

K-2 Kernel Specification

Authors

sgaweda(20143503) and spratap(20638036)

Make and Run

Make – Run make in the {cloned repo}/k2

Run – In GTK terminal run load ARM/spratap/k2.elf

Changes in the kernel

In this kernel we have implemented the sending and receiving of messages based on Send Receive Reply pattern which provides both synchronization and data access to shared resources.

Our kernel now supports the functions – Send, Receive and Reply which it uses for inter-process (task) communications. It also has a server for playing Rock Paper Scissors.

For blocking the tasks on the various queues, we use Ring Buffers with FIFO functionality, same for the Rock Paper Scissor.

For the name server, we just copy the name onto the Task Descriptor of the process itself!

Git Repository

<https://git.uwaterloo.ca/sgaweda/cs452-rtos-aka-trains.git>

Priorities chosen for game tasks

For the priorities chose for the game tasks, we chose to set them lower than the server itself as this helped run the games each time the server received on them.

Output produced by our game tasks

Since we create 6 different tasks and they play only 1 fixed move for 3 times and then quit, hence the results of our games were quite straightforward.

First up we had the match between Paper and Rock in which Paper won every time.

Second, we had the match between Rock and Scissor in which the Rock won every time.

Finally, we had the match between Paper and Scissor in which the Scissor won every time.

Performance measurements

We ran Timer 3 at 508 KHz and did 100000 iterations of SRR (using the sendTask and receiveTask functions in TimingTasks.cpp). After that we got our results on various settings and the numbers have been posted in a file called performance.txt in the repo itself as well.