UMass Lowell 16.480/552 Microprocessor Design II and Embedded Systems

Lab 4: Multithreaded Programming

Due Date: See the course schedule web page.

Objectives

- Learn how to design multithreaded programs for embedded multicore systems
- Understand the principles of synchronization and mutual exclusion
- Learn POSIX thread libraries.
- Learn HTTP protocol and libcurl library

Description

You should by now have both a working sensor device (interfaced with the Atom development board via a parallel port) and essential software modules (a device driver and user applications). You will need to design a multithreaded program that communicates with both the sensor and a web server. You can use libcurl APIs to send http requests and parse html pages.

Your program should meet the following requirements:

1. Use at least three threads to perform the following tasks respectively
   (a) User command interface: this thread allows for local user commands such as resetting the sensor, checking status, setting up interrupts and reading ADC values. This thread also facilitates debugging the program;
   (b) Sensor control: this thread is responsible for sending commands to the sensor and obtaining sensor responses. This thread also needs to check the sensor's status periodically and report its status to a web server (provided by the instructor);
   (c) Communication: this thread communicates with the provided web server using HTTP protocol and *libcurl* library. The sensor related communication protocol is defined as follows.

2. The sensor application (more specifically the communication thread) reports sensor status and data using HTTP POST method. That is, the senor application use the following URL to supply status, data and timestamp:

   http://servername:port/sensor.py?id=xxx&passwd=xxx&name=xxx&status=xxx&data=xxx&timestamp=xxx

   Note:

   "http://servername:port/sensor.py" is the CGI script on the web server (servername) listening at port number (port)

The sensor application must provide the following information as part of the URL request

Id	A unique numerical identifier. To be assigned to each project group
Passwd	A unique password to authenticate the sensor app
Name	The project group name
status	The current status of the sensor as defined in "sensor.h" file
data	An integer value of the ADC value
time	timestamp in the format of yyyymmdd-hh:mm:ss

3. (Optionally) Parse the html pages replied from the web server:
The web page replied from the web server has commands embedded in it. The sensor application needs to perform appropriate actions for servicing the command received from the server.

4. (Optionally) Analyze the multithreaded program using Intel® VTune Amplifier to observe the activities of the threads and identify hot spots in the threads.

Deliverables

A zipped file containing
1. Source code (for both PIC and Atom)
2. Reports

References

[1] Linux Device Driver, 3rd ed. Available at http://lwn.net/Kernel/LDD3/
[2] Posix Thread programming. Available at https://computing.llnl.gov/tutorials/pthreads/
[3] libcurl APIs. Available at http://curl.haxx.se/libcurl/
[4] Intel® VTune Amplifier XE 2011. http://software.intel.com/en-us/articles/intel-vtune-amplifier-xe/