CSE 344

HOMEWORK 3

BERKE BELGİN

171044065

In the assignment I used named semaphores, shared memory and fifos for synchronization. I used allocated shared memory as an integer array to not struggle with char arrays, segmentation faults etc. Given fifo file must contain distinct file names. When program is executed after parsing command-line arguments it first opens semaphores and shared memories and then chooses itself nth fifo file inside the fifopaths file from top to bottom where nth is determined by checking the index of the last process inside the shared memory who have chosen a fifo file for itself (For instance, the firs process created will check the last index and wil get 0. Then it will increment the index inside the shared memory to 1 and reads the path of 0^{th} fifo from fifopaths file). It increments the index inside the shared memory and proceeds to fifo opening section.

In order to synchronize fifo openings and to prevent deadlocks while opening fifos, I forced all processes to follow the same order while opening fifo files either for reading or for writing. For example, lets say we have files, fifo1, fifo2, fifo3. All the processes will first open fifo1. If it is the owner process of fifo1, it will open it for reding if it is it will open it for writing. After every process completes its opening connection phase, if the process has a potato it will send it to a random process through its fifo and then wait for potatos to receive.

When a process receives a potato it decrements the hotness value of that potato by one inside the records in shared memory and sends it to another process if it is not cold yet. If it is cold it checks if all the potatos are cold now. If it is, it sends exit message to all fifos to leave the program and exits, if it is not then it does nothing.

After receiving exit signal, every process frees the resources it use and terminates the program.