Homework 4 - Q4:
Provide a write-up on a multi-threaded application that you would like to implement as part of the final project

Topic - Health Data logging and Storage system.

Description - Health Monitoring application (like FitBit) which tracks heart rate and foot steps and sends this data using socket communication to logger as well as storage system. Backup of this data to be maintained on a USB drive.

TIVA Tasks -

- Heart Rate Monitor task (Sparkfun AD8232)retrieves value from the heart rate sensor.
- ADC conversion task, converts analog data received from heart rate sensor to digital counterparts.
- Pedometer task (Sparkfun LSM6DS3) retrieves data from pedometer sensor (I2C communication).
- Socket task to send above data values over TCP sockets.
- Message API task, that creates the data payload.
- Main task to monitor all above tasks and create them.

BEAGLEBONE tasks -

- Socket task to receive data over sockets.
- Logger task to log this data into multiple files depending on what data it is (heart rate or footstep count).
- Notification task, which notifies user of his health condition based on calculations of threshold values. It also gives him history of his footstep count(distance travelled effectively) over a period of time.
- Storage task which saves above logs to USB pen drive. This task communicates with a USB driver implemented using loadable kernel modules.
- Main task to monitor all above tasks and create them.

Components:

1. BeagleBone Green
2. Linux OS on BBG
3. TIVA Board
4. Free RTOS on TIVA
5. TIVA ware HAL library
6. Heart Rate Monitor task (Sparkfun AD8232)
7. Pedometer task (Sparkfun LSM6DS3)
8. USB (2.0/3.0) Pendrive

9. BSD Sockets API

Data structures -

```
typedef enum i2c_states
{
  SUCCESS,
  ERROR_READ,
  ERROR_WRITE,
  ERROR_OPEN,
  ERROR_ADDRESS,
  ERROR_VALUE,
  NULL_POINTER,
}i2c_state;

typedef enum loglevel
{
        ALERT,
        WARNING,
        INITIALIZATION,
        INFO,
}LogLevel;

typedef enum{
  PULSE_TASK,
  STEP_TASK,
  ADC_TASK,
  PARENT_TASK
}tivaSources;

typedef enum{
  LOG_PULSE_TASK,
  LOG_STEP_TASK,
  NOTIFICATION_TASK,
  STORE_DATA_TASK,
  PARENT_TASK
}beagleboneSources;


typedef struct logger
{
        Sources sourceId;
        reqCmds requestID;
        LogLevel level;
        time_t timestamp;
        char payload[100];
}LogMsg;
```

Mechanisms -

Userspace Mechanisms -

- Sockets API for TCP sockets
- TIVA ware HAL library for interfacing with all sensors
- I2C libraries to define our own I2C communication API
- Pthreads API(mutex, condition variables) to synchronize communication between multiple thread tasks

Kernel -

- USB Linux Kernel Driver
- Socket system calls (open, close, read, write)
- Semaphores to synchronize any kernel module data