Jeff Morin

CISC 3320

Fall 2017

HW2

1. Author: Jeff Morin. I was the sole author of the assignment. My role was utilizing the ProcessIDManager code written in the previous assignment, to allocate PIDs to threads in the driver file. I also made a custom class that inherits attributes from a thread to further customize the way the threads (processes) allocate and release PIDs.

2. The problem was basically to create a multi-threaded program that utilizes the PIDManager class created from HW1. Several threads would request a PID if one is available, sleep for some time for the next thread to allocate a PID. When the time is elapsed, the re-awoken thread would release the PID, and then terminate.

3. I assumed that the creation of so many threads to request IDs from the PIDManager would be sort of unwieldy, considering there wasn’t any synchronization mechanisms involved with accessing the object. I also assumed that simply creating threads using the base Thread class provided by the Java API wouldn’t be sufficient to properly represent the threads, or “processes” read about in chapter 4. Therefore, I extended the Thread class in a class I made called Process, and overrode the methods *interrupt()* and *getId()*; as *interrupt()* would simply kill the thread without notice, and *getId()* would return the default ID assigned by the JVM. So I overrode *interrupt()* to properly notify a thread death, and the *getId()* method to return the proper PID assigned by the PIDManager class.