Assignment 2 Report

Ismail Faruk 260663521

Introduction

This assignment was about implementing a key-value store in shared memory using hash mapping algorithm to position the key and values and using semaphores to read/write to that shared memory space.

unsigned long hash(unsigned char \*str) – This function was used to pass the key value as argument and returned a hashed integer to map the key and its value into the shared memory.

int kv\_store\_create(char \*kv\_store\_name) – This function was used create a new shared memory, or write to an existing shared memory with the same name. We also passed the beginning of the shared memory space into a variable for further use.

int kv\_store\_write(char \*key, char \*value) – This function was used to access the existing shared memory, use hash map and pointer shifts to access the space of the key passed as argument and copy the values of key and value into the appropriate space.

char \*kv\_store\_read(char \*key) - This function was used to access the existing shared memory, use hash map and pointer shifts to access the space of the key passed as argument and copy the values of key and value into the appropriate space.

char \*\*kv\_store\_read\_all(char \*key) – This function was used to return all the values that was represented by a particular key value. Like kv\_store\_read, whereas, this time it returned all the existing value in the pod.

int kv\_delete\_db() – This function acted as a cleanup function for memory unmapping and unlinking semaphores.

int bookkeeping() – This function was used to initialize semaphores, certain global variables and the global SharedMemoryStruct object SMS.

Conclusion

I wish I had started earlier. There was more to learn in this than I got to.