***Project Proposal***

**Project title:**

Exploration of Multiprocessing and Multithreading in Python.  
  
  
**Group members**

Ersum Moghis(60616)

Omer Shahab(60154)

Saad Siddiqui (60181)

Usama-Ul-Haq(60608)

Raheel Saleem (60682)

**Project description:**

**Multiprocessing: (Omer)**  
In computing, a mode of operation in which two or more processors in a computer simultaneously process two or more different portions of the same program (set of instructions). Multiprocessing is typically carried out by two or more microprocessors, each of which is in effect a central processing unit (CPU) on a single tiny chip. Supercomputers typically combine thousands of such microprocessors to interpret and execute instructions.

**Multithreading: (Saad)**

Multithreading is a type of execution model that allows multiple threads to exist within the context of a process such that they execute independently but share their process resources. A thread maintains a list of information relevant to its execution including the priority schedule, exception handlers, a set of cpu registers, and stack state in the address space of its hosting process.

**Advantages of Multithreading: (Usama)**

There are many advantages some of them are as follows:

* **Improved performance and concurrency**  
    
  For certain applications, performance and concurrency can be improved by using multithreading and multicontexting together. In other applications, performance can be unaffected or even degraded by using multithreading and multicontexting together. How performance is affected depends on your application.
* **Simplified coding of remote procedure calls and conversations**

In some applications it is easier to code different remote procedure calls and conversations in separate threads than to manage them from the same thread.

* **Reduced number of required servers**  
    
  Because one server can dispatch multiple service threads, the number of servers to start for your application is reduced. This capability for multiple dispatched threads is especially useful for conversational servers, which otherwise must be dedicated to one client for the entire duration of a conversation

**What We Do:**In Beginning linux Programming there are examples of multithreading(chapter 12) and multiprocessing(chapter # 11), which are written in language C, so we will implement those examples in python with explanation.

**Implementation till interim presentation**:

We will do atleast one example of multiprocessing(chapter # 11).

**Final demonstration:**

We will do the remaining examples of multiprocessing and multithreading.