</td>
<td>80 total (points listed on questions)</td>
<td></td>
</tr>
</tbody>
</table>

No electronic submission required this week!
Paper report due date is Thursday, not Friday! (Friday is a holiday)

Total: _____________ of 80 for lab report
Purpose: The purpose of this lab is to reinforce and deepen your understanding of the role of operating systems in managing how your computer runs. In addition, you will be introduced to an operating system that is an alternative to Windows. There are a set of questions throughout this document that you should answer as you encounter them and submit by the due date.

Choose a Partner
You should work in pairs for this lab. One computer should be used to answer the questions that deal with the Windows operating system, while the other computer should be used to answer the questions that deal with the Ubuntu operating system.

Starting Ubuntu
Each student whose laptop is going to run the Ubuntu operating system should request the CD from the lab instructor. Please use this CD during lab and then return it at the end of lab so it can be used by students in the later sections. An introduction to starting Ubuntu and using it will be provided at the start of lab.

Ubuntu is an operating system that is based on the Unix operating system. Linux, which you heard about in lecture, is also descended from Unix. As you know Windows was developed by Microsoft. Unix on the other hand was initially developed by employees at AT&T – the phone company! This operating system has a lineage that is quite separate from Windows and it is actually closer to the current Apple operating system in design. We will compare and contrast Windows and Ubuntu to see in which ways they act very similar (since they are both operating systems and must fulfill the role of an operating system) and ways that they are distinct.

To run Ubuntu, you should insert the CD in your laptop and reboot. When the window prompts you, choose the option “Run Ubuntu without changing your computer”. BE CAREFUL THAT YOU ONLY MAKE THIS CHOICE. Also be patient with Ubuntu – since we will be running it from a CD instead of from your hard drive, it may take a little longer to respond than you are used to.

The Operating System’s Supervisor Role
WINDOWS: As was presented in the pre-lab, the Task Manager program in Windows can reveal information about the programs that are being executed on your computer. Start the Task Manager application by hitting the Ctrl, Alt, and Delete buttons together and then selecting Start Task Manager from the screen that appears (on an older laptop, Ctrl, Alt, and Delete together should just bring up the Task Manager).

Select the Processes tab from the Task Manager window. Remember that this window shows a table with several columns: the name of the program that is running, the user who started the program, the amount of CPU time that is being used by the program, the amount of memory being used by the program, and, on new laptops, a brief description of the program.

Looking into the User Name column, you should see two user names – your username/login for the laptop and the name SYSTEM. The SYSTEM user is a fake user which has full administrative rights to the machine and is responsible for execution of much of the built in operating system software. Many of the rest of the programs are listed as being executed by your username. This may be surprising, since you may have only
started a few programs manually. However, many programs spawn other programs to help them complete their job, and many programs are started (such as those in the lower right-hand side system tray) when you login and thus are attributed to you as the user.

In the pre-lab, we looked at which programs were making most use of the CPU. If we want to see which programs use the most memory, we can sort on that column. Let’s sort highest to lowest – look back in the pre-lab if you don’t remember how to sort the columns.

**Question 1: (6 points)** On your computer, sort the table by the Memory column so the values in that column are in descending order (highest to lowest).

Part 1) What is the Image Name of the top three programs listed on your computer.

1) ____________________________ 2) __________________________
3) ____________________________

Part 2) What is the User Name associated with each of the programs you listed in Part 1?

1) ____________________________  2) _________________________
3)_____________________________

Part 3) How stable is the memory usage of these programs? (Are the values changing a lot or a little?)

**UBUNTU:** You will now examine how Ubuntu keeps track of and presents similar program CPU and memory usage. When you have Ubuntu running, go to the System menu, Administration submenu, and then choose the System Monitor application. You should see a window similar to the one at right appear.

Under the View menu in the System Monitor application, choose All Processes. Under the Edit menu, choose Preferences. Click the box next to User and hit Close.

**Question 2: (6 points)** On your Ubuntu running computer, sort the table by the CPU column so the values in that column are in descending order (highest to lowest; the small ^ symbols should be pointing up).

Part 1) What is the Process Name of the top three programs listed?

1) ____________________________ 2) __________________________
3) _____________________________

Part 2) What is the User associated with each of the programs you listed in Part 1?

1) ____________________________  2) _________________________
3)_____________________________

Part 3) How stable is the CPU usage of these programs? (Are the values changing a lot or a little?)
Supervisor Job Scheduling

WINDOWS: On the Windows laptop in your pair, open the web-browser that worked for you in running the simulator from the last lab and go to the following webpage:
http://www.cs.wfu.edu/~turketwh/CS101/Spring2010/lab8/

You should see a window come up similar to the one at right. Using this webpage, you will examine different ways that an operating system might schedule jobs. There are three commonly used scheduling algorithms, of which we will look at the first two.

First come, first served (FCFS): Programs are scheduled on the CPU in the order they arrive, and each program, when it starts, executes on the CPU until it completes. Only then does another program get to start on the CPU. This is very similar to waiting in line to checkout at a grocery store. If you are stuck behind someone with a lot of items, you have to wait until they are done before you can be checked out. However, once the cashier starts helping you, no one could interrupt you until you are done.

Round robin: Each program gets a small amount of time on the CPU (a quantum) and then another program gets to run. Programs are moved off and on the CPU, allowed to run for the quantum each time they are moved on, until they complete. If only one program is left to run, then it basically stays on the CPU (getting a lot of quantums in a row). Using the grocery store checkout metaphor, this is equivalent to the case where, if there are multiple people in line, one item is scanned per person, repeatedly. Eventually all of your items will get scanned, but you have to share the cashier with everyone else in line at the same time.

Shortest job first: This approach executes programs in order of remaining completion time. Using the grocery store metaphor, this would be if they asked everyone in line to re-order based on how many items were in their cart. At that point, it turns into first-come, first served.

Question 3: (4 points) On your Ubuntu running computer, now sort the table by the Memory column so the values in that column are in descending order (highest to lowest; the small ^ symbols should be pointing up).

Part 1) What is the process name of the top three programs listed?
1) ____________________________ 2) ____________________________ 3) ____________________________

Part 2) How stable is the Memory usage of these programs? (Are the values changing a lot or a little?)

Question 4: (4 points) On the Windows operating system, you should have seen two users running programs: SYSTEM and your username. How many different users do you see on Ubuntu?

Argue whether or not you believe that Ubuntu has a user to run administrative programs similar to the Windows SYSTEM user.

✓ If you get here during lab, have the instructor or TA checkoff that you have completed this step of the lab.

Supervisor Job Scheduling

WINDOWS: On the Windows laptop in your pair, open the web-browser that worked for you in running the simulator from the last lab and go to the following webpage:
http://www.cs.wfu.edu/~turketwh/CS101/Spring2010/lab8/

You should see a window come up similar to the one at right. Using this webpage, you will examine different ways that an operating system might schedule jobs. There are three commonly used scheduling algorithms, of which we will look at the first two.

First come, first served (FCFS): Programs are scheduled on the CPU in the order they arrive, and each program, when it starts, executes on the CPU until it completes. Only then does another program get to start on the CPU. This is very similar to waiting in line to checkout at a grocery store. If you are stuck behind someone with a lot of items, you have to wait until they are done before you can be checked out. However, once the cashier starts helping you, no one could interrupt you until you are done.

Round robin: Each program gets a small amount of time on the CPU (a quantum) and then another program gets to run. Programs are moved off and on the CPU, allowed to run for the quantum each time they are moved on, until they complete. If only one program is left to run, then it basically stays on the CPU (getting a lot of quantums in a row). Using the grocery store checkout metaphor, this is equivalent to the case where, if there are multiple people in line, one item is scanned per person, repeatedly. Eventually all of your items will get scanned, but you have to share the cashier with everyone else in line at the same time.

Shortest job first: This approach executes programs in order of remaining completion time. Using the grocery store metaphor, this would be if they asked everyone in line to re-order based on how many items were in their cart. At that point, it turns into first-come, first served.
Almost all modern computers make use of round-robin scheduling, as this supports multi-tasking – each application you are running gets a small bit of time and then another one runs. The quantums are small enough on modern computers that the user is unable to tell that their program is not actually “running” at any given instant.

We want to look at two scheduling approaches: First Come, First Served (FCFS) and Round Robin. In the window you just opened, select Example 1 in the drop down box and make sure FirstCome,FirstServed is in the Algorithm box. You should see four processes in the Process List, named 1-4 and with times 20, 80, 35, and 60 respectively. Hit the Run button and monitor the Messages and Statistics box.

When all four jobs finish, answer the following questions:

**Question 5: (8 points)** For FCFS scheduling, complete the following table (you may have to scroll up in the Messages and Statistics box to find the Start and Finish values). You won’t find the Total Time data – instead, you should compute that using the Start and Finish values you record.

<table>
<thead>
<tr>
<th>Process Number</th>
<th>Start Time (first time you see ‘switching to process X’ or ‘Starting process X’)</th>
<th>Finish Time (when you see ‘Finishing process X’)</th>
<th>Total Time Executing (Finish – Start)</th>
<th>Total Time Since Arrival (Finish – 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What is the average total time of execution (the average of the next-to-last column)?

What is the average total time since arrival (the average of the last column)?

Now hit the Reset button, choose RoundRobin in the Algorithm box, and set the Quantum size entry to 10. Hit the Run button and monitor the Messages and Statistics box.

**Question 6: (8 points)** For RoundRobin scheduling with a quantum of 10, complete the following table (you may have to scroll up in the Messages and Statistics box to find the Start and Finish values).

<table>
<thead>
<tr>
<th>Process Number</th>
<th>Start Time (first time you see ‘switching to process X’ or ‘Starting process X’)</th>
<th>Finish Time (when you see ‘Finishing process X’)</th>
<th>Total Time Executing (Finish – Start)</th>
<th>Total Time Since Arrival (Finish – 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What is the average total time of execution (the average of the next-to-last column)?

What is the average total time since arrival (the average of the last column)?

✔ If you get here during lab, have the instructor or TA checkoff that you have completed this step of the lab.
Finally, hit the Reset button, choose RoundRobin in the Algorithm box, and set the Quantum size entry to 1 (instead of 10). Hit the Run button and monitor the Messages and Statistics box (which will fill up pretty fast!) This is the most realistic scheduling algorithm with respect to modern computers.

### Question 7: (8 points)
For RoundRobin scheduling with a quantum of 1, completing the following table (you may have to scroll up in the Messages and Statistics box to find the Start and Finish values).

<table>
<thead>
<tr>
<th>Process Number</th>
<th>Start Time (first time you see ‘switching to process X’ or ‘Starting process X’)</th>
<th>Finish Time (when you see ‘Finishing process X’)</th>
<th>Total Time Executing (Finish – Start)</th>
<th>Total Time Since Arrival (Finish – 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What is the average total time of execution (the average of the next-to-last column)?

What is the average total time since arrival (the average of the last column)?

### Question 8 (4 points):
Most people are interested in minimizing the total amount of time it takes between the time they first ask a CPU to work on their job and the time it ends.

Part a) Which of the two averages you came up with is more useful as a measure of what people are interested in? Make sure that you understand what it means for a job to arrive on the CPU and to actually start executing on the CPU under the different scheduling algorithms. You can assume all jobs arrive at time 0.

Part b) Given your answer to Part (a), which scheduling approach would your averages suggest is best to be used: FCFS, RoundRobin with Quantum 10, or RoundRobin with Quantum 1?

---

### Memory Management
Now we want to examine the role of the operating system in memory management. Remember from lecture that memory is allocated to each program individually. The operating system itself gets a large chunk of memory, and then each program gets memory for the program itself (the instructions) and then for the data being manipulated by the program. In lecture, the image to the right was shown depicting this process.

To see this process in action, on both your Windows laptop and your Ubuntu laptops, start the Firefox browser if it is not already open. Once it has been opened, answer the following questions by viewing the Task Manager in Windows and the System Monitor in Ubuntu.
The Ubuntu Operating System

The last part of this lab will explore the Ubuntu operating system. We will examine a little bit more of how it does file management, process management (its supervisor role), and its interface.

Interface:
On the laptop running Ubuntu, under the Places menu, choose the Home Folder menu option. This should open a window similar to the one at right and should be similar to what you typically work with in Windows.

Now choose the Applications menu, Accessories submenu, then the Terminal menu option. This should open a window similar to the one below and to the right.

Question 9 (12 points):
Part a) On Windows, how much memory in MB is reported to be taken up by the Firefox program? The number reported by the Task Manager is the total memory allocated for application and data. _________________

Part b) On Ubuntu, how much memory is reported in MB to be taken up by the Firefox program? The number reported by the System Monitor is the total memory allocated (application and data). _________________

Part c) On Windows, let’s find the actual Firefox program. On my computer, it is found in: C:\Program Files\Mozilla Firefox. Record the size of the firefox.exe program in MB here: _________________

Part d) Assuming that the file size in part (c) is the size of memory in MB required for the program instructions on both Windows and Ubuntu, how much memory is being used for data?
   On Windows: ________________________  On Ubuntu: ________________________

Part e) On your Windows laptop, now open a new tab or window in Firefox and go to your favorite website.

Did the amount of memory allocated to Firefox increase, decrease, or stay the same? _________________

If the memory changed, how much did the memory change (in MB)? ________________________________

The Ubuntu Operating System

The last part of this lab will explore the Ubuntu operating system. We will examine a little bit more of how it does file management, process management (its supervisor role), and its interface.

Interface:
On the laptop running Ubuntu, under the Places menu, choose the Home Folder menu option. This should open a window similar to the one at right and should be similar to what you typically work with in Windows.

Now choose the Applications menu, Accessories submenu, then the Terminal menu option. This should open a window similar to the one below and to the right.

Question 10: (3 points) Write first image or second image in the blanks to answer the questions below.

Which image above best represents the Ubuntu command line interface? ________________________

Which image above best represents the Ubuntu graphical interface? ____________________________

Question 11: (3 points) In the 2nd window you opened (the Terminal application), type ls –l | grep drwx (those are the letter l as in lisa, not ones). Answer the following question – you answered it in the pre-lab for the Windows operating system – this time it’s for Ubuntu: Looking at these two views, are there any folders present in one view that aren’t in the other view? Remember you only need to look at folders, not files.
   a. Yes, almost all the folders were different
   b. A few extra folders were present in one interface that weren’t in the other interface
   c. No, the exact same folders were present
Supervisor Role:
Now, in the Terminal application, type the following command: `top`. You should see the window change to look similar to the following:

```
    Tasks: 114 total,  1 running, 113 sleeping, 0 stopped, 0 zombie
    Cpu(s): 11.6%us, 25.1%sy, 0.0%ni, 62.4%id, 0.0%wa, 1.0%hi, 0.0%si, 0.0%st
    Mem: 380056k total, 367406k used, 12656k free, 46148k buffers
    Swap: 409616k total, 16688k used, 392928k free, 115456k cached
```

The highlighted line shows how heavily the CPUs on your computer are being used. The combination of the `us` (11.6% in the example above) and `sy` (25.1%) entries together say that the CPU is approximately 36.7% busy.

Question 12: (4 points) Answer the following question – you answered it in the pre-lab for the Windows operating system – this time it’s for Ubuntu.

Part a) Observe the CPU usage for 20 to 30 seconds. Choose the range below in which your CPU usage tended to fall while you were watching. Remember to estimate the sum of the `us` and `sy` entries.
   a) 0 to 25%  b) 26 to 50%  c) 51% to 75%  d) 76 to 100%

Part b) As we were able to do in Windows by viewing the per-CPU graph (you did this in the pre-lab) we can see per CPU usage rates in Ubuntu. With this `top` program still running, hit the 1 key on your keyboard. You should see the highlighted line above turn into two lines: `Cpu0` and `Cpu1`.

Choose the answer below which best fits what you observed:
   a) The data from both CPUs looked very similar, suggesting programs are running on both CPUs
   b) One CPU looked like it was used more often than the other
   c) My computer only had one CPU, so this question doesn’t relate to me. (This should not be the case unless you have a very old laptop or forgot to change the option described above!)

Question 13: (4 points) Help me justify this statement by filling in the blank: It appears that the CPU(s) on my computer are always running some process (program), given that the following is true about the `us` and `sy` times I am seeing:

_____________________________________________________________________________

Question 14: (6 points)
Part a) What was the PID of your firefox process? ____________________________________________

Part b) Describe what happened to firefox when you issued the kill command:

Part c) Are you aware of a way to do something similar in Windows? If so, how do you do it in Windows?
If you get here during lab, have the instructor or TA checkoff that you have completed this step of the lab.

At this point, you are done with the lab. On the laptop running Ubuntu, click on the O/1 circle near the top of the window you are working in (similar to the image shown at the right). Choose the **Shut Down**... menu option. This will stop the Ubuntu program, ask you to eject the CD, and then to reboot your computer.

Submission: There are no electronic/digital files to submit online this week. Just make sure to turn in your answers to these on-paper questions and to make sure that you include your partner’s name on the grade-sheet.

Note the due date for this lab is Thursday at 3:00pm, not Friday – Friday is a University holiday!