



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
НОВИ САД
Департман за рачунарство и аутоматику
Одсек за рачунарску технику и рачунарске комуникације

ЗАВРШНИ (BACHELOR) РАД

Кандидат: Иван Милосављевић

Број индекса: 12711

Тема рада: РАЗВОЈ ДЛНА КОНТРОЛЕРА ЗА ТАБЛИЧНЕ РАЧУНАРЕ НА
БАЗИ АНДРОИД ОПЕРАТИВНОГ СИСТЕМА

Ментор рада: Проф. др Јелена Ковачевић

Нови Сад, август, 2012.



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:	
Идентификациони број, ИБР:	
Тип документације, ТД:	Монографска документација
Тип записа, ТЗ:	Текстуални штампани материјал
Врста рада, ВР:	Завршни (Bachelor) рад
Аутор, АУ:	Иван Милосављевић
Ментор, МН:	Проф. др Јелена Ковачевић
Наслов рада, НР:	РАЗВОЈ ДЛНА КОНТРОЛЕРА ЗА ТАБЛИЧНЕ РАЧУНАРЕ НА БАЗИ АНДРОИД ОС
Језик публикације, ЈП:	Српски / латиница
Језик извода, ЈИ:	Српски
Земља публикавања, ЗП:	Република Србија
Уже географско подручје, УГП:	Војводина
Година, ГО:	
Издавач, ИЗ:	Ауторски репринт
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6
Физички опис рада, ФО: <small>(поглавља/страна/ цитата/табела/слика/графика/прилога)</small>	
Научна област, НО:	Електротехника и рачунарство
Научна дисциплина, НД:	Рачунарска техника
Предметна одредница/Кључне речи, ПО:	дЛНА, МУЛТИМЕДИЈА, НАМЕНСКИ РАЧУНАР, УРНП, АНДРОИД, ЈНИ
УДК	
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад
Важна напомена, ВН:	
Извод, ИЗ:	У овом раду је приказано решење модула за контролу и репродукцију мултимедијалног садржаја у ДЛНА мрежи. Решење је реализовано за Самсунг Галакси таблични рачунар на верзији 2.3 Андроид ОС.
Датум прихватања теме, ДП:	
Датум одбране, ДО:	
Чланови комисије, КО:	Председник: др Илија Башичевић
	Члан: др Иштван Пап
	Члан, ментор: др Јелена Ковачевић
	Потпис ментора



KEY WORDS DOCUMENTATION

Accession number, ANO :	
Identification number, INO :	
Document type, DT :	Monographic publication
Type of record, TR :	Textual printed material
Contents code, CC :	Bachelor Thesis
Author, AU :	Ivan Milosavljević
Mentor, MN :	PhD prof.dr Jelena Kovačević
Title, TI :	Development DLNA controller for tablet based Android OS
Language of text, LT :	Serbian
Language of abstract, LA :	Serbian
Country of publication, CP :	Republic of Serbia
Locality of publication, LP :	Vojvodina
Publication year, PY :	
Publisher, PB :	Author's reprint
Publication place, PP :	Novi Sad, Dositeja Obradovica sq. 6
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)	
Scientific field, SF :	Electrical Engineering
Scientific discipline, SD :	Computer Engineering, Engineering of Computer Based Systems
Subject/Key words, S/KW :	DLNA, multimedia, personal computer, UPnP, Android, JNI
UC	
Holding data, HD :	The Library of Faculty of Technical Sciences, Novi Sad, Serbia
Note, N :	
Abstract, AB :	This paper describes the implementation of modules for control, reside and playback of multimedia content to DLNA network. The decision had been realized for the Samsung Galaxy Tablet on Android OS version 2.3.
Accepted by the Scientific Board on, ASB :	
Defended on, DE :	
Defended Board, DB :	President: Ilija Basiccevic, PhD
	Member: Istvan Pap, PhD
	Member, Mentor: Jelena Kovacevic, PhD
	Mentor's sign

Zahvalnost

Zahvaljujem stručnim saradnicima prof.dr Jeleni Kovačević, Tomislavu Maruni i Nikoli Crvenkoviću na pruženoj pomoći tokom izrade završnog (*bachelor*) rada

SADRŽAJ

1. Uvod	1
2. Teorijske osnove.....	3
2.1 Sadržaji i uređaji	3
2.1.1 Infrastruktura programska podrška.....	4
2.1.2 UPNP protocol.....	4
2.1.3 Uređaj za deljenje multimedijalnog sadržaja	5
2.1.4 Uređaj za reprodukciju multimedijalnog sadržaja u lokalnoj mreži	5
2.1.5 Uređaj za reprodukciju multimedijalnog sadržaja primljenog od DMC.....	6
2.1.6 Uređaj za kontrolu i reprodukciju multimedijalnog sadržaja	6
2.2 DLNA ograničenja.....	7
2.3 Android operativni sistem.....	7
2.4 Java programski jezik	9
2.4.1 Sprega Java programskog jezika i C koda.....	10
2.4.1.1 Kada koristiti JNI.....	10
2.4.1.2 Prednosti JNI arhitekture.....	11
2.4.1.3 Nedostatci korištenja JNI aritekture.....	12
2.5 Eclipse razvojno okruženje	12
3. Koncept rešenja	13
3.1 Komponente sistema.....	13
3.1.1 Struktura Android aplikacije	14
3.1.1.1 Komponente aplikacije.....	14
3.1.2 Meta-jezik za označavanje tekstualnih dokumenata.....	15
3.1.3 Aplikacijski resursi.....	15
4. Programsko rešenje.....	16

4.1	Gradenje izvornog koda.....	16
4.1.1	Kreiranje sablona za pisanje android.mk datoteka	16
4.2	JNI sloj.....	17
4.2.1	JNI realizacija.....	18
4.2.2	Modul lubunp	18
4.2.3	Modul liblma_rel	19
4.2.4	Modul mmlib.....	19
4.2.5	Paket android.os	19
5.	Rezultati.....	20
5.1	Ispitivanje za tablet Samsung Galaksi	20
6.	Zaključak	24
7.	Literatura	25

SPISAK SLIKA

Slika 2.1 DLNA okruženj.....	3
Slika 2.2 DLNA kućna mreža	4
Slika 2.3 Komunikacija UPNP.....	5
Slika 2.4 Komunikacija uređaja u mrži.....	6
Slika 1.5 JNI u procesu povezivanja Jave i C/C ++ programskog jezika.....	11
Slika 1.1 Komponente DLNA sistema	14
Slika 4.1 Primer Android.mk modula fs-lib.....	17
Slika 4.2 Primer Android.mk modula mmlib.....	19
Slika 1.1 Izgled aplikacije nakon startovanja.....	21
Slika 1.2 Izgled aplikacije nakon odabira pretraživanja uređaja	22
Slika 1.3 Prikaz dostupnog sadržaja.....	23

SKRAĆENICE

- AIDL** - *Android Interface Definition Language*, Android jezik za definisanje sprege
- DLNA** - *Digital Living Network Alliance*, Protokol za deljenje digitalnog sadržaja između multimedijalnih uređaja
- JNI** - *Java native interface*, Sprega Java programskog jezika i C koda
- IPC** - *Inter-process communication*, međuprocena komunikacija
- OS** - *Operating system*, Operativni sistem
- CAM** - *Conditional – Access module*, Modul za uslovni pristup
- XML** - *Extensive Markup Language*, proširivi metajezik za označavanje tekstualnih dokumenata
- CE** - *Consumer electronics*, Korisnička elektronika
- DMC** - *Digital Media Controller*, Modul za kontrolu rada DMP i DMS u lokalnoj mreži
- DMP** - *Digital Media player*, Modul za reprodukciju u lokalnoj mreži
- DMS** - *Digital Media server*, Modul za deljenje sadržaja klijentima u lokalnoj mreži
- DMR** - *Digital Media renderer*, Modul za reprodukciju sadržaja
- HTTP** - *Hypertext Transfer Protocol*, protokol za prenos hiper teksta
- UPNP** - *Universal Plug and Play*, Skup mrežnih protokola koji dozvoljavaju multimedijalnim uređajima da ostvare međusobne veze i uspostave razne mrežne usluge
- IP** - *Internet Protocol*, Internet protocol
- URL** - *Uniform Resource Locator*, Jedinstvena adresa resursa
- NAS** - *Network Attached Storage*, Uređaj koji u sebi sadrži hard disk
- LAN** - *Local area network*, Lokalna računarska mreža
- TCP** - *Transmission Control Protocol*, Protokol kontrole toka
- MPEG** - *Motion Picture Expert Group*, Standard za kodovanje
- JPEG** - *Joint Photographic Experts Group*, Standard kodovanja slike

VM - *Virtual Machine*, Virtualna masina.

DVM - *Dalvik Virtual Machine*, Dalvik virtuelna masina

ARM - *Advanced RISC Machine*, RISC arhitectura

AVI - *Audio Video Interleave*, Zvuk i pokretna slika sprega

HD - *High-definition*, Visoka rezolucija

SOAP - *Simple Object Access Protocol*, Jednostavni pristup objektu protokol

1. Uvod

U ovom radu je realizovana programska podrška namenjen za kontrolu i reprodukciju multimedijalnog (zvuk, pokretna i nepokretna slika) sadržaja (DMC - *Digital Media Controler*)[1] na udaljenom kompatibilnom uređaju (DMR - *Digital Media renderer*). Programska podrška je realizovana za tablične računare i mobilne telefone na bazi android opeativnog sistema.

Prenos multimedijalnog sadržaja između uređaja za deljenje, kontrolu i reprodukciju multimedijalnog sadržaja definiše DLNA (*Digital Living Network Alliance*)[2] protokol. DLNA objedinjuje proizvođače namenskih računarskih struktura koji poštuju određeni standard oblika prenosa. Članovi DLNA grupe razvili su koncept žične i bežične interoperabilnosti mreža, u kojima se digitalni sadržaj može koristiti iz personalnih računara, tabličnih računara i mobilnih uređaja, u i izvan kuće.

Prilikom razvoja programske podrške za Android operativni sistem, prirodno okruženje za razvoj programeru predstavlja Java programski jezik. Ovaj jezik obično zahteva fizičke arhitekture sa više resursa, što često nije slučaj sa sistemima u realnom vremenu. Da bi iskoristili sve dostupne resurse, programeri koriste programske jezike poput C/C++, ali ovim rešenjima nedostaje olakšana prenosivost koda i pojednostavljeno otklanjanje grešaka [3]. Takođe je prikazan način komunikacije Java servisa sa programskom podrškom korišćenjem sprege Java programskog jezika i C koda (JNI - *Java native interface*).

Ovaj rad je sačinjen od sedam poglavlja.

Drugo poglavlje opisuje DLNA, programsku podršku, osnove android operativnog sistema i java programskog jezika.

U trećem poglavlju dat je koncept i opis rešenja rada.

Četvrto poglavlje sadrži detaljan opis uvedenog Android servisa, prikaz sprege Java i C koda, kao i način korišćenja servisa u Android aplikaciji.

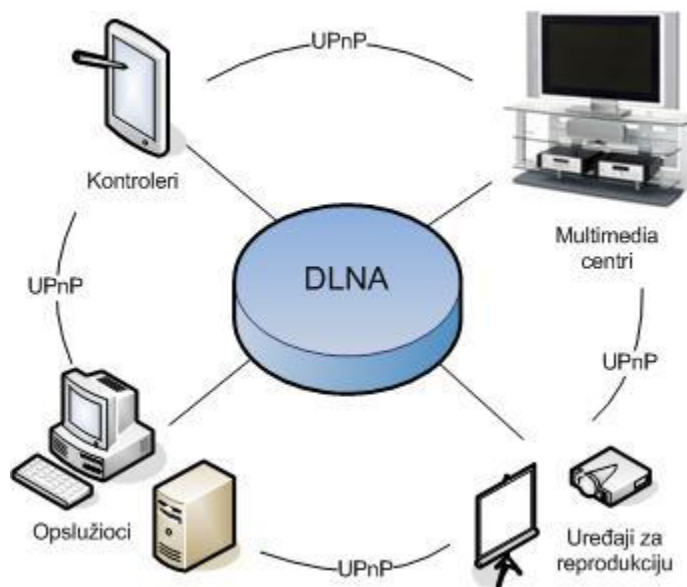
U petom poglavlju su opisani načini ispitivanja i verifikovanja svih modula u Java servisu.

Šesto poglavlje sadrži pregled onoga što je urađeno u ovom radu, kakvi su dalji pravci razvoja i moguća unapređenja.

U sedmom poglavlju dat je spisak literature korišćene za izradu rada.

2. Teorijske osnove

Sony i Intel osmislili su DLNA standard 2003. godine, a u međuvremenu se priključilo više od 200 firmi iz oblasti mobilne telefonije, elektronike i informacione tehnologije. Zajednički cilj je da proizvodi potrošačke elektronike budu međusobno kompatibilni, odnosno da definišu zajednička pravila koja će omogućiti različitim grupama proizvođača (pametni telefoni, tablični računari, personalni računari, digitalni foto-aparati i kamere, pametni televizori) da koriste sadržaje koji su skladišteni unutar mreže.

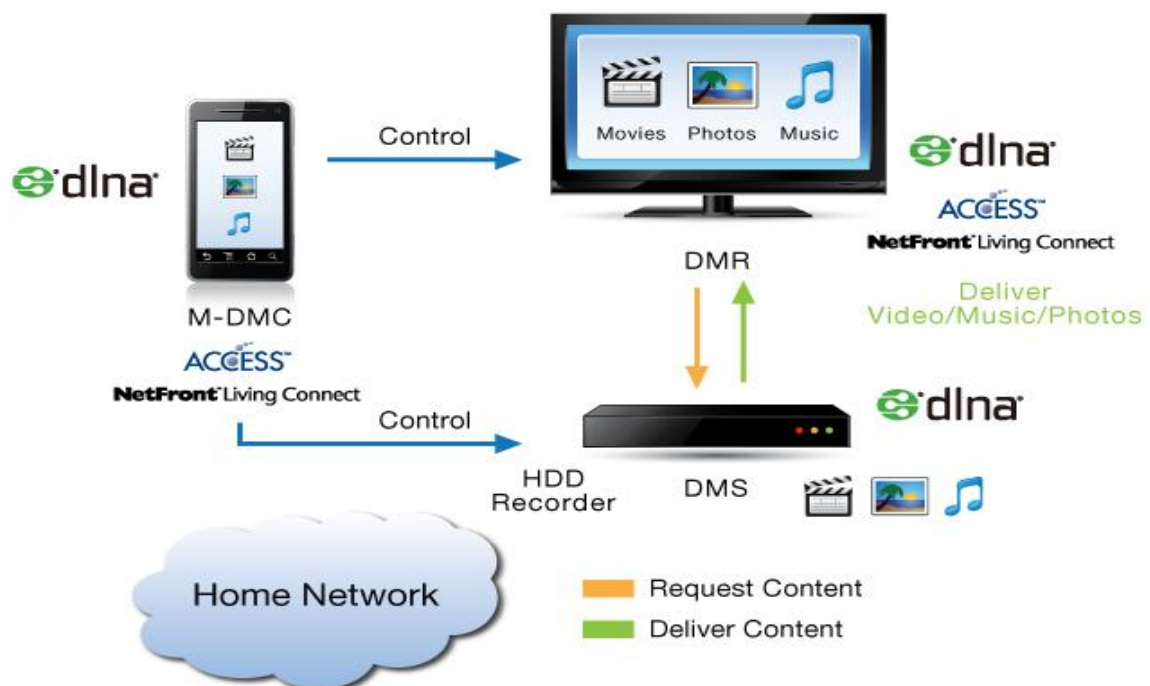


Slika 2.1 DLNA okruženje

2.1 Sadržaji i uređaji

DLNA s jedne strane razlikuje sadržaje zvuk, pokretnu i nepokretnu sliku, kao i različite tipove uređaja, kao na primer uređaje za deljenje, kontrolu i reprodukciju multimedijalnog

sadržaja, kako bi drugi uređaji koji su uključeni u mrežu mogli da zatraže odgovarajuću uslugu. Na DLNA sertifikatu, koji se dobija nakon što uređaj prođe sve ispite i provere, može se videti za koje uređaje i medijume postoji podrška. U standardne formate spada standard za kodovanje slike (JPEG - *Joint Photographic Experts Group*), za zvuk, a za pokretnu sliku - sadržaje (MPEG2 *Motion Picture Expert Group*). Osim tih minimalnih zahteva, većina uređaja, naravno, podržava mnogo više formata. Svako ko želi da se informiše pre nego što kupi novi hardver, može to da uradi preko pretrage proizvoda na zvaničnoj DLNA web stranici.



Slika 2.2 DLNA kućna mreža

2.1.1 Infrastruktura programska podrška

Preduslov za zajedničko korišćenje medija je da svi uređaji moraju da podržavaju UPNP (*Universal Plug and Play*)

2.1.2 UPNP protocol

UPNP potvrđuje da postoji mogućnost komunikacije s uređajima koji ne moraju nužno da budu od istog proizvođača, bez obzira na to da li se povezivanje odvija preko Ethernet mreže ili bežično. Sertifikati nas informišu i o mogućnostima povezivanja, a više o tome možemo takođe da saznamo preko DLNA web stranice. U skladu s DLNA standardima, proizvođač obezbeđuje sprege u obliku aplikacija i komandi za funkcije. Samsung je jedan od članova DLNA alijanse koji se već uveliko služi ovim konceptom. UPNP je otvorena

arhitektura mrežnih protokola i omogućava komunikaciju između mrežnih uređaja pomoću već afirmisanih standarda kao što su protokol kontrole toka (TCP/IP - *Transmission Control Protocol*), protokol za prenos hiper teksta (HTTP - *Transfer Protocol*) i jednostavni pristup objektu protokol (SOAP - *Object Access Protocol*). Uređaji koji poseduju UPNP mogu dinamički da se prijave na lokalnu mrežu, preuzme Internet protocol (IP - *Internet Protocol*) adresu, objavi svoje ime, a može i da sazna imena već prisutnih uređaja. Uređaj može takođe da napusti mrežu bez dodatnih poruka o svom stanju. DLNA protokl definiše klase uređaja (uređaji za reprodukciju, pretraživanje i kontrolu multimedijalnog sadržaja) koji su u podskupu UPNP uređaja.



Slika 2.3 Komunikacija UPNP

2.1.3 Uređaj za deljenje multimedijalnog sadržaja

Uređaj za deljenje multimedijalnog sadržaja DMS (*Digital Media server*) je deo programske podrške za tablične računare i mobilne telefone namenjen za deljenje multimedijalnog sadržaja. Realizovan je pomoću dva rukovaoca, adresar multimedijalnog sadržaja i nadzornika sprege mrežnih priključaka definisan UPNP arhitekturom.

2.1.4 Uređaj za reprodukciju multimedijalnog sadržaja u lokalnoj mreži

Uređaj za reprodukciju multimedijalnog sadržaja na lokalnoj mreži DMP (*Digital media player*) je deo programske podrške za tablične računare i mobilne telefone namenjen za pretragu i reprodukciju multimedijalnog sadržaja deljenog DMS kompatibilnim uređajima. Realizuje tri rukovaoca: pretraživač adresa multimedijalnog sadržaja, dobavljiava multimedijalnih podataka i rukovaoc reprodukcije multimedijalnog sadržaja. Međusobna

saradnja i razmena informacija omogućava celoj programskoj podršci pronalaženje uređaja za deljenje multimedijalnog sadržaja i reprodukciju preko android operativnog sistema (OS – *Operation system*) na tabličnim računarima.

2.1.5 Uređaj za reprodukciju multimedijalnog sadržaja primljenog od DMC

DMR je deo programske podrške namenjen za reprodukciju multimedijalnog sadržaja kontrolisanog od strane udaljenog DLNA DMC kompatibilnog uređaja. Realizuje tri rukovaoca: poslužioca multimedijalnih zahteva, dobavljača multimedijalnih podataka i rukovaoca reprodukcije multimedijalnog sadržaja. Za razliku od DMP dela programske podrške ne poseduje sposobnost pretraživanja multimedijalnog sadržaja, zadržavši podršku reprodukcije istog. Realizovanu reprodukciju pruža svakom DLNA DMC kompatibilnom uređaju uz ograničenja.

2.1.6 Uređaj za kontrolu i reprodukciju multimedijalnog sadržaja

DMC je deo programske podrške realizovan je u ovom radu. Namenjen je za kontrolu i reprodukciju multimedijalnog sadržaja na udaljenom DMR kompatibilnom uređaju. DMC predstavlja programsku nit koja realizuje dva rukovaoca, pretraživač multimedijalnog sadržaja i upravljač reprodukcijom multimedijalnog sadržaja. DMC poseduje mogućnost pretrage multimedijalnog sadržaja ali ne poseduje mogućnost reprodukcije suprotno DMR.



Slika 2.4 Komunikacija DMS, DMC, DMR uređaja

2.2 DLNA ograničenja

DLNA ima svoja ograničenja. Za zvuk, formati koji mogu da budu reprodukovani su (MP3 – *Multimedia Play3*, MPEG4 - *Motion Picture Expert Group4*) i drugi. Da bi DLNA standard mogao da rukuje sa slikom, ona mora da bude u standardom obliku (JPEG - *Joint Photographic Experts Group*). Od video formata podržani standardi kodovanja (MPEG 1, MPEG 2, MPEG 4). Ukoliko multimedijalni sadržaj nije po specifikaciji, DLNA potokola ne može se koristiti. Stručnjaci iz oblasti obrade zvuka, pokretne i nepokretne slike predlažu da se problemi sa previsokom restrikcijom prevaziđu tako što bi se unapredio princip koji važi kod UPNP standarda na personalnim računarima. Istina, tada će veliki broj uređaja teže moći da dobije DLNA sertifikat, jer bi ovaj standard tada podrazumevao daleko veći protok i zahtevao bi bržu obradu informacija.

2.3 Android operativni sistem

Android operativni sistem je trenutno najrasprostranjenij operativni sistem za mobilne telefone, zasnovan je na Linux jezgri i prilagođen je tako da se može koristiti na većini mobilnih uređaja, uključujući pored mobilnih telefona i tablet računare, laptop računare, čitače elektronskih knjiga, pa čak i ručne satove.

Iako je Android Linux distribucija, on po nekim stvarima odstupa od većine standardnih Linux distribucija[4]. Pokretanje samih aplikacija se, kod Androida, ne vrši direktno, već se aplikacije pokreću u okruženju odvojenom od ostatka sistema gde dobijaju samo određeni deo sistemskih resursa, pa tako nemaju pristup delovima sistema koji su im nepotrebni, što donekle poboljšava sigurnost i stabilnost sistema, takođe pri instalaciji aplikacija korisnik dobija listu svih dozvola koje jedna aplikacija zahteva da bi se instalirala, što korisniku daje mogućnost da uoči potencijalno štetne aplikacije i obustavi njihovu instalaciju pre nego što dođe do oštećenja.

Sa tehničke strane Android predstavlja Linux operativni sistem razvijen za ARM i x86 arhitekturu i sastoji se od modifikovanog monolitnog Linux jezgra zaduženog za podršku hardvera i funkcija niskog nivoa, skupa biblioteka zaduženih za dodatne podrške kao što su iscertavanje grafike, podrška za dekodovanje video snimaka, u sklopu biblioteka se nalazi i odvojeni Android start vreme koji sadrži osnovne, bazne, biblioteke i Dalvik virtualna mašina zadužena za pokretanja aplikacija višeg nivoa napisanih u Java programskom jeziku. Na višem nivou od biblioteka su sistemske aplikacije neophodne za upotrebu sistema od strane korisnika i tu se nalaze, menadžer resursa, menadžer instalacionih paketa, kao i aplikacije zadužene za obavljanje osnovnih funkcija vezanih za mobilne telefone ili uređaj na kom je

instaliran Android, na najvišem nivou se nalaze krajnje korisničke aplikacije, odnosno aplikacije koje direktno koristi korisnik. Android poseduje i ugrađenu podršku za multitasking.

Kroz svoju istoriju Android je imao nekoliko verzija od kojih je svaka donosila neku novinu i poboljšanje, tako je npr. verzija 1.0 bila prva zvanično dostupna verzija Android operativnog sistema. Sa pojavom verzija 2.0 i 2.1 pod nazivom Eclair ispravljene su mnoge postojeće greške u samom sistemu i dodate dodatne podrške za rad sa kamerom, kao i poboljšana virtualna tastatura. Verzija 2.2 Froyo je prešla na novo jezgro 2.6.32, ubrzala je rad sa memorijom i poboljšala performanse samog sistema, V2.3 (Gingerbread) takođe prelazi na novo jezgro 2.6.35 i dodatno poboljšava korisničku spregu, takođe donosi sa sobom i podršku za veće ekrane kao i za neke dodatne senzore. Verzija 3.0 poznata i kao Honeycomb bila je zasnovana na jezgru 2.6.36 i bila je prilagođena tablet računarima, dodati su joj sprege elementi kao što su sistem meni i akcioni bar koji su prilagođeni za tablične računare, takođe pojednostavljena je i upotreba multitaskinga, redizajnirana je i virtualna tastatura tako da omogući lakše i brže kucanje a uklonjeni su i neki sigurnosni propusti.

Verzija 4.0 koja nosi naziv Ice Cream Sandwich je trenutno najnovija verzija Android OS. Ova verzija donosi redizajnirani korisničku spregu prilagođen za ekrane visoke rezolucije i dizajniran je tako da omogući čist i jasan prikaz teksta i grafičkih elemenata na ovakvim ekranima

Android 4.0 poseduje veoma razvijen interaktivni sistem obaveštenja, kod većine uređaja ova obaveštenja, vezana najčešće za dolazeće poruke ili trenutni rad neke aplikacije.

Treba naglasiti i da je samo ispisivanje poruka pojednostavljeno, tastatura je redizajnirana i poboljšana tako da je njen odziv znatno bolji, pored toga dodata je i poboljšana mogućnost provere ispravnosti ispisanog teksta, tako da sada tekst koji je pogrešno napisan biva podvučen crvenom linijom a klikom na njega dobija se tri ispravne ponuđene mogućnosti za zamenu pogrešnog teksta ispravnim, kao i mogućnost da uklonite odabranu reč ili da je dodate u rečnik kao ispravnu reč. Dodatne rečnike namenjene za druge jezike moguće je naknadno preuzeti i instalirati.

Android 4.0 daje i mogućnost da tekst poruke diktirate, što je omogućeno posebnim softverom namenjenim za prepoznavanje glasa. Ovaj softver ne samo da vam daje mogućnost skoro neograničeno dugog diktata, već Vam daje mogućnost i da diktirate tekst na bilo kom jeziku, kao i da diktirate interpunkcijske znake kako bi sročili ispravnu rečenicu. Pošto je upotreba interneta nezaobilazni detalj svakodnevice bilo je neophodno dodatno unaprediti mogućnosti komunikacije i deljenja sadržaja na globalnoj mreži.

Veliki broj aplikacija napisanih za Android zahteva vezu sa internetom, kontrola protoka podataka je postala neophodna, pa je ona sada ugrađena u vidu aplikacije koja grafički prikazuje količinu dolazećih i odlazećih podataka, kao i listu aplikacija koje su odgovorne za slanje i primanje podataka.

Ova aplikacija sem što prikazuje podatke vezane za protok podataka u mreži daje Vam i mogućnost da pojedinim aplikacijama ograničite količinu protoka ili u potpunosti ugasite pristup internetu, što može biti veoma korisno u slučaju kada ste ograničeni protokom od strane Vašeg provajdera. Pored povezivanja Android uređaja na internet, omogućeno je i njihovo povezivanje sa drugim uređajima, bez potrebe za internetom, putem bežične mreže ili Bluetooth tehnologije, što daje mogućnost brže i direktne razmene sadržaja između Vašeg Android uređaja i uređaja na koji ste povezani. Veliki broj Android telefona je više od običnog telefona, i zapravo prelazi u male multimedijalne uređaje, sa ovim na umu razvijen je i set aplikacija zaduženih za obradu fotografija i video snimaka i njihovo organizovanje. Kako je pristupačnost sistema veoma bitna stavka, sa verzijom 4.0 Android je dobio poboljšanu podršku za slepe i slabovide osobe, tako da je sada moguće uvećati fontove na nivou sistema i uključiti zvučne signale koji se aktiviraju pri prvom pritisku na neki od sprega elemenata, dok pri drugom pritisku na isti element dolazi do aktiviranja samog elementa.

Iako Android sada predstavlja veoma zreo i pouzdan sistem iza koga stoji ogroman broj zadovoljnih korisnika, on to ne bi postao bez velike zajednice koja neprestano razvija aplikacije namenjene za ovu platformu i na taj način je čini upotrebljivom i živom, takođe ova zajednica sem što predstavlja životnu podršku za ovaj sistem ujedno predstavlja i glavnu pokretačku silu koja ovaj sistem, sa svakom novom verzijom, čini boljim i jednostavnijim za upotrebu.

Pored velike podrške korisnika neizostavna je i podrška velikih firmi koje Android uvode za sastavni deo svojih uređaja i na taj način ga čine dominantnim sistemom za mobilne uređaje.

2.4 Java programski jezik

Java je objektno-orijentisan programski jezik razvijen u timu predvođenim James Goslingom u kompaniji Sun Microsystems početkom 1990.god. Ideja je bila da se stvori programski jezik koji bi bio nezavisan od operativnog sistema, baziran na C++ programskom jeziku, ali sa pojednostavljenom sintaksom, stabilnijim radnim okruženjem i pojednostavljenom kontrolom memorije. Java pripada skupini viših programskih jezika i ima svojstvo prenosivosti. Prenosivost označava mogućnost izvršenja jednog te istog programa na

različitim platformama. Program koji je pisan u višem programskom jeziku ne može biti izvršen direktno na računaru. Takav program se prevodi u mašinski jezik. To obavlja poseban program prevodilac. Nakon što je program jednom preveden, može se izvršavati neograničen broj puta, ali samo na jednom računaru. Da bi se mogao izvršavati na drugim računarima, potrebno ga je ponovno kompajlirati. Umesto korišćenja kompajlera, moguće je koristiti interpreter, koji prevodi naredbu po naredbu, prema potrebi. Interpreter omogućuje izvršavanje programa kompajliranog za jednu vrstu računara na drugom.

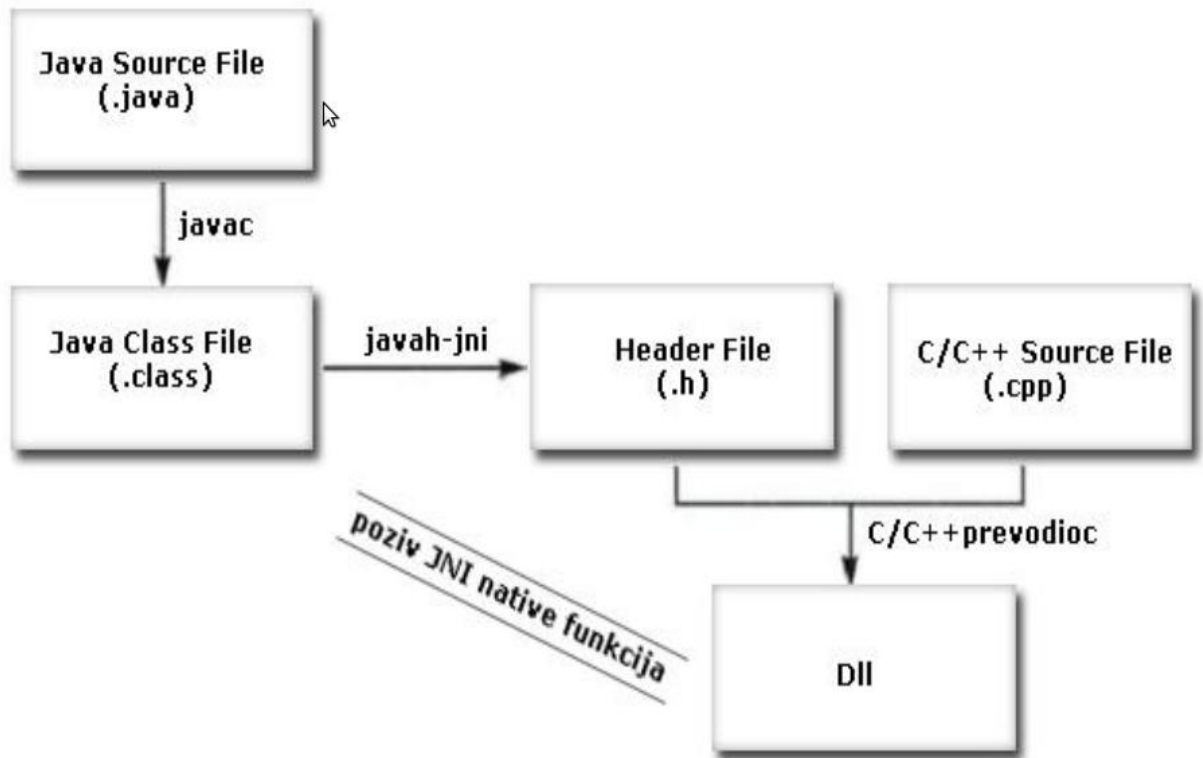
U Java programskom jeziku su objektno-orijentisani principi obavezni. Sve je u Javi objekt, a sav izvorni kod je pisan unutar klasa. U programskom jeziku stanje se opisuje sa promenljivom, a ponašanje se definiše sa metodama. Klasa predstavlja nacrt objekta. Na temelju klasa se proizvode objekti. Klasa može označavati deo programskog koda ili celi programski kod. Po pravilu, svaka klasa je deklarirana unutar datoteke sa istim imenom i ekstenzijom .java. Pravilo je da su imena klasa i datoteka u kojima se klase nalaze ista. Sve klase jedne aplikacije spremaju se u paket sa nazivom domena. Paket predstavlja hijerarhijsko grupisanje razreda, a ime paketa odgovara strukturi direktorija na disku. Drugi pojam u Javi je metoda. Klase koje pokreću program moraju imati Main metodu. Metode su zasebne celine unutar pojedine klase koje izvršavaju određene operacije. Metode su u pojmu objektno-orijentisanog programiranja objekti koji imaju svoje ponašanje, drže podatke i mogu međusobno delovati.

2.4.1 Sprega Java programskog jezika i C koda

JNI je jedini standardni mehanizam koji omogućava interakciju programskog jezika Java sa jezicima čije se naredbe prevode direktno u mašinski kod. Mašinski kod se izvršava direktno u kontrolnoj procesnoj jedinici (CPU – *Control process util*).

2.4.1.1 Kada koristiti JNI

Postoje velik broj aplikacija napisanih u drugim programskim jezicima i namenjenih izvršavanju na jednom računaru. Razvojem interneta, javila se potreba korišćenja tih aplikacija i na internetu. Zbog kompleksnosti većine napisanih aplikacija i zbog velikih troškova njihove reimplementacije, koriste se mehanizmi koji mogu implementirati između web tehnologija i programskih jezika koji se direktno prevode u mašinski kod, od kojih je jedan JNI.



Slika 2.4 Prikaz korišćenja JNI u procesu povezivanja Jave i C/C++ programskog jezika

Poziv JNI funkcija se odvija na Java sloju. JNI funkcije pozivaju deljenje biblioteke u kojem su zapisane prevedene metode korišćene na C/C++ sloju. Na taj način je ostvarena veza između Java i C sloja.

Proces prikazan na slici 2.4 možemo zapisati po koracima:

- Kreiranje deljenih biblioteka na temelju metoda napisanih na C/C++ sloju
- Generiranje JNI klasa
- Implementacija JNI metoda na Java sloju
- Prevođenje programskog koda napisanog u Javi i njegovo izvršavanje

2.4.1.2 Prednosti JNI arhitekture

- Binarna kompatibilnost: JNI programski kod je kompatibilan sa svakom Java virtuelnom mašinom (JVM – *Java Virtuel Masin*) istog operativnog sistema. To znači da se Java aplikacija može prevesti na jednoj JVM a izvoditi na bilo kojoj drugoj JVM koja podržava JNI.
- Ograničenje pristupa: JNI ograničava pristup JVM-u iz programskih jezika koji se direktno prevode u mašinski kod. Napredni mehanizam regulacije zagušenja toka programa

2.4.1.3 Nedostatci korištenja JNI aritekture

Portabilnost korištenje JNI u Javi ograničava korištenje Java aplikacije na određeni operativni sistem. Za korištenje Java aplikacije na windows operativnom sistemu potreban je drugačiji format zapisa.

- Kompleksnost – Aplikacija koja koristi JNI je kompleksna iz razloga što zahteva od programera dobro poznavanje Java programskog jezika i C/C++ jezika.
- Sigurnost – Java aplikacije koje koriste funkcije koje se direktno prevode u mašinski kod nisu toliko sigurne kao čiste Java aplikacije.

2.5 Eclipse razvojno okruženje

Programski kod rada je delom pisan u Eclips-u razvojnom okruženju. Eclipse je višejezično, softversko razvojno okruženje koje se sastoji od integriranog razvojnog okruženja i proširivog sistema. Eclipse je razvijen od strane (OTI - *Object Technolog International*) kompanije kao Java softver otvorenog koda. Može se koristiti za razvoj aplikacija u Javi i drugim programskim jezicima putem različitih softverskih dodataka. Moguće je Eclipse nadograditi sa dodacima za programske jezike C/C++. Razvojno okruženje Eclipse-a je vrlo prilagodljivo okruženje. Za izradu rada razvojno okruženje Eclipse-a je prošireno na programski dodatak za razvoj Android aplikacija, omogućava kreiranje, uređivanje i uklanjanje grešaka kod Android aplikacij. Nadogradnja sadrži paket softverskih razvojnih alata u kojem su uključene sledeće komponente:

- Emulator koji služi za izvršavanje programa na računaru.
- Pronalaženje i uklanjanje grešaka, kontrolu i nadzor nad aplikacijama (DDMS - *Dalvik Debug Monitor Server*).
- Stvaranje i distribuciju Androidovog programskog paketa u .apk formatu.

U Android je uključen emulator pokretnog uređaja koji omogućava razvoj i ispitivanje aplikacija. Jedino ograničenje je nemogućnost uspostave dolaznih i odlaznih poziva, uključuje celi Androidov sistemski stek, skup unapred instaliranih aplikacija kojima se može pristupiti korišćenjem vlastitih aplikacija (korisničko grafičko okruženje), prilagodljivo je dodeljivanju vrednosti, mnoštvo naredbi i mogućnosti za kontrolu ponašanja ispitivanog okruženja.

3. Koncept rešenja

Integracija DMC u Android operativni sistem zahteva postojanje više slojeva programske podrške. Svaki od slojeva predstavlja prilagođen servis za druge slojeve

Na najvišem nivou nalaze se Android aplikacije koje obavljaju grafičko prikazivanje realizaciju tri rukovaoca: pretraživač adresa multimedijalnog sadržaja, dobavlivača multimedijalnih podataka i rukovaoc reprodukcije multimedijalnog sadržaja. Različite aplikacije se od sada mogu pisati na identičan način kao i ostale Android aplikacije korišćenjem standardnog Android okruženja.

Sloj JNI omogućava razmenu podataka između Android aplikacija i programske podrške DLNA protokola. Postojanjem ovog sloja programer više ne mora da bude upoznat sa realizacijom programske podrške DLNA sertifikata. Time se omogućava potpuno nezavisna izrada ovih slojeva.

Najniži sloj sadrži programsku podršku za tablične računare, specifičan za svakog proizvođača, koji oni mogu da prilagode tako da najviše odgovara njihovim potrebama. U najnižem sloju se nalazi programska podrška koja je pisana u C/C++ jeziku. Poziv JNI funkcija se odvija na Java sloju. JNI funkcije pozivaju deljene biblioteke u kojem su zapisane prevedene metode korištene na C/ C++ sloju. Na taj način je ostvarena veza između Java sloja i C sloja. Da bi dobili deljenje biblioteke moramo napisati android makefile za svaki deo programske podrške.

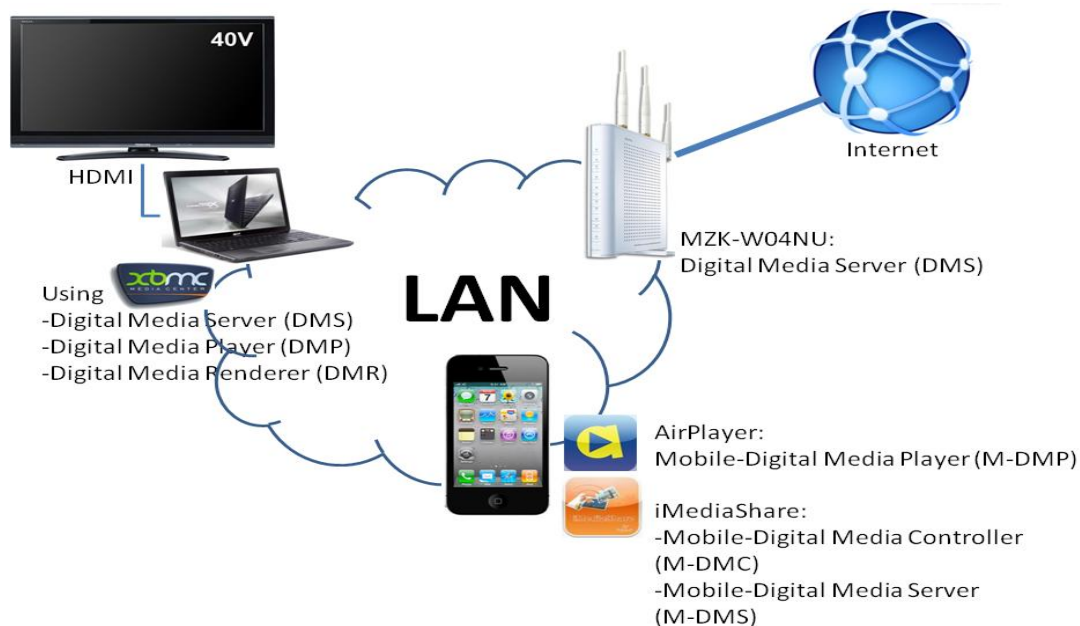
Svaki od slojeva komunicira sa susednim, a zajedno čine kompsktnu i fncionalnu celinu.

3.1 Komponente sistema

Programska podrška za deljenje, kontrolu i reprodukciju multimedijalnog sadržaja zasnovana na DLNA protokolu, i deli se na četiri zasebne celine: DMS (*Digital Media*

Server), DMP (Digital Media Player), DMR (Digital Media Renderer) i DMC (Digital Media Controller).

Svaka celina je realizovana kao posebna programska nit u C programskom jeziku. Svi delovi programske podrške koriste libupnp biblioteku.



Slika 3.1 Komponente DLNA sistema

3.1.1 Struktura Android aplikacije

Android aplikacije su pisane u Java programskom jeziku. Android SDK kompajlira programski kod u izvršni kod. Izvršni kod je arhivska datoteka sa ekstenzijom .apk. Takva datoteka je aplikacija koja se može instalirati na Android-ovom mobilnom uređaju. Pravilo je da se pri instalaciji svakoj aplikaciji dodjeljuje jedinstveni Linux identifikator. Prema tom identifikatoru aplikaciji se daje pristup servisima, sensorima i drugim delovima uređaja i operativnog sistema. Android platforma primjenjuje princip najmanje privilegije, aplikacija ima samo pristup onim komponentama koje su joj potrebne za rad. Svaka se aplikacija pokreće u DVM-u i deluje izolovano od drugih aplikacija.

3.1.1.1 Komponente aplikacije

Aplikacijske komponente su temeljni sastavni delovi Android aplikacije. Svaka komponenta je samostalna celina, igra specifičnu ulogu u radu aplikacije i ima svoj način interakcije sistema sa aplikacijom. Postoje različiti tipovi aplikacijskih komponenata, gde svaki ima svoj ciklus trajanja koji određuje proces pokretanja i prekida aplikacije. Aktivnosti

komponenta aktivnost predstavlja glavnu ulaznu tačku korisnika u program. Aktivnost je najčešće korištena komponenta. Zadatak aktivnosti je prikaz korisničkog sučelja programa i omogućavanje interakcije korisnika sa programom (npr. slanje e-mail-a, i dr.). Aplikacija se obično sastoji od više aktivnosti koje su međusobno povezane. Servisi predstavljaju aplikacijsku komponentu koja izvršava zadatke u pozadini programa tokom daljeg vremenskog perioda. Servisi nemaju grafičku predstavu. Ostale aplikacijske komponente mogu sa servisima uzajamno dejstvovati i izvoditi međuprocesnu komunikaciju.

3.1.2 Meta-jezik za označavanje tekstualnih dokumenata

Pre pokretanja aplikacije, sistem mora znati od kojih se komponenti aplikacija sastoji. Registracija aplikacijskih komponentata se radi u Android-ovom dokumentu (XML - *Extensive Markup Language*). XML datoteka zove `AndroidManifest.xml` za sve aplikacije. Funkcije ovog dokumenta su identifikiranje aplikacijskog pristupa sistemu uređaja, definisanje minimalne verzije operativnog sistema na kojem će aplikacija raditi, deklarisanje delova uređaja ili aplikacija koje program koristi, deklarisanje eksternih biblioteka klasa s kojima je program povezan.

3.1.3 Aplikacijski resursi

Za grafički prikaz aplikacije koristi se direktorijumski resursi u kojem mogu biti spremljene slike, ikone, zvuk datoteke, boje, fontovi i ostalo što ima veze sa vizualnom prezentacijom. Korišćenje resursa olakšava ažuriranje zvuk-grafike bez modifikovanje koda. Za svaki resurs koji je uključen u Android projekt, Android definiše jedinstveni identifikator, koji se koristi kao poveziivač aplikacijskog koda i drugih izvora definisanih u XML-u. Svi identifikatori se prikazuju u automatski generisanoj datoteci `R.java`. Datoteka `R.java` služi kao indeks svih resursa korišćenih u projektu. Na primer, aplikacija sadrži datoteku sa slikama. Android generise identifikator resursa pod nazivom `R.drawable.logo` koji se koristi u programskom kodu za prikazivanje slike korisniku.

4. Programsko rešenje

U ovom poglavlju je dat detaljan opis građenja izvornog C koda, prikaz sprege Java i C koda, kao i način korišćenja i funkcionisanja Android aplikacije.

4.1 Građenje izvornog koda

Za građenje izvornog C koda za rad na Android operativnom sistemu potrebno je po konvenciji kreirati datoteku sa ekstenzijom “.mk”.

Android.mk je poput karte koja daje instrukcije kompajleru kako da prevede izvorni kod u binarni oblik. Sadrži podatke poput optimizacija koje treba koristiti, gde treba instalirati finalni proizvod, mod fajla koji treba instalirati. Povremeno se ovi fajlovi modifikuju ručno, ali uglavnom uz pomoć skripta za konfiguraciju. Da bi DLNA struktura funkcionisala potrebno je izgraditi Android.mk za sve module, a zatim ih objediniti jednim Android.mk. Android podržava dve vrste biblioteka statičke i deljene. Posebna pažnja posvećena je prilagođavanju programskog koda fizičkim platformama (ARM - *Advanced RISC Machine*).

4.1.1 Kreiranje sablona za pisanje android.mk datoteka

Svaki Android.mk sadrži labele:

- LOCAL_PATH:= \$(call my-dir) - pozivom funkcije my-dir vratiće nam path od Android.mk
- include \$(CLEAR_VARS) - briše sve promenljive započete sa LOCAL
- LOCAL_SRC_FILES:= filelib.c - Izvorna lista fajlova za prevođenje
- LOCAL_MODULE:= foo - Ime izvršne binarne biblioteke
- include \$(BUILD_EXECUTABLE) - prevođenje izvršne binarne biblioteke
- LOCAL_C_INCLUDES:= \

- `$(LOCAL_PATH)/../fs_lib/` - naredba za uključivanje svih `*h` hedere iz projekta
- `include $(BUILD_STATIC_LIBRARY)` – prevođenje statičkih biblioteka
- `include $(BUILD_SHARED_LIBRARY)` – prevođenje deljenih biblioteka
- `LOCAL_WHOLE_STATIC_LIBRARIES` – `libupnp`, `libxml`, `libsqlite`, `libmedia` uključujemo potrebne statičke biblioteke.

Android.mk datoteka sadrži naredbe za izgradnju i povezivanje odvojenih modula

```
LOCAL_PATH:= $(call my-dir)
LOCAL_CFLAGS+=-DANDROID_LIBDLNA_BUILD
include $(CLEAR_VARS)
LOCAL_SRC_FILES:= \
    /android_dtv_media_explorer_MediaExplorerControlNative.cpp \
    /filelib.c \
    /file-local.c
LOCAL_C_INCLUDES:= \
$(LOCAL_PATH)/../fs_lib/
$(LOCAL_PATH)/../libupnp/upnp/inc/
$(LOCAL_PATH)/../libupnp/ixml/inc/
LOCAL_MODULE_TAGS:=optional
LOCAL_PRELINK_MODULE:=false
LOCAL_MODULE := libfs
include $(BUILD_SHARED_LIBRARY)
```

Slika 4.1 Primer Android.mk modula fs-lib

4.2 JNI sloj

JNI sloj programske podrške se sastoji od deklaracija metoda koje su prethodno opisane u Java paketima i njihove realizacije u C kodu. Deklaracije metoda se nalaze u okviru Java paketa, dok se sama realizacija nalazi u zasebnim C modulima koji su Javi dostupni u obliku deljenje biblioteke. Ovde je realizovano dvosmerno prozivanje metoda:

- smer od Java aplikacije ka programskoj podršci DLNA;
- smer od programske podrške DLNA protokola ka Java Android aplikaciji.

Sve native metode moraju da budu implementirane kao funkcije u deljenoj biblioteci (.so datoteka). Deljenje biblioteka se učitavaju metodom za učitavanje biblioteka (`System.loadLibrary`) u statičkom bloku aplikacije. Da bi virtualna mašina mogla da prepozna JNI funkciju u deljenoj biblioteci, potrebno je da ima odgovarajući potpis u prototipu, odgovarajući naziv argumente i vrednosti. Potpis se generiše pozivom javah alata koji je standardni deo Java SDK.

4.2.1 JNI realizacija

U Android projektu napravimo dodatni direktorijum JNI. U njemu će biti implementacija svih JNI metoda.

JNI direktorijum sadrži:

- Mašinski generisanu `android_media_explorer_MediaExplorerControlNative.h` datoteku sa prototipovima.
- `Android_media_explorer_MediaExplorerControlNative.cp` - datoteka sa implementacijom JNI funkcija.

Sve C/C++ datoteke se smeštaju u JNI direktorijum kreiran u projektu radi lakše preglednosti i prenosivosti koda.

Klasa `android_media_explorer_MediaExplorerControlNative.cpp` sadrži sledeće funkcije:

- `initNative()` – inicijalizacija JNI metoda
- `deinitNative()` – deinicijalizacija JNI metoda
- `registerEntry(int fileHandlingType)` -Registruje file handler pozivom native metoda
- `unregisterEntry(int fileHandlingType)`
- `setCurDir(String newCurrentDir)` – Setuje trenutni direktorijume
- `getCurDir()`
- `getNext(String URI)` – implementuje sredstva iz trenutnog direktorijuma sve dok ne bude prazan poziva se pole `getFirst`
- `getFirst(String URI)` - implementuje sredstva iz trenutnog direktorijuma sve dok ne bude prazan, prvi se poziva
- `getFileProperty(String URI, int ID)` – daje iformacije o fajlovima
- `getNextDev(String URI)` - implementira sredsta iz odabranog modula za reprodukciju
- `getFirstDev(String URI)`
- `openPlayer(String name)` - hendler za otvaranje modula za reprodukciju

4.2.2 Modul libupnp

Sastoji se od nekoliko podmodula od kojih su najvazniji `ixml` i `theardutil.Libupnp` kao i njegove module izgrađuju se statički da bi se te biblioteke mogle povezati.

4.2.3 Modul libdlna_rel

Glavni modul u kojem se nalaze sve funkcije neophodne za rad DMC. Njega smo pomoću android.mk izgradili da bude deljena biblioteka. Deljene biblioteke su u “*.so” obliku, odnosno njih može da poziva svaka aplikacija kojoj su potrebne funkcije iz te biblioteke.

4.2.4 Modul mmlib

Omogućava i podržava multimediju.

```
LOCAL_PATH:= $(call my-dir)
include $(CLEAR_VARS)
LOCAL_CFLAGS := -DTMP_SETTINGS_PATH="/tmp/dms/"
LOCAL_SRC_FILES:= \
    /src/medialib.c \
    /src/pc/media-mplayer.c \
    /src/pc/media-img-feh.c \
    /src/mediainfo-parser.c
LOCAL_C_INCLUDES:= \
    $(LOCAL_PATH)/src/ \
    $(LOCAL_PATH)/src/pc
LOCAL_MODULE_TAGS:=optional
LOCAL_PRELINK_MODULE:=false
LOCAL_MODULE := libmedia
include $(BUILD_STATIC_LIBRARY)
```

Slika 4.2 Primer Android.mk modula mmlib

4.2.5 Paket android.os

Ovaj paket sadrži sve module koji su neophodni za normalno funkcionisanje jedne Android aplikacije. Uključuje sledeće module u sebe:

- BrowseRenderers – pretraživanje uređaja za reprodukciju u mreži.
- FsLibActivity – početna aktivnost koja se sastoji od dva dugmeta (Browse Rendere i Servers) koji pokreću aktivnost.
- BrowseServers - realizuje pretraživanje uređaja za deljenje multimedijalnog sadržaja kao i pretraživanje njihovih sadržaja u mreži.

5. Rezultati

U ovom poglavlju biće ispitana funkcionalnost DMC Android aplikacije na tabličnom računaru.

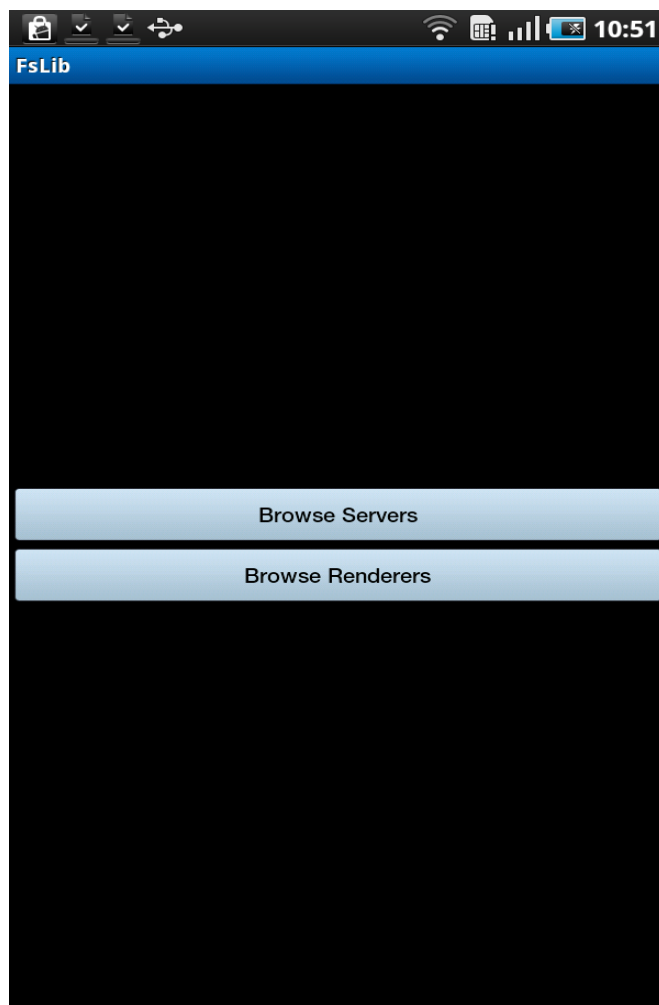
5.1 Ispitivanje za tablet Samsung Galaksi

Glavno ispitivanje i provera funkcionalnosti rada realizovana je na tabličnom računaru proizvođača Samsung modela Galaksi sa verzijom Android operativnog sistema 2.3 Gingerbread.

Grupe funkcionalnosti koje su ispitane u ovim slučajevima su:

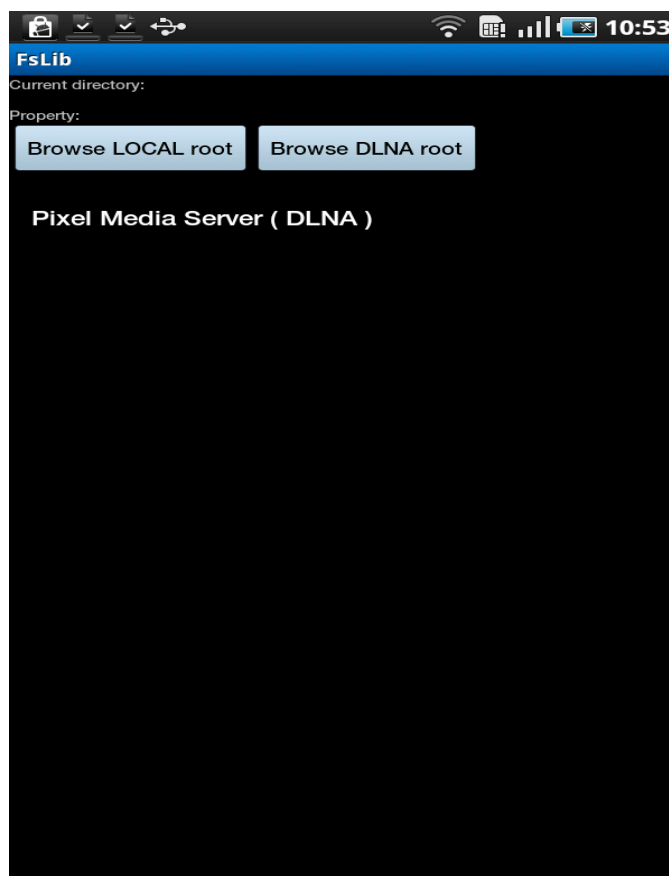
- odziv aplikacije na određene pobude;
- pretraživanje sadržaja za deljenje i reprodukciju multimedijalnog sadržaja;
- prikaz informacija o odabranom fajlu pozivanjem povratne metode;
- odabir više uređaja za deljenje multimedijalnog sadržaja
- prikaz informacija o načinu rukovanja aplikacijom

Aplikacija se od nekoliko aktivnosti. Glavna aktivnost sadrži dva polja čijim odabirom se pokreće aktivnost pretraživanja uređaja za deljenje multimedijalnog sadržaja, kao i za reprodukciju istog. Odabirom pretraživača uređaja za deljenje sadržaja omogućuje se odabir da li da sadržaj traži u lokalno ili na dostupnim uređajima za deljenje sadržaja.



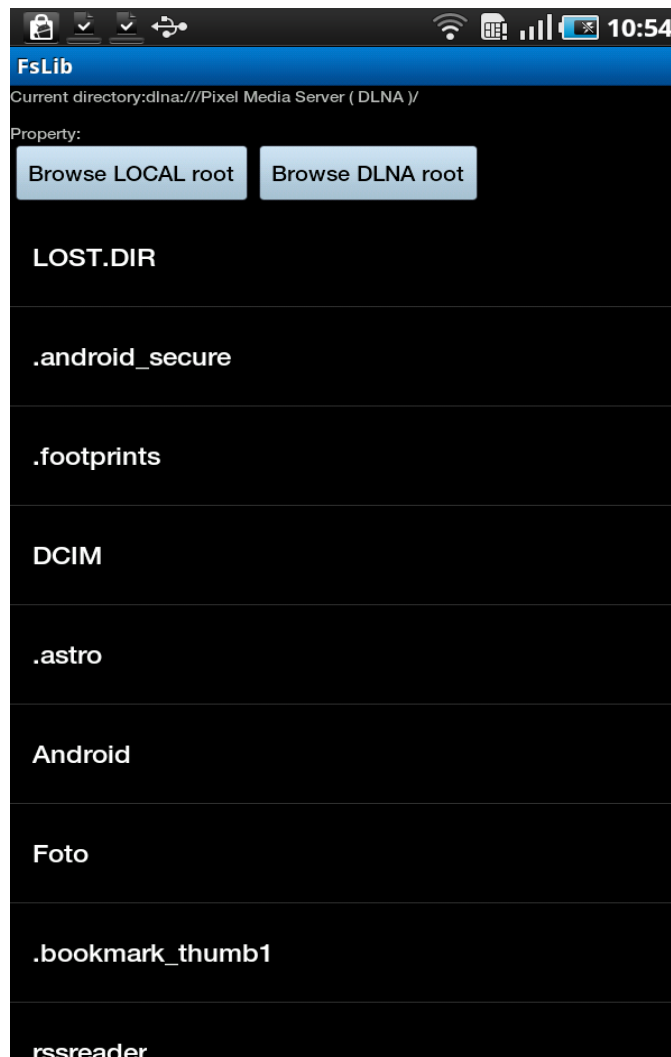
Slika 5.1 Izgled aplikacije nakon pokretanja

Pretražuju se svi uređaji u bežičnoj mreži kao i njihovog sadržaja (zvuk, pokretna i nepokretna slika). Odabrani sadržaj koji može biti zvuk, pokretna ili nepokretna slika se prosleđuje odabranom kompatibilnom uređaju za reprodukciju.



Slika 5.2 Izgled aplikacije nakon odabira pretraživanja uređaja

Kako bi aplikacija radila neophodno je da i odabrani uređaj za reprodukciju sadržaja bude u istoj mreži. Omogućeno je i biranje različitih uređaja za reprodukciju ukoliko ih ima više.



Slika 5.3 Prikaz dostupnog sadržaja

Da bi aplikacija pronašla u mreži sve uređaje neophodno je u Java okruženju setovati internet prava pristupa. Također svi uređaji moraju biti u istoj mreži. Uređaj mora podržavati verziju Android operativnog sistema. Na primer, ako je aplikacija izgrađena na 2.3 platformi Android operativnog sistema može raditi samo na takvoj verziji. Za sve promene eventualne izmene neophodno je adekvatno ponovo izgraditi aplikaciju.

6. Zaključak

U ovom radu realizovan je aplikacija za kontrolu i reprodukciju multimedijalnog sadržaja (zvuka, pokretne i nepokretne slike) u DLNA mreži. Rad je realizovan za tablične računare na bazi Android operativnog sistema verzije 2.3 Gingerbread.

Izrada ovakog rada zahtevala je detaljnu analizu tehnologija potrebnih za povezivanje C/C++ tehnologije i java programskog jezika, jer je programska podrška pisana u C programskom jeziku, a grafičko okruženje u java programskom jeziku. Najveća pažnja posvećena je na zahteve i pravila korišćenja tehnologija za prebacivanje funkcija C programskog jezika i Java. Posebna pažnja posvećena je prilagođavanju programske podrške Android operativnog sistema.

Aplikacija DMC je ispitana na tabličnom računaru proizvođača Samsung i utvrđena je njena funkcionalnost.

Dalji razvoj aplikacije zasniva se na prenosivosti rešenja i mogućnosti rada na različitim platformama. Poboľšanje grafičkog okruženja aplikacije i proširivanju formata za reprodukciju. Takođe i na proširivanje programske podrške modulima za zaštitu multimedijalnog sadržaja.

7. Literatura

- [1].....Nikola Kuzmanović, Member, IEEE, Tomislav Maruna, Member, IEEE, Darko
.....Kličković, Željko Lukač, Member, IEEE Jedno rešenje realizacije programske
.....podrške za deljenje kontrolu i reprodukciju multimedijalnog sadržaja pomoću
.....DLNA protocol steaka ,Telfor 2010
- [2]..... Sajt podrške DLNA, Living Network Alliance, www.dlna.org, učitano 20. 09.
.....2012
- [3].....Vladimir Kovačević, Miroslav Popović: Sistemska programska podrška realnom
..... vremenu, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka, 2002.
- [4]..... Sajt Android podrške za razvoj, www.developer.android.com, učitano
.....18.09.2012.
- [5]....M.Vidakovic, N.Teslic, T.Maruna, and V.Mihic: *Android4TV: a proposition for*
.....*integration of DTV in Android devices*, IEEE 30th International Conference on
.....Consumer Electronics (ICCE), Las Vegas, January 2012, pp. 441-442