CS 349

Assignment 2: Socket Programming

Problem 3: Calendar Server

Name : Nitesh Singhal
Roll No. : 10010139
Date : 27th February, 2013

ASSUMPTIONS:

- Date format is assumed to be in "MMDDYY" format.
- The "YY" in date format assumed to be years in the twenty-first century.
- Used strptime function from ctime library which considers month number between 1-12, day of month between 1-31 & year between 0-99.
- Time format is assumed to be in "HHMM" format.
- When a client is getting events from getall command, at that time no events are getting expired.
- No change in entries (addition, removal or updation) happens when one of the client is running getall.

HOW TO COMPILE:

Just run "make" command in terminal. This will compile all the three server files and the client file.

DIRECTORY STRUCTURE:

- Root directory contains two directories namely Client & Server & three files namely definitions.h, Makefile & README.pdf.
- Server directory contains three different implementations of server file namely server.c, server_select.c & server_multithreading.c
- Client directory contains client.c

PROCEDURE OF EXECUTING CODE:

1. Add a new calendar event:

```
./client hostname port myname add date start_time end_time type_of_event
```

2. Remove a calendar event:

```
./client hostname port myname remove date start_time
```

3. Update an existing calendar event:
./client hostname port myname update date start_time end_time type_of_event

4. Get the events for a specific time or time range:

a. To get all events of a specific user of a specific date & specific start time
./client hostname port myname get date start_time

b. To get all events of a specific user of a specific date
./client hostname port myname get date

c. To get all events of a specific user
./client hostname port myname getall

SAMPLE INPUT/OUTPUT:

When ./server file is running on port 1100

1. Adding an event
Input: **./client localhost 1100 nitesh add 121213 1111 2222 study**
Output: **Event Added!!**

If we get a conflict in events then
Output : **Conflict detected!**

2. Removing an event
Input: **./client localhost 1100 nitesh remove 121214 1111**
Output: **Event Removed!!**

If for user nitesh & date 121214 no event has start time 1111 then
Output: **None of event removed!**

3. Updating an event
Input: **./client localhost 1100 nitesh update 121214 1111 2221 study**
Output: **Event Updated!!**

If no user/date/start_time event matches the inputed data then
Output: **None of event updated!**

If conflict occurs during updation
Output: **Conflict detected!**
None of event updated!

4. Getting event(s)
Input: **./client localhost 1100 nitesh get 121214 2222**
Output: **Event Date : 121214**
Event Start Time : 2222
Event End Time : 2224
Event Type : study

Input: **./client localhost 1100 nitesh get 121214**
Output: **Event Date : 121214**
Event Start Time : 1111
Event End Time : 2221
Event Type : study

Event Date : 121214
Event Start Time : 2222
Event End Time : 2224
Event Type : study

When ./server_select file is running on port 1100

Only getall is added functionality in server_select. Rest of operations inputs are same as in ./server file.
1. Getting all events of a user
Input: **./client localhost 1100 nitesh getall**
Output: **Event Date : 121214**
Event Start Time : 2222
Event End Time : 2224
Event Type : study

```
Event Date : 121215
Event Start Time : 2222
Event End Time : 2224
Event Type : study

Event Date : 121213
Event Start Time : 2222
Event End Time : 2224
Event Type : study

Event Date : 121213
Event Start Time : 1122
Event End Time : 1224
Event Type : study
```

We will get each event at every 2 seconds delay.

If date check return false then
Output: **Syntax error : Date should be in MMDDYY format.**

If time check return false then
Output: **Syntax error : Time should be in HHMM format.**

When executing ./server_multithreading file

All procedures of functions are same as in ./server and ./server_select executable files.

NOTE:

To overcome the data addition, removal & updation when one client is using getall command, we used mutex lock in server_multithreading part.