Design Doc

The DesignDoc file that contains the description of the major features of your program that is not

specified in the project specification, including

The list of program files you have altered, and the changes you made in each program file that

are beyond the “\*\*\* ADD CODE” segments,

Altered programs:

* All files required the #include <unistd.h>.
* System.c-added calls for start\_client\_reqhandler for the server socket functionality and end\_client\_submission to terminate sockets.
* Term.c- changed terminal\_output to take in int sockfd in order to send back to client. terminal\_output function no longer creates terminal.out as per specs and instead sends the sockfd and character pointer to send\_client\_result function found in process.c. Changed the struct of TermQnode to contain a character array instead of character pointer. Synchronization was achieved with insert\_termio and handle\_one\_termio as both are contained within mutexes in order to achieve mutual exclusion with terminal queue contents. Insert\_termio furthermore posts semaq to allow for handle\_one\_termio to go past the sem\_wait(&semaq) found in the while loop of termIO(). start\_terminal inits semaq to 0 and mutex to 1 as so handle\_one\_termio will not loop on an empty queue. End\_terminal function posts to &semaq in order to for the thread to exit termIO() to join.
* Submit.c was changed in order to accommodate reqhandler’s functions and to initialize sockets. Start\_client\_reqhandler replaced submission manager initialization called in system.c and creates a thread to manage socket client interface.
* Process.c was changed for context\_out in that CPU register information and execution status as well as process id were saved into the process control block. Furthermore, code was added to added to run the idleprocess by changing it’s execution status in the PCB and context switching in, with an added timer and accommodate execution status change to eReady which would lead to context switching out the idle process.
* Cpu.c was changed for handling opcode cases. Appropriate breaks were added.
* Client.c was altered accordingly to spec to not require the submission of a client id as it is assigned a process id. Client.c no longer continues onto next file name request after one read from the socket as multiple print statements should be read per file.
* Simos.h-small changes from original, such as reconciling function names and function parameters. Consolidated new added function names and parameters. PCB structure was altered slightly as sockfd of the client was added.
* Code from Project 1 was obtained from standard original files.
* Issues: This was not in the specifications, just personal feelings- client does not terminate with simos.exe. Could be fixed for client.c using threading if I had time to be able to detect server termination and thus automatic client self-termination. Wanted to get rid of the warnings from queue.c (unchanged from the original file).