

A. Download Scripts

1. Scripts can be found here: <http://cseweb.ucsd.edu/classes/wi13/cse237A-a/project/proj3/scripts/>

Try to keep your scripts updated with the latest release. Whenever a new release appears, you will be notified through piazza. **New scripts will not invalidate the logs you have collected so far.** They will just provide additional useful information or fix bugs.

B. Script Descriptions

First login to adb shell and run the following commands to create necessary directories on the phone's file system.

- ➔ `mkdir -p /mnt/sdcard/seestats`
- ➔ `mkdir -p /data/seestats`

Then copy the scripts to your phone, more specifically, to `/data/seestats`. You can do this in one of 2 ways:

- a. *Preferred*: Run `adbpush_execs.sh` from the source location, having first connected the phone to your computer
- b. Copy manually using the `adb push` (from host) and the set specific permission using `chmod` (within adb shell, i.e from within the device).

The description of the scripts is as follows

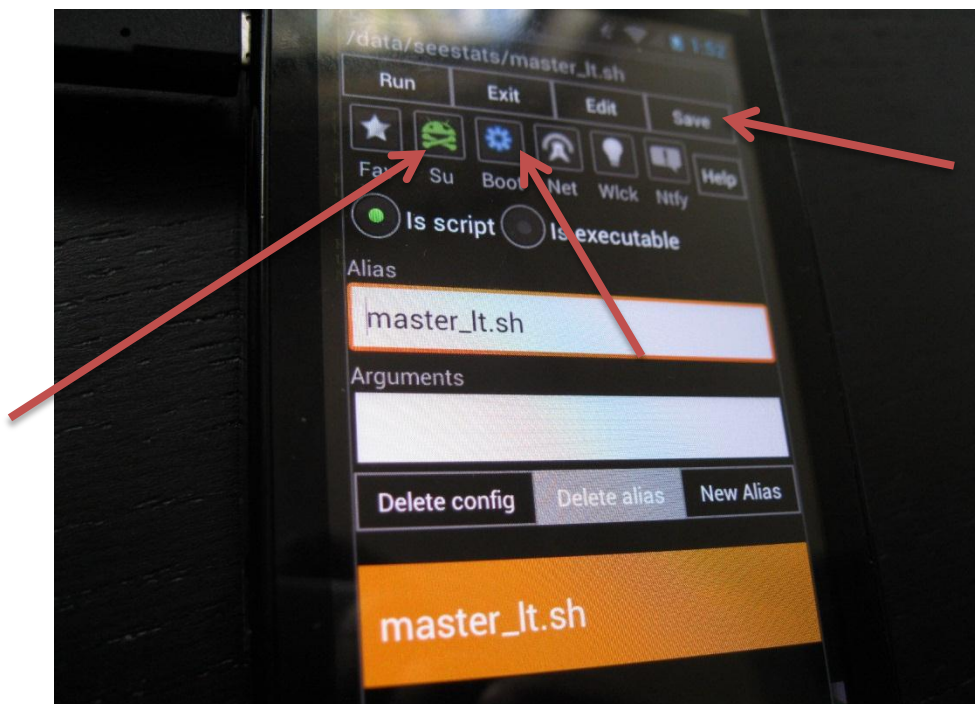
1. **`adbpush_execs.sh`** – this script will copy over the necessary files to your phone. This will need to be run every time scripts are updated
2. **`master_lt.sh`** – Used for long term data collection. You have to set the Su and Boot option for this script (and Save the options). **The script will start automatically at boot time (i.e. every time your phone reboots). You do not have to explicitly run this script. Usage of this script may sometimes lead to discontinuous logs due to android suspend events.**
3. **`master_sched_lt1.sh` and `master_sched_lt2.sh`** : These scripts can be used to perform long term stats collection. **When run from the script manager scheduler facility, these scripts will survive better the android system suspend events. Add these to the scheduler list of script manager . Both these scripts should be granted root access.** While the first should be run at the granularity of 15 minutes (00:15), the other should be run with the granularity of 1 hour (01:00). Instructions on how to add to the script manger's scheduler list is explained at the end of this section.

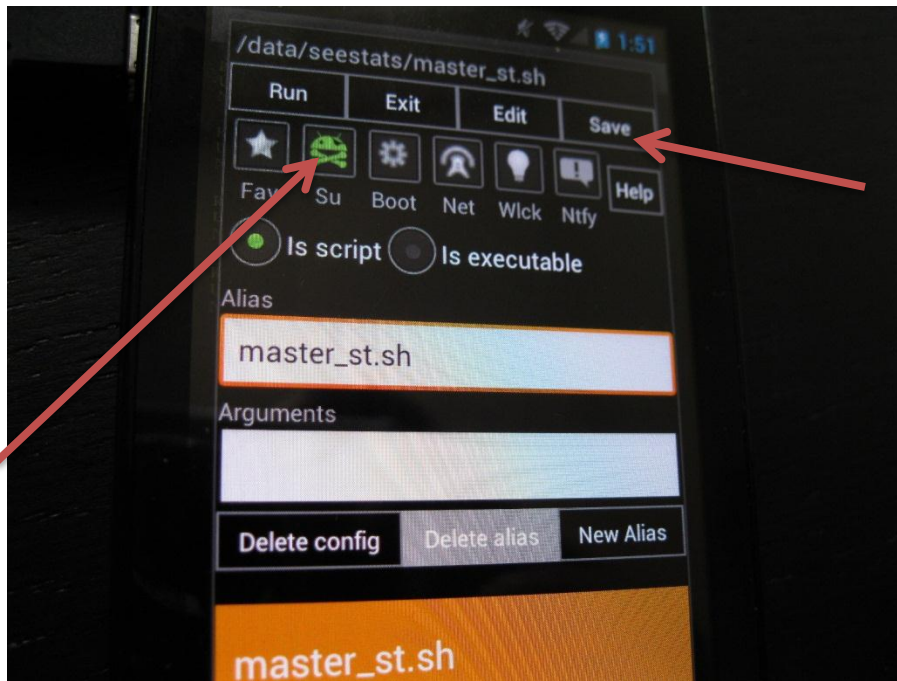
Please note: These logs may not be generated at exact intervals.

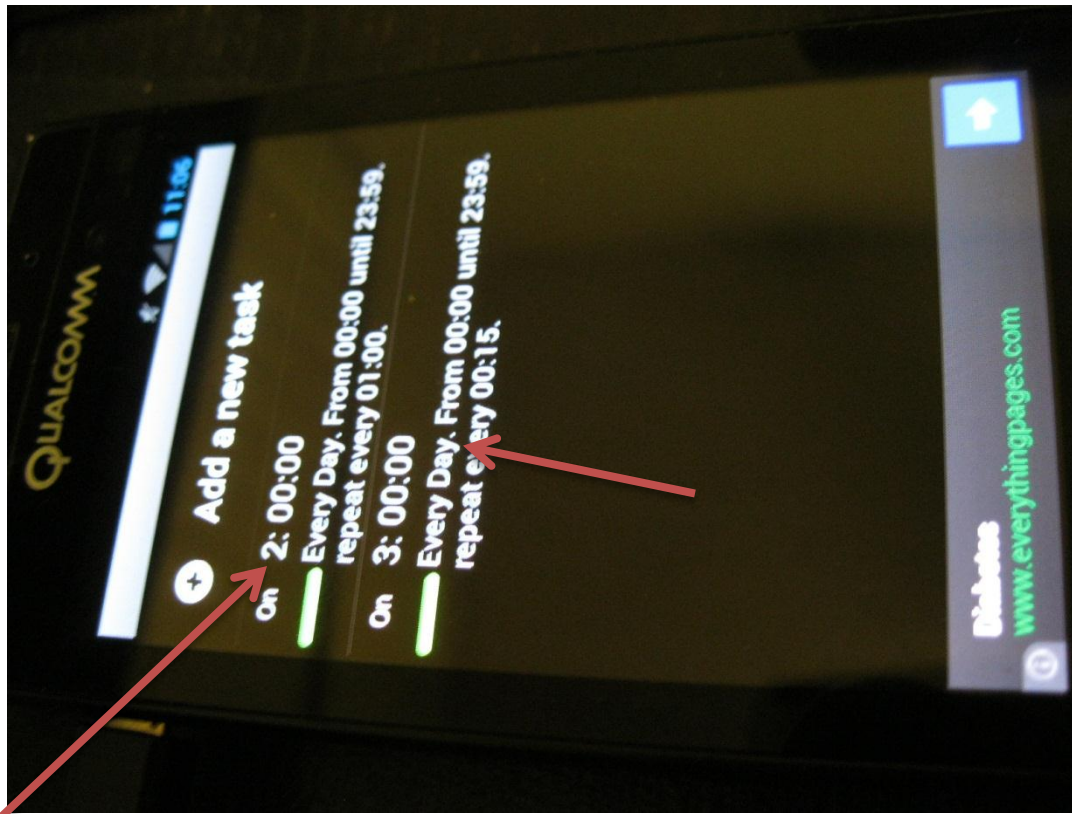
4. **master_st.sh** – this is the script that must be run to start each short term session. It will execute all subordinate scripts as needed. Only Su must be set (and Saved) within smanager for this script. Move this script to background once you are done with running the script. DO NOT press the stop button until the stop button turns white from red.
5. **stop_st.sh** – this is the script to end each session. It will shut off all running short-term scripts. This must be done after each session. If you do not stop the short term session your sdcard will get filled too rapidly and you might run out of storage space. No Su or Boot options needed.
6. ***.out files** – these binaries are wrappers to various Android performance monitors, and called by *master_st.sh* automatically. They will output to appropriately named *.txt log files, and data is collected every 100 ms
7. **shorterm1.sh, longterm1.sh, etc** – scripts for collecting other useful data. You are not supposed to run any of these scripts.

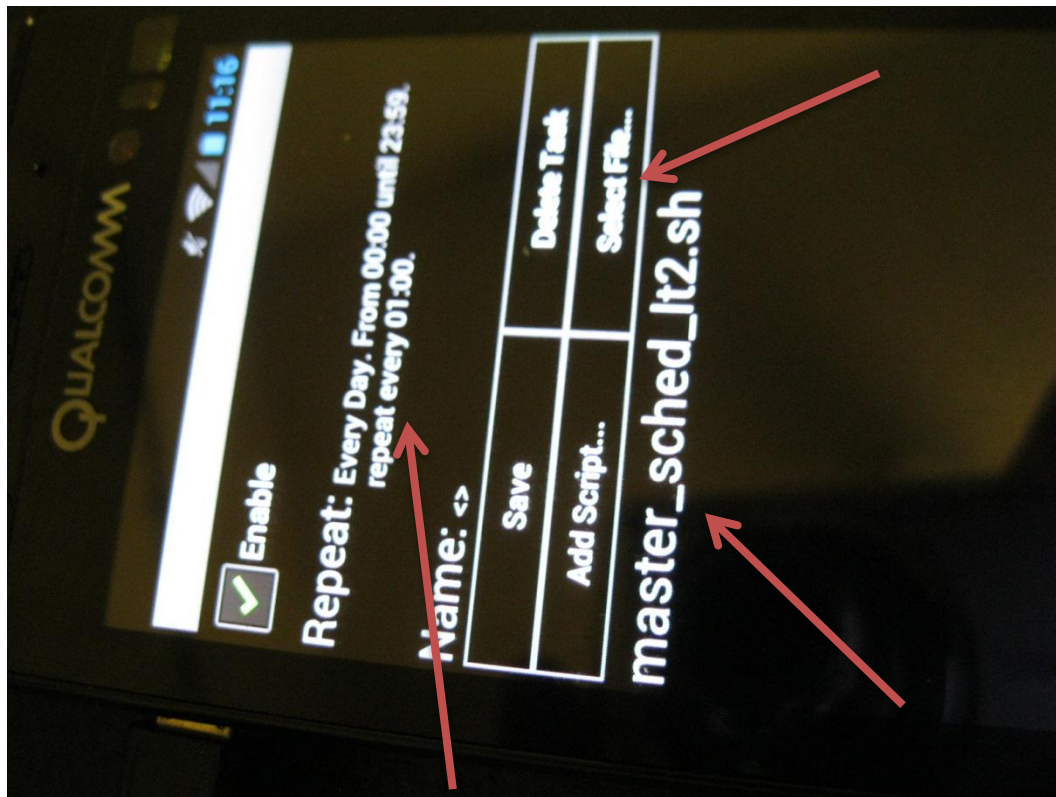
Set Su or Boot options only to the scripts mentioned above (ie. start_st.sh and start_st.sh). If some scripts need these options we would tell you explicitly.

Check attached pictures.









All the logs can be found in /mnt/sdcard/seestats. You can use adb pull command (from host) to pull the logs to your computer (host).

How to add to scheduler list:

Go to script manager

Then, click on settings 'on' the phone (the symbol looks like this)

(the button between the back button and the home button)

Then inside that click, on More -> Advanced -> Scheduler

Then, add a new task

Followed by Add file -> Then add script

First file/script to be added:

master_sched_lt1.sh (from latest release. do adbpush_execs.sh for this)

Click on Repeat and then set 00.15 -> i.e. 15 minutes

Then Save and enable script. **Make sure the Su option is set for this script.** Also make sure you save and enable the script.

Second script/file to be added:

Repeat for master_sched_lt2.sh -> Set 01:00 for this. **Su for this script too.** Make sure you save and enable the script.

(Attached pictures to give you an idea)

So finally u should have two sets of tasks (as shown in the first picture from my phone)

(Note: Each task will have some number. In my case it is 2 and 3, for you it may be 1 and 2)

If you notice, the new master_lt.sh is not doing anything useful currently (as per the 18th Feb 1 am release). So even if you keep it the way it is (ie. both Su and Boot option), it is fine. It acts as a placeholder for now. We will use it in future if we want to enable specific options at boot time.

C. Default Settings

A: Verify the following ClockSync Settings

NTP server – pool.ntp.org

Timeout (seconds) - 10 seconds

Enable Automatic synchronization – Enable

Interval – 15 mins

Use Offline Databases - Enable

Select Time Zone - America/LA (Los Angeles)

Sync on boot – Enable

Only on WiFi – Disable

Only when awake – Disable

Strict Interval – Disable

High Precision mode – Disable

Time Zone Fixer – Disable

Rootless Mode – Disable

Date and Time - Shortcut to system settings

<Select this then>

Automatic Data and Time – Disable

Automatic Time Zone – Disable

Strict Interval , high precision mode, only on wifi, only when awake, show notification, haptic feedback should **not** be enabled

B: Verify the following OS settings on the phone

Inside adb shell ,

- `ps | /data/data/com.qualcomm.qview/qview/busybox grep mpdecision`

you must see a mpdecision process listed

- `ps | /data/data/com.qualcomm.qview/qview/busybox grep thermald`

you must see a thermal daemon listed

- `cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor`

you must see ondemand

- `cat /sys/devices/system/cpu/cpu1/cpufreq/scaling_governor`

you must see ondemand. This may depend whether or not the core1 is online. If u don't see this, it is fine.

C: Make sure you disable GPS .

Go to Settings app -> Location services-> Disable google's location services and GPs satellites.

D. Description of log files

Here is the description in brief.

- a. **cpuusage:** this data is taken from `/proc/stat`, and contains CPU state information:

```
cpu <col1> <col2> <col3> <col4> <col5> <col6> <col7> <col8> <col9>
<col10>
cpu0 <col1> <col2> <col3> <col4> <col5> <col6> <col7> <col8> <col9>
<col10>
cpu1 <col1> <col2> <col3> <col4> <col5> <col6> <col7> <col8> <col9>
<col10>
intr *
```

where:

col1: 1147 - user: normal processes executing in user mode
col2 - nice: "nice" processes executing in user mode
col3 - system: processes executing in kernel mode
col4 - idle: CPU not executing anything
col5 - iowait: waiting for I/O to complete
col6 - irq: handling interrupts
col7 - softirq: servicing software interrupts
col8 - steal: involuntary wait
col9 - guest: running a normal guest
col10 - guest_nice: running a "nice" guest

These are helpful to build a profile of a session – that is, how much of the CPU was in each state? You can effectively ignore the intr sets, as they do not help us here.

- b. **cpufreq.out** – outputs the time spent in different frequencies
Easy enough to parse – just match the frequency (first column) with the time spent in that particular frequency (second column)
- c. **cpuonoff.out** – outputs which cores are currently active (every 100 ms)
0 → only core 0 is active
0,1 → both cores 0 and 1 are active.
- d. **gpubusy.out** – please ignore this file unless further described
- e. **gpufreq.out** – single output, frequency of the GPU
- f. **temperature.out** – single output, representing SoC temperature in Celsius.

E. Special settings

For the progress report, it would be good to stick to default settings and focus more on delivering the logs (short term and long term) which align with your usage pattern.

F. Debugging scripts

top10.sh, df.sh, dumphsys_battery.sh

➔ More details on piazza.

