Homework 2: Dispatcher/Worker model with Linux pthreads

Submitters:

Omer Bahary 315329946

Almog Cohen 308223585

GitHub link: https://github.com/omerbahary/os_homework/tree/main/hw2_315329946_308223585

Solution notes:

- We started thinking about what way is the best for this model, and learned a lot along the way. We started with the approach of filling in the work queue, and then start the threads and let them take care of the pending jobs. When we got to the “dispatcher wait” part we have realized that the right way is that all of the threads are being initialized at the start of the program, “listening” and waiting for incoming jobs.

- There is an extent usage of mutex`s and synchronization mechanisms, making sure not to waste any CPU cycles.

- **On a high level this is the mechanism of the program:** dispatcher initialize threads, then reads command file. If it is a worker command it is being added to the queue, else it is a dispatcher command and it handles it. After the dispatcher finished reading the commands in file, a clean-up process is running to make sure free all allocated memory and let the threads exit gracefully using “pthread join”.

To sum up, it was a hard but very meaningful exercise, and we enjoyed the learning process. It also combines everything we learned so far regarding C language – pointers, NULL instances handling, structs, header files, declarations, etc. We feel that we understand the topics of threads and particularly the shared resources issues that can occur, and how to handle them.