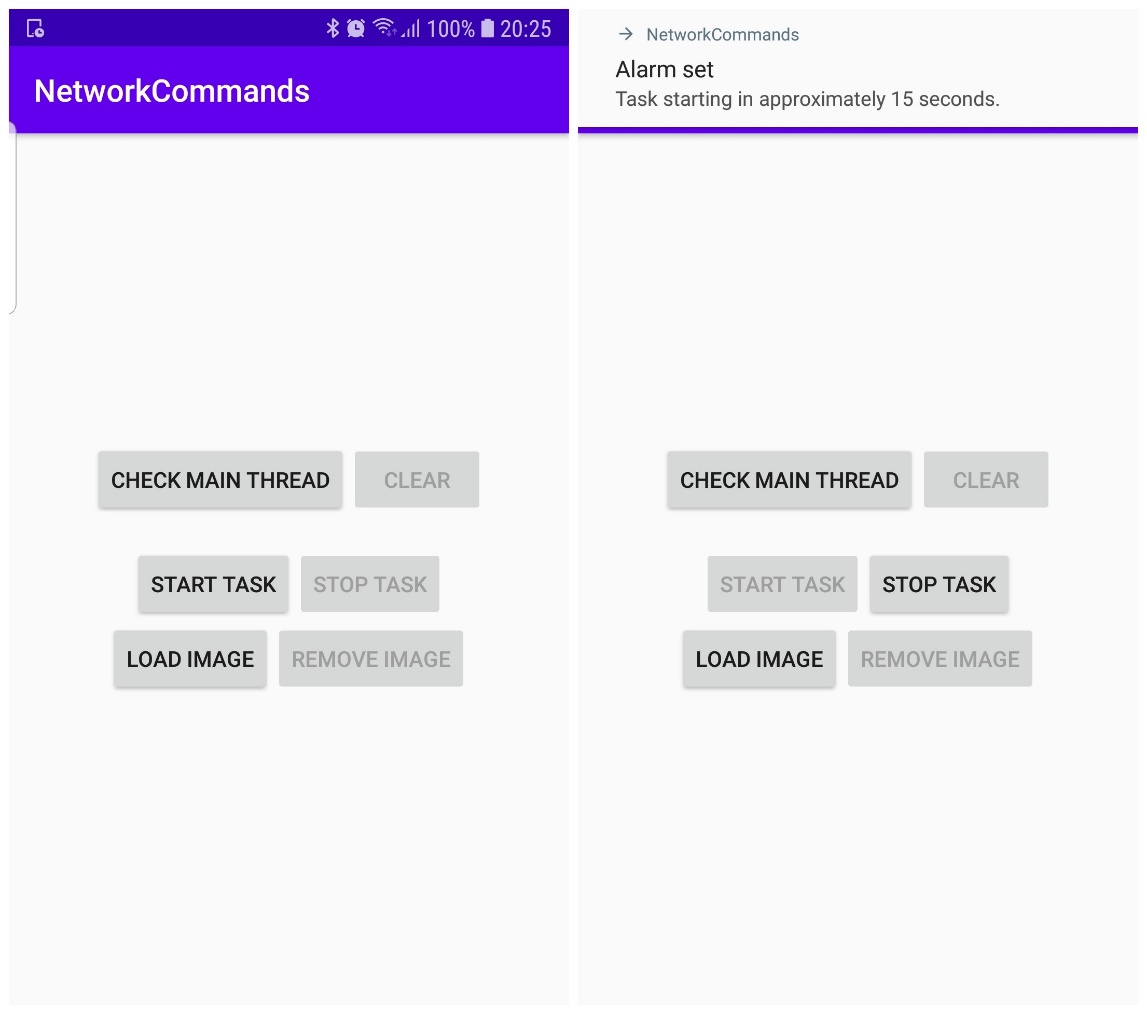
**Проект по предметот Мобилни сервиси со *Android* програмирање**

*Жанета Тренчева 9/2017*

Мојата апликација се вика *Network Commands*. Во истата ги искористив стекнатите знаења од предметот Мобилни сервиси со *Android* програмирање, а опфатив и слични теми од областа кои не беа изучувани во предметот. Паралелно со објаснувањето во продолжение, поставував и *screenshot*-ови и слики од *Logcat*, кои освен функционалностите на апликацијата, го демонстрираат и однесувањето на апликацијата временски. За тестирање користев мобилен телефон *Samsung* *Galaxy* *S7* *Egde* (*Android* верзија 8.0).

**Опис на апликацијата**

Со стартување на апликацијата, се отвара *MainActivity* која е и единствената активност во програмата. Оваа активност има неколку копчиња со соодветни функционалности. Со притискање на копчето *Start task*, се повикува функцијата *setAlarm*() во *MainActivity*. Овде најпрво се испраќа нотификација која го известува корисникот дека алармот е поставен и задачата ќе почне да се извршува за околу 15 секунди (би сакала да напоменам дека кодот кој го користам за нотификацијата се разликува од оној кој го учевме, прилагоден е на поновите верзии на *Android*). Се поставува аларм кој е повторувачки, од типот *ELAPSED\_REALTIME*, со време на тригерирање од 15 секунди и периода на повторување од 15 минути*.*



*Broadcast intent*-от од алармот го прифаќа *broadcast receiver*-от *AlarmReceiver*, кој е регистриран за акцијата *ACTION\_ALARM* во *onCreate()* во *MainActivity*. *AlarmReceiver*-от во *onReceive*() користи *JobScheduler* за да закаже *job* кој побарува било каква мрежна конективност и е отпорен на ребутирање. Соодветниот *JobService* e *CommandJobService*, кој во *onStartJob()* го стартува сервисот *CommandService*. При тоа, доколку верзијата на *Android* e 7.0 или поголема, сервисот се стартува со *startForegroundService()*, а во спротивно со *startService()*. Во сервисот, доколку тој е повикан со *startForegroundService()*, се испраќа нотификација која го известува корисникот дека моментално задачата се извршува во позадина. Нотификацијата е *ongoing*, односно корисникот не може да ја избрише, што е карактеристика на еден *foreground service*. Задачата што ја извршува сервисот е повикување на *CommandAsyncTask* 3 пати, со тоа што *thread*-от спие 5 минути помеѓу повикувањата. Со тоа се постигнува да во рок од 15 минути (при секое стартување на алармот) се извршуваат командите што се дефинирани во *Async* *Task*-от. Методот *doInBackground()* на *CommandAsyncTask* прима *String* кој всушност претставува команда која треба да се изврши. При повикувањето на *CommandAsyncTask*, најпрво како *String* се предава „*ifconfig*“, потоа „*netstat –n*“ и на крај „*getprop*“, со тоа што претходно се проверува дали уредот е поврзан на интернет. Резултатите од извршувањето на командите се испишуваат во *Logcat*.

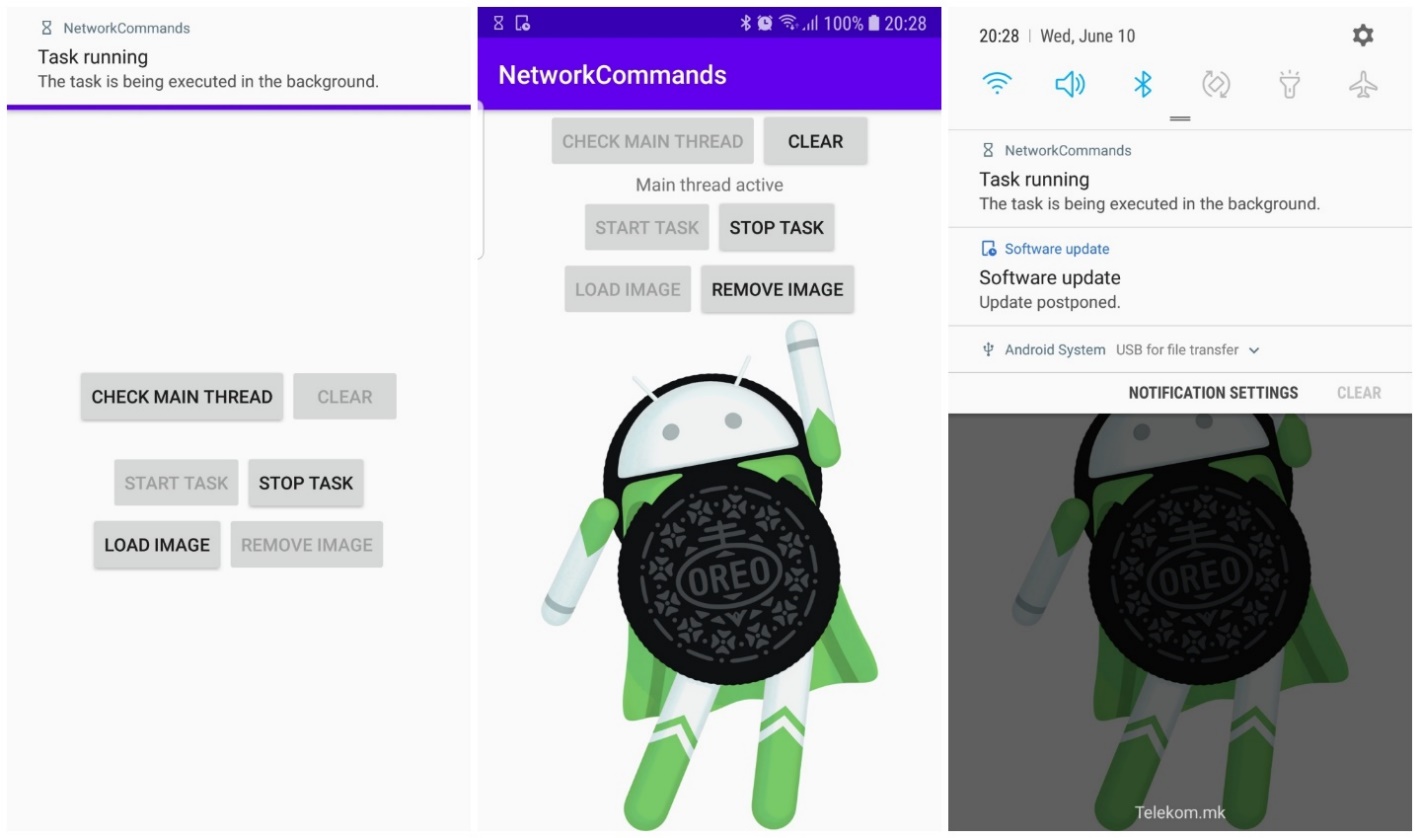
Со притискање на копчето *Stop task* се повикува функцијата *stopAlarm()* во *MainActivity*. Овде се откажува постоечкиот аларм, но и се праќа *broadcast intent* со акција *ACTION\_STOP\_TASK*. Овој *intent* го прифаќа *Broadcast* *Receiver*-от *NotificationReceiver* којшто е имплементиран како подкласа во *CommandService*. Овде се поставува променлива *flag* на *false* со што *Аsync* *Task*-от престанува да се извршува, доколку се користи *foreground* *service* се брише *ongoing* нотификацијата и конечно се прекинува сервисот. Истата логика е имплементирана и во *onDestroy()* методот на *MainActivity*, кадешто дополнително и се дерегистрира *AlarmReceiver*-от.

Од друга страна, со притискање на копчето *Load image* се иницијализира *AsyncTaskLoader ImageLoader*. Функцијата на овој *Loader* е да *load*-ира слика поставена на интернет, чиј *URL* се испраќа како *String* во *Bundle*-от. Потоа сликата се поставува во соодветниот *ImageView*.

Копчето *Check main thread* служи за демонстрација на тоа дека главниот *thread* на апликацијата е постојано активен додека се извршуваат позадинските *task*-ови.

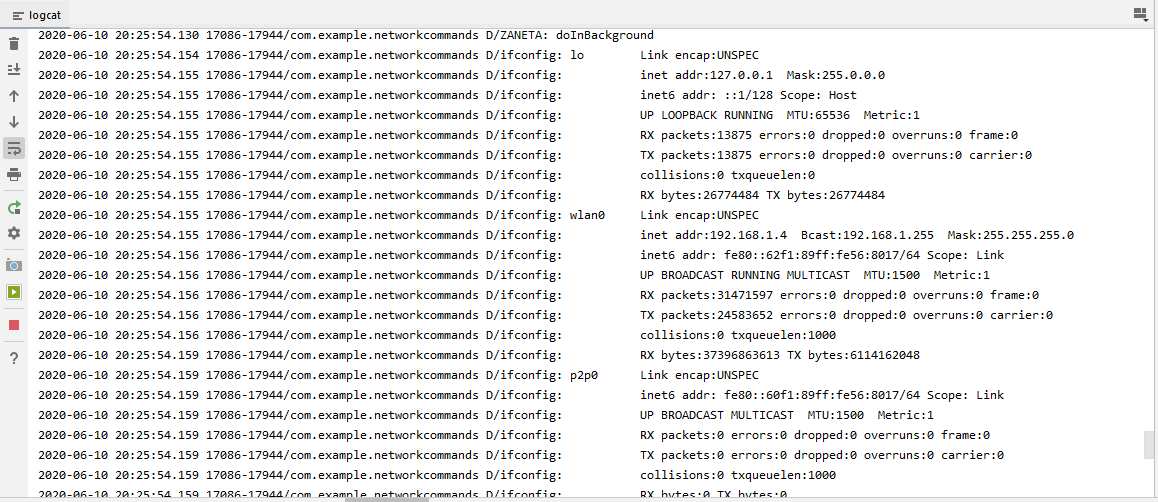
Апликацијата работи само во *portrait mode*, *landscape mode* на активноста е оневозможен.

Исклучувањето на апликацијата (активноста не е видлива, но апликацијата е пуштена) не влијае врз нејзината работа. Ако целосно ја исклучиме апликацијата, таа престанува со работа (објаснето погоре во делот со *onDestroy()*).

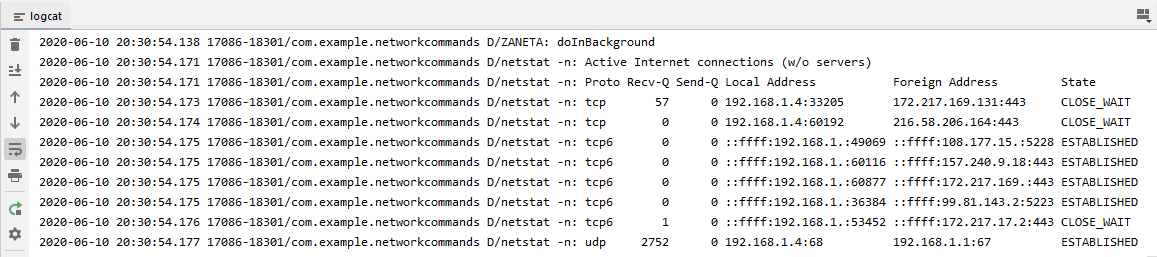
****

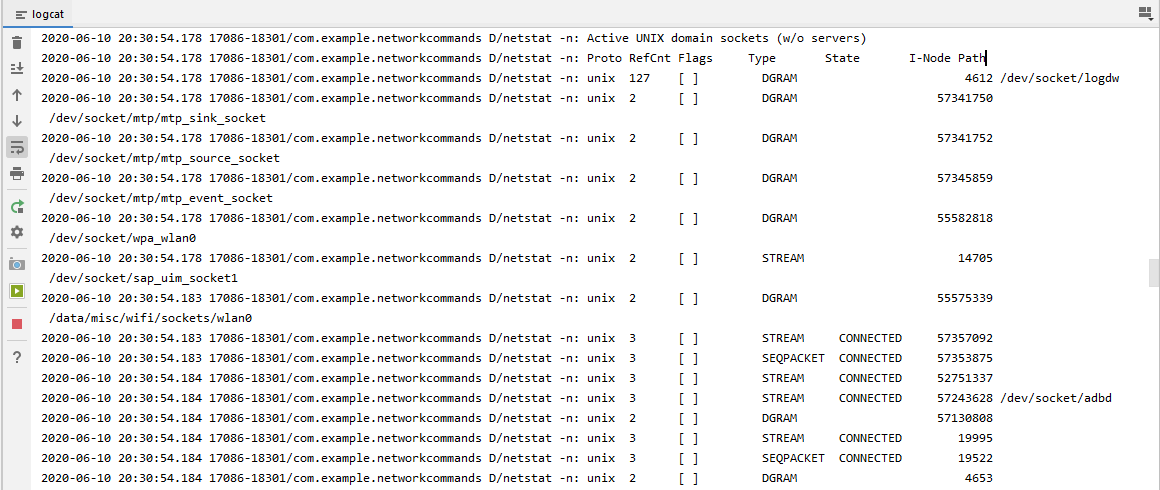
**Опис на командите**

Командата *ifconfig* (*interface* *configuration*) овозможува преглед и промена на конфигурацијата на мрежните интерфејси на системот. Командата без дополнителни параметри дава информации за сите моментално активни мрежни интерфејси.

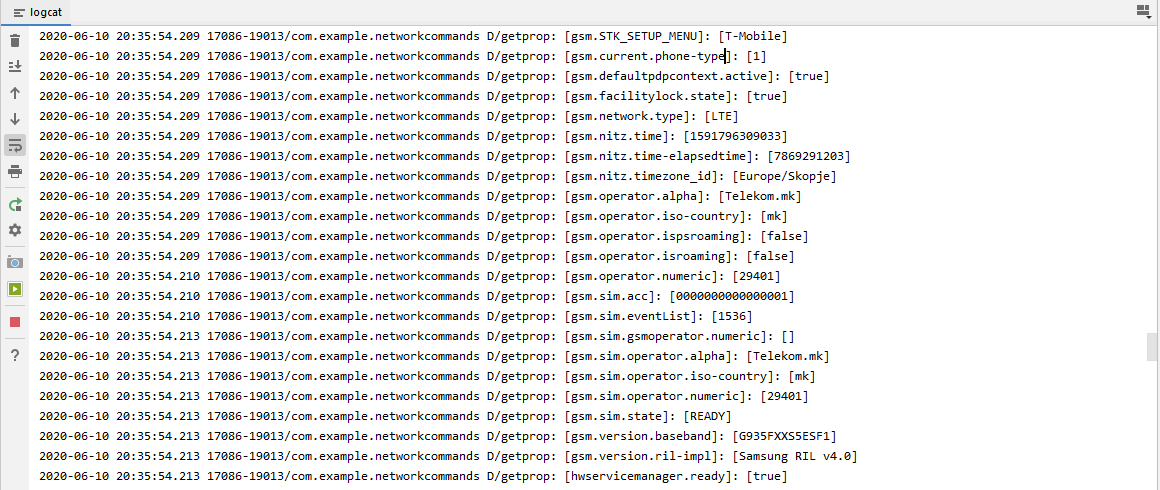


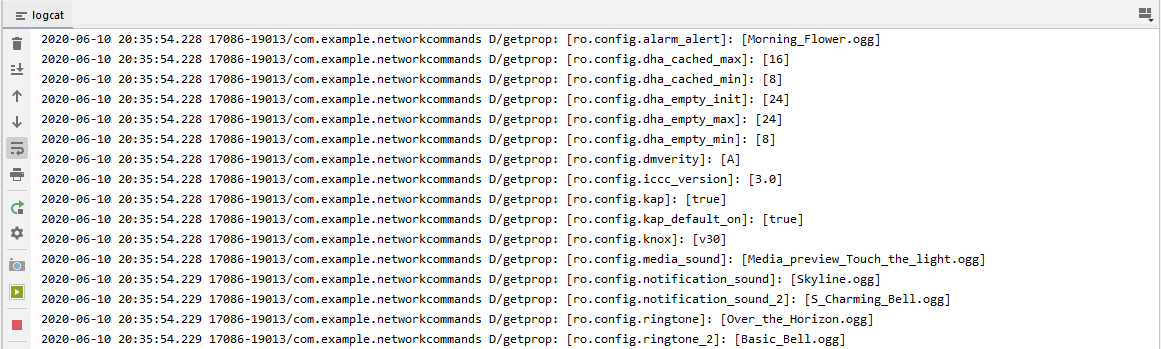
Командата *netstat* (*network* *statistics*) прикажува детални информации за тоа како нашата машина комуницира со други машини или мрежни уреди. Поспецифично, деталите се однесуваат на индивидуални мрежни конекции, мрежна статистика според протокол и слично. *–n* се користи за да не се разрешуваат *host* имињата за туѓи (*foreign*) *IP* адреси, што може значително да го намали времето на извршување на командата.





За разлика од претходните, командата *getprop* (*get* *properties*) не е мрежна команда, туку *ADB* (*Android* *Debug* *Bridge*) команда која се користи за добивање на сите карактеристики на *Android* уредот.





Подетални информации за овие команди можe да се најдат на линковите подолу.

**Користени линкови**

<https://www.tutorialspoint.com/android-asynctask-example-and-explanation>

<https://stackoverflow.com/questions/43093260/notification-not-showing-in-oreo>

* <https://stackoverflow.com/questions/582185/how-to-disable-landscape-mode-in-android>
* [https://android.stackexchange.com/questions/3715/is-there-a-command-or-application-similar-to-ipconfig](https://android.stackexchange.com/questions/3715/is-there-a-command-or-application-similar-to-ipconfig#:~:text=Android%20comes%20with%20ifconfig%2C%20install,root%20to%20use%20ifconfig%20though.&text=it%20will%20lists%20the%20interface,itself%2C%20or%20remotely%20via%20adb.)
* <https://www.computerhope.com/unix/uifconfi.htm>
* <https://www.lifewire.com/netstat-command-2618098>
* <https://stackoverflow.com/questions/11255568/how-to-read-output-of-android-process-command>

Дополнително ги искористив и линковите за *foreground service* што ни беа посочени во *workshop*-от.

**Линк од G*itHub*:** [**https://github.com/Zaneta14/NetworkCommands**](https://github.com/Zaneta14/NetworkCommands)