INFO I211 - Information Infrastructure II

Spring 2014

Assignment 1

Building a distributed application using Python

(to be completed in your I211 Student Team - and documented by your *partner I211 Team*)
Total: 100 points (and extra tasks for bonus 15+20+15 points, see below)
Please see Additional comments in green typeface, added on 2014-04-05 for clarification.

Due Date:

Tuesday, April 08 2014, 11:59PM, on IU OnCourse.

How to work on your Assignment 1:

The Assignment 1 isto be completed in your I211 Student Team, where each team member is to pick one of three tasks. The tasks are:

1. Task 1 = Part A of the distributed application
2. Task 2 = Part B.1 of the distributed application
3. Task 3 = Part B.2 of the distributed application (which generates the page implementing Part C of the distributed application)
4. *if* there are four (4) students in your I211 Student Team, the task for the fourth team member is to *document the process*, i.e. provide a day-by-day journal of how the teamwork and individual work proceeded, and to provide a *final summary* with the results of testing their Assignment 1 implementation.

When working on these files, exchange them with your *partner I211 Team* periodically. The partner team's responsibility is to *document* each file, by providing *code comments* for all programs and scripts (including the parts that are already provided to you in the "a1startingcode.zip" file contents). Your role is to do the same for your partner team's files.

Assignment Tasks:

The task for Assignment 1 is to complete the "Security Camera" distributed application, as presented at I211 lectures 18 and 19. To begin:

- read the notes for *Lecture 18 and 19*, which you can find in the *Oncourse I211 Resources* folder;
- Additional:
    a. all references to lecture notes refer to the lecture slides/notes that are available in the *Oncourse I211 Resources* folder, e.g. *20140331-20140401-I211-lecture-18.pdf*, etc.
- on your local system running either Windows or Mac OS X (e.g. a laptop computer, or desktop computer, running Python as used for previous I211 assigments), download the file "a1startingcode.zip" from the *Oncourse I211 Resources-->Assignments* folder;
- on your local system, extract the contents of the "a1startingcode.zip" file into a folder named *a1startingcode* located on your local system's desktop.

Assignment 1 consists of 3 parts. These are nearly completed in the provided *starting code*, and will require some editing and configuration to make them work properly.

A. on the *remote data generation* system: Python-based code that is used to periodically take a snapshot from a security camera (or a webcam) connected to the remote system, and automatically upload it using a *secure ssh connection* to the server running CGI scripts as in point B below. The *remote data generation* system is going to be your laptop or desktop computer running Python.

B. on the *server* system running CGI scripts and an HTTP server:

   1. a Python program executed as *cron script*, with related *crontab* configuration, to rename incoming image files according to their timestamp, and move them to a directory which can be safely accessed by CGI scripts.
   2. a Python CGI script to collect all the image filenames and dynamically generate a web page to display the images, organized in a table format.

C. on the *web client* (HTML browser), the web page generated by the CGI script running on the server, as in point B above.

For part A, follow *Lecture 18 and 19* notes. Specifically:

- test the webcam on your laptop computer.
    - On Windows systems, from the Command Prompt, run the *CommandCam.exe* command-line program included in the *a1startingcode* folder. If your computer does not have a built-in webcam or camera attached to it, the program will print the message: "No video devices found". In that case, use the provided "snapshot.bmp" sample image to complete your assignment.
    - On Mac OS X systems, from the Terminal, run the *imagesnap* command-line program included in the a1startingcode folder. Some webcam hardware require an optional parameter to correctly save an image:
        - `localhost$  ./imagesnap -w 1.00`
- SSH keys setup.
    - Set up the SSH configuration on your laptop computer.
        - On Mac OS X systems, "OpenSSH" is already included and you don't need to do extra configurations to run it.
        - On Windows systems, use PuTTY. If you don't have it yet installed on your laptop computer, download the *putty-0.63-installer.exe* file containing the Windows installer from the official PuTTY site (http://www.chiark.greenend.org.uk/%7Esgtatham/putty/download.html) and install PuTTY on your computer.
    - Generate SSH keys for your account on the *burrow* cluster. If you don't have a working SSH key pair setup, do the following:
        - Login to *silo.soic.indiana.edu* by using "WinSCP" on Windows or "OpenSSH" on Mac OS X as per I211 *Lab notes*. Following instructions at the IU KB page Generating SSH keys, generate the public/private key pair for your account.
        - While logged in to *silo.soic.indiana.edu*, append the content of the new *public key* you just created to the file *authorized_keys*, which will allow remote login by SSH key, thus:
            - `silo.soic.indiana.edu%  cd ~/.ssh`
            - `silo.soic.indiana.edu%  cat id_rsa.pub  >> authorized_keys`
    - On your laptop computer, get the *private* SSH you just generated from *silo.soic.indiana.edu* and store it into the proper location for your local SSH client:
        - On Mac OS X systems, from the Terminal run the following:
            - `localhost$  mkdir ~/.ssh`
            - `localhost$  cd ~/.ssh`
            - `localhost$  scp -p mitja@tank.cs.indiana.edu:/u/mitja/.ssh/id_rsa .`
            - `localhost$  scp -p username@tank.cs.indiana.edu:/u/username/.ssh/id_rsa`

    - Modify the *windowspscpfile.bat* file included in the *a1startingcode* folder: adjust the *local* path, substituting *username* with the username you use on your laptop computer, and *a1remote* with the directory where you'll be running this batch script.
  - On Windows systems:
    - Modify the *windowspscpfile.bat* file included in the *a1startingcode* folder: adjust the *local* path, substituting *username* with the username you use on your laptop computer, and *a1remote* with the directory where you'll be running this batch script.
    - If you are running a 32-bit windows system, you'll also have to modify the *windowspscpfile.bat* file included in the *a1startingcode* folder to cd to the *Program Files* directory, instead of the *Program Files (x86)* directory. On *Windows XP* systems, you'll also have to replace the *Users* directory in the path with *Documents and Settings*, and include the resulting path in "double-quotes".
    - Using "WinSCP", from *silo.soic.indiana.edu* copy the file *id_rsa* contained in your home directory's *~/.ssh/* directory on silo to your laptop computer, into the *a1startingcode* folder where the other scripts are located.
    - On your laptop computer, start the *PuTTYgen* program (part of the PuTTY install), select the *Conversions->Import Key* menu, load the *id_rsa* private key file you just copied from *silo.soic.indiana.edu*, select "Save private key", and save it as *id_rsa_putty.ppk* into the same *a1startingcode* folder.
    - Modify the *windowspscpfile.bat* file included in the *a1startingcode* folder: to the *pscp* command, add the option *-i "\path\to\id_rsa_putty.ppk"* (including the double quotes), with the correct path to where you saved the *id_rsa_putty.ppk* file.
      Additional:
      1. place the option *-i "\path\to\id_rsa_putty.ppk"* right after the *pscp* command, and before the remote file description, for the *id_rsa_putty.ppk* file to be properly found by the *pscp* command.
      2. the *"\path\to"* is the complete directory(folder) path to the *id_rsa_putty.ppk* file on your system, not the words *path/to...*

Once your webcam and SSH key setup is configured properly, proceed with editing the *a1remote.py* file on your laptop computer:

1. include *your own username* for login to *silo.soic.indiana.edu* instead of the generic username word.
2. modify the main code of the *a1remote.py* file, so that the entire process (snapshot from webcam and ssh copy of the file to *silo*) is repeated in an *infinite while loop*, sleeping for *1 hour* between each photo shoot/upload. Hint: use the sleep() function in Python.
3. Additional:
   a. if you are running the the *a1remote.py* file from command-line on your laptop (or STC lab) computer, the .bat (for Windows) and .sh (for Mac OS X) files ought to be located just fine by providing the filename of those files in the python script. If you are running the *a1remote.py* file from IDLE or other non-command-line environments however, you may have to prefix the filename with "./" to be properly found by the *a1remote.py* program.

For part B, follow *Lecture 18* notes at pages 5-6-7. Specifically:

- On *silo.soic.indiana.edu*, prepare the directories that will be used by the server-side scripts:
  - ~/tmp/ for uploading the images from the remote data generation system as in part A.
  - ~/bin/ for the *crontab*-run Python program
  - ~/cgi-pub/i211/a1/ for the CGI script *index.cgi*
  - ~/cgi-pub/i211/a1/data/ to store the timestamped image files.
- Read the *crontab.pdf* file included in the *a1startingcode* folder, and follow the instructions to set up a *crontab* entry on *silo.soic.indiana.edu*. Copy the *timestampfile.py* file included in the *a1startingcode* folder to the ~/bin/ directory on *silo.soic.indiana.edu*, and rename it to *movefile.py*. Then edit *movefile.py* so that

it refers to files in your home directory.

- Note: additionally, if you are using a Windows computer for your *remote data generation* system as in part A, replace all references to *snapshot.jpg* with *snapshot.bmp*.

For part B.2 -- which enables Part C -- follow *Lecture 18* notes at pages 8-9. Specifically:

- Copy the *index.cgi* file included in the *a1startingcode* folder to the ~/cgi-pub/i211/a1/ directory on *silo.soic.indiana.edu*.
- Edit *index.cgi* to display all stored images:
- Either from newest to oldest, or from oldest to newest
- In a properly formatted HTML table: compute table dimensions (rows and columns) to fit images on a page to be displayed on a web browser running a standard-sized laptop screen.
- Resize the displayed images in the table. Hint: use the <img src="..." width=""> tag.
- Remove all *debugging* text from the HTML page generated by *index.cgi*.
- The page should be visible at *your username* location on the CGI SoIC server, e.g. similar to http://cgi.soic.indiana.edu/~mitja/i211/a1/, but with the above modifications completed.

Bonus tasks:

i. (15 bonus points) verify the integrity of EXIF metadata in the image captured on the *remote data generation* system, as mentioned on page 2 in *Lecture 18* notes.
ii. (20 bonus points) implement a simple "lock" mechanism as mentioned on page 3 in *Lecture 18* notes.
iii. (15 bonus points) in the CGI script *index.cgi*, extract image metadata (e.g. EXIF) to add captions with time, date, etc. to each image in the generated HTML table.

Assignment Submission:

When submitting the application, each team member is to submit all required files for their part of the application, as well as a (common) team *README* file and a (distinct) individual *README* file.

1. Include a plain-text file named `readme-a1-i211-`*`yourteamnumber`*`.text` in which you explain:
   - the parts of the assignment *your team* has completed
   - any extra functionality added to the code
   - any suggestions/enhancements/contributions you may have received from your partner team
2. Include a plain-text file named `readme-a1-i211-`*`yourusername`*`.text` in which you explain:
   - the parts of the assignment *you* have completed
   - the parts of the assignment *your team* has completed
   - any particularly clever code you added
   - (optional) any question you may have about the tasks or the provided template code
3. On the i211 / Spring 2014 site, turn in all your files by 11:59PM on Tuesday, February 10, 2014.
4. P.S.: *yourusername* should in fact be your username, not the word *yourusername*... and *yourteamnumber* should in fact be your team number, not the word *yourteamnumber*...

Include the following information as a `#comment` at the top of every Python file you submit, *right after* the special comment that starts Python:

```
# your name (First, Last)
# your IU email address
# your I211 team number
# the names of all your I211 team members
# your partner team's number
# the names of all your partner team members
```

Include the following information as a `<!--comment-->` at the top of the HTML file you generate from *index.cgi*:

```
<!-- your name (First, Last) -->
<!-- your IU email address -->
<!-- your I211 team number -->
<!-- the names of all your I211 team members -->
<!-- your partner team's number -->
<!-- the names of all your partner team members -->
```

Complete the "I211 Student Team Feedback Form.doc" (which you can find in the *Oncourse Resources->Assignments* folder) about your experience in the student team, and turn it in on Oncourse with your Assignment 1.

Before you turn in the files on *Oncourse*, test the correct functioning of your distributed system, in all three parts (A, B and C, as described above). Don't forget the appropriate chmod settings for your files. Your page should be visible at *http://cgi.soic.indiana.edu/~yourusername/i211/a1/*.

Turn in all the modified files on Oncourse to receive a grade for Assignment 1. Upload the following files to *Oncourse* under *Assignments->Assignment 1*.

- Part A: *a1remote.py* (and *windowssnapshot.bat* and *windowspscpfile.bat* if you used Windows)
- Part B.1: *movefile.py*, and a file named *crontab.txt* containing your *crontab -l* configuration (just copy and paste its content from the connection to *silo.soic.indiana.edu*).
- Part B.2 - Part C: *index.cgi*
- "I211 Student Team Feedback Form.doc" - about your experience in your I211 student team
- if there are four (4) students in your I211 Student Team, the fourth team member will turn in the documentation to the process, i.e. the day-by-day journal of how the teamwork and individual work proceeded, and a final summary with the results of testing their Assignment 1 implementation. Please use either plain-text or PDF for the file you're turning in for this part, and name the file documentation-journal-a1-i211-yourteamnumber.text (or .pdf).

Notes about assignment completion:

1. Your Python programs need to be written using the Python 2.x programming language (they will *not* be considered if written in Python 3.x or any other programming language).
2. To receive full credit for the work done, you have to submit:
   a. All *source code* files that you modified for Part A, Part B and Part C.
   b. The *I211 Student Team Feedback Form* document for your experience in the student team.
3. Assignments lacking the *I211 Student Team Feedback Form* and/or the will lose 50% of the assigned grade.

Good luck!