The folder contains a folder “scripts”. To perform long-running experiments, we make use of the scripts in “*scripts/expExecution*”. There is a *runtest.sh* file that is the only file that should be configured and run.

A set of scripts is launched in background by this file, which are described in the following. To include/exclude some scripts just comment the corresponding line. To change the workload, you can edit the workload files in the “scripts/expExecution/workload” directory.

To run the experiment, open a terminal and run the “runTest.sh”. Note that output files are written within the same directory where “runTest.sh” is located; to write in a different directory, move the “runTest.sh” file into a new directory and change the first two lines of the “runTest.sh” file, by correctly setting the directory where the other scripts are located.

The scripts launched by “runtest.sh” are:

* *dumpOnlyMeminfo.sh.* It collects detailed memory information statistics of the applications’ processes used as workload.
* *logcat\_displayed.sh.* It collects “displayed” events from *logcat* used for activities launch time computation.
* a “*WorkloadFixed.sh*” and a “*WorkloadVariable.sh*” script that contain the *monkey* commands to launch the experiment. The command for the fixed workload is “adb shell monkey” and contains, as arguments, the list of the APK to run during the experiment, the configuration of the “events” type factor (namely, which percentage of “*touch*”, “*motion*”, “*navigation*”, “*trackball*”, and “*appswitch*” events are set in the experiment), and the number and frequency of the events. To change the fixed workload, just edit this command according to the *monkey* utility guidelines.

The variable workload have more commands, in order to switch from a workload state to another.

* A *LaunchTimeMeasurement.sh* script, that kills and launch the set of apps under test in order to get the measurement of the Launch Time (every 60 seconds)
* *killMonkey.sh*, which kills the monkey execution. It currently is set up at 6 hours; to change the duration, edit this file.
* *logcat\_art.sh.* It collects ART events from *logcat* used for garbage collection times computation
* *dumpTask.sh.* It collects info at task level; we used it for *system\_server*, *media\_server,* *surfaceflinger* and *systemui* processes’ tasks

To process data, there is a series of scripts (folder “*scripts/analysis*”):

* **Memory**
  + Shell scripts
    - *generate\_Meminfo\_data.sh.* This file generates plots of PSS for each process running during the experiment.
    - *generate\_Meminfo\_data\_Slopes.sh.* This file computes the Slope of each PSS data series for each process running during the experiment, and prints the output on the console.
    - *generate\_Global\_Meminfo\_data.sh.* This file generates plots of PSS for each process running during the experiment.
  + R scripts. The following scripts are used by the *.sh* scripts to compute trends and generate plots.
    - *analyze\_data\_slopes.r*
    - *analyze\_data.r*
    - *analyze\_other\_mem\_data.r*
  + Python scripts. The following script is used by the *.sh* scripts to extract CSV files from meminfo.txt file.
    - *excel\_gen.py*
* **Launch Time**
  + Shell scripts
    - *generate\_Time\_data.sh.* This file generates plots of the Launch time per activity and textual files reporting the slope for each activity launch time.
  + R scripts. The following script is used by the *.sh* script to compute trends and generate plots.
    - *analyze\_time\_data.r*
  + Perl scripts. The following script is used by the *.sh* script to pre-process the raw file displayed.txt generated by the monitoring script *logcat\_displayed.sh*
    - *displayed.pl*