The purpose of this program is to create NS semaphores in a semaphore set, where NS is an integer value between 1 and 100 (100 was an arbitrary chosen number). There are two ways to try to attempt to create the semaphore set. The first way is to remove any existing semaphore set prior to creating a new set and the second way is to attempt to create a new semaphore set without removing a currently existing set. If the second method is chosen and the set already exist, the program will display an error message that the semaphore set already exist. Either ‘r’ or ‘n’ is the second argument on the command line, and the value for NS is the third argument. The character ‘r’ stands for “remove existing semaphore” and the character ‘n’ stands for “do not remove existing semaphore”. The fourth argument through the NS + 3 argument are the values given to each individual semaphore in the semaphore set. The number of values passed must equal NS. For example, if NS = 2, then only 2 values can be passed. If 2 values are not passed, the program will display an error message that the arguments do not match.

Each individual semaphore value is initially placed into an array called sem\_array. This array will later be used to initialize each semaphore in the set. After the semaphore values are placed into the array, the program will check for any existing semaphore sets that have been created with the given key value, called ipc\_key. The key is generated from the ftok UNIX function. The program uses the semget UNIX function to return any existing semaphore identifiers that have been created with ipc\_key. If the semaphore identifier does not exist, semget returns a value of -1; otherwise it returns the semaphore identifier. If a semaphore identifier associated with the key already exists, the program will respond in two different ways, depending on whether ‘r’ or ‘n’ was given as an argument. If ‘r’ was chosen, the program will remove the existing semaphore set. If ‘n’ was chosen, then the program will not remove the existing semaphore set. The program will then attempt to create a new semaphore set, but if the semaphore set was not previously removed (currently exist), then the program will display an error message that the semaphore set already exist. If the semaphore set was removed, then the program creates the semaphore set. The semget function with flags IPC\_CREAT and IPC\_EXCL are used to create the semaphore set and return an error message if the set already exist. Once the semaphore set has been created, the program will display the semaphore identifier. After the semaphore identifier is displayed, the semctl UNIX function (this function requires the use of a union of type semun, called arg in the program) uses the integer command value, IPC\_STAT, to copy the contents of the semid\_ds structure, sum\_buf, identified by sem\_id, into arg.buf. The program then prints the time and date the semaphore set was created. The remainder of the program uses the UNIX function semctl in order to initialize and view the existing semaphore set. First, the function uses the integer command value, SETALL, to initialize each semaphore in the set with the values held in sem\_array. After all the semaphores have been set, the function uses the integer command value, GETVAL, to display each individual semaphore value. The output to the program is the semaphore identifier, the date and time the semaphore set was created and the value given to each individual semaphore in the semaphore set.