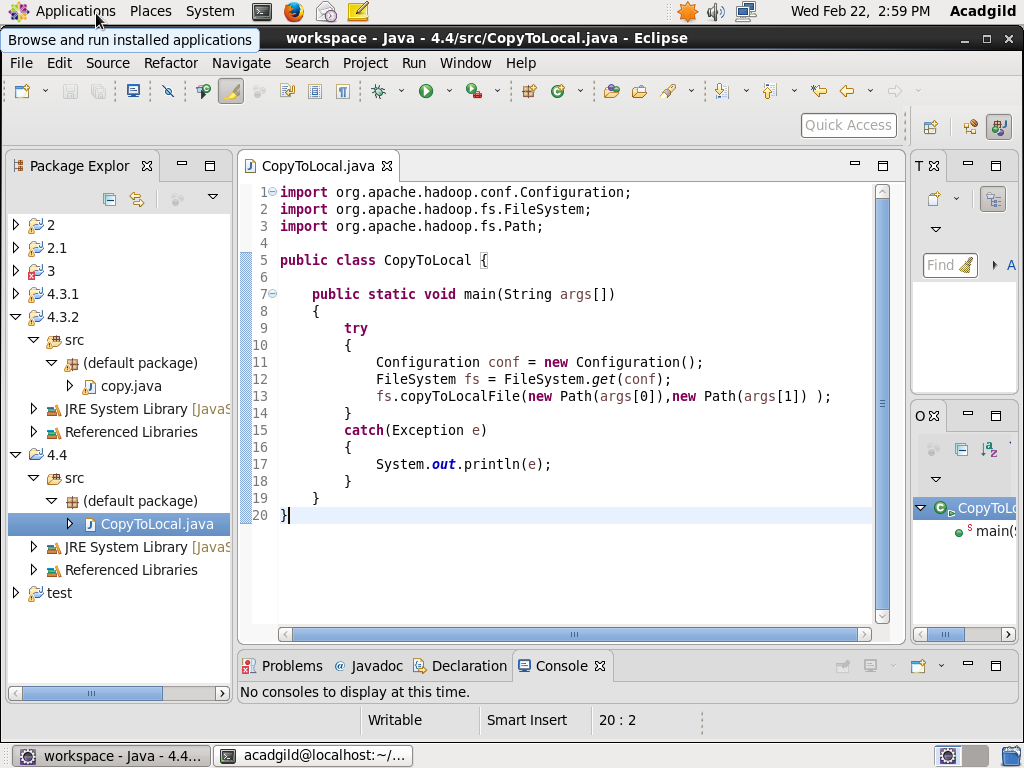
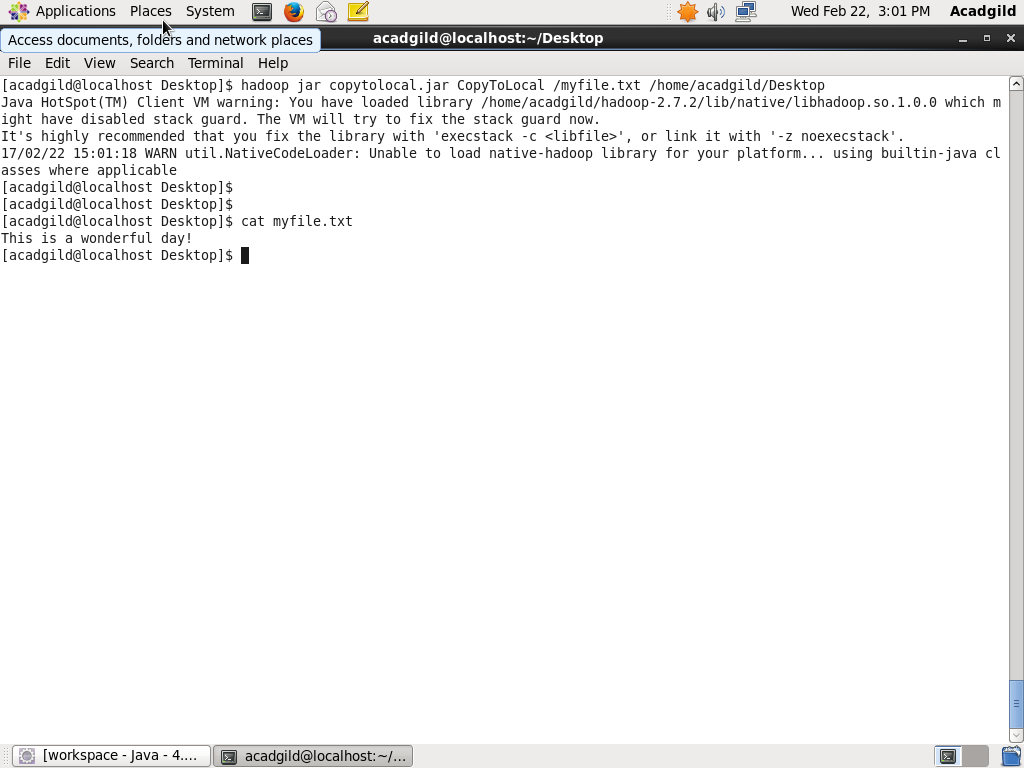
Q1-Write a Java program to copy a file from HDFS to LFS.

PROGRAM:



Output :



Q2-2. Explain the importance and usage of the below terms in details

● DFSInputStream

● DFSOutputStream

● FSDataInputStream

● FSDataOutputStream

FSDataInputStream:

1. The DistributedFileSystem returns an FSDataInputStream (an input stream that supports file seeks) to the client for it to read data from. FSDataInputStream in turn wraps a DFSInputStream, which manages the datanode and namenode I/O.
2. A data input stream lets an application read primitive Java data types from an underlying input stream in a machine-independent way. An application uses a data output stream to write data that can later be read by a data input stream. In Hadoop the same relation applies to FSDataInputStream and FSInputStream

DFSInputStream:

1.When the client calls read() function, the DFSInputStream stores the address of first few blocks in the file, then connects to the first (closest) datanode for the first block in the file.

2. Data is streamed from the datanode back to the client, which calls read() repeatedly on the stream.

3. When the end of the block is reached, DFSInputStream will close the connection to the datanode, then find the best datanode for the next block.

4. It will call the namenode to retrieve the datanode locations for the next batch of blocks as needed. When the client has finished reading, it calls close() on the FSDataInputStream.

5. During reading, if the DFSInputStream encounters an error while communicating with a datanode, then it will try the next closest one for that block.

6. It will also remember datanodes that have failed so that it doesn’t needlessly retry them for later blocks.

7. The DFSInputStream also verifies checksums for the data transferred to it from the datanode. If a corrupted block is found, it is reported to the namenode before the DFSInput Stream attempts to read a replica of the block from another datanode.

FSDataOutputStream:

1. The DistributedFileSystem returns an FSDataOutputStream for the client to start writing data to.
2. Just as in the read case, FSDataOutputStream wraps a DFSOutput Stream, which handles communication with the datanodes and namenode.

DFSOutputStream:

1. As the client writes data (step 3), DFSOutputStream splits it into packets, which it writes to an internal queue, called the data queue.
2. The data queue is consumed by the Data Streamer, whose responsibility it is to ask the namenode to allocate new blocks by picking a list of suitable datanodes to store the replicas.
3. DFSOutputStream also maintains an internal queue of packets that are waiting to be acknowledged by datanodes, called the ack queue.