1)

Since the device file name is currently stored as a global environment variable, you are able to load it when you start your program. In order for all programs to work on the environment, the program must pull the data from the environment. So to do this, when the program starts, it will load the device file name from the environment, to use for i/o operations.

2)

Lots of the implementation between the System-V, and POSIX memory sharing functions are the same. The changes in the program need to be made when dealing with semaphores and memory sharing. To start the memory sharing, we should use shm\_open() instead of shmget(). We should use mmap() to share the memory space instead of shmat(). To unlink the shared memory, we should use shm\_unlink() instead of shmdt(). Creating and closing the semaphores should be done by using sem\_open() and sem\_close(). There are various functions to manipulate the semaphore in the POSIX api.