

3.2

- a. The output to the console is a large bunch of A's then a large bunch of B's but never "ABA" or "BAB" due to the quantum being 50ms resulting in a context switch every 50ms
- b. At 25ms quantum, the amount of consecutive A's and B's is much shorter, but the reason why is the same (from a.). At 75ms quantum, the amount of consecutive A's and B's is much longer. This is due to the context switch happening either more often (25ms) or less often (75ms), depending on the time of the quantum.
- c. I let the XINU session run for a long while to see if the sndB process would ever artificially have a higher priority than sndA, but after discussing with Professor Park, I've realized XINU doesn't have artificial priority increase from amount of time waiting. To conclude: since the priority of A is higher than B, sndA will always be running over sndB.
- d. Before "process two" (sndB) is created, "process one" (sndA) has a chance to run and A is printed for one quantum, but once "process two" is created, the priority is higher than "process one" so it will always be running over "process one". (I understand it's not pid 1, solely name of the process)
- e. After main is created: sndA is created and runs, then main runs, then sndB is created and runs, then sndA runs, then main runs, which changes the priority of sndA to -3, and then B prints forever because processes are queued by priority, and sndB is higher than sndA

Bonus

I have created a system call that instead of dumping the registers to the console like Xtrap, this system call (named procinfo) 'dumps' all the process information to the terminal. Its input is pid32. It can be important to determine many things very easily, especially when a shell is unavailable, such as: creating processID, stack size, state, semaphore it is waiting on (useful for debugging) etc. There is error checking to ensure that the call fully executes if and only if the PID is linked to a valid process (a.k.a. not PR_FREE). When it has completed it returns OK. It is saved in bonuslab2.c under system/.