Report

Android OS Monitor

Developer: Hengming Dai & Xinzhe Yu

Description

This app can show usages for every running process about CPU, memory, and battery in Android operating system and let users optimize there operating system.

Implementation

We use class *AppData* to store the information of each app(its package name), including process it has used(PID), CPU time, memory use and battery use. In class *app_list*, we have some functions to get CPU time, memory and Battery use of each process.

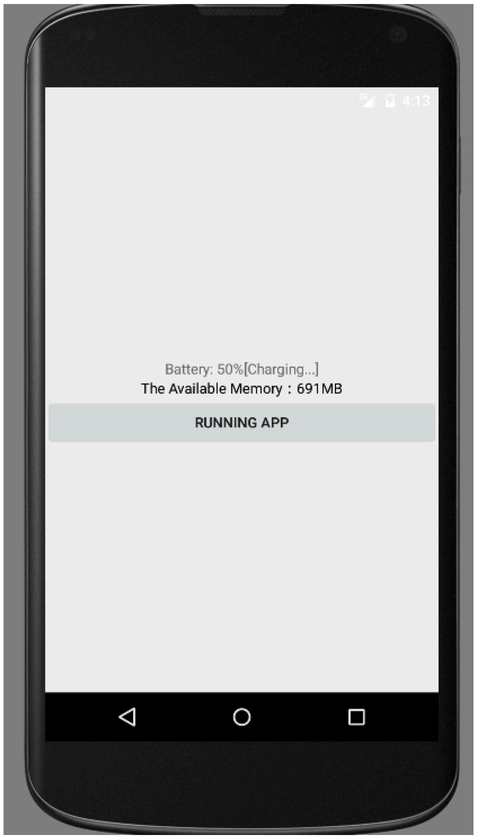
We can get CPU time of each process by calling function *getAppProcessTime(*), and get total CPU time by calling *getTotalCpuTime()*. Then the battery usage in percent of a process is Process time divides Total CPU time, in percent. In real life, CPU frequency is not a constant, we can get the frequency of CPU and calculate corresponding current I. Then energy consumption equals U*I*T. For memory, we use *getProcessMemoryInfo()* to get each process's memory use.

Run app

AVD Information
Version: Nexus 4
Target: Android 5.0.1
CPU: x86
Size on Disk: 1GB
API: API 21

Android Mobile Device Information
Version: LGE LG-VS450PP
Target: Android 4.4.2
CPU: 1.2GHz Qualcomm MSM8610
Size on Disk: 1GB
API: 18

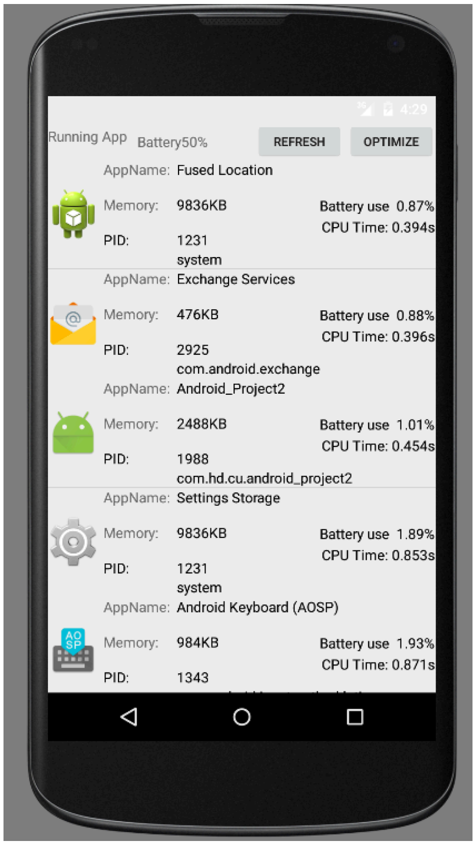
The user interface:



4:13

Battery: 50%[Charging...]
The Available Memory : 691MB

RUNNING APP



4:29

Running App Battery50% REFRESH OPTIMIZE

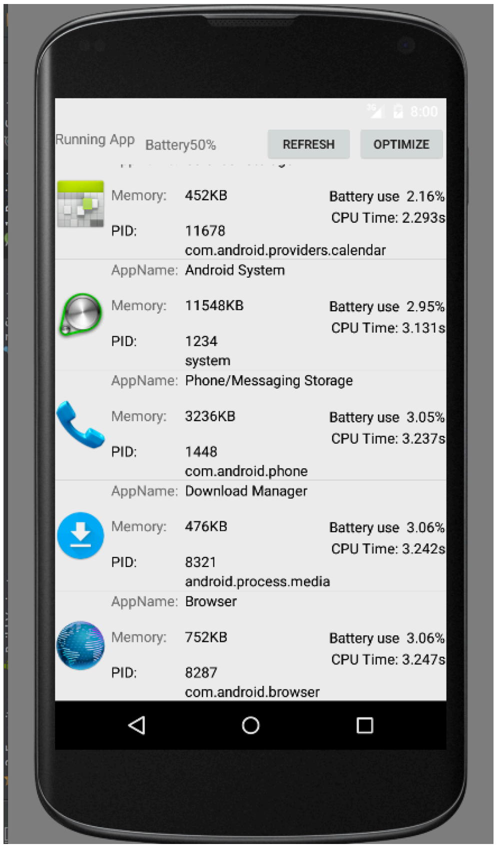
AppName: Fused Location
Memory: 9836KB Battery use 0.87%
CPU Time: 0.394s
PID: 1231
system

AppName: Exchange Services
Memory: 476KB Battery use 0.88%
CPU Time: 0.396s
PID: 2925
com.android.exchange

AppName: Android_Project2
Memory: 2488KB Battery use 1.01%
CPU Time: 0.454s
PID: 1988
com.hd.cu.android_project2

AppName: Settings Storage
Memory: 9836KB Battery use 1.89%
CPU Time: 0.853s
PID: 1231
system

AppName: Android Keyboard (AOSP)
Memory: 984KB Battery use 1.93%
CPU Time: 0.871s
PID: 1343



8:00

Running App Battery50% REFRESH OPTIMIZE

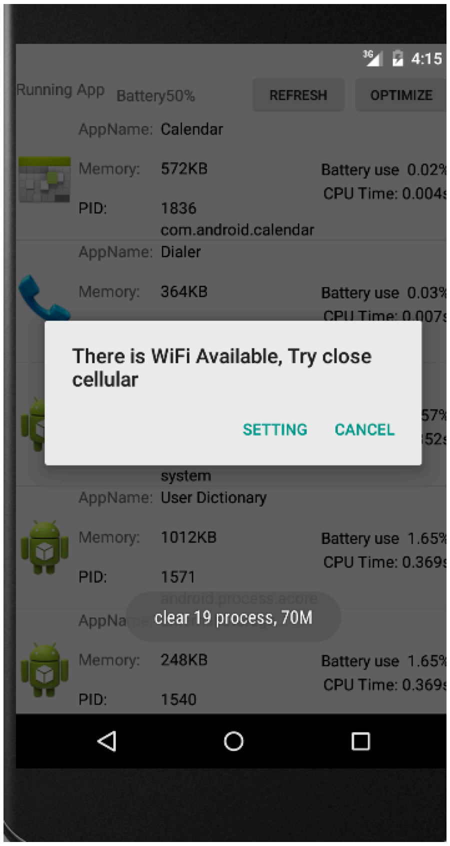
Memory: 452KB Battery use 2.16%
CPU Time: 2.293s
PID: 11678
com.android.providers.calendar

AppName: Android System
Memory: 11548KB Battery use 2.95%
CPU Time: 3.131s
PID: 1234
system

AppName: Phone/Messaging Storage
Memory: 3236KB Battery use 3.05%
CPU Time: 3.237s
PID: 1448
com.android.phone

AppName: Download Manager
Memory: 476KB Battery use 3.06%
CPU Time: 3.242s
PID: 8321
android.process.media

AppName: Browser
Memory: 752KB Battery use 3.06%
CPU Time: 3.247s
PID: 8287
com.android.browser



4:15

Running App Battery50% REFRESH OPTIMIZE

AppName: Calendar
Memory: 572KB Battery use 0.02%
CPU Time: 0.004s
PID: 1836
com.android.calendar

AppName: Dialer
Memory: 364KB Battery use 0.03%
CPU Time: 0.007s

There is WiFi Available, Try close cellular

SETTING CANCEL

system

AppName: User Dictionary
Memory: 1012KB Battery use 1.65%
CPU Time: 0.369s
PID: 1571
android.process.acore

clear 19 process, 70M

Memory: 248KB Battery use 1.65%
CPU Time: 0.369s
PID: 1540

In main screen, it will show charging state and available memory. Click the button then app will start, and it will show the list of process, including their PID, CPU time, memory use and battery use.

Analysis of Energy Consumption

In the list of energy consumption, apps that consume most energy are Browser and Download Manager. The reason is: both these process need to use cellular or Wi-Fi, which needs a lot of CPU usage. Apps that consume least energy are fused location and Android Project 2. For Android Project 2, app will use little CPU and memory when listening on-Click event. And when user click the Refresh button, it will use CPU and memory to calculate. So the total CPU and memory usage is little. Besides, it doesn’t call wakelock mechanism. So energy consumption is low. And for fused location, since AVD is on computer, the location is known, doesn’t need navigation, which leads to low energy consumption.

System Optimization

For each task, we have several CPU frequencies to run it. With higher CPU frequency, we can get better performance but larger energy consumption. So for the tasks that doesn’t need better performance and need long running time, such as music player, navigation, we can choose lower CPU frequency. For download task, if user downloads something background, CPU will use lower frequency. If download task is not background, CPU will run in higher frequency. For tasks such as games, we need to satisfy their time constraint and find a CPU frequency with lowest energy consumption, using dynamic programming.

Dynamic programming is a method for solving a complex problem by breaking it down into a collection of simpler sub-problems, solving each of those sub-problems just once, and storing their solutions - ideally, using a memory-based data structure. The next time the same sub-problem occurs, instead of re-computing its solution, one simply looks up the previously computed solution, thereby saving computation time at the expense of a (hopefully) modest expenditure in storage space.

In mobile device, reduce the energy consumption is a complex problem, since there are many process in an Android system. However, we can use dynamic programming to reduce each process’s energy consumption, in terms of their priority. In that way, we can reduce the whole energy consumption of Android operating system.

To use dynamic programming, we need to get the priority of each process. Then we can have a sequence of processes that CPU need to deal with. To get running time of each process, we can use class *PkgUsageStats.* After we have running time of each process, we can choose CPU frequency for each process, using CPU governor. Android provides us some cat command to get choices of CPU frequency and set CPU frequency. By using dynamic programming, we can choose the best CPU frequency for each process.

Users can terminate some useless processes manually if they don’t want to use it, and this can be done on the OS Monitor app. On our OS Monitor, users can optimize their Android operating system. By clicking the “OPTIMIZE” button, app will terminate some useless process and let you close the cellular if there is Wi-Fi available, since cellular cost more energy than Wi-Fi in a time unit.

Besides, app can help you terminate some useless process. Android has wakelock mechanism to keep some process using CPU after the screen is locked. However, wakelock of some processes is useless to user. These processes lead to meaningless energy consumption. So we terminate these processes by stopping the wakelock mechanism of these processes.