Assignment 2 Marking Scheme

1. (10 points)

Correct Observation: The CPU utilization should be between 95% to 100% CPU utilization.

Explanation: Basically the assignment of different nice values should not affect the amount of CPU time given to cpuTimeWaste. Since cpuTimeWaste is the only process demanding CPU time it is not competing with other processes for CPU time and hence it is almost always selected. The result is close to 100% CPU utilization.

To receive full marks students should have the correct observation (5 points) and a correction explanation (5 points).

Another Interpretation: I believe some of the students may have interpreted the question to mean that they were to simultaneously instances of cpuTimeWaste at each of the nice levels. In that case they should see different CPU utilizations.

The instance running at nice value -5 should get 2/3 of the CPU;
The instance running at nice value 0 should get 20-21% of the CPU;
The instance running at nice value 5 should get 7-8% of the CPU;
The instance running at nice value 10 should get 2-3% of the CPU;
The instance running at nice value 15 should get 1% of the CPU;
The instance running at nice value 19 should get less 1% of the CPU;

The exact numbers will depend on their system so just make sure the trend you see above is similar.

2. (10 points)

- (5 points) For the process with the default nice value to receive 10% of the CPU the nice value of the other process should be -10.
- (5 points) For the process with the default nice value to receive 95 % of the CPU the nice value of the other process should be 13(will accept 14/15)

Notes:
Students may show this experimentally or by applying a formula. Both should be accepted. Deduct -2 for each of the above if there isn’t any evidence of either.
3. (10 points) Run the program to make sure that getpid() needs more time than the minimal function call. Look at the code to make sure that it is timing a function a simple function and getpid(). If so students should receive 10 marks.

If the observations are not as expected then look at the code – If there is some sort of code give 3 points.

4. (15 points)

Basically give 8 points for the correct processor type and 7 points for the correct kernel version. The answer depends on your VM. The processor type is in /proc/cpuinfo. I am assuming they print the value associated with “model name” but if they also include the vendor_id that is fine. The kernel version is in /proc/version. The correct answer is something like “Linux version 3.19.0-49-generic”. I think many people are just printing everything. Go ahead and accept it.

5. (20 points)

You should run the program for cpuTimeWaste and IOBound (these are on the web page). For cpuTimeWaste the amount of time spent in user mode should always be quite a bit higher than that of kernel mode. For IOBound the user mode time should be quite a bit less than kernel mode time. I suggest running the code for 10-15 seconds. Please remember to kill cpuTimeWaste and IOBound after marking each individual assignment.

Just remember you are looking for trends.

Each observation is worth 10%.

If the programs don’t work just give 3 points for compiling correctly.

6. (20 points) With a CPU intensive process:

You should run the program to confirm their observations. For the interactive program please open another window and type commands.

The observations are the following:

**OTHER: (6 points)** – Should feel normal

**FIFO (6 points):** 95% percent of the CPU is reserved for the FIFO process (5% of the CPU is reserved for non-FIFO tasks). The system is slow to user input since comparison1 consumes 95% of available processor time. Since it sleeps for one second it improves for short bursts of time.
**RR (6 points):** Similar comments to FIFO but the students may note that it isn’t has “bad” for interactive programs as FIFO.

In marking give half the points to the observation and the other half to the explanation.

**Conclusion (2 points):** FIFO, RR should not be used for CPU intensive programs

7. (15 points)

**OTHER: (6 points) –** Should feel normal

**FIFO (5 points).** A little sluggish; The CPU bursts are shorter; so more time for other processes

**RR (5 points)** A little sluggish; the CPU bursts are shorter; so more time for other processes

**Conclusion (5 points):** FIFO, RR can be used for programs with short CPU bursts.