

# Operativni sistem Android i brza pretraga podataka

Roman Pavlović, Petar Jovanović, Nikola Teslić

**Sadržaj** — U radu su istražene mogućnosti realizacije programa za brzu pretragu korisničkih podataka na operativnom sistemu Android [1],[2]. Ciljevi rada su upoznavanje sa novom mobilnom platformom i realizacija programa za brzu pretragu. Prikazane su osobenosti samog operativnog sistema i postupci pri realizaciji brze pretrage. Realizovan je program brze pretrage sistema datoteka i adresara kao ogledni primer.

**Ključne reči** — Android, brza pretraga, sistem datoteka, baza podataka.

## I. UVOD

MOBILNE platforme su poslednjih godina prevazišle svoju prvobitnu namenu i približile se radnim stanicama. Zahvaljujući značajnom povećanju performansi, ove platforme pružaju veliki broj funkcija i mogu da pohrane velike količine podataka. Potrebni su novi pristupi pretraga kako bi podaci bili dostupniji i brže pruženi korisniku. U ovom radu je obrađena ideja da se kroz jedinstveni program korisnicima pruži uvid u što veći broj podataka pohranjenih na operativnom sistemu Android (u daljem tekstu, Android OS). Podaci obuhvataju sistem datoteka, adresar, kalendar, funkcije sistema, telefonske pozive i poruke. Na tržištu postoji nekoliko kompanija koje se bave brzom pretragom kao što su Nuance i Zi Corporation [5]. Rešenja ovih kompanije ne pokrivaju Android OS.

Rad daje uvid u Android OS i potrebnu organizaciju podataka za brzu pretragu. Dat je primer realizovanog programa na samom sistemu.

## II. OPERATIVNI SISTEM ANDROID

Android je programska podrška za mobilne platforme koja uključuje operativni sistem i ključne programe. Iako se često naziva operativnim sistemom on je uistinu mobilna platforma zasnovana na Linux-u. Pojavio se krajem 2007. godine. Dostupan je samo kroz Android SDK (*software development kit*) skup alata za razvoj programa. SDK pruža alate i sprege za programiranje (*API*) potrebne za razvoj programa na platformi Android koristeći programski jezik Java [3]. Sistem pruža sledeće funkcije:

1. Application framework - omogućuje ponovno korišćenje i zamenu komponenti.
2. Dalvik virtual machine - Java virtualna mašina optimizovana za mobilne platforme.
3. Integrated browser - pretraživač zasnovan na konceptu otvorenog koda WebKit.
4. Optimizovan prikaz dvodimenzionalnih grafičkih komponenti zasnovanih na grafičkoj biblioteci; trodimenzionalna grafika zasnovana na OpenGL ES 1.0 specifikaciji.
5. SQLite - strukturirani pristup bazama podataka.
6. Media - podrška za audio, video i formate slika (MPEG-4, H.264, MP3, AMR, JPG, PNG, GIF).
7. GSM Telefonija.
8. Bluetooth, EDGE, 3G i WiFi.
9. Kamera, GPS, kompas i akcelerometar.



Sl. 1. Arhitektura operativnog sistema Android.

Android OS se sastoji iz tri sloja (prikazani na Sl. 1.):

### A. Programsko okruženje (Framework)

Programeri imaju pun pristup istom API razvojnog okruženja kao i sistemskim programima. Programski sloj je razvijen sa ciljem da pojednostavi korišćenje komponenti nižih nivoa. Svim programima je omogućeno da međusobno dele resurse.

### B. Izvršni sloj (Android runtime)

Android uključuje skup sistemskih biblioteka putem kojih se nudi većina funkcionalnosti sistemskih biblioteka napisanih u Java programskom jeziku. Svaki program je predstavljen jednim procesom, sopstvenom instancom Dalvik virtualne mašine (DVM) [2]. DVM je napisan tako da omogućuje pokretanje više virtuelnih mašina sa efikasnim iskoriscenjem resursa. DVM pokreće izvršne datoteke u .dex formatu koji je optimizovan za minimalno memorijsko zauzeće. DVM je virtualna mašina zasnovana na registrima i oslanja se na Linux jezgro za rad sa nitima i memorijom na nižem nivou.

— Ovaj rad je delimično finansiran od Ministarstva za nauku Republike Srbije, projekat 11005, od 2008. god

Roman Pavlović, Fakultet tehničkih nauka, Novi Sad, Srbija (e-mail: roman.pavlovic@rt-rk.com).

Petar Jovanović, RT-RK, Sistemi zasnovani na računaru, Novi Sad, Srbija (e-mail: petar.jovanovic@rt-rk.com)

Nikola Teslić, Fakultet tehničkih nauka, Novi Sad, Srbija (e-mail: nikola.teslic@rt-rk.com)

### C. Jezgro sistema Linux (kernel)

Android se oslanja na jezgro operativnog sistema Linux, v2.6 kada je u pitanju sigurnost, rukovanje memorijom, rukovanje procesima, mrežni stek i model upravljačkih programa (*driver*). Jezgro Linux-a predstavlja apstraktni nivo koji povezuje elemente fizičke arhitekture sa programskom podrškom.

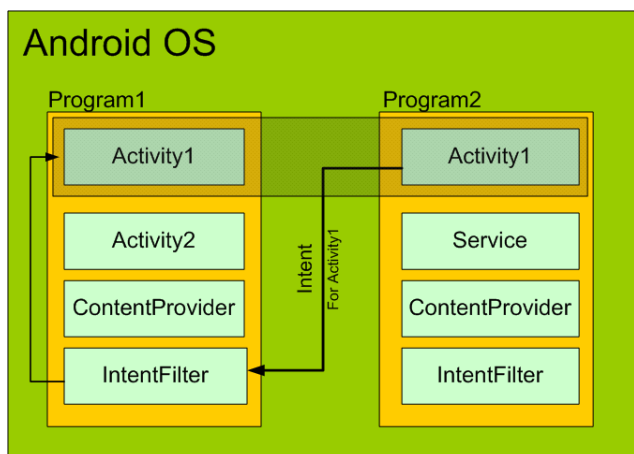
### III. BLOKOVİ ANDROID OS-A

Postoje četiri glavna gradivna bloka Android programa:

1. Aktivnost (*Activity*);
2. Blok koji reaguje na spoljne događaje (*Broadcast Intent Receiver*);
3. Usluga (*Service*);
4. Davalac sadržaja (*Content Provider*).

Svaki program treba da sadrži neku kombinaciju ovih blokova. Ne moraju se koristiti svi blokovi (Sl. 2.).

Komponente koje se koriste u programu nabrojane su u datoteci *AndroidManifest.xml*. To je XML (*Extensible Markup Language*) datoteka koja sadrži informacije o korišćenim komponentama, njihovim ograničenjima i spoljnim događajima koje je program spreman da obradi.



Sl. 2. Blokovi Android OS-a

#### A. Aktivnost

Ovo je najčešće korišćen Androidov blok. Aktivnost se najčešće predstavlja kao korisnička sprega programa. Svaka aktivnost se realizuje kao klasa koja nasleđuje osnovnu klasu *Activity*. Ova klasa se sastoji od korisničke sprege sastavljenog od Pogleda (*Views*) i reakcija na događaje. Većina programa se sastoji od više korisničkih grafičkih sprega tj. ekrana (*npr. program za poruke sadrži ekrane Inbox, Outbox, Sent ...*). Svaki od ovih ekrana mora biti implementiran kao posebna aktivnost. Prelazak na drugi ekran (*Screen*) se izvršava pokretanjem nove aktivnosti. Kada se pokrene novi ekran, stari se pauzira i stavlja na poseban stek koji predstavlja istoriju rada pokrenutih programa na sistemu (*History Stack*). Sistem može da se kreće unazad po ovom steku. Android zadržava na steku svaki program koji je pokrenut sa glavnog ekrana (*Home Screen*). Rad sa stekom rezervisan je isključivo za operativni sistem.

#### B. Blok koji reaguje na spoljne događaje

Za kretanje između ekrana Android koristi posebnu klasu *Intent*, koja opisuje zahteve za odgovarajućim akcijama. Sastoji se iz dva glavna dela, akcije i dela za podatke na osnovu kojih se reaguje. Sa njom je povezana klasa *IntentFilter* koja definiše tipove zahteva koje odgovarajući program može da obradi.

Klasa *IntentReceiver* omogućuje reakciju odgovarajućeg programa na određene događaje operativnog sistema ili drugih programa. Klasa *IntentReceiver* ne koristi korisničku spregu (*UI*). Može koristiti neka od sistemskih obaveštenja, prozore obaveštenja koje pruža operativni sistem. Sistemski događaji na koje je program spreman da reaguje moraju biti registrovani u datoteci *AndroidManifest.xml*. Moguće ih je registrovati i iz koda pozivanjem *registerReceiver()* funkcije klase *Context*.

#### C. Usluga

Usluga je program nezavisan od UI. Više aktivnosti mogu koristiti istu uslugu. Dobar primer za ovo je program za rad sa audio datotekama. Komande programa i prikaz liste mogu biti prikazani na zasebnim ekranima. Usluga omogućuje pokretanje audio datoteka nezavisno od ekrana. Sistem se brine o uslugama, memoriji koje zauzimaju i procesorskom vremenu.

#### D. Davalac sadržaja

Pogradi mogu da pohranjuju svoje podatke u datoteke, SQLite baze podataka [4] ili neke druge strukture. Ova klasa sadrži standardni skup metoda za skladištenje i slanje podataka koji su joj dodeljeni.

### IV. PRIKAZ MODELA PRETRAGE PODATAKA

Nakon analize mobilne platforme Android, specifičnosti i novina koje unosi, modelovan je sistem pretrage. Zadatak sistema je dobavljanje svih dostupnih podataka po zadatom kriterijumu korisnika. AndroidOS, u svojoj drugoj verziji, pruža korisniku informacije adresara i sistema datoteka na kojima se zasniva pretraga. Sistem se zasniva na pretragama tabela baza podataka koje sadrže informacije od interesa za korisnika. Table se popunjavaju informacijama pre početka rada programa. Ovaj princip se naziva indeksiranje unapred informacija sistema. Na njemu se zasnivaju sve brze pretrage. Zadatak sistema za pretragu je održavanje baza sa unapred indeksiranim informacijama konzistentnima. Informacije se dobijaju postavljanjem upita nad bazom a ne pretragom celog sistema.

#### A. Organizacija podataka

Android OS ima snažnu podršku za SQLite3 baze podataka. Ova novina na mobilnim platformama olakšava posao čuvanja i pristupa podacima na sistemu. Korišćenje SQL (*Structured Query Language*) za rad sa bazama ubrzava unapred indeksiranje informacija sistema. Korisnički podaci na sistemu se čuvaju u bazama. Sistemске baze sadrže informacije o telefonskim kontaktima i pozivima, korisnicima i sistemu datoteka. O njihovoj konzistentnosti se brine operativni sistem. Sistemška baza za rad sa datotekama pokriva samo audio,

video i grafičke tipove datoteka. Pretraga sistema datoteka podrazumeva pretragu svih datoteka. Problem se prevazilazi stvaranjem baze podataka koja treba da sadrži informacije o svim datotekama na sistemu. Informacije koje opisuju datoteke na sistemu su puna putanja, ime, veličina i tip datoteke. Baza podataka se popunjava prolaskom kroz stablo sistema datoteka i dobavljanjem informacija o datotekama. Sistem datoteka je organizovan po uzoru na Unix sistem datoteka. Baze podataka sistema datoteka je potrebno popuniti samo jednom, pri samom pokretanju operativnog sistema ili pri pokretanju programa po prvi put. Prilikom realizacije rešenja izabran je drugi pristup. Na taj način je obezbeđena konzistentnost sadržaja baze u trenutku pokretanja programa za pretragu. Mana ovog pristupa se ogleda u sporijem pokretanju programa. Android poseduje ranije pomenuti programski stek na kome se čuvaju pokrenuti programi. Čak i nakon njihovog zatvaranja programi ostaju na ovom steku. Sistem uklanja programe sa steka kada ostane bez raspoložive memorije. Kriterijum uklanjanja programa je vreme neaktivnosti programa. Iz prethodnog se vidi da program za pretragu neće kompletno pretraživati sistem kada se zatraži njegova usluga, pošto ostaje aktivan u memoriji. Izuzetak je prvo pokretanje programa (ili slučaja posle duge neaktivnosti kada operativni sistem sam ugasi program).

Android ne pruža programerima funkcionalnost primanja sistemskih obavesti (*Notifications*). Ukoliko u toku rada dođe do izmene sistema datoteka ili adresara nemoguće je da program primi obaveštenje o tome. Ovaj nedostatak onemogućava praćenje sistema datoteka u toku rada programa. Problem se prvazilazi implementacijom modula za praćenje sadržaja direktorijuma sistema datoteka. Ovakav pristup zahteva mnogo resursa. Kompromis između količine zauzetih resursa i vremena potrebnog za osveženje baze podataka se postiže postavljanjem modula za praćenje sadržaja direktorijuma na one koji se nalaze direktno ispod korena stabla sistema datoteka. Svaka izmena sistema datoteka na nekoj od ovih grana pokreće novu pretragu te grane sistema datoteka. Brišu se stari i popunjavaju se novi podaci o datotekama te grane.

### *B. Pretraga podataka*

Zadavanjem željenog niza slova se započinje pretraga. Pretraga se vrši nad bazama podataka. Podaci su sadržani u sistemskoj bazi sa informacijama o adresaru i korisničkoj bazi sa informacijama o sistemu datoteka. Upiti na baze se postavljaju korišćenjem SQLite3 biblioteke. Rezultati pretrage se prikazuju na ekranu mobilne platforme, a zavise od prethodno podešenog filtera. Filter označava tip podataka koji se pretražuju. Može biti podešen na pretragu adresara, sistema datoteka, samo lokalne memorije sistema datoteka ili dodatne memorijske kartice.

## V. KONCEPT REŠENJA OGLEDNOG PROGRAMA

Na osnovu prethodne analize problema realizovani ogledni program je podeljen u više modula. Potrebni moduli za realizaciju programa su glavni modul, modul za rad sa bazom podataka, modul za praćenje stanja sistema datoteka i moduli Adaptera za prikaz rezultata na ekranu.

### *A. Koraci pri realizaciji programa*

1. Realizacija korisničke sprege. Iscrtavanje potrebnih komponenti na ekran i uspostavljanje kontrole nad njima.
2. Realizacija modula za rad sa bazom podataka i sistemom datoteka.
  - 2.1. Stvaranje baze podataka.
  - 2.2. Implementacija osnovnih operacija nad bazom (*ubacivanje i brisanje elemenata iz baze*).
  - 2.3. Realizacija funkcije pretrage sistema datoteka.
  - 2.4. Realizacija Adaptera u koje treba da se smeste rezultati pretrage. Potrebno je realizovati dva adaptera. Prvi adapter treba da objedini prikaz rezultata pretrage kontakata i baze sistema datoteka. Drugi adapter treba da smesti rezultate pretrage baze sistema datoteka.
  - 2.5. Realizacija funkcija za pretragu baze podataka.
3. Realizacija modula za praćenje sistema datoteka.
4. Funkcije za pokretanje i zaustavljanje rada modula za praćenje sistema datoteka.
5. Realizacija modula aktivnosti za prikaz informacija o određenom rezultatu.

### *B. Kratak opis modula i njihova zaduženja*

#### *1.1. Glavni modul*

Zadatak glavnog modula se ogleda u upravljanju radom komponenti korisničke sprege, modulom za praćenje stanja sistema datoteka i ostalim modulima. Na odabrani rezultat pretrage poziva jednu od dve aktivnosti za detaljniji pregled rezultata. Ukoliko se radi o telefonskom kontaktu poziva se sistemski program za pregled telefonskih kontakata. Na odabran rezultat pretrage nad datotekama pokreće se aktivnost za prikaz informacija o datotekama. U zavisnosti od stanja programa modul reaguje na odgovarajući način: ažurira bazu, pokreće i zaustavlja module za praćenje sistem datoteka.

#### *1.2. Modul za rad sa bazom podataka sistema datoteka*

Predstavlja programsku osnovu. Ostvaruje kompletnu podršku za pretragu i rad sa podacima na sistemu datoteka. Obavlja osnovne operacije nad bazom: otvaranje i zatvaranje baze, dodavanje i brisanje elemenata baze, pretragu nad elementima baze. Da bi baza u svakom trenutku bila ažurirana ovaj modul sadrži modul za praćenje stanja sistema datoteka. Ne prate se sistemski direktorijumi i direktorijumi za privremeno smeštanje informacija o sistemu: /data, /system, /dev i /tmp.

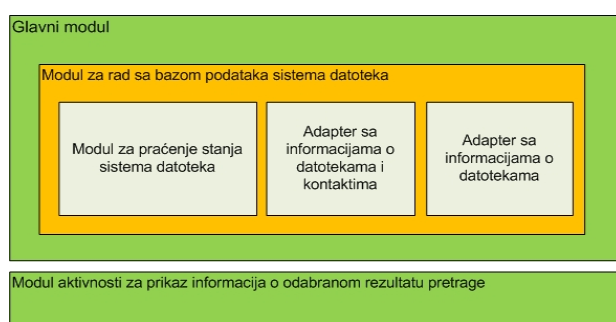
#### *1.3. Adapteri sa informacijama o kontaktima i datotekama*

Realizacija adaptera je potrebna da se uspešno prikažu rezultati pretrage. Oni predstavljaju vezu između baze podataka telefonskih kontakata, baze podataka datoteka i liste na korisničkoj sprezi u kojoj se prikazuju rezultati. Adapter se sastoji od liste elemenata određene klase. Obezbeđuje korektan prikaz elemenata na ekranu unutar nekog polja korisničke sprege. Program sadrži dva ovakva adaptera pri čemu svaki ima elemente liste druge klase

(jedan sadrži instance klase koja sadrži informacije o rezultatima pretrage baze sistema datoteka, dok drugi sadrži i informacije o rezultatima pretrage kontakata). Adapteri su realizovani u dve klase AllViewAdapter i FileViewAdapter.

#### 1.4. Modul za praćenje stanja sistema datoteka

Postavlja se na direktorijume čiji se sadržaj želi pratiti i prati promene na zadatoj grani sistema datoteka. Ukoliko dođe do promene ažurira se baza na način da se sva polja koja potiču od tog direktorijuma brišu i ponovo se pretražuje direktorijum. Ovaj modul je uključen u modul za rad sa bazom sistema datoteka. Klasa koja realizuje ovaj modul nasleđuje klasu FileObserver iz paketa android.os.FileObserver.



Sl. 3. Grafički prikaz modula oglednog programa.

## VI. ZAKLJUČAK

Android je operativni sistem namenjen mobilnim platformama. Još uvek je u procesu razvoja što u mnogome ograničava razvoj programske podrške. Većina standardnih programa kao što su Kalendar, Pretraživač datoteka, Program za rad sa slikama još uvek nisu uključeni u razvojno okruženje Android SDK. Ugrađena baza datoteka na sistemu prepoznaje samo osnovne tipove media datoteka. Shodno tome, realizovana je baza koja pruža informacije o datotekama na sistemu, kao i neophodne funkcije kako bi se baza održavala ažurnom. Podrška za sistemski obaveštenja o promenama na sistemu datoteka u sadašnjoj verziji nije dostupna korisniku, tako da je prilikom razvoja programa korišćena manje optimalna metoda praćenja sadržaja direktorijuma, a sve u cilju održavanja baze ažuriranom. Prilikom izrade

programa korišćene su preporuke od strane proizvođača Androida (Google). Preporuke se odnose na strukturu programa i način primene komponenti. Mnoge novine koje uvodi Android OS olakšavaju posao programera. Neki primeri za to su realizovanje programa u Javi, unapred definisana grafičke sprege, adapteri, blokovi za reagovanje na spoljne događaje...

Android je mobilna platforma sa kodom otvorenog tipa (open source) sa besplatnim razvojnim alatima (Eclipse, SDK, Linux). Kombinacija Linux i DVM (*Dalvik Virtual Machine* - optimizovana verzija JVM za mobilne platforme) daje mnoge prednosti kao što su mogućnosti C sistemskog programiranja ali i Java objektnog razvoja programa. Sistemski programski podrška je već na zavidnom nivou što znači da su realizovane biblioteke za sve dodatke za mobilne platforme.

Na razvoju ove platforme proizvođaču Androida Googlu su se pridružile firme članice *Open Handset Alliance*-a (kao što su HTC, Intel, Texas Instruments, Samsung). Postoje razna predviđanja razvoja Androida. Autori ovog rada smatraju da je prvi sledeći korak u razvoju prenošenje sistema na fizički uređaj, podrška za sistemski obaveštenja i poboljšanje UI-a.

## LITERATURA

- [1] Dušica Marijan, "Pretraga podataka na mobilnim uređajima sa operativnim sistemom Linux", Etran 2008.
- [2] <http://code.google.com/android>, Google co.
- [3] David Flanagan, "Java in a Nutshell: A Desktop Quick Reference", O'Reilly, 2005.
- [4] M. Maslakowski, "Sam's Teach Yourself MySQL in 21 Days", 2000.
- [5] George Voulgaris, ITM Analyst Partner, "Activating the Idle Screen: Uncharted Territory"

## ABSTRACT

**This paper describes possible solutions of implementation programs for fast user data search on AndroidOS mobile platform. New Android platform is open source operating system for mobile phones that combines power of Linux kernel and easy Java programming. The paper gives an overview of new features that Android brings to mobile platforms and how they can be used for speeding up user data search. It describes one solution to this problem.**

## Operating system Android and fast user data search

Roman Pavlović, Petar Jovanović, Nikola Teslić