

## CSL202 | Assignment-4 | Due 26/Mar/2018 11:59 PM | 100 points

- Important instructions for coding submission are here: <https://goo.gl/IMWvdF>
  - Grading scheme to be followed is available here: <https://goo.gl/52D82g>
  - Assignment description may be underspecified to allow some room for exploration and creativity.
  - Your submission should be packaged as a zip file named **exactly** in this format:  
CSL202-[your entry no.]-[assignment no.].zip.
- 

The **top** command on UNIX like operating systems provides a dynamic real-time view of a running system. We need to write a program in Java which can make use of **top** command to find all those processes which have violated a predefined limit on usage of resources such as CPU and memory. The limit on CPU and memory usage can be specified in a properties file which the Java program will read at the start. The high-level logic of the program can be described as:

1. Read the configuration properties file (say, **settings.properties**) at the start. A sample file is shown in the end below.
2. Initialize an instance of a suitable data structure (say, we call it **procData**) which will store the historical information about resource usage by different processes.
3. Periodically invoke the **top** command and process its output for identifying violators as follows:
  - a. For each line of the output do:
    - i. Extract: PID, USER, %CPU, %MEM, TIME+ and COMMAND fields.
    - ii. Check (in **procData**) whether the process whose data is shown in this line has violated the resource usage limits.
    - iii. If limits are violated then note down the process information (say, in another data structure instance called **violators**)
  - b. Collect all the noted details of violating processes and send an email to the address(es) mentioned in the configuration file **settings.properties**.
  - c. Clear the entries older than past **n** minutes as specified in the configuration file **settings.properties**.

Your program should be able to identify violating processes even if your program or the machine on which it is running gets restarted.

Example configuration file format:

```
# ===== This file contains the settings for the program =====
# File: settings.properties
# Set to 24 hours
quota.window.minutes = 1440
# Sustained max. CPU usage duration limit in minutes
sustained.max.cpu.usage.duration.limit = 60
# Limit on sustained CPU usage in percentage.
sustained.max.cpu.usage.limit = 70

# Sustained max. memory usage duration limit in minutes
```

```
sustained.max.memory.usage.duration.limit = 60
# Limit on sustained memory usage in percentage.
sustained.max.memory.usage.limit = 10

# Email addresses to be notified in case of violations. Value is a comma-separated list
notify.emails = notify1@test.com,notify2@test.com,notify3@test.com,
```

#### HINTS:

1. You can run the top command in batch mode every n seconds.
2. You may make use of Java's [ScheduledExecutorService](#) class or something similar to periodically run the command.