**PROJECT REPORT**

****

Submitted to

**Dr. Lawrence J. Osborne**

By

Nidhi S Patel(L20410813)

Bibek Paudel(L20398654)

**Acknowledgment**

We are very much thankful to Dr. Lawrence J. Osborne for giving us a great opportunity to learn and implement various topics like concurrent thread programming, RPCs, synchronization through message passing, implement the mutual exclusion by using RPCs, and the problems caused by time logical of transactions. We are grateful for providing us with all the facilities and guiding us. We have gained a valuable amount of knowledge and information by working on the project “Replicated Reliable Banking System”.

**Abstract**

The aim of the project is to develop fault-tolerance, resynchronization using logs and reliable. In this system, there are three replicated servers and coordinators. The main purpose of the coordinator is to act as an interface between the bank servers. And we use Python 2.7 to build this reliable banking system and we implemented on Linux Environment. Each client sends its request to coordinators which forward the request to the all three servers. And all the data are stored in each server’s log file. Which will update every time.

**Introduction**

In this the basic understanding and motivation behind the project along with overview of the technical report for this project.

* **BACKGROUND:**

Replicated Reliable banking system is the system based on all the accounts of the bank are replicated on three servers. And all the information of accounts is to be stored and updated in that server’s log file. Each server will maintain a log for each operation which are performed. When any server will fail and it will in working condition then it needs to resynchronize with the other servers to restore a consistent state for each account.

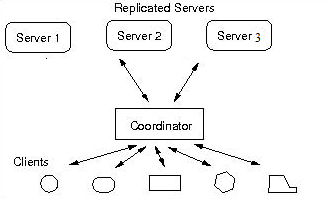
The system will use the bank database is distributed and replicated.

* **PROJECT DESCRIPTION**

This project is developed in a distributed environment. A client and a server process is implemented in a Linux machine. They will communicate with each other using UDP sockets using Python 2.7. There are three replicated servers in the system and that will coordinate with each other using coordinator file and in the client part client will operates by deposit some amount, Balance, and withdrawal commands.

**DESIGN AND IMPLEMENTATION PART**

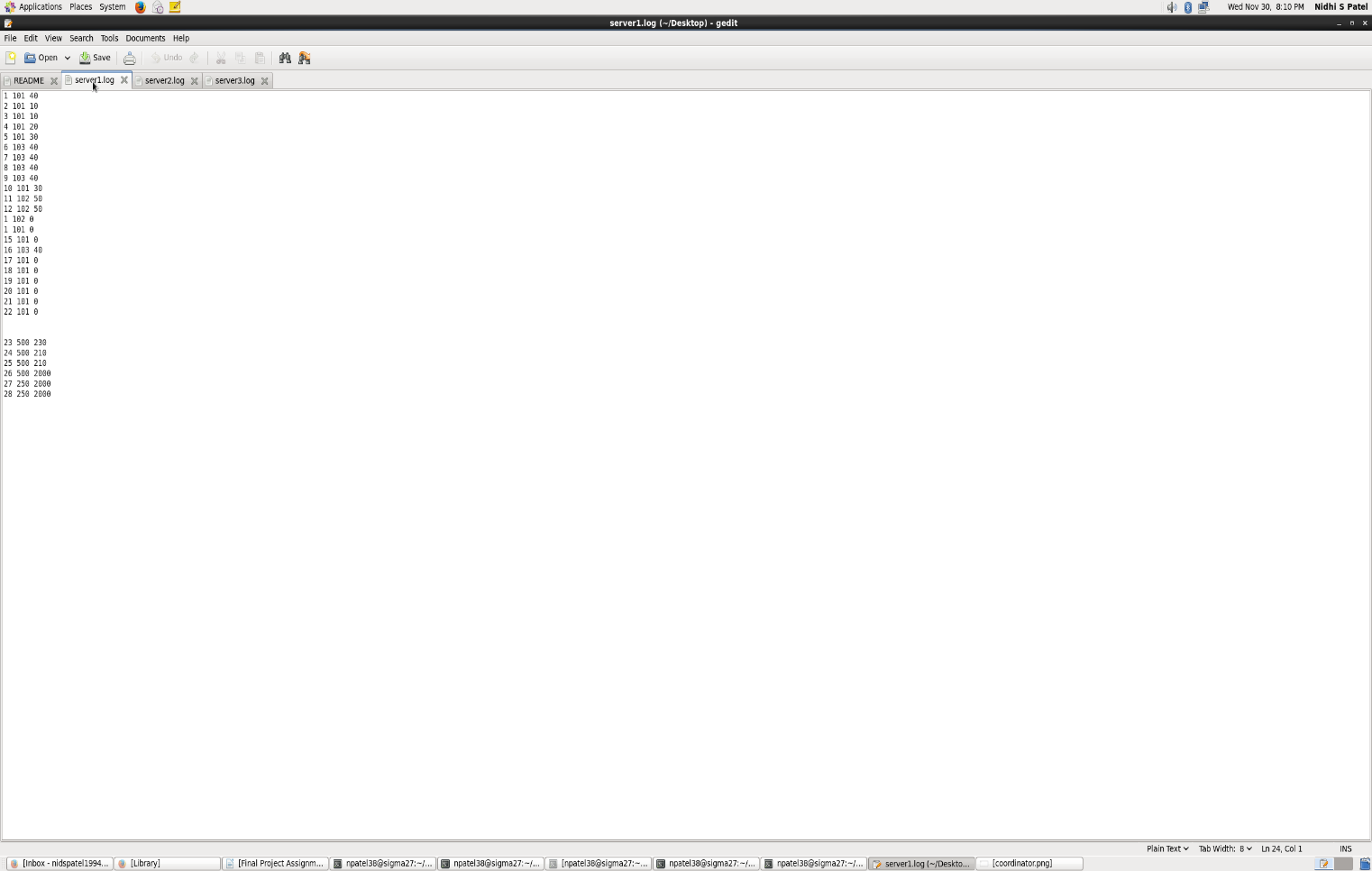
Following one is diagram, from which we created system



* **HARDWARE**
* Five computer (Three for replicated server, one for coordinator and one for client)
* **SOFTWARE**
* Python 2.7

**IMPLEMENTATION:**

* **REPLICATION:** All accounts of the bank are replicated on three servers. For the assignment assume that all accounts are created. Further, information of accounts is to be stored and updated in a file. If one server will crash and there are some data in that server when this server will wake up and that server have all the data otherwise it will take updates from server 2 and server 3.
* **LOGGING:** Each server maintains a log for each operation performed, the log must be stored/updated in a file for persistence. Each server uses a different log file. They do not share a single log file. Following is screenshot of logging.



**FIG 1 – server 1 log file**



**FIG 2 - server 2 log file**

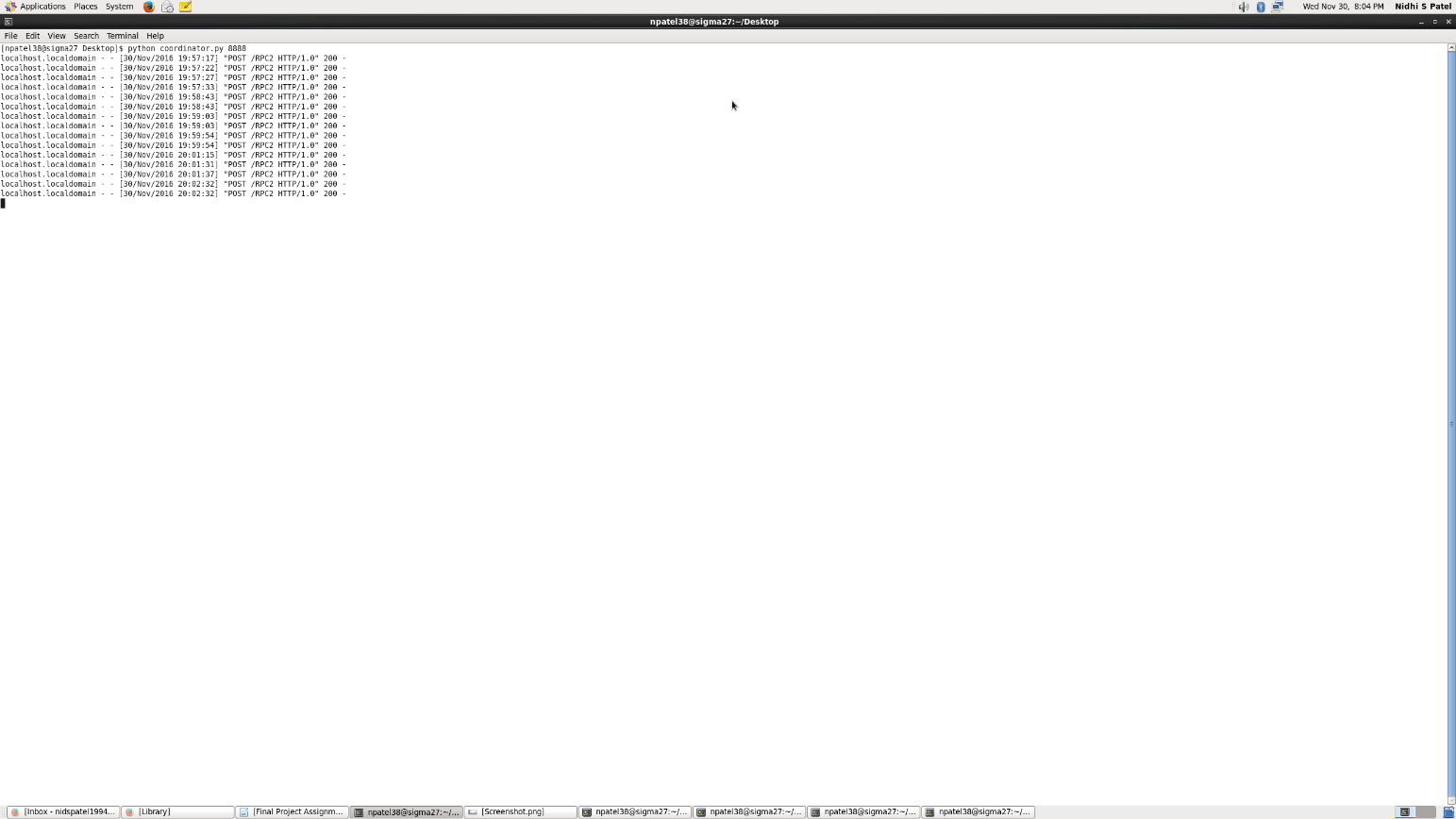


**FIG 3 - server 3 log file**

* **RESYNCHRONIZATION AND WAKEUP:**When a server fails and wakes-up it needs to resynchronize with the other replicas in-order to restore a consistent state for each account. When we will wakes-up its send ALIVE.

The server sends a resynchronize command to remaining server with latest operation number in log file of that server. The server had not failed then look up their Account number and balance.

* **COORDINATOR:** Each client request is forwarded to the coordinator, which operates replicated server. Following one is screenshot of coordinator.



**FIG 4 – coordinator system**

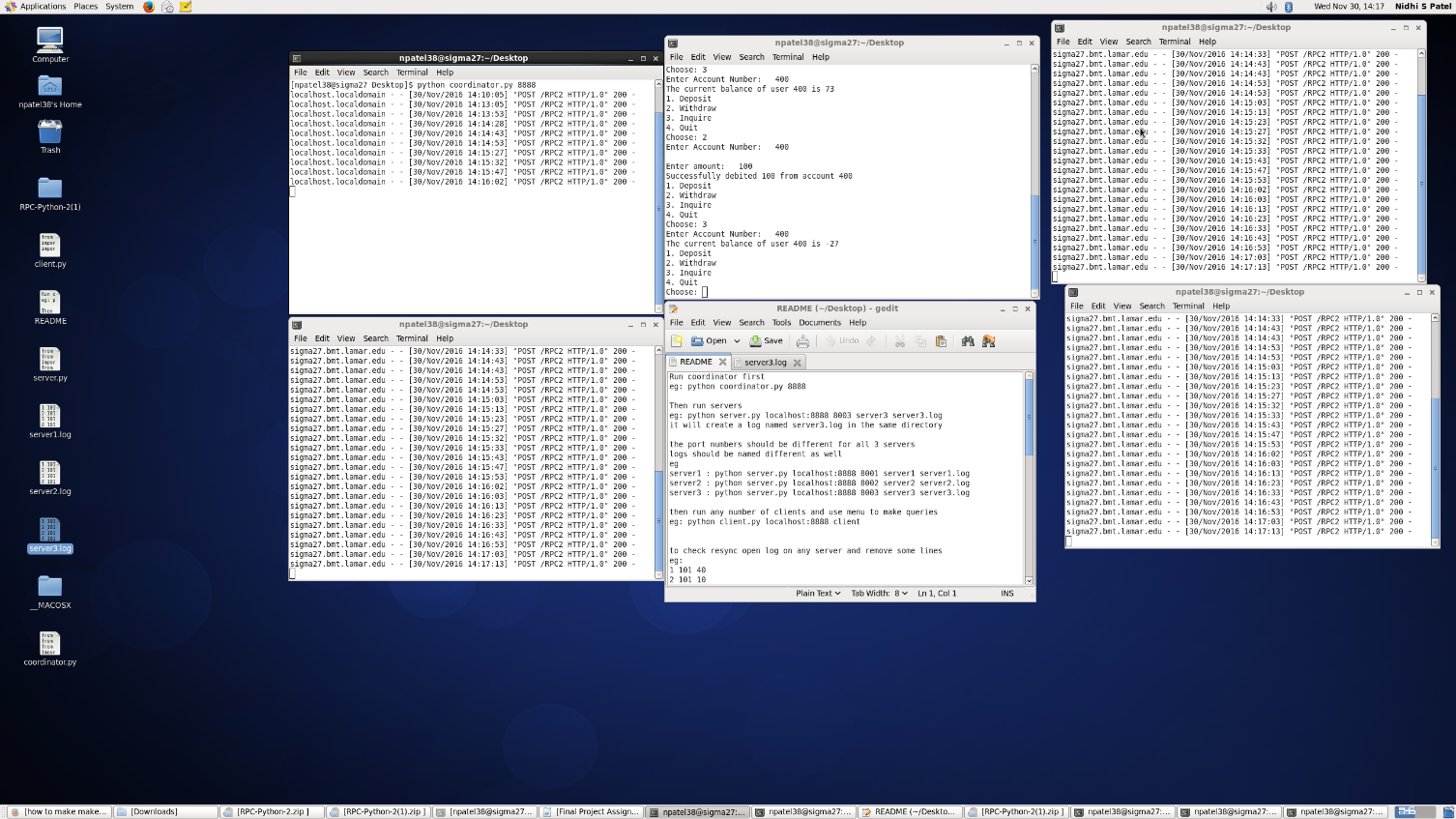
* **HOW TO RUN/COMPILE:**
* First, we must run Coordinator file. Write in command prompt “**python coordinator.py 8888”**
* Then run Server file and create log file of three servers.

For server1: python server.py localhost:8888 8001 server1 server1.log

* After this run client file.

“**python client.py localhost:8888 client**”

**SCREENSHOTS:**

****

**FIG 5**

**CONCLUSION:**

The above shown outputs show that the program works as expected. Being able to do this assignment, I feel we’ve gained valuable knowledge of Python socket programming, connection protocols, UDP, TCP and understood banking operations in general. Most importantly we understood concepts of distributed systems such as failure handling, fault tolerance, replication and logging and implemented them in the project.

**REFERENCE:**

* <https://www.python.org/doc/>
* Python network programming Tutorial:<https://www.tutorialspoint.com/python/python_networking.htm>