

Homework 1 – Chapter 4

- 1) The CPU utilization should be 100%. When we run with the flags -c and -p, we see that when any process is running, it is on the CPU as indicated by the "RUN:CPU" indicator. Furthermore, the stats show that the CPU is busy 100% of the time.
- 2) Since our default tick is 5, I would expect it to take 10 time units to complete. This is due to the fact that the IO process is called after the CPU process. Since processes that use IO will have to be paused at some point to facilitate the action of accessing IO, those paused time units are wasted, since the CPU process is already complete.

The -c and -p flags confirm this. We see that the total time used is 10.

- 3) The time is a lot lower. This is because we call the IO process first. While the process is waiting, the CPU process is free to do its thing. After it is done, we return to the IO process and it should be ready to be done.

The -c and -p flags confirm this. The total time is only 6.

- 4) With SWITCH_ON_END, the behavior is similar to question (2). The default behavior is to switch on IO, but now we chose to only switch on end. This means the CPU will not switch to another process when the current process is accessing IO. Thus the end result is the same, where ticks where the process is accessing IO is wasted.
- 5) The behavior will be similar to that of question (3). The -c -p flags confirm this. The time, CPU and IO busy stats are all the same. This is because we are using the default switching behavior, to switch when IO is being accessed. The waiting ticks are not wasted.
- 6) The IO process runs first, waits, then the other processes start running. However, the CPU will not switch back to the IO process that is waiting until the other non-IO processes are done, as indicated by the "IO_RUN_LATER" flag. Since our IO process will attempt 3 times, we can see that we have a ton of wasted ticks. Our system resources are not being effectively utilized.
- 7) Now, the CPU will switch back to the IO process once it is ready. This means we don't have any wasted ticks. This is similar to the difference between question (2) and (3), but just kind of multiplied. Our resources are being utilized much more effectively. It is a good idea simply just to reduce the time cost, and perhaps to improve snappiness. It would suck if a process would wait till every other non-IO processes to finish, when there are hundreds of processes going on.
- 8) There is no difference with the IO flags. This is probably because of the random nature of our instruction probability (CPU vs IO).
Between SWITCH_ON_IO and SWITCH_ON_END, switch on IO is still quicker, which is consistent with my observation on question (7).