1. I used the synchronization primitives condition variables and locks. The condition variable “freebowls” I used to make the thread wait until there was bowls availables and no “enemies” eating. I used the locks to protect the critical conditions. I used several other global variables: “catsturn”, a binary value I used to determine if it was the cat’s turn to eat or not. “cateating” and “mouseeating” were integers that I used to determine the number of mice or cats eating at any time. I also had “static int *bowls”, a global array I used to determine which bowls were occupied – 1 if occupied and 0 if not.
2. The condition variable “Freebowls” was used to make the thread wait until it was safe for the organism to eat. Furthermore, once several members of the species got their turn to eat, freebowls would broadcast to the “enemy” species, letting them know it was their time to eat. This ensures that both sides get to eat. The lock “mutex” was used to support this condition variable. “Catsturn” was used as a switch to determine which species’s turn it was to eat. “Cateating” and “mouseeating” was used for 2 purposes: to determine if there were any enemy species on the field and to determine if it should let the other species eat.
3. It is not possible for 2 species to eat at the same time because I have the switch variable “Catsturn”, which is either 1 or 0. The cat_simulation thread will only run if “Catsturn” is 1 and the mouse_simulation thread will only run if “Catsturn” is 0.
4. It is not possible for mice to be eaten by cats because 2 species will never eat at the same time. Furthermore, the mouse_simulation thread will not run as long as the cateating > 0. This ensures that mice will never encounter cats.
5. Cats nor mice will not starve under my synchronization technique because it always ensures that the other species will be given a turn using several techniques described in this PDF.
6. My synchronization is quite balanced and fair for both species. What I did was (in the case of cats): If (NumCats == Cats eating at the moment OR NumBowls == Cats eating at the moment) then stop allowing more cats to eat and give the mice a turn to eat. This means that, if every single cat has eaten in a round, or every single bowl has been used, then the species will switch.