**Homework (Simulation) Wan Huzaifah bin Wan Azhar**

This program, process-run.py, allows you to see how process states change as programs run and either use the CPU (e.g., perform an add instruction) or do I/O (e.g., send a request to a disk and wait for it to complete). See the README for details.

**Answer:**

1. Using the flag 5:100,5:100, the OS will create two process. Process 1 will run first before giving the CPU to process 2. The CPU utilization will be 100% since there is no I/O operation.
2. The total time to complete each process is higher than 4 since the process 0 will run and finished. Process 1 will then run io and wait for IO operation to finish. How long it will take depends on the hardware. Since IO is always slower than CPU, it will take longer than 1 instruction to finish the IO. In this case, it will take 11 instruction to finish both processes.
3. Switching the order of process which does IO operation and process that does not do IO matters. If an IO operation starts on first process, the OS will switch the first process to blocked until it finishes, which will start second process. After the second process finish or interrupted, control will be given back to first process. The total time should be less than 11 instructions since it should be faster.
4. If the process cannot switch until IO is finished (SWITCH\_ON\_END), the command will run the first process of IO operation until it is finished before CPU switch to the second process. In this case, the time taken will be 11 instructions.
5. This is the same as 3., the first process that does IO operation will be blocked and the second process will run. After the second process is finished, control is given back to the first process.
6. With (IO\_RUN\_LATER) flag, the CPU is not effectively utilized in managing the process. This is because the first process starts IO operation but cannot finish or start another IO operation until the second and the third process is finished. In effect, the next IO operation will be done after that, which will waste CPU resources on waiting for IO operation to finish because there is no process to switch to.
7. With (IO\_RUN\_IMMEDIATE), the IO process can start IO operation, and switch to second or third process. If one process is finished, the OS will switch to IO process to finish previous IO or start another IO operation before switching to the next process. This will efficient as instruction can run when IO operation is being done.
8. With “-s 1 -l 3:50,3:50” or “-s 2 -l 3:50,3:50” or “-s 3 -l 3:50,3:50”, there is no way to predict how the trace will turn out as it is randomized. In the 3 instruction, there is many possibilities, such as:
   1. Two IO instruction and One CPU instruction
   2. Three CPU instruction
   3. Three IO instruction
   4. Two CPU instruction and One IO instruction.

IO\_RUN\_LATER will run IO Process after all the process is finished while IO\_RUN\_IMMEDIATE will return the control to immediate IO process after one process is finished.

SWITCH\_ON\_END will switch IO process to another process after IO operation is finished while SWITCH\_ON\_IO will switch IO process to another process after IO process is blocked.