Preston Peck

Prof. Simina Fluture

Distributed Systems 344 Mo/We

April 26, 2017

**Homework 4: Pipes**

A. **ConnectionManager** starts by creating **2 PipedInputStreams**, **2 PipedOutputStreams**, 1 **ObjectOutputStream**, and 1 **ObjectInputStream**. The reasoning for this is because there will be **2** transfers, and each transfer requires a pipe which is comprised of **2** parts. Each **PipedInputStream** (destination) will be connected to a **PipedOutputStream** (source). The **ObjectOutputStream** and **ObjectInputStream** will be used because objects will be sent between threads. They will attach to a pipe of a corresponding type (I/O) when necessary to either extract or inject a **Message** that will be comprised of a string (mess), a string array (line), and an integer (num). No matter how many transactions, the number of **ObjectOutputStreams** and **ObjectInputStreams** can remain as low as **1** each because they can be recycled and used again whenever an any object moves. After **ConnectionManager** creates all of the objects it needs and makes the appropriate connections, a **Sender** and **Receiver** thread are started. One of the first things **Sender** does is look to see whether or not there is a **Message** waiting for it in the **Receiver->Sender** pipe using an **ObjectInputStream**. If it doesn’t see anything, it **blocks** until it does. **Receiver**, either before or after **Sender blocks**, creates a **Message** to write to the **Receiver->Sender** pipe using an **ObjectOutputStream**. Afterwards, **Receiver** sends an **integer** to **Sender**, which it had been **blocking** for following **Receiver’s** **Message**, and this time **Receiver** will **block** as it waits for a response from **Sender**. At the point where both threads have entered their respective while loops, **Sender** rapidly sends text from an input file character by character to **Receiver** via the **Sender->Receiver** pipe, and **Receiver** in turn tries to reconstruct **Sender’s** text piece by piece as they trickle one by one in from the very same **Sender->Receiver** pipe. The text is eventually printed to console and the cycle continues until the specified values are read in.