

3.3

I chose to make modifications in `resched.c` because I found from the control flow of XINU that `resched.c` is where the context switch occurs.

5.2

I observe that all the 5 process (CPU bound) print similar `x` values (in the range of 825588495, 825577283, 825578189, 825578634, 825578694 for all 5 processes respectively) and the cpu time is 29.026 seconds which happens to be the time the while loop runs for.

5.3

Same `x` values for all 5 process are created (1450) and this value is way less compared to the CPU bound process. The cpu time for all the 5 process will be the same. The 5 processes are created in 29 seconds.

5.4

Same values for `x` are posted for cpu bound and io bound processes. The values for `x` for CPU are process are significantly higher than the IO processes. The IO process also execute first because they have higher priority.

5.5

I estimated that all 5 process would receive $29000 + 5$ ms of CPU time however we observe that the first process receives 29019 ms of CPU time followed by 29026, 29031, 29060 and 29065 ms.

Bonus problem: `bonusapp.c`

By printing priority of the process in test case B and C we will observe that the priority of the IO bound processes is higher than the CPU bound processes. This means that the IO bound process will be given more CPU time which is what we want. Thus, in the bonus problem I have printed the priority of the IO bound process the we create.