

Mart 2014.



LIBRE!

Časopis o slobodnom softveru

broj

23



9. mart
Redovna *openSUSE*
konferencija održaće se
u Dubrovniku.



31. mart
25. aprila 2014. i Novom
Mestu održaće se *Open*
Way.



Creative Commons Autorstvo-Nekomercijalno-Deliti pod istim uslovima.



Dan posle

Sama činjenica da sada čitate reči LiBRE! urednika, znači da smo preživeli najavljenju katastrofu. To takođe znači da je ponovo „Reč urednika” u množini a ne u jednini kao u prošlom broju.

Šok terapija koja se ogledala u najavi katastrofe i gašenja LiBRE! projekta, imala je za cilj da mobilise sve unutrašnje i spoljašnje snage da bi projekat opstao. Nije bio u pitanju prosti trik kao iz bajke o čobaninu koji je zavitlavao seljane vičući „Vuk! Vuk!”. Časopisu su pomoć i reorganizacija zaista bili potrebni. Zahvaljujemo se svima koji su pravilno razumeli naš signal za uzbunu i pritekli u pomoć.

Da je vrag odneo šalu, srećom, shvatili su i članovi redakcije. Nismo morali da se „prodajemo” i da dozvolimo da neko drugi upravlja časopisom. Drastično smo podmladili urednički tim i to sada daje rezultate. Moramo javno da se pohvalimo da je cela redakcija pravilno shvatila mobilizaciju i doprinela da novi broj izađe u kraćem roku (zbog kašnjenja prethodnog broja) i nikad lakše.

Žao nam je, ako smo nekoga razočarali našom šok terapijom. Neki su

nas već prežalili, ne verujući u žilavost projekta. Bilo je i predloga da ne izlazimo redovno, mesečno. Do nekog olakšanja posla, barem privremeno, moralo je doći. Pošto smo se morali nečega odreći, odlučili smo da to bude *ePUB* izdanje. To izdanje još uvek nije toliko popularno da bi mnogo nedostajalo. Ovo izdanje ćemo vratiti kad definitivno budemo rešili problem grafike i kad budemo rešili problem direktnog preuzimanja ovog izdanja u mobilne telefone.

Periodiku izlaženja, jednom mesečno, ne želimo da menjamo. Ovaj broj dokazuje da je moguće opušteno sa malom mobilizacijom ispoštovati ovu periodiku. Produženje roka na dvomesečnik ili još gore na izlaženje po principu objave „kad bude gotovo” nije dobro. Časopis mora imati disciplinu. Opušten pristup ne bi doneo ništa dobro u budućnosti ovog projekta. Kašnjenja časopisa bila bi sve veća i na kraju bi se prosto ugasio.

Iako je reorganizacija uspela i sad se lakše diše, nismo rešili sve probleme. Ni posle dve godine nismo našli rešenje ni zamenika za jedinog grafičara u projektu. Koliko je opasno da projekat ima samo jednog



urednika, toliko je još veća opasnost, imati samo jednog grafičara. Bez grafičara koji će složiti PDF, džaba nam svi urednici, autori i lektori. Ovaj jedan grafičar ne sme ni da se razboli ni da ide na godišnji odmor, a da ne govorimo da nas napusti. Zato sad apelujemo na sve koji se bar malo razumeju u grafiku da nam se jave i pomognu. Učvršćivanje te pozicije je sad prioritet projekta. Volonterima u grafici možemo ponuditi, za sada, obuku, praksu i potvrdu o volontiranju što može da im pomogne u traženju posla.

Pored grafičara, koji su nam trenutno prioritet, uvek su dobrodošli novi autori, ali i svi oni koji prate novosti o FLOSS-u, koji bi mogli da budu dopisnici i javljaju šta se novo dešava i šta je aktuelno u ovoj oblasti u svetu i njihovom kraju. Najbolji način da stupite u kontakt s nama, jeste preko naše već poznate internet adrese libre@lugons.org.

Do čitanja.

LiBRE! Tim

Moć slobodnog softvera



Broj: 23

Periodika izlaženja: mesečnik

Glavni i odgovorni urednik:
Nikola Hardi

Izvršni urednik:
Stefan Nožinić

Lektura:
Jelena Munčan
Katarina Čehić
Maja Panajotović
Aleksandra Ristović
Aleksandar Božinović
Aleksandar Stanisavljević

Redakcija:

Goran Mekić	Sandrina Dimitrijević
Džoni Promis	Aleksandar Todorović
Željko Šarić	Milovan Krivokapić
Danilo Đokić	Dalibor Bogdanović
Darko Stantić	Aleksandar Brković
Petar Simović	Gavrilo Prodanović
Zlatan Vasović	Mihajlo Bogdanović
Ivan Bulatović	Vladimir Cicović
Željko Popivoda	Aleksandar Vesić
Bojan Bogdanović	Veljko Simić

Saradnici:

Nikola Nenadić	Momčilo Medić
Nenad Mijatović	Velimir Baksa
Tamara Đorđević	Stefan Stojanović

Grafička obrada:
Dejan Maglov

Dizajn:
Mladen Ščekić

Zoran Lojpur

Kontakt:
IRC: #floss-magazin na [irc.freenode.net](irc://irc.freenode.net)
E-pošta: libre@lugons.org

<http://libre.lugons.org>



LiBRE! vesti str. 6

Vesti



Puls slobode str. 8

OpenSUSE ili *osCE14*,
konferencija u Dubrovniku str. 8



Ugovor Republike Srbije
sa *Microsoftom* (5.deo) str. 10

aquaticy.com str. 14

Predstavljamo str. 17

Serbian 2014 (1. deo) str. 17



Kako da...? str. 23

Uvod u programski
jezik *C* (1. deo) str. 23

Learn C Programming

Oslobađanje str. 26

Linux unatraške str. 26



Slobodni profesionalac str. 31

OpenCV str. 31





Internet, mreže
komunikacije

str. 35

Tor mreža:
Online anonimnost i
sigurnost (2.deo)

str. 35



Mobilni kutak

str. 38

Android ispod haube

str. 38



Network Spoofer

str. 45



LIBRE! prijatelji



REGIONALNI
LINUX PORTAL

linuxzasve.com





Redovna godišnja openSUSE konferencija

9. mart 2014.



Ovogodišnja *openSUSE* konferencija nazvana „Snaga promene” (*The Strength to Change*) koja će se održati od 24. do 28. aprila u Dubrovniku, okupiće članove *openSUSE* zajednice kao i ostale članove *F(L)OSS* zajednice.

Koristan link: <http://j.mp/1mvlmSH>

Puppy Linux 5.7 „Slacko”

12. mart 2014.



Nova verzija *Slacko Puppy Linuxa* je objavljena. *Slacko Puppy* je izgrađen od *Slackware 14.0* binarnih *TXZ* paketa, otuda ima binarnu kompatibilnost sa *Slackware* i pristup *Slackware*, *Salix* i *Slacky* riznicama.

Koristan link: <http://j.mp/1pyObuB>

Blender fondacija radi na novom animiranom filmu

15. mart 2014.



Blender fondacija je 9. marta pokrenula kampanju u trajanju od četrdeset dana sa ciljem prikupljanja 500 000 € kojima bi se finansirao novi *open source* animirani film.

Koristan link: <http://j.mp/1lmzXzt>

PC-BSD 10.0.1

16. mart 2014.



Prvo *PC-BSD 10.0* kvartalno ažuriranje je pred nama, sada je dostupan 10.0.1. Ovaj *update* donosi ispravke grešaka, novije pakete *KDE 4.12.2*, *Cinnamon 2.0* i ažuriran *GRUB*.

Koristan link: <http://j.mp/1gYjNog>

Tiny Core Linux 5.2.1 „piCore”

17. mart 2014.



Objavljen je *piCore 5.2.1 Tiny Core Linuxa* za *Raspberry Pi single-board* kompjuter. *Tiny Core Linux* nije tradicionalna distribucija, već je alat koji omogućava da napravite sopstveni prilagođeni sistem.

Koristan link: <http://j.mp/1lmA2TT>

Evropski parlament prelazi na ODF i Ubuntu-based Distro

19. mart 2014.



Nakon Britanije koja se obavezala da će preći na *Open Document Format*, nekoliko članica Evropskog parlamenta pokreće istu inicijativu u svojim administracijama. Prelazak na *ODF* je prvi korak u prelasku EU administracije na *open source*.

Koristan link: <http://j.mp/1dOzLkR>

Tails 0.23

20. mart 2014.



Tails je Debian-based „živa“ CD distribucija, prekonfigurisana za anonimno pretraživanje web-a. Nova verzija 0.23 je objavljena. Ovo izdanje

rešava brojna pitanja sigurnosti i preporučuje se da svi koji su do sada koristili Tails, odmah ga nadgrade na novu verziju.

Koristan link: <http://j.mp/1iB0yFD>

Udruga Osijek Software City organizuje The Geek Gathering 2014.

20. mart 2014.



Udruga Osijek Software City organizuje drugi po redu The Geek Gathering, trodnevnu godišnju konferenciju koja će se održati od 8. do 11. maja 2014. godine u hotelu „Osijek“.

Brojna poznata imena već su potvrdila dolazak.

Koristan link: <http://j.mp/1fk0WnY>

ATM operateri žele Linux umesto XP-a

20. mart 2014.



Neke kompanije koje nude finansijske usluge, žele da svoje ATM uređaje prebace sa Windowsa na Linux u pokušaju da imaju bolju kontrolu nad hardverom i standardne cikluse updatea softvera. Potez koji ih gura u ovom pravcu,

očigledno je odluka Microsofta da okonča podršku za Windows XP 8. aprila ove godine.

Koristan link: <http://j.mp/NHDQA2>

Open Way

31. mart 2014.



U petak, 25. aprila 2014. godine u Novom Mestu (Slovenija) održaće se Open Way - konferencija posvećena open source

rešenjima.

Koristan link: <http://j.mp/1hvb3ur>

LiBRE! prijtelji

LUTHERUS
Et in Arcadia ego!


Think about this


 info i tutorijali na srpskom
lubunturs.wordpress.com




OpenSUSE ili osCE14, konferencija u Dubrovniku



Autor: portal „Linux Za Sve”

Konferencija će se održavati u razdoblju od 24. do 28. travnja (aprila) u prostorijama Dubrovačkog Sveučilišta. Prisustvovanje konferenciji je potpuno besplatno. Potrebno je samo registrirati se na službenim stranicama konferencije (https://conference.opensuse.org/osem/accounts/sign_in). Registracije su otvorene do 24. travnja (aprila) uključujući i sam dan, no preporuča se da ipak napravite svoju prijavu ranije da se zna koliko će ljudi otprilike doći.



Na konferenciji možete prisustvovati zanimljivim predavanjima i radionicama, te sudjelovati u razmjeni znanja i iskustva s okupljenim sudionicima.

Konferencija se održava jednom godišnje, a mjesto održavanja se svake godine mijenja. Tako su se ove godine, na prijedlog *HrOpena*, *HULK*-a i Dubrovačkog Sveučilišta, u *openSUSE*-u odlučili posjetiti Dubrovnik.

OpenSUSE konferencija tradicionalno ima temu, a ove godine to je „Sposobnost za promijeniti se”. Promjena je konstanta slobodnog softvera, a sa rastom relevantnosti mobilnih uređaja i operacijskih sustava poput *Androida* i *Chromebookova*, treba se prilagoditi na razini čitavog *openSUSE* projekta.

Strategije o tome kako to najbolje izvesti, diskutirala se i diskutira se putem raznih digitalnih medija, ali i uživo. Zbog toga ova konferencija dolazi u četiri paralelna segmenta i namijenjena je različitim interesnim skupinama s nadom da će se naći zajednički jezik i rješenje za promjene u okolišu. Promjena nije nikada lagana, ali je uvijek važna.



Krajnji korisnici

Ovaj segment pruža priliku naprednim korisnicima bilo kojih aplikacija da dijele svoje znanje i trikove kako bi uspjeli postići najviše određenim alatom koji koriste. Ukoliko (ne) znate za malo poznate ali korisne funkcije ili programe, ovdje ih možete podijeliti sa drugima ili saznati nešto novo. Teme uključuju, ali nisu ograničene na aplikacije, *desktop* okruženja, multimediju i igre.



nego prije, najbolje će biti da u njoj i sudjelujete.

Poslovni korisnici

Ovaj segment je namijenjen ljudima koji koriste *openSUSE* i/ili *FOSS* u poslovnim okruženjima, da opišu sa čime se sve susreću, te riješenja i savjete za bolji rad. Teme uključuju, ali nisu ograničene na probleme i rješenja vezana za komunikaciju sa državnim agencijama, regulatornim agencijama, drugim poslovnim subjektima, obukom osoblja te promjenom tehnologije kao takve.

Tehnologija i razvoj

U ovom dijelu će se koncentrirati na sistemsku tehnologiju i razvoj distribucije, uključujući paketni sustav (*rpm*), razvoj, testiranje, *debugiranje*, praksu i razvojne metode, infrastrukturu, izdavanje i nadzor. Saznat ćete i bolje razumijeti kako se koriste alati u razvoju *SUSE* distribucije ali i drugih projekata nevezanih direktno za distribuciju.

Zajednica i projekat

Ovaj segment se odnosi i fokusira se na projekte i aktivnosti zajednice. Teme uključuju, ali nisu ograničene na upravljanje samim projektom, marketing i grafički dizajn. U mnogo slučajeva ovaj dio naglašava zajedništvo zajednice i kooperaciju, te jedinstvene probleme sa kojima sa *openSUSE* zajednica suočava. Ukoliko imate ideje kako pomoći zajednici da bude jača

Što još reći osim: skupite ekipu, nađite si neki jeftiniji smještaj, uživajte u predsezoni u ovom prekrasnom gradu, družite se s Linuksašima iz cijelog svijeta i dođite uživo vidjeti legende *FLOSS* i *Linux* svijeta! Jamčimo vam da ćete se odlično zabaviti!

Korisni linkovi:

- [1] <http://conference.opensuse.org/>
- [2] <http://www.linuxzasve.com/ovogodisnja-opensuse-konferencija-odrzat-ce-se-u-dubrovniku>



Ugovor Republike Srbije sa *Microsoftom*

(5. deo)

PC = Windows?

Autor: Dejan Maglov

Moramo prvo da napravimo malu digresiju i da dopunimo prethodni nastavak jednom informacijom. Pomenuli smo da postoje drajveri za čitače sertifikacionih kartica i prpratni softver za *Linux*. Po našim saznanjima, još niko nije uspeo da sa ovim drajverom i softverom preko *Linuxa* komunicira sa sajtom Poreske uprave Republike Srbije. Sve radi u lokalnu, ali prilikom povezivanja sa Poreskom upravom komunikacija sa sertifikatom se ne prepoznaje. Ukoliko ipak neko nađe rešenje za ovaj problem, voleli bismo da nam to javite i podelite rešenje sa ostalim našim čitaocima.

Istorijat

Vraćamo se na temu ovog nastavka. Činjenica je da u našoj zemlji važi jednačina *PC = Windows*. Sve statistike potvrđuju da *Windows* pokreće preko 97% *PC*-a. Pod *PC* (personalnim računarima) ovog puta računamo desktop i

laptop računare. Ovo je svetska statistika. U razvijenim delovima sveta (SAD, Kanada i Evropa) udeo *Windowsa* je nešto manji zbog većeg udela *Mac OS X*-a i donekle *Linuxa*, ali u principu *Windows* svuda vlada na *PC*-u.

Da bismo shvatili zašto je to tako, morali bismo se malo vratiti u istoriju. Nešto o toj istoriji smo već pisali u članku o pojmu *FLOSS*-a u broju 11. Već samim čitanjem istorijata *PC*-a može se steći prvi utisak u čemu je problem. Prvih deset godina *PC*-a (1980-1990) *Microsoft* sa *MSDOS*-om nije imao praktično nikakvu konkurenciju na toj platformi. Narednih deset godina (1990-2000) nije imao dostojnu konkurenciju i učvrstio je svoju poziciju sa *Windows 9x* koji je svojom pojavom pokrenuo *PC* revoluciju. U to vreme *Apple Power MAC* i *IBM* kompatibilni *PC* bili su hardverski različiti uređaji (imali su različite procesore), tako da se nisu mešali i praktično nisu bili jedan drugom konkurencija u softverskom pogledu.

Koliko je PC revolucija na čelu sa *Microsoft Windows 95* bila jaka, pokazuje i činjenica da je *Apple* morao da dozvoli da se *MS Office* portuje na *Power MAC*. *Appleovo* zaostajanje za *IBM* kompatibilnim PC-em kulminira prelaskom *Mac Power PC*-a sa *Motorola* na *Intel* procesore po principu „ako ih ne možeš pobediti, pridruži im se”.

Prvih deset godina PC-a *Linux* nije ni postojao. Narednih deset godina je bio suviše mlad i slab. *Linuxova* ekspanzija počinje tek sa pojavom širokopojasnog interneta. Mnogo je manji bio hendikep imati *Windows* bez interneta nego *Linux* bez interneta. *Linux* se potpuno oslanja na internet. Preko interneta se distribuira, nadograđuje i snabdeva programima. Sve informacije o *Linuxu* i *OSS*-u su na internetu, na internetu je i tehnička podrška. Tehničku podršku,

programe i informacije za *Windows* možete naći i kod prvog komšije.

U našoj zemlji masovnije širenje *FLOSS*-a i *Linuxa* počinje tek 2005. godine kad su se pojavile prve *wireless* mreže i *ADSL* priključci.

Dvadeset pet godina apsolutne vladavine *Windowsa* se odrazilo na percepciju da je *PC = Windows*. U poslednjih deset godina iako je *Windows* dobio dostojnu alternativu, statistika se nije znatno promenila u korist *FLOSS*-a. Informacije o postojanju alternative se još uvek teško i stidljivo pojavljuju.

Pogledajte i danas bilo koji kompjuterski časopis, TV emisiju ili udžbenik informatike - o *FLOSS*-u ili ništa nećete moći da čujete, odnosno da pročitate, ili će to biti manje od 1% ukupnog sadržaja.





FLOSS u medijima

Ovde opet dolazimo do začaranog kruga. Nema informacija o FLOSS-u u medijima zato što ne postoji dovoljan broj publike za taj sadržaj koji bi značajno povećao čitanost ili gledanost tog medija. Sa druge strane, korisnika FLOSS-a ima toliko malo zato što ne postoji masovnija informacija o postojanju i upotrebljivosti FLOSS-a.

FLOSS aktivisti su prečesto skloni da za sve nevolje FLOSS-a optuže firme koje stoje iza vlasničkog softvera. Češći uzrok za ovakvo stanje je mnogo prozaičniji. LiBRE! nema članke o BSD-u, ne zato što urednici blokiraju BSD i protežiraju Linux, već zato što nema autore za BSD teme. Slična situacija je i sa medijima. Ne verujemo da bi ijedan urednik odbio dobar tekst ili zanimljivog sagovornika za teme o FLOSS-u, ako mu se ponudi besplatno, a da medij ne sponzorise neposredno Microsoft. Novac ipak okreće svet. Komercijalni mediji bi mogli da angažuju autore za bilo koju oblast pa i za FLOSS, ali za to nemaju interes jer im neće doneti veću čitanost (gledanost). Besplatni članci i gostovanja su druga priča. Svaka popuna zanimljivim sadržajem je dobrodošla uredniku, ako baš nije vezan ugovorom koji bi mogao biti ugrožen pričom o FLOSS-u.

Komercijalni mediji se uvek trude da im sadržaj bude što popularniji i zanimljiviji. Osamdesetih godina okosnica sadržaja je bila priča o novom hardveru. Priče o hardveru devedesetih godina zamenila je priča o Windows 9x i ostalom softveru za ovu platformu.

Početakom 21. veka internet preuzima okosnicu priča u medijima. Danas, okosnica tema medija su mobilni uređaji. *Mainstream* teme medija su uvek oni proizvodi koji se najbrže menjaju i koji su najprodavaniji u tom trenutku. Tako se stvara interesni krug: gledaoci žele da gledaju (čitaju) o zanimljivim aktuelnim temama, mediji žele veću gledanost (čitanost), a sponzori žele da se pojave u gledanim (čitanim) medijima.

Uprkos tome što je razvoj OSS-a najbrži u poslednjih deset godina u odnosu na druge informacione tehnologije, to nije privuklo medijsku pažnju komercijalnih medija jer se ne očekuje da to privuče sponzora (reklame).



U početku kompjuterizacije mediji sa temama o kompjuterima nisu mogli da računaju ni na sponzore, a publiku je tek trebalo da stvore. Svojim entuzijazmom i upornošću prvi autori su uspeli da se probiju u moru konzervativnih novina i TV emisija i da pridobiju urednike koji su na kompjutere gledali samo kao na



igračke. Oni su imali više problema da se medijski probiju, nego što bi sad FLOSS autori imali. Kao što prvi autori kompjuterskih medija nisu mogli da računaju na podršku starih urednika, tako ni FLOSS autori ne mogu da računaju na to da će bez njih urednici naviknuti na već etablirani softver zatvorenog koda početi preko noći da uvode FLOSS u svoje medije bez garancije da će im to doneti profit. Komercijalni mediji već imaju svoju publiku i sponzore koje ne žele da izgube, a pogotovo ne na takav način da ulažu svoje pare u neizvestan eksperiment kao što je uvođenje FLOSS tema.

Druga decenija 21. veka je idealna prilika da OSS probije medijske blokade. Zahvaljujući mobilnim uređajima OSS je već postao *mainstream* tema komercijalnih medija. O *Androidu*, *Firefox OS-u*, *Tizenu* se uveliko priča u svim medijima. Doduše vrlo se stidljivo govori o tome kao o OSS-u. Zanimljivo je da autora ovog teksta vrlo često pitaju zašto koristi *Linux* a ne *Windows* kao sav „normalan svet”? Onda ih on pita, da li koriste *Android* na svom telefonu? I on je *Linux*, da li to znači da ni oni nisu „normalan svet”? Interesantno je da sad svi koriste *Linux* i to im je normalno, iako ne znaju da ga koriste.

LiBRE! je drugi (prvi je *GNUzilla*) pokušaj FLOSS zajednice da se probije medijska blokada za OSS. Za razliku od komercijalnih medija LiBRE! ekskluzivno pokriva samo teme o OSS-u. Takođe *mainstream* tema u LiBRE! je *PC*, naročito, desktop koji nije više u

fokusu medija jer prodaja te platforme pada, uprkos tome što je to i dalje glavna poslovna platforma.

Umesto zaključka

Za sve nevolje ne treba kriviti druge. Stanje u medijima je takvo kakvo jeste. Mediji svoj sadržaj popunjavaju onim temama koje su popularne i donose korist. Njihova uloga je da zabave. Ako uz zabavu donesu i neku korisnu informaciju, odlično, a ako ne, nisu u obavezi.

Ono što je popularno, nije uvek i najbolje. Popularisati nešto što nije popularno, a bolje je od trenutno popularnog, uvek je mučan i pionirski rad koji se materijalno, kratkoročno, ne isplati. Kada bi bili sigurni da će se dugoročno isplatiti, verovatno bi im bilo lakše započeti takav pionirski rad, ali garancija nikad nema. Verovatno su se isto ovako pitali i entuzijasti koji su osamdesetih sastavljali naš prvi računar Galaksija. To im se sigurno nije isplatilo. Nisu ni nastavili svoj rad u tom smeru, ali su sad urednici komercijalnih časopisa o kompjuterima i na neki način im se entuzijazam ipak isplatio.

Mediji nemaju obavezu da bez naknade daju informacije o nečem novom što nije popularno, ali škole bi morale. Trebalo bi da Škola bude prvi i osnovni izvor informacija o bilo čemu pa i o FLOSS-u. Analizu stanja u obrazovnom sistemu ćemo ostaviti za neki drugi nastavak ovog serijala.

Nastaviće se.



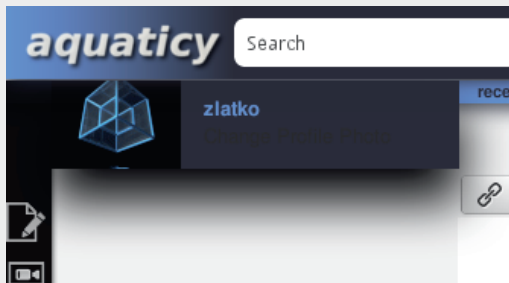
aquaticy.com

Autor: Zlatko Barnak

Svet otvorenog kôda je zaista šarolik, od raznih skripti koje se konstantno razvijaju, preko aplikacija pa do ozbiljnih distribucija, te je zaista sve to teško ispratiti. Naravno tu su *mailing* liste, RSS agregatori, razni portali, blogovi, novine i drugo. Možda je vreme da se učini jedan mali korak napred i da se sve to objedini na jednom mestu, kako bi nam uvek bilo dostupno.

Predstavljamo vam *aquaticy* aktivan *web* portal (skraćeno *AWP*), namenjen preciznom i aktivnom informisanju iz oblasti slobodnog softvera. U sebi objedinjuje sve ono što svako od nas koji pratimo razvoj slobodnog kôda, potajno priželjkuje - najnovije informacije na jednom mestu.

Aquaticy se oslanja na *RSS feedove*. Preuzima najnoviji sadržaj sa raznih *web* sajtova, foruma i blogova i plasira ih kao direktnu informaciju korisniku. Međutim, za razliku od drugih portala, svojim korisnicima plasira tačno one informacije koje su njima zanimljive. Pored toga, kako je ipak reč o nekom vidu društvene mreže, zanimljiv sadržaj



moгу da vam preporuče i vaši prijatelji, a možete i da ispratite sve njihove komentare ili da se uključite s njima u diskusiju, nezavisno od toga da li pratite taj profil ili ne.

Prilikom ulaska na *aquaticy*, na samom startu nas dočekuje polje za pretragu, koje omogućava uvid u sadržaj *AWP*-a. Da bi se videla puna snaga *aquaticy*a, potrebno je izgraditi svoj profil, kao i graditi sadržaj u okviru profila, koji ćete pratiti ili deliti na mreži. Postoji mogućnost automatske izrade profila povezanog sa *facebook* profilom, ali u tom slučaju se nalazite, takoreći, u *stealth* modu (nevidljivi u popisu). Bitno je spomenuti da sve vaše aktivnosti, kao i registracija na *aquaticy*u **neće** prouzrokovati propratni sadržaj na vašem *facebook* profilu. Nakon izrade profila, dočekuju vas dva prazna zida (*ActiveWall*). Jedan *ActiveWall* (*Whats new*) vas obaveštava o svim novostima koje ste naznačili da su vam interesantne, dok drugi *ActiveWall* (*What say friends*) je namenjen praćenju aktivnosti (u vidu komentara ili preporuka) vaših prijatelja.

Pojasnićemo vam nekoliko stvari.

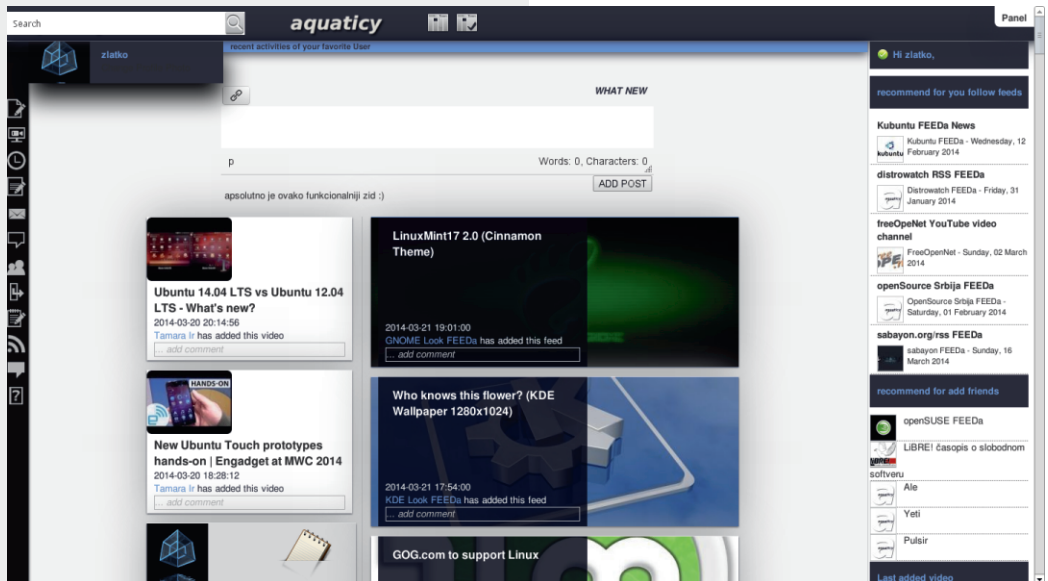


Nakon što izaberete zanimljive profile koje želite da pratite (*Follow*), *ActiveWall* (*Whats new*) prikazuje sadržaj na zasebnim karticama, tj. svaka informacija se nalazi na kartici specifičnog izgleda. Vremenska linija je podjeljena na dva dela. S leve strane se nalaze objave koje su ručno dodate (*microblog*, blog, video i najave), dok se na desnoj strani nalazi sadržaj koji je korisnik omogućio da se automatski dodaje sa izvornog sajta (*feeds*). Iznad vremenske linije se nalazi prostor za *microbloging*, kao i zadnja uneta poruka.

ActiveWall (*What say friends*) prati aktivnosti vaših prijatelja. Da biste sa nekim postali prijatelj, potrebno mu je uputiti zahtev, i tek nakon njegovog odobravanja bićete u mogućnosti da se međusobno pratite. Na „*What say friends*” zidu videćete njegove komentare ili njegove preporuke o zanimljivom sadržaju.

Na oba zida sa leve strane se nalazi funkcionalni *dock*. Omogućava lako i brzo postavljanje novog sadržaja (blog, video i najava), bez potrebe da se napušta stranica *ActiveWall*. Sem dodavanja sadržaja, *dock* za vas prikazuje i prapratne informacije, poput zahteva za prijateljstvo, privatnih poruka, komentara vezanih za sadržaj koji ste vi objavili itd. Sa desne strane zida se nalazi preporuka za praćenje ili za dodavanje prijatelja, zatim još neki prapratni sadržaji.

Naravno, sem ova dva *ActiveWalla*, koja smo ovde ukratko opisali, postoji još mnoštvo korisnih stranica za prikaz ili unos sadržaja, kao i stranica za podešavanje raznih opcija. Najbolji način da upoznate *aquaticy AWP* je da ga istražite zajedno s prijateljima, a ukoliko ih ne vidite tamo, pozovite ih da vam se pridruže.



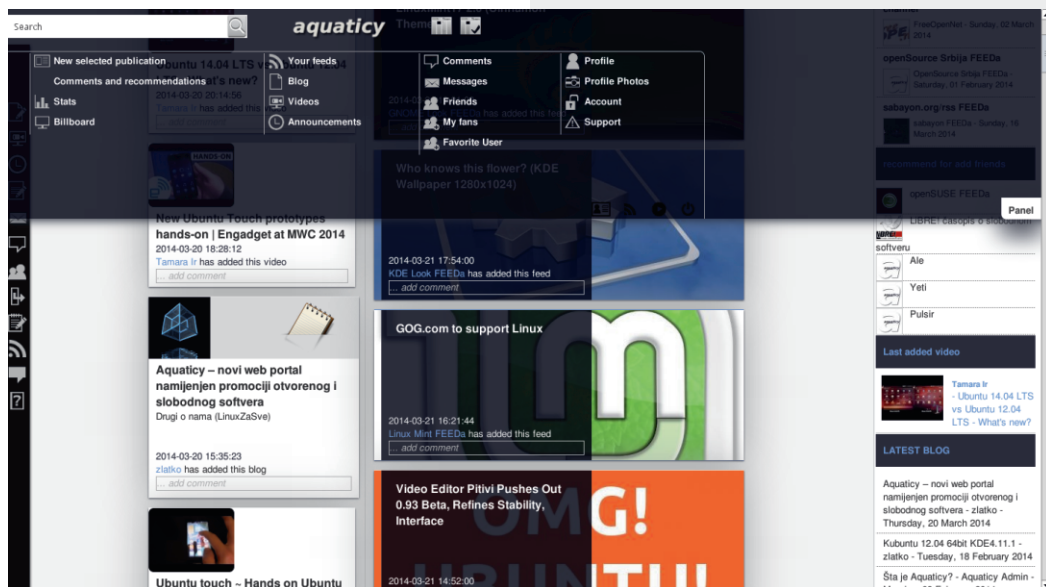
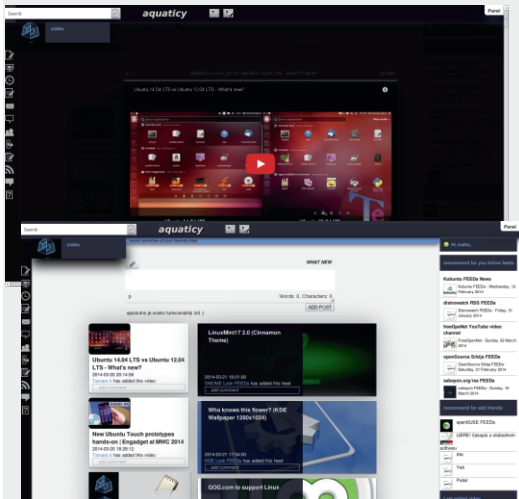


Za programere, administratore sajtova, blogere i ostale urednike raznih stranica, portala, foruma i video kanala, *aquaticy* je zaista mesto gde bi trebalo da imaju aktivan profil. Ne zahteva takoreći nikakvo održavanje s vaše strane, potpuno je automatizovan i jedino što možete da imate od njega, jeste korist. Prikazuje sav novi sadržaj koji dodajete (u formi koja podstiče korisnika da poseti vaše stranice), a za uzvrat prosleđuje korisnike na vaše stranice. Jedino što treba da uradite, jeste da dodate *URL (RSS Feed)* vaše *web* stranice (blog, sajt, forum, portal, video kanal i drugo).

Aquaticy je još uvek u svojoj razvojnoj beta fazi iz koje uskoro izlazi, pa neke od dodatnih opcija nisu aktivne, ali to ne umanjuje vrednost ili način korišćenja *aquaticy* AWP-a. Budite među prvima, jer to je jedna od garancija da

će velik broj budućih korisnika posetiti upravo vaš profil i vaše stranice.

www.aquaticy.com





Serbian 2014

(1. deo)

Autor: Dejan Maglov

Uvod

Ovo neće biti klasična recenzija *Linux* distribucije jer Serbian nije obična distribucija. Serbian je srpska nacionalna distribucija nastala uglavnom komponovanjem, a ne razvojem. Baš zato što je srpska, kad je autor ovog teksta u svojstvu urednika pitao u redakciji, „Ima li dobrovoljaca da pišu o ovoj distribuciji?“, dobio je odgovor da je to škakljiva tema i da distribuciju nije prihvatila *Linux* zajednica Srbije. Kad je tako, autor ovog teksta je pomislio: „Onda je to baš zanimljiva tema za mene. Ja volim škakljive i kontroverzne teme.“ (prim.aut.).

LiBRE! ima obavezu da populariše domaće projekte. Zbog ograničenja prostora za jedan članak i zanimljivosti teme možete očekivati više nastavaka o ovoj intrigantnoj distribuciji.

Serbian

Kao prvo moramo utvrditi da li je to uopšte tema za nas. Definitivno, na prvi i na svaki drugi pogled se vidi da je to slobodni softver sa malim primesama

besplatnog vlasničkog softvera (*flash-plugin*, *kodeci* i *Skype*). Takođe, jasno je da je to *Linux* distribucija, originalni *Linux* kernel je osnova ovog operativnog sistema. Licence nisu povređene neovlašćenim potpisivanjem ispod dela koji je neko drugi uradio.

Da li je ovaj operativni sistem nešto novo, što bi priređivaču ovakve distribucije dalo za pravo da promeni originalno ime? Ovde treba stati i razmisliti. Šta je definicija nečeg novog? Za vlasnički softver je primenjiva kreacionistička teorija razvoja. Po toj teoriji vlasnik softvera je njegov Bog i stvoritelj. Jedino stvoritelj ima pravo na izmenu svoje kreacije i dalje umnožavanje. Prema tome stvoritelj je taj koji će reći kada je nešto novo.

Slobodan softver prati evolucionističku teoriju razvoja. Po evolucionističkoj teoriji novi život je plod kopiranja, modifikovanja i kombinovanja. Postoje evolutivni skokovi koji su plod većih promena spoljašnjih uslova, ali u većini slučajeva promene su male. Samo najbolje kombinacije kopiranja, modifikovanja i kombinovanja opstaju, a jedino vreme može da pokaže da li je ta kombinacija uspešna.



Serbian ima sve elemente evolucije. To je kopiran *Debianov* model modifikovan simpatičnim originalnim grafičkim elementima kao i podešavanjima koji odgovaraju zadatom projektnom zadatku, kombinovan sa aplikativnim softverom koji, takođe, zadovoljava projektni zadatak.

Čak je i ime Serbian, slučajno ili namerno, vrlo primereno ovoj distribuciji. Ako je tačna legenda da je ime *Debian* izvedeno od imena Jana Mordoka, osnivača *Debian* pokreta i njegove žene Deb, onda je Serbian odlična analogija. Serb kao Srbin i Ian kao Jan Mordok. Deb će, nadamo se, oprostiti što je „izgurana” iz imena.

Projektni zadatak

Projektni zadatak ove distribucije je jednostavan - komponovati potpuno

lokalizovanu srpsku distribuciju sa, takođe, lokalizovanim aplikativnim softverom koji pokriva sve osnovne oblasti primene računara i podesiti taj sistem tako da omogućava filozofiju „instaliraj i odmah koristi”.

Mišljenja smo da ovakav projektni zadatak ima više smisla od projektnog zadatka *Pear OS*. Da podsetimo, *Pear OS* je sad već bivša francuska distribucija čija je osobenost bila samo u tome da je podsećao na *Mac OS X*. *Pear OS* je bila zvanična *Linux* distribucija na *Distrowatchu*. Ako je *Pear OS*-u zvanična *Linux* distribucija, ne vidimo zašto i Serbian ne bi bila prihvaćena distribucija.

Kompozitori Serbiana nisu uradili nikakvo čudo. Sve komponente Serbiana su već bile lokalizovane. Ovo ne umanjuje spretnost i napor kompozi-



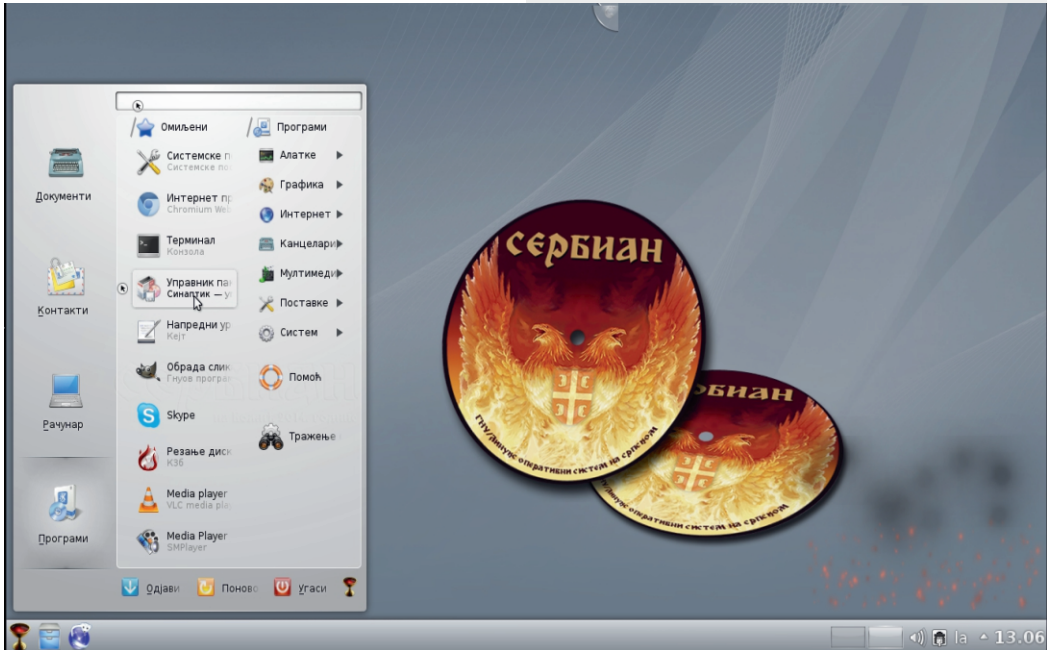
tora da pronađu najbolje lokalizovane komponente i uklope ih u jednu skladnu celinu.

Svesno ili nesvesno Serbian je postavio neke repere koji će ga postaviti na značajno mesto u istoriji FLOSS-a u Srbiji, bez obzira na to da li će biti prihvaćen i da li će nastaviti sa razvojem ili ne. Primarno, ovo je prva srpska distribucija koja je uspela da dođe do nulte verzije (analogno nultom broju LiBRE!) bez novčanih subvencija. Sekundarno, svojom pojavom i konceptom komponovanja Serbian je dao jasan presek trenutnog stanja aktivizma u FLOSS zajednici Srbije.

Činjenica da je Serbian prvi uspeo da

dođe do nulte verzije, daje mu prednost i veću šansu da opstane. Takođe, pokazuje da je taj koncept uspešniji od nekih drugih. Za korisnika je važan njegov koncept „instaliraj i odmah koristi”. Ono što mu se najviše zamera, jeste da nije ništa novo nego obični remiks. Upravo to je njegov najveći plus i interes autora. Serbian je remiks već postojećeg softvera bez većih izmena i to ga čini samoodrživim. Autori praktično nemaju posla oko održavanja, Serbian će se sam održavati bar do nove stabilne verzije *Debiana*.

Ovakav koncept je idealan za lokalnu distribuciju opšte namene sa malom lokalnom zajednicom koja je održava. Da nije tako i da su autori radili veće

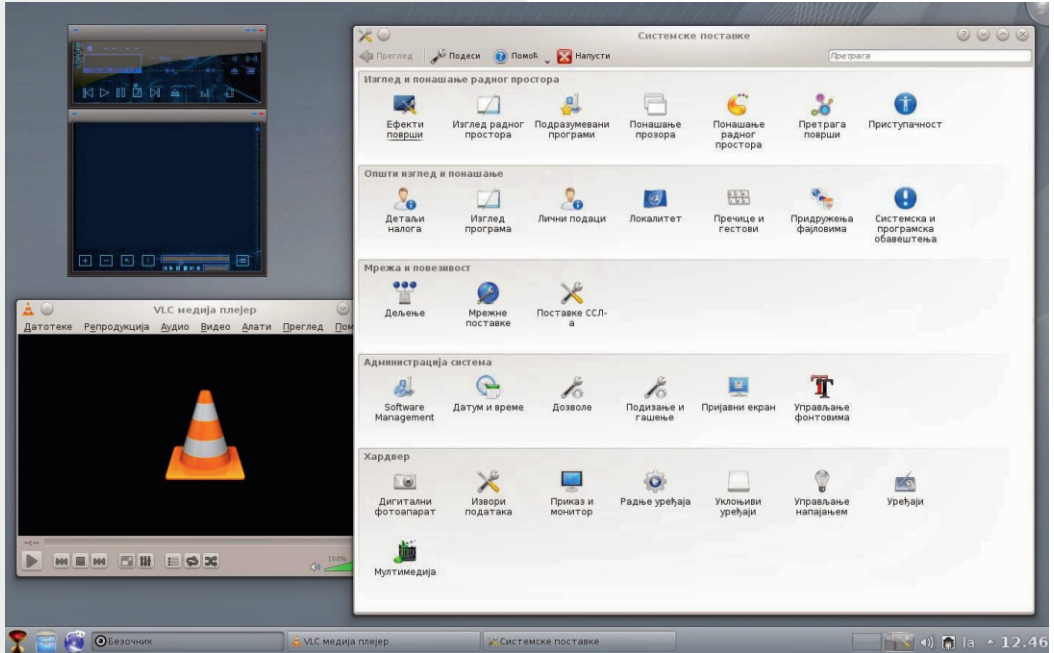


izmene na izvornom softveru, imali bi velikog posla oko održavanja i to bi funkcionisalo samo dok ih drži entuzijazam.

Sledeća zamerka na račun Serbiana - čemu još jedna *Linux* distribucija, pogotovu što to nije revolucionarno nova distribucija? Odgovor je vrlo jednostavan - prvo zato što slobodni softver podstiče na ovakve projekte, a drugo zato što je dobro da postoje nacionalne distribucije. Nacionalne distribucije popularišu slobodan softver, čuvaju jezik i posebnost jedne nacije. Istina je da svaki iole iskusniji *Linux* korisnik može sebi da iskomponuje ovo što Serbian nudi. Za to je potrebno izdvojiti nekoliko sati, a Serbian to nudi za manje od pola sata. Serbian je sad postavio standard šta treba da ima jedan lokalizovan OS da bi bio

kompletan. Ovakvo skupljen lokalizovan slobodni softver na jednom mestu je pokazao sve mane i poneku vrlinu dotadašnje lokalizacije softvera. Pokazuje i koje oblasti primene računara nisu pokrivenе lokalizovanim softverom.

Ne samo da je jednoj naciji potrebna jedna ovakva distribucija *Linuxa* opšte namene, već je potrebna još jedna strogo kontrolisana distribucija poslovne namene. Sama stihijska priroda razvoja slobodnog softvera nameće potrebu da nacija koja želi da ga koristi u poslovne svrhe, mora da ima svoju strogo kontrolisanu distribuciju. U poslovnom okruženju ne sme se desiti da neki nekontrolisani *update* dovede do prestanka rada bitnog poslovnog softvera. Zadatak nacionalne distribucije poslovne namene je da kontroliše



razvoj operativnog sistema i prijavljenog aplikativnog softvera te da na vreme interveniše i spreči bilo kakvu mogućnost nestabilnosti u radu. Naravno, ovo je već neka druga priča za drugi članak. Ovde samo napominjemo kao ilustraciju da ne samo da nije besmisleno imati još jednu distribuciju u moru različitih distribucija, nego nam čak nije ni dovoljna za sve naše potrebe.

Mane Serbia

Mane Serbia analiziramo kroz prizmu projektnog zadatka. Osnovni projektni zadatak je lokalizacija. Pošto je Serbian komponovana distribucija, sve mane „idu na dušu” uzvodno na lokalizaciju komponenti. Naše zamerke su usmerene u dva pravca.

Prva zamerka se odnosi na to da mnogi slobodni softveri uopšte nisu lokalizovani. Autor ovog teksta, recimo, ne koristi lokalizovan OS zbog toga što mu smeta kad na istom sistemu ima pomešane programe na engleskom jeziku, na srpskoj latinici i na srpskoj ćirilici (prim.aut.). Zato je prava umetnost bila komponovati Serbian koji neće imati ovih problema. U Serbian ne mogu ući komponente koje nisu već lokalizovane na srpsku ćirilicu. Zato nije primerna primedba, zašto neki popularniji program nije ušao u spisak programa Serbia? Odgovor je jednostavan - nije lokalizovan. Zato smo već rekli da je značaj Serbia u tome da je napravio svojevrsni presek trenutnog stanja lokalizacije slobodnog softvera.

Poboljšanje ovog segmenta može da se



ostvari samo uzvodno. Što više budemo imali lokalizovanog softvera, Serbian će biti bogatiji i korisniji. Neka Serbian zbog toga bude podsticaj za bolju lokalizaciju softvera.

Druga zamerka odnosi se na nepostojanje standarda za lokalizaciju. Serbian je i tu dao jasnu sliku. Na istoj stranici izbornika (favoriti) imamo *Skype*, „K3b” i „Gnuov program za obradu slika”. Ova nedoslednost „bode oči”. Ovaj problem je imao i LiBRE!, kako pisati imena programa? Da li pisati u originalu latinicom, pisati po Vukovom pravilu „piši kako se čita” ćirilicom ili prevesti ime na srpski jezik? Ovo nije standardizovano i ne zna se kad će biti. Bilo bi dobro da se iskoristi Serbian i jednom za svagda usvoji standard kako ćemo to pisati.

Sličan je problem i sa prevodom komandi i alata u programima. Dok su lokalizacije programa bile projekti svaki za sebe, nije bio uočljiv problem nepostojanja standarda. Kad se ovako upakuje u jednu distribuciju, sve mane isplivaju na površinu.

Za kraj

Šta reći za kraj prvog nastavka? Ovaj nastavak se poprilično odužio, a uspeli smo tek da prokomentarišemo koncept. Sadržaj, *artwork*, upotrebljivost, stabilnost, performanse i upoznavanje sa autorima (autorom) moramo da ostavimo za neki drugi nastavak.

Zaključak ovog nastavka: Serbian ne samo da je ispunio očekivanja, već ih je

i premašio. Značaj Serbiana je nemerljiv za *FLOSS* zajednicu Srbije i to bez obzira šta će biti sa njim u budućnosti. Ne bismo voleli da u budućnosti autori Serbiana odustanu od komponovanja nacionalne distribucije zarad pravljenja svetski priznate i prihvaćene distribucije. To bi bio veliki gubitak za nacionalni *FLOSS* a mali ili nikakav doprinos svetskom. Trebalo bi da autori budu zadovoljni činjenicom da su „prvi u svom selu”. Sada su u poziciji da svojim radom utiču na sve nacionalne *FLOSS* projekte. Šta više od toga može da poželi neki *FLOSS* aktivista, ako već dostigne poziciju „dići ruku, a puk da ga sledi” (stihovi Duška Trifunovića iz pesme „Šta bi dao da si na mom mestu?”).

Ovaj nastavak bi bio nedorečen kad osim pohvala na račun kompozitora Serbiana ne bismo spomenuli i sve uzvodne projekte kao što su *LibreOffice* Srbija, *Mozilla* Srbija, zatim ljude koji su pomogli lokalizaciju *KDE*-a, autora *Lancelota*, *Great little radio playera* i mnoge druge koji su omogućili da se Serbian pojavi.

Koristan link:

<http://www.debian-srbija.iz.rs/p/serbian.html>





Uvod u programski jezik C

(1. deo)

Autori: Veljko Simić, Stefan Nožinić

Reč, dve o nastanku

Programski jezik C nastao je osamdesetih godina prošlog veka, možemo reći iz krajnje potrebe za programskim jezikom, koji je prenosiv, efikasan i lak za učenje. Pre C-a, programeri su morali da biraju jezik u kome je jedan skup osobina bio favorizovan u odnosu na drugi. Npr. iako se *Fortran* može koristiti za pisanje dobrih programa koji imaju primenu u nauci, on nije najpogodniji za pisanje sistemskih programa. Zatim *Basic*, lako se uči, ali nije previše moćan i veoma je nepogodan za pisanje dugačkih programa. *Assemblerom* se mogu pisati veoma efikasni programi, ali se on teško uči i zadaje ozbiljne glavobolje programerima i nije isti za svaku arhitekturu računara. Početkom sedamdesetih godina, kada je računarska revolucija počela da se zahuktava, potražnja za softverom je uveliko prevazišla mogućnost programera da ih otkuca. Napokon, programski jezik C projektovao je *Dennis Ritchie* 1972. godine u *Bellovim* laboratorijama i ugradio ga u mašinu *DEC PDP-11*.



GNU C prevodilac

Pre nego što napišemo naš prvi program, potrebno je da imamo instaliran *GCC (GNU C Compiler)* prevodilac, kako bismo naš program mogli da pokrenemo. Ako imate *UNIX*-oliki sistem, *GCC* je verovatno već instaliran. Da biste to proverili, upišite `gcc --version` u konzoli. Ako je ispis na vašem ekranu sličan sledećem, to znači da je *GCC* instaliran.

```
$ gcc --version
gcc (Debian 4.7.2-5) 4.7.2
Copyright (C) 2012 Free Software
Foundation, Inc.
This is free software; see the
source for copying conditions.
There is NO
warranty; not even for
```



MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Ukoliko nemate instaliran *GCC* izvorni kôd prevodioca, možete ga dobiti od organizacije *Free Software Foundation*. Instalaciju možete pogledati na: <http://gcc.gnu.org/install/>.

Naš prvi program

Kôd programa možete napisati u bilo kom uredniku teksta, a najpopularniji su *Gedit*, *VIM*, *Nano*. Potom ga je potrebno prevesti nekim prevodiocem, kao što je to gore pomenuti *GCC*. Takođe, postoji mnoštvo razvojnih okruženja koji olakšavaju pisanje programa i njegovo prevođenje i pokretanje (*Eclipse*, *CodeBlocks* i *NetBeans*). Može se reći da je standard da prvi program u

nekom programskom jeziku bude *Hello World!* koji na standardni izlaz ispisuje poruku **Hello World!**. Tako će biti i u ovom tekstu.

```
#include<stdio.h>

int main()
{
    printf ("Hello World!\n");
    return 0;
}
```

Prvu liniju našeg kôda vam nećemo u ovom broju časopisa objasniti. Zasad je dovoljno reći da **#include** direktiva služi da se u vaš trenutni program učitaju dodatne biblioteke, odnosno zaglavlja, pa se time omogućava modularnost. Zaglavlje **stdio.h** sadrži mogućnosti za rad sa ulazno-izlaznim





funkcijama. `int main()` je zaglavlje funkcije `main`, dok sve što se nalazi između vitičastih zagrada, označava njeno telo. `int` znači da je njena povratna vrednost ceo broj, odnosno u našem slučaju `0` (`return 0;`). Svaki program mora imati funkciju `main`, jer je to ulazna tačka programa, a ako je nema, prevodilac će prijaviti grešku. U telu naše jedine funkcije se nalaze dve naredbe, svaka naredba se mora završavati tačka-zarezom („;“). Jedna naredba ispisuje na ekran **Hello World!** i to čini pomoću funkcije `printf(...)` koja je definisana u `stdio.h` biblioteci, dok druga naredba `return 0;` vraća vrednost funkcije `0`. Ovde je bitno napomenuti i `\n` na kraju stringa za ispis. Ovo je specijalan karakter koji govori konzoli da pređe u novi red.

Prevođenje programa

Mi još uvek nismo napravili nijedan program. Napisali smo njegov kôd koji računar ne razlikuje od obične tekstualne datoteke, sve dok ga ne prevedemo na jezik razumljiv računaru. Za to nam služi prevodilac (*Compiler*), o kome smo pričali u ovom tekstu. Da biste preveli (kompajlirali) program, dovoljno je da ukucate sledeću naredbu:

```
gcc helloworld.c
```

U ovom slučaju `helloworld.c` je ime izvorne datoteke u kome je smešten kôd našeg programa. Ako se vaša datoteka zove drugačije, ukucajte ime vaše datoteke. Obratite pažnju na ekstenziju `.c`. Primetićete da se stvorila nova datoteka pod nazivom `a.out`. To je izvršna

Uvod u programski jezik C



datoteka i to je program. Pokrenućemo ga sledećom naredbom:

```
./a.out
```

Kao rezultat našeg programa, ispisaće se na ekranu poruka: **Hello World!** U slučaju da ne želite da se vaš program zove `a.out` (morate priznati, nije zanimljivo da se tako zove program), naredba pri prevođenju izgleda ovako:

```
gcc helloworld.c -o helloworld
```

Sada se vaš program zove `helloworld`. Potrebno je obratiti pažnju na dodatnu opciju koja je zadata prevodiocu.

Ovo je bio kratak uvod u tematiku. U narednim brojevima vam pišemo detaljnije i mnogo zanimljivije stvari.

Learn C Programming



Linux unutraške

Autor: Nikola Hardi

Uvod

Najverovatnije ovaj tekst čitate na nekom računaru ili ste barem koristili računar da biste preuzeli naš časopis. Pisali smo već mnogo puta o programima koji su namenjeni vama kao korisnicima računara. Pisali smo i o operativnim sistemima i raznim distribucijama, ali ovaj put želimo da ispričamo priču o tome šta se sve obično događa na računaru unutraške od trenutka kada ste počeli da čitate ovaj tekst, pa sve do trenutka kada ste pritisnuli dugme za uključivanje računara. Akcentat će biti baš na onim stvarima o kojima se retko kada razgovara, o onome šta se ne vidi.

PDF reader i pokretanje programa

Pretpostavićemo da pred sobom upravo imate PDF čitač (ugrađen u browser ili kao poseban program). Krajnji rezultat je slika na monitoru, ali pre toga je PDF fajl morao biti preuzet na računar, pokrenut program koji učitava taj fajl, zatim bila je potrebna mreža za pristup internetu. Jedan od zadataka operativnih sistema je da pruže upravo takav

vid podrške aplikacijama. Videćemo malo kasnije da na kraju svi putevi nekako vode do kernela. Ovaj nivo obuhvata krajnje aplikacije koje najčešće koristimo, u ovom slučaju *evince*, *okular*, *apvlv* i *xpdf*. Naravno, u ovu grupu programa spadaju i razni audio *playeri*, internet pregledači, kancelarijski paketi i ostalo.

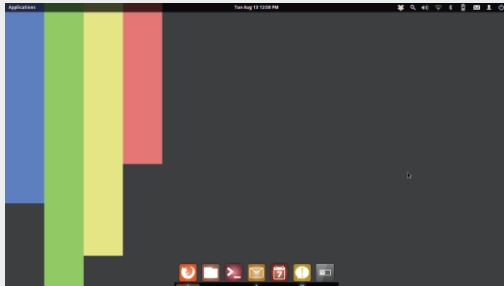


Okruženje radne površi

PDF čitač ste mogli da pokrenete na mnogo načina – recimo klikom na datoteku u nekom pregledaču datoteka ili direktnim pokretanjem programa. Da bismo se lakše snašli u zbrnci datoteka i direktorijuma, vremenom su razvijeni skupovi programa za tu namenu - *desktop*, odnosno radna okruženja. Njihova namena je, u širokom smislu, da nam omoguće da pokrećemo programe,



da pronađemo datoteke, da poredamo prozore po našoj želji itd. Pojam okruženja radne površi je vrlo širok i nije jasno definisan. U ovaj nivo spadaju GNOME, Unity, MATE, KDE, Xfce i mnogi drugi. Naravno da ne moramo koristiti kompletna okruženja radne površi za ove zadatke, ali ovo je klasična postavka na današnjim kućnim računarima koje pokreće Linux.



Session manager

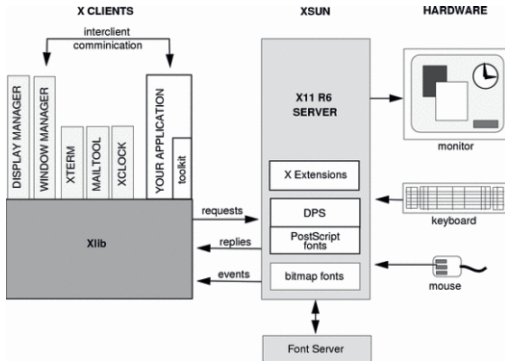
Jedan računar može koristiti više korisnika i mogu biti dostupna razna okruženja radne površi istovremeno. Da bi računar znao koje grafičko okruženje da pokrene i koji korisnik trenutno želi da koristi računar, moramo na neki način da mu prenesemo tu informaciju. Za ovaj zadatak je najčešće zadužen *session manager* (nekada se naziva i *desktop manager*). Uslovno rečeno, *session manager* vidimo kada se prijavljujemo na računar, unosimo korisničko ime, lozinku i biramo sesiju (*desktop* okruženje). Neki od primera *session managera* su *LightDM*, *GDM*, *KDM* i *slim*. Ovo je još jedan običan program na računaru i njega je takođe morao neko da pokrene. Za pokretanje *session managera* uglavnom je zadužen *init system* o kojem će biti reči malo kasnije. Bitno je

pomenuti da je za grafički *session manager* potrebna podrška za grafička okruženja koju (još uvek) pruža *X server*.



X server

U redu, sada smo već stigli do granice iza koje prosečan korisnik računara ne vidi šta se sa sistemom dešava. *X server* je program koji sistemu pruža podršku za grafička okruženja. *X* nije jedini, ali je još uvek *de facto* standard za *Unixlike* sisteme. O alternativama i životu *X.org* projekta je bilo više reči u našem prethodnom broju kada smo pisali o *Waylandu*. *X* se nalazi na putu od aplikacije koja želi da iscrtava nešto na monitoru, a posle njega su *driver* za grafičku kartu i podrška u kernelu. Podrška koju *X* pruža, uglavnom se svodi na to da opsluži zahteve za iscrtavanje aplikacije. Kao rezultat dobijemo sliku koja predstavlja stanje te aplikacije, a tu sliku potom *window manager* prikazuje u nekom delu monitora. *Window manager* može da funkcioniše kao deo okruženja radne površi ili zasebno, a o ovome je takođe bilo više reči u prethodnom broju u već pomenutom tekstu. *X* je takođe jedan od procesa koje najčešće pokreće *init system*.

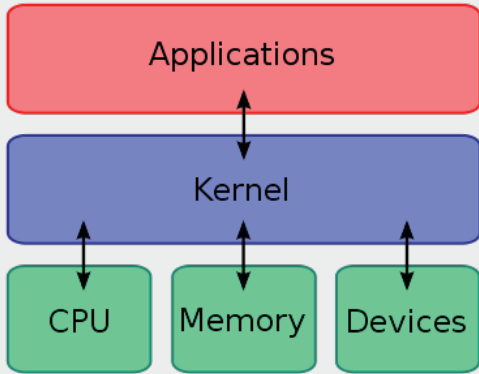


Init

Već nekoliko puta smo u ovom tekstu pomenuli *init system*, što naravno znači da je ovo vrlo bitan deo celog sistema. O *init* procesu i *init* sistemima možemo da napišemo i celu knjigu, ali ovaj put ćemo samo pokušati da opišemo njihovu namenu i zadatak. *Init system* je (jedan od) sistema za automatsko pokretanje servisa i programa. Možemo reći da su njegove dve glavne komponente *init* proces i *init* skripte. *Init* proces je prvi proces koji se pokreće na našem računaru, pokreće ga sam kernel, njegov *ID* je 1 i iz njega nastaju svi ostali procesi. Druga komponenta su *init* skripte koje se gotovo uvek nalaze u `/etc/init.d/` direktorijumu. *Init* skripte predstavljaju uputstva za pokretanje servisa (tzv. *daemon*) i drugih programa. *Init system* funkcioniše otprilike tako što nakon pokretanja *init* procesa, kernel pokrene servise i to putem izvršavanja već pomenutih *init* skripti. Kao i u drugim delovima sistema, i ovde postoji niz alternativa. Neki primeri su *SystemV*, *OpenRC*, *upstart* i *SystemD*.

Kernel

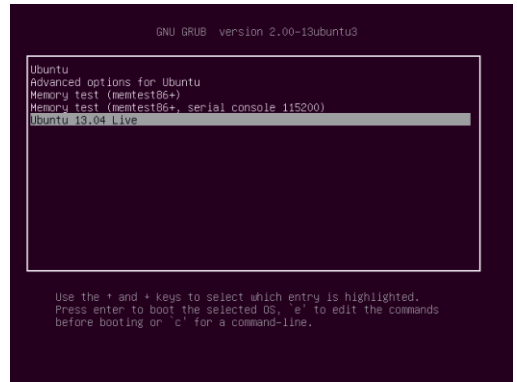
Kernel je jedna od magičnih reči koja se često spominje u *Linux* svetu i svetu drugih operativnih sistema i spada u domen crne magije za mnoge korisnike računara. Kada smo već do kolena zagazili u crnu magiju, red je da razjasnimo o čemu je tačno reč. Naravno, i ovo je tema koja zaslužuje poseban tekst, knjigu i život. Kada bismo operativni sistem morali da podelimo na nekoliko delova, to bi bili kernel i sve ostalo. Kernel vodi sasvim drugačiji život u računaru u odnosu na ostale procese. Upravlja hardverom, odlučuje koji proces će se izvršavati, šta će procesor da radi u nekom trenutku i upravlja *RAM* memorijom. Kernel je zapravo jedino mesto gde se novi procesi mogu kreirati. U kernelu se nalaze *driveri* (zovu se još i upravljački programi) koji nam omogućuju da koristimo naš računar, a još važnije je da omogućuju našim programima rad i na drugim računarima bez obzira na to koliko se razlikuju po hardverskim karakteristikama. Kernel obezbeđuje interfejs za upravljanje fajlovima, pristup mreži i još mnoštvo toga, a proces inicijalizacije, „raspakivanja” i pokretanja kernela takođe izlaze iz okvira ovog teksta. Najbitnija stvar koju treba da znamo, jeste da kernel predstavlja spregu između hardvera, upravlja radom računara, pokreće se procesom „*bootovanja*” za koji je zadužen *bootloader* i nakon što kernel postane funkcionalan, pokreće se proces *init*, a ostatak već znate. Osim *Linux* kernela postoje i drugi, recimo *Mach*, *Hurd* i *BSD*. Kernel se često zove i jezgro, što bi bilo bukvalno značenje te reči.



Bootloader

Procesori imaju krajnje jednostavan i dosadan način rada. Možemo ih smatrati crnim kutijama koje na osnovu jednog niza nula i jedinica (programske reči, instrukcije, dela koda programa) znaju u kakvo stanje treba da se podese, a potom da od nekog drugog niza nula i jedinica (podaci), naprave treći niz nula i jedinica (rezultat) i taj rezultat zabeleže negde (zapisivanje u memoriju). Kada uključimo računar, dešava se upravo to. Procesor počne da izvršava neki program, a program koji učitava operativni sistem, tačnije prvo kernel, pa onda sve ostalo, naziva se *bootloader*. Ako se pitate zašto ne pokrećemo odmah kernel, odgovor je jednostavan. Kernel je prevelik da bi ga procesor mogao odmah pokrenuti pa prvo koristimo jednostavniji program čija namena je učitavanje kernala i priprema za njegovo pokretanje. Neki od najpopularnijih su opšte poznati *GRUB* (da, da, ono čudo sa kojim ste imali problema), *LILO* (njega verovatno ne koristite, osim ako niste *slacker*) ili *u-boot* (ako se srećete sa

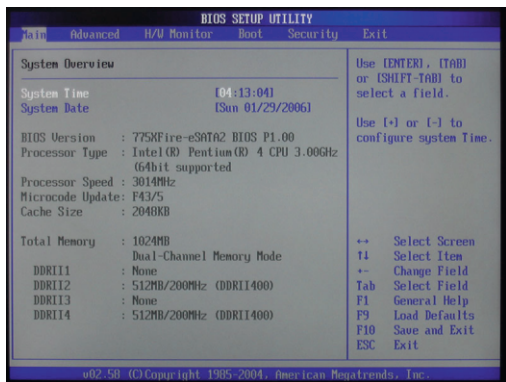
embeded sistemima ili imate *raspberry*). Međutim, ni ovo nije kraj jer možemo da imamo više *bootloadera* na računaru, više uređaja na kojima se nalaze operativni sistemi, a procesor ne može direktno da komunicira ni sa hard diskom, ni sa monitorom, a opet na neki način možete da vidite *bootloader* ili da izaberete uređaj sa kojeg želite da startujete sistem. Za to je zadužen *BIOS*.



BIOS

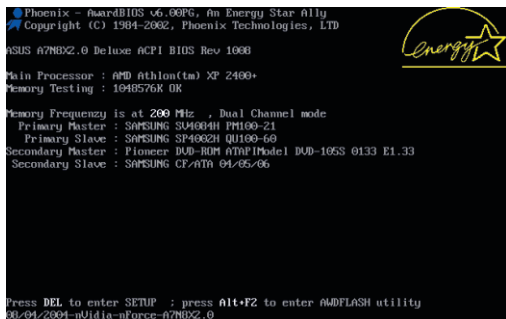
BIOS (*Basic Input/Output System*) je prvi i osnovni set programske podrške na računaru. Obezbeđuje ga proizvođač matične ploče, a njegova namena je da pruži vrlo jednostavna podešavanja računara i komunikaciju sa računarom na nekom osnovnom nivou. *BIOS* pruža podršku za ispis na monitor, unos sa tastature i pristup memorijskim uređajima. Ovo znači da je *BIOS* dovoljan da biste pokrenuli neki program na računaru, bez potrebe za operativnim sistemom. *Bootloader* je jedan takav program, ali vi možete da napišete i svoj program koji će se izvršavati direktno na procesoru. Za ovakve igrarije preporučujemo *osdev wiki* (<http://wiki.osdev.org>).

Zanimljivo je reći da se ovi programi moraju nalaziti u prvih 512 bajtova uređaja (odnosi se na CD, hard disk ili flash memoriju) i ne mogu biti veći od 1MB. Ovde takođe postoje alternative, postoje *open source BIOS*-i, a za više detalja o tome potražite *open firmware*, *openbios* i *coreboot*.



POST

Ostao je još jedan mali detalj. Pre nego što BIOS postane sasvim funkcionalan, računar prvo pokreće POST (*Power On Self Test*) da bi proverio svoje stanje. POST je najčešće deo BIOS-a, odnosno deo *firmwarea* matične ploče. To su ona prva slova koja se pojave kada pritisnete dugme za uključivanje i ono što pišti kada memorija, procesor ili grafička kartica nisu na svojem mestu.



Za kraj

Recimo da smo sada prošli kompletan put od preuzimanja našeg časopisa do pritiska dugmeta za uključivanje. Bolji izraz bi možda bio da smo proleteli kroz ovu proceduru jer mnogo toga nije napisano. Ima više načina kako se sistem može startovati, ali ovo je jedan klasičan primer. Dakle, da se podsetimo ali sad hronološkim redom. POST izvršava osnovni vid provere hardvera, proverava da li su sve komponente računara na broju. BIOS je prvi nivo apstrakcije koji omogućava procesoru da komunicira sa ostatkom računara i govori procesoru na kojem uređaju da potraži program koji treba da izvrši. Taj prvi program se nalazi u *bootsectoru* (prvih 512 bajtova nekog memorijskog uređaja) i naziva se *bootloader*. To je mali program koji učitava kernel i započinje njegovu inicijalizaciju. Nakon što je kernel funkcionalan, on pokreće prvi proces na sistemu - *init* proces, i njemu prepušta da dalje aktivira sistem. *Init system* potom pokrene osnovne servise, između ostalog i *X* koji pruža usluge za grafička okruženja, pokreće se *session manager* kako bi se korisnik prijavio za rad i potom se pokreće okruženje radne površi. U okruženju radne površi već imamo ikonice koje predstavljaju fajlove, dokumenta, direktorijume i izvršne datoteke. Kada izaberemo neku od ikona, pokreće se potreban program i on se pojavljuje na monitoru. Nadamo se da smo uspeali da vas zainteresujemo. Vrlo rado ćemo pisati o sličnim temama i time vam pomoći da rešite neke nedoumice. Javite nam se i izaći ćemo vam u susret.



OpenCV

Autor: Stefan Nožinić

Šta je računarska vizija?

Računarska vizija (eng. *computer vision*) je oblast računarskih nauka koja se bavi obradom i analizom slike i višedimenzionalnih prostora (najčešće trodimenzionalni i četvorodimenzionalni). Ova oblast je veoma važna i ima veliku primenu u industriji i nauci.

Ova oblast je zastupljena u:

- kontroli robota u industriji
- navigaciji
- detekciji objekata i događaja
- indeksiranju baza slika
- prepoznavanju lica
- analizi slike u medicini
- popravci slike
- optičkom prepoznavanju teksta (OCR)

Proces se najčešće sastoji od:

- dobijanja slike sa kamere ili nekog drugog optičkog senzora
- obrade slike (otklanjanje šumova, konverzija u potreban sistem boja i

drugo)

- segmentacije (razdvajanje bitnih stvari od nebitnih)
- ostalih obrada (prepoznavanje objekata, specifičnosti, transformacije i drugo)

Kao što se može primetiti, ovo je dosta složen proces i razvijanje sistema baziranih na računarskoj viziji postaje sve izazovnije kao što i problemi u ovoj oblasti postaju sve učestaliji i složeniji.

Zbog gore navedenih razloga, potrebno nam je rešenje koje će nam omogućiti viši nivo apstrakcije i omogućiti da razmišljamo o novim rešenjima koristeći već postojeća. U razvijanju ovakvih sistema tražimo rešenje koje će nam povećati produktivnost i kreativnosti naše mogućnosti za rešavanje problema preusmeriti na zanimljivije stvari od stalne brige o tome da li smo dobro implementirali već neko postojeće rešenje.

Šta je OpenCV?

OpenCV (*Open Source Computer Vision*)



je razvojna biblioteka pre svega za C++ koja omogućava olakšanu implementaciju sistemabaziranih na računarskoj viziji. Ova biblioteka je razvijena pod BSD licencom što omogućava visok nivo otvorenosti i slobode korišćenja, menjanja i distribuiranja. Ova biblioteka ima ogromnu primenu u industriji i najčešće je prvi i jedini izbor većine onih koji se bave ovom i srodnim oblastima. Dostupna je na mnogim platformama kao što su *Linux*, *Windows*, *OSX*, *iOS* i *Android*.

Iako je pre svega namenjena kao C++ biblioteka, postoji interfejs za *Python* i *Java*-u.

Zbog svoje efikasnosti i brzine, primenjuje se u sistemima koji zahtevaju brz odziv, odnosno u sistemima koji rade u realnom vremenu. Ovo je omogućeno zbog mogućnosti korišćenja optimizacija za različit hardver. Takođe ima podršku za višeprocensnu obradu što omogućava iskorišćenje punih resursa

sistema koji mogu da vrše obradu paralelno.

Zbog ogromnog broja mogućnosti i funkcija, *OpenCV* je podeljen u više modula, a svaki modul ima svoju namenu. Kombinacijom ovih modula je moguće napraviti zaista korisne aplikacije. U nastavku teksta vam dajemo objašnjenje za neke najbitnije module.

Osnovni *core* modul

Kao što mu i samo ime kaže, ovo je osnovni modul u biblioteci. Ovaj modul sadrži neke osnovne strukture i funkcije za biblioteku.

Ovde se nalaze strukture za reprezentaciju slike kao matrice piksela, boja, tačaka, vektora i drugog. Od funkcija tu su osnovni algoritmi za operacije nad nizovima, grafovima i vektorima.





Modul za upravljanje grafičkim interfejsom

Ovaj modul u biblioteci ima naziv *highgui*. Pored toga što ima mogućnosti upravljanja grafičkim interfejsom, on ima i mogućnosti učitavanja slika iz memorije, tvrdih diskova, kamere i ostalih uređaja u strukture koje su navedene u *core* modulu.

Tu su mogućnosti i prikazivanja slika kao i neki dodatni grafički elementi koji pružaju lako uklanjanje grešaka (eng. *debugging*).

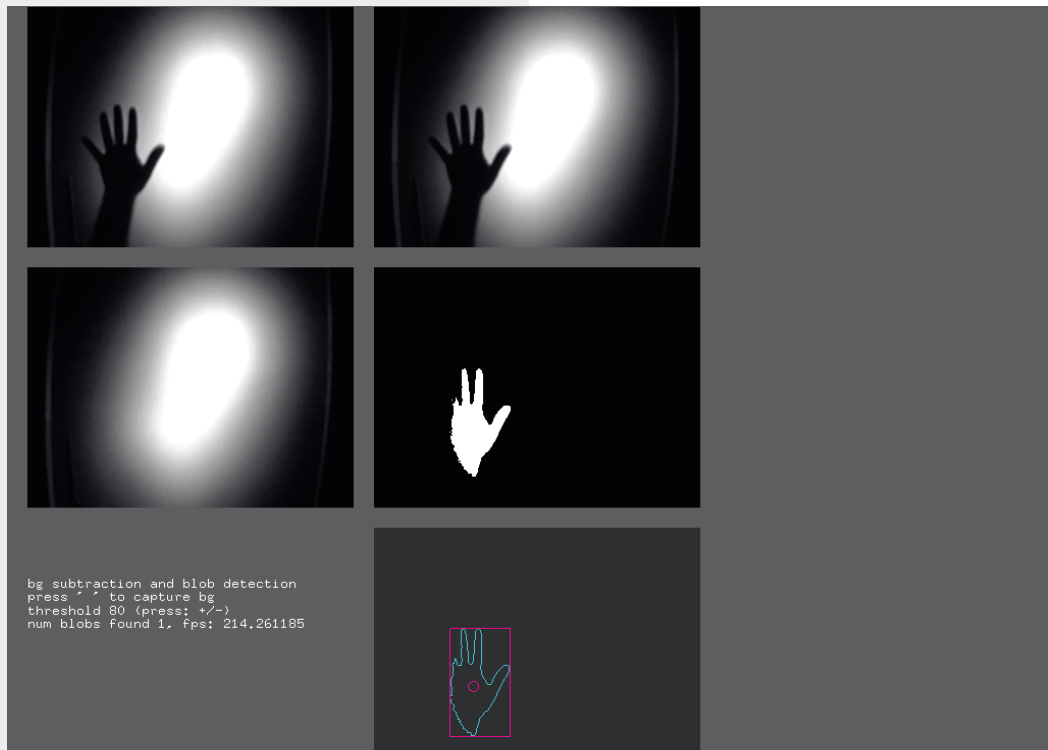
Modul za obradu slika

Ovaj modul se u biblioteci naziva

imgproc. Služi za obradu slike. U ovom modulu se mogu pronaći razni algoritmi kao što su *blur* efekat, Hafova transformacija, menjanje veličine slike, rotacija, adaptivni trešhold *threshold*, *floodfill*, izračunavanje histograma i mnogi drugi.

Ostali moduli

Pored gore navedenih modula, postoje još mnogi korisni moduli kao što su video moduli, moduli za kalibraciju, moduli za detekciju specifičnih objekata i moduli za učenje. Ovde je potrebno napomenuti da postoje i *non-free* moduli koji nisu besplatni.



Dokumentacija

Ako ste mislili da je ovo previše komplikovano, razmislite ponovo! Možemo slobodno da kažemo da je dokumentacija za ovu biblioteku jedna od boljih dokumentacija, ne samo u sferi slobodnih razvojnih biblioteka već i globalno. *OpenCV* nudi kompletan opis svih modula i funkcija kao i imena algoritama koji se koriste u datoj funkciji. Nije redak slučaj da naiđete na primere koda gde se koristi data funkcija, a i često se mogu videti slike koje predstavljaju demonstraciju rezultata neke funkcije. Pored ovoga, dostupni su i *OpenCV* tutorijali kao i priručnik koji se može preuzeti u *PDF* formatu ili čitati onlajn.

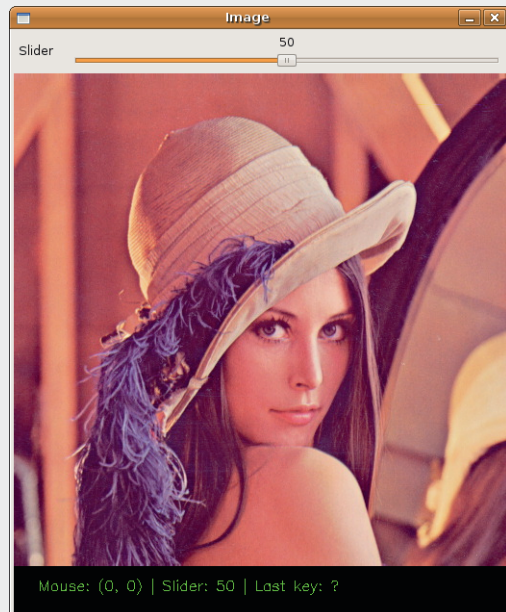
OpenCV za *Android* platformu

OpenCV ima svoj servis za *Android* platformu koji omogućava deljenje *OpenCV* binarnih dinamičkih biblioteka između aplikacija pa time aplikacije štede memoriju. Takođe su dostupne razne optimizacije za različite platforme. *OpenCV Android* upravnik jedostupan u *Googleovoj* prodavnici aplikacija i besplatan je. Po instalaciji ove aplikacije dobićete mogućnost da instalirate *OpenCV* za vašu platformu. Posle ove instalacije moći ćete da koristite vaše aplikacije bazirane na ovoj biblioteci.

Ko je Lena?

Ako ste pregledali *OpenCV* dokumentaciju ili ako ste pročitali bilo kakav tutorijal iz ove biblioteke, sigurno ste primetili da se često pojavljuje slika

devojke koja je prikazana na slici 2. Nije redak slučaj da se neko zapita ko je ona zapravo i zašto se toliko puta pominje. Ova sekcija u ovom tekstu je namenjena da se malo zabavimo, ali i da ispričamo zanimljivu priču.



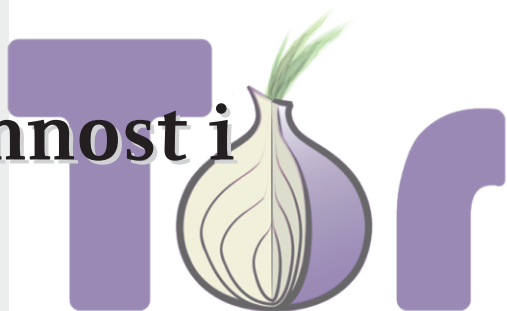
Devojka sa slike je *Lena Söderberg*. Bila je model za *Playboy* i ova slika vuče poreklo iz davnih sedamdesetih godina kada jebilo dosta teško doći do skenirane fotografije. Razvojni tim koji se bavio algoritmima za kompresiju slike i njenu obradu, uspeo je da dođe do ove slike u digitalnoj formi i na njoj testirao svoje algoritme. *Lena* je zbog ovoga nazvana prvom devojkom internetai bila je gost na pedesetoj *IS&T* konferenciji 1997. godine koja se bavi računarskom vizijom i na njoj se predstavila.



Tor mreža:

Online anonimnost i sigurnost

(2. deo)



Autor: Petar Simović

Tor, iako jedan od najsigurnijih načina komunikacije u današnjim vremenima neprekidnog hladnog sajber rata i sve prisutnije masovne špijunaže kojih smo u današnje vreme sve svesniji, ima i neka svoja ograničenja i potrebno je tačno znati šta nam to *TBB* omogućava, a šta ne.

Prvo, ako se koristi samo *TBB* kao alat za anonimno surfovanje netom na nekom operativnom sistemu (pretpostavićemo da je to *Linux* jer inače nema mnogo smisla, bar što se privatnosti tiče), onda valja znati da *Tor* koristi samo određen broj portova (od mogućih $65536=2^{16}$ tj. 0-65535) za sigurnu komunikaciju sa anonimnom mrežom, a da su ostali portovi „izloženi” na milost i nemilost ostalim programima sa mogućim bagovima koje neko može da otkrije i počne da ih zloupotrebljava.

Druga stvar je to što ne mora da se bude, bog zna kakav, haker da bi se shvatilo da običan program za snimanje pozadine *desktopa*, ako zaobiđe

sigurnosne provere operativnog sistema kroz zloupotrebu njegovih eventualnih propusta, može u potpunosti da obesmisli upotrebu *TBB* samim snimanjem aktivnosti *desktopa* koje bi slao napadaču i time izbegne svu silnu enkripciju *Tor* mreže. Naravno, ovo su ekstremni slučajevi, jer se najčešće mogu izbeći redovnim ažuriranjem sistema, ali nisu nemogući jer su izvodljivi i česti, ukoliko je meta iz nekog razloga važna i ima pristup poverljivim podacima (novinari u neliberalnim državama, političari, špijuni i drugi).

Ako se osvrnemo na smisao poslovice: „Bolje sprečiti, nego lečiti.” i „Znanje je moć.”, zaključićemo da, kako sada znamo za ovu mogućnost napada - možemo od njega da se efikasno odbranimo upotrebom operativnih (*open-source*) sistema u kojima je glavni akcenat na anonimnosti i u čiju grupu spadaju *Whonix* (<http://goo.gl/b1swp>) i *Tails* (<http://goo.gl/isQIn>). *Tails* se zasniva na *Debianu*, a sve komunikacije se primoravaju da koriste *Tor* mrežu.

Ovde je važno napomenuti da nije

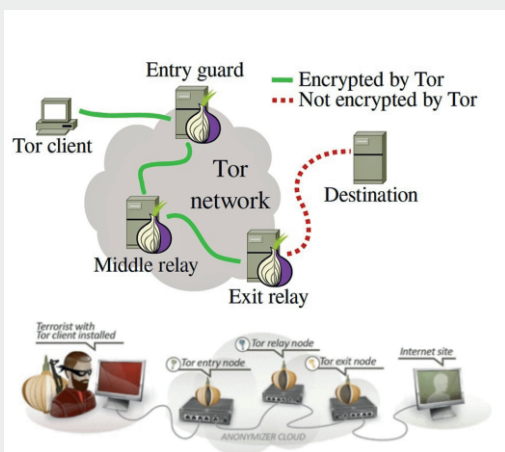


pametno koristiti programe koji se zasnivaju na P2P komunikaciji tj. *torrent* programe (*utorrent*, *bittorrent* i ostali) preko *Tor* mreže. To je zato što su moderni trakeri (*trackers*) dizajnirani da koriste *UDP* protokol koji ignoriše *proxy* koji koristi *Tor*, a opet *Tor* podržava samo *TCP* protokole, pa je šteta višestruka jer ne samo da niste prikriili identitet i *IP* adresu, već prouzrokuje ranjivosti *Tor* mreže i vašeg sistema. Ranjivost je u tome što ti *torrent* programi jednostavno uzmu vašu *IP* adresu i prekopiraju je u traker koji se šalje drugima u P2P mreži. *Tor* i ovde obavlja svoj deo posla te niko ne zna odakle je *IP* i odakle se podaci šalju, ali to je daleko od anonimnosti koju biste želeli (više na: <http://goo.gl/Es5xL>).

Osim ove, *Tor* ima i drugih negativnih osobina koje se možda ne vide na prvi pogled, ali se brzo mogu uočiti iz dizajna. Informatičarima je sigurno poznata rečenica: „Svaki sistem je jak koliko je jaka njegova najslabija karika.” *Tor* nije nikakav izuzetak. Naime, slaba tačka *Tora* su njegovi izlazni čvorovi (*exit nodes*) koji dekrriptuju sadržaj, tako da izlazni čvorovi mogu imati informaciju o vama, ako se ne koristi takozvana *end-to-end* enkripcija

kao, na primer, *SSL* (više na <http://goo.gl/YUqqq6>).

Imajući ovo na umu, ne čudi što svetske sigurnosne organizacije poput *NSA* ili *GCHQ* instaliraju hiljade, pominje se podatak od devetnaest hiljada *Tor* releja i izlaznih čvorova u vlasništvu *NSA*, pokušavajući da prate i analiziraju sav protok kroz *Tor* mrežu, kontrolišući najveći deo izlaznih čvorova koje se malo ko usudi da pokrene zbog mogućih sukoba sa zakonom i problema u koje može da zapadne zbog različitog sadržaja koji prolazi neenkriptovan jednim delom kroz izlazni čvor.

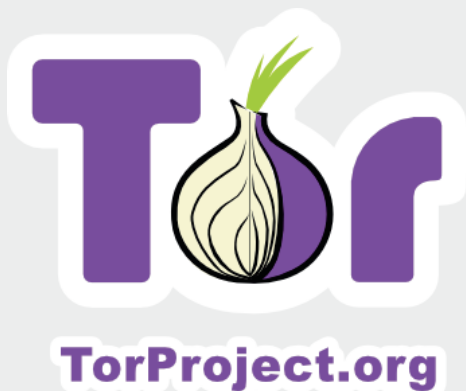




Za sada ovo nije veliki problem, te nikako ne pozivamo da se odustane od Tor mreže, jer dokle god se pridržavate navedenih pravila, koristite SSL i uopšte znate šta radite, enkripcija je dovoljno jaka da niko neće moći da ugrozi vašu anonimnost, a NSA bi morala da tačno zna sve čvorove kroz koje vaša komunikacija prolazi da bi otkrila vaš identitet. Sav trud koji NSA ulaže u „razbijanje” Tor mreže govori o uspešnosti i sigurnosti same mreže.

Brojni eksperimenti i naučni testovi pokazuju da je NSA u mogućnosti da prisluškuje veoma mali deo Tor mreže, ali da nikada neće biti u stanju da u realnom vremenu prati i dešifruje sav saobraćaj. Svi pokušaji ugrožavanja Tor komunikacija na nivou mreže su propali, pa je za očekivati i druge sofisticiranije načine špijuniranja u bliskoj budućnosti.

U sledećem delu ćemo videti nekoliko zanimljivih programa za korišćenje u okviru Tor mreže i kako možemo da poboljšamo sigurnost našeg sistema tamo gde nas Tor ne štiti.



Pregled popularnosti GNU/Linux /BSD distribucija za mesec mart

Distrowatch

1	Mint	3630<
2	Ubuntu	2436>
3	Debian	1783<
4	openSUSE	1384>
5	Fedora	1262=
6	Mageia	1196<
7	Arch	1170>
8	elementary	1138>
9	Puppy	1057>
10	Manjaro	896<
11	SparkyLinux	886>
12	Zorin	864>
13	PCLinuxOS	828<
14	CentOS	789<
15	Lubuntu	750>
16	CrunchBang	655<
17	GhostBSD	618>
18	Kali	601<
19	antiX	584>
20	Simplicity	582>
21	Xubuntu	578<
22	Bodhi	564<
23	Kwheezy	533>
24	FreeBSD	532<
25	Ultimate	520>

Pad <

Porast >

Isti rejting =

(Korišćeni podaci sa Distrowatcha)



Android ispod haube



Autor: Nikola Hardi

Da li je *Android* istovremeno i *Linux*?

Kao što već sigurno dobro znate, *Android* je jedan od najpopularnijih operativnih sistema za pametne telefone i slične uređaje i zasnovan je na *Linux* kernelu. Razvija ga kompanija *Google* koja je u saradnji sa proizvođačima mobilnih telefona osnovala grupu za razvoj *Androida*, pod nazivom *Open Handset Alliance*. Ova grupa je više formalnog karaktera i do sada nije imala značajnog uticaja na razvoj *Androida*. *Android* nije čistokrvni *Linux* i jako se razlikuje od klasičnih *Linux* sistema, a da li imamo pravo da ga svrstavamo u *Linux* operativni sistem ili ne, procenite sami. U duhu teksta „*Linux* unatraške” i nekih budućih tekstova o *Linux* kernelu i operativnim sistemima, u ovom tekstu ćemo se baviti tematikom koja je na nekom nižem nivou, dalje od korisnika i bliže hardveru.

Gde je tu *Linux* i šta je tačno *Android*?

Jedan *Android* uređaj karakterišu veliki ekran osetljiv na dodir, aplikacije pisane u *Java*-i, moderan grafički korisnički interfejs, prodavnica za aplikacije i drugo. *Android* projekat se bavi pravljenjem programske podrške koja pruža ove mogućnosti. Napravljen je *API* koji programerima pruža način da iskoriste *Android* uređaj. Takođe se napravljeni dodatni sistemi koji omogućuju izvršavanje *Android* aplikacija i pružaju razne servise. U takve sisteme spadaju *Dalvik* virtuelna mašina, *Android API* i biblioteke, sistem za kreiranje i upravljanje aplikacijama kao i razne izmene u *Linux* kernelu. Dakle, *Android* uređaji ne koriste čistokrvni *Linux* kernel, a o svim ovim temama više detalja možete da pročitate u nastavku teksta.

Dalvik

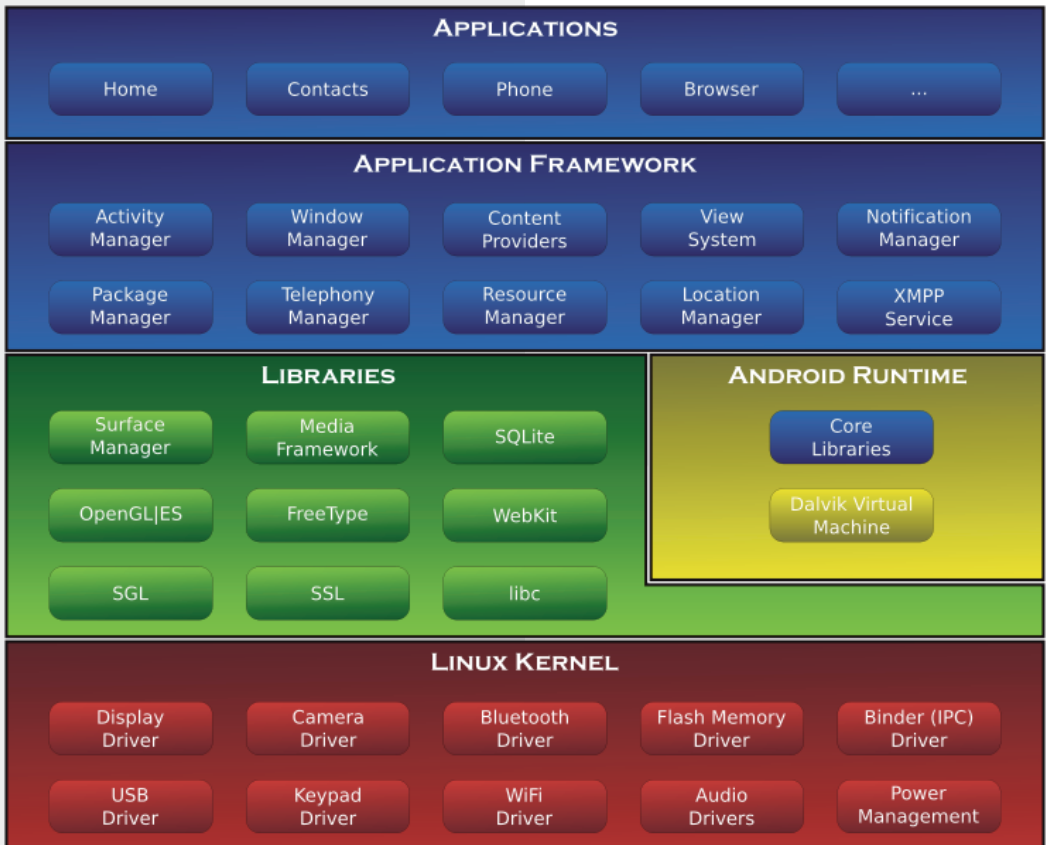
Dalvik je *Java* virtuelna mašina koja je zadužena za izvršavanje *Java* kôda na

Androidu. Pre svega, *Java* je jedan vrlo specifičan programski jezik. *Java* izvorni kôd se prevodi u tzv. *Java bytecode* koji potom interpretira i izvršava *Java* virtuelna mašina. Pitanje zašto je izabrana baš *Java*, ostavićemo za neko drugo mesto i vreme, ali ostaje pitanje zašto nije izabrana neka od već postojećih implementacija *Java* virtuelnih mašina. Za ovo postoji nekoliko odgovora, a dva najvažnija su *Oracle* i prenosivi uređaji.

Java i *Solaris* postali su vlasništvo kompanije *Oracle*, nakon što je *Oracle* 2009. godine otkupio kompaniju *Sun*

Microsystems. Ovo je dovelo do mnogih pravnih sporova i problema oko licenciranja koje su tvorci *Androida* želeli da izbegnu, a dovelo je i do drugih bitnih prekretnica u izboru tehnologija, kao što je izbor *Harmony Java* standardne biblioteke koju razvija *Apache* a ne *Oracle*. Drugi bitan faktor je činjenica da je *Android* namenjen prenosivim uređajima, što podrazumeva ograničene resurse po pitanju memorije, procesorske snage i baterije, pa je *Dalvik* stvoren baš za ovakve uslove rada.

Dalvik se od klasične *Java* virtuelne mašine razlikuje po nekim formalnim





stavkama oko implementacije i arhitekture, ali i po formatu datoteka koje izvršava. Među virtuelnim mašinama postoje dve klase, one čiji se rad bazira na *stacku*, i one čiji se rad bazira na registrima. Neki primeri *stack based* virtuelnih mašina su klasična *Java* virtuelna mašina i *.NET* virtuelna mašina. *Dalvik* je *register based* i sasvim je drugačije organizovan. Osim po arhitekturi, razlikuje se i po tome što se *Java* kôd prevodi u *.class* datoteke, a *Dalvik* radi sa *.dex* datotekama. Neke od važnih osobina *.dex* datoteka su da omogućuju bolju kompresiju, manje su i smanjeno je ponavljanje podataka unutar njih u odnosu na *.class*.

Bionic i Harmony

Bionic je zamena za standardnu biblioteku *C* programskog jezika koju je razvio *Google*, a zasniva se na *BSD*-ovoj biblioteci. *Bionic* se razvija odvojeno od *Androida*, ali se vrlo čvrsto oslanja na *Linux* kernel. Objavljen je pod *BSD* licencom za razliku od *GNU*-ove varijante (*glibc*) i time je razvoj formalno slobodniji. Ovo naravno možemo da protumačimo i drugačije, jer je *BSD* licenca, u odnosu na *GPL* familiju licenci, zapravo pogodnija za izbegavanje objavljivanja kôda i izmena. Ovo jeste vid slobode, ali da li je prava stvar ili nije, to ostaje za neku drugu diskusiju. Osim licence, *Bionic* se razlikuje i po tome što je manji od *glibc* i optimizovan je za rad na sporijim procesorima. Iz sličnih razloga zbog koji su nastali *Dalvik* i *Bionic*, izabrana je *Harmony* biblioteka. Pored razlike u licenci, za *Harmony* je bitno navesti da

je projekat pod *Apache* fondacijom i da je napisana iz početka. Vrlo je slična sa zvaničnim *JavaSE* i *JavaME* standardima, ali ne poklapa se sa njima potpuno.

Zygote

Android koristi poseban sistem za pokretanje sistema i *Android* aplikacija. Razlika u odnosu na standardne *init* sisteme je u tome što se rad *Zygota* ne završava po pokretanju svih servisa, već je *Zygot* aktivan, te i dalje kreira procese za novopokrenute *Android* aplikacije.



Servisi

Pošto je *Android* namenjen prenosivim uređajima, najčešće mobilnim telefonima i tabletima, on pruža i niz servisa za upravljanje periferijom. Servisi su različiti, od onih koji su zaduženi za razna obaveštenja, tajmere, alarme i drugo, do onih za upravljanje pokrenutim aplikacijama. Način pokretanja i upravljanja aplikacijama je takođe vrlo karakterističan. Dok na drugim sistemima imamo samo jednu aplikaciju koja je trenutno pokrenuta ili imamo pokrenute samo one aplikacije koje trenutno koristimo, *Android* koristi način koji funkcioniše po principu *stacka*. Sve aplikacije koje smo pokrenuli, ostaju aktivne u pozadini i uvek se možemo vratiti na njih, a kada ponestane resursa, tada se stanje onih najstarijih aplikacija čuva i one se zaustavljaju.

Android biblioteke

Android je vrlo brzo stekao popularnost kako među krajnjim korisnicima, tako i među programerima, a za to je u velikoj meri zaslužan i deo *Androida* koji programerima omogućava da na što jednostavniji način upravljaju raznim mogućnostima telefona. Ove biblioteke se jednim imenom zovu *Android API* i obuhvataju različite nivoe, od onih za definisanje izgleda i funkcionalnosti aplikacija do onih za komunikaciju sa drugim aplikacijama, servisima i hardverom.

Android kernel

Ovo je za mnoge bolna tačka. Da li je *Android* stvarno *Linux* ili baš i nije? Ne postoji tačan odgovor, a mi možemo opisati problem. Strogo gledano, *Linux* je samo kernel, dok je ostatak sistema proizvod *GNU* projekta. Dakle, možemo da pređemo na pitanje da li *Android* koristi *Linux* kernel? Da, *Android* koristi *Linux* kernel, s tim što su u potpunosti izmenjeni njegovi pojedini delovi. Ove izmene su dostupne i čak se nalaze u skladištu programskih paketa *Linux* kernela (za kernel hakere, `/drivers/staging/android`). Za sada ne postoje inicijative da ovaj kôd postane sastavni deo standardnog *Linux* kernela. *Linux* kernel na koji su primenjene ove izmene, često se naziva „androidizovan” *Linux* kernel, a procesom „androidizacije” od standardnog *Linux* kernela proizvoljne verzije možete napraviti kernel za *Android*, ukoliko vam je to iz nekog razloga potrebno.

Binder

Binder je zadužen za komunikaciju između procesa (*IPC - inter process communication*). Vrlo je čudno što se našao u samom kernelu. Mnogo češći slučaj je da se ovakva podrška implementira na višim nivoima, međutim, pošto je reč o sistemu sa ograničenim resursima i pošto se koristi za svakakve vidove komunikacije, ovakav potez je opravdan. Neki od primera za šta se sve koristi *binder* su: upravljač aplikacija i procesa, iscrtavanje na ekranu, unos podataka, pokretanje koda u drugim aplikacijama i drugo. *Binder* je u samom srcu *Androida* i bilo bi vrlo teško



zameniti ga na drugačiji način, barem u datom okruženju.

Ashmem

Na sistemima u kojima se izvršava više procesa istovremeno (*multitasking*), javlja se pojam deljenje memorije (*shared memory*). *POSIX* standard definiše deljenu memoriju, međutim, *Android* koristi drugačiji pristup problemu i ova implementacija se zove *ashmem* (*Android shared memory*). Naravno, i ovde je reč o štednji resursa, a glavna razlika između *POSIX SHM* i *ashmem* ogleda se u tome što se *ashmem* drugačije ponaša u sistemima sa malo radne memorije, a između ostalog, pruža i jednostavniji način za rad sa datotekama.

Alarm

Treća bitna karakteristika „androidizovanog” kernela su alarmi. Ovde nije reč o aplikaciji koja uznemirava vaše snove, već budi procese. U teoriji operativnih sistema postoji pojam signala kojima možemo obavestiti određeni proces o nekom događaju. Recimo, kada u terminalu pritisnemo kombinaciju tastera **CTRL + C**, tada u stvari trenutnom procesu šaljemo signal koji će proces da obradi i uradi nešto, najčešće će se zaustaviti. Drugi primer je signal *SIGKILL* kojim možemo da poručimo da se zaustavi, najčešće na destruktivan način. Sličnu funkcionalnost predstavljaju alarmi i *SIGALARM*. Program može da obavesti sistem da želi da se zaustavi i poruči od sistema buđenje u nekom trenutku. Tada taj proces prelazi u stanje čekanja, a po isteku zadatog vremena operativni

sistem će tom procesu poslati signal *SIGALARM* i na taj način ga probuditi, odnosno nastaviti njegovo izvršavanje. Problem se javlja kada uređaj pređe u stanje pripravnosti (*standby*). U običnom *Linux* kernelu, tada vreme staje, a time se obustavlja i računanje vremena za ovakve alarme. Kada *Android* uređaj pređe u stanje pripravnosti (kada pritisnemo dugme za zaključavanje, na primer), o ovim alarmima se i dalje vodi računa.

Još nekoliko razlika

Android ne koristi *X* za iscrtavanje grafičkog okruženja, već se za to koristi direktno *OpenGL ES (GLES)*, verzija *OpenGL*-a za *embedded* sisteme. *GLES* se razvija odvojeno od *Androida* i za njegov razvoj je zadužena *Kronos* radna grupa, koja razvija *OpenGL*, *OpenCL* i druge nama već poznate biblioteke.

Mobilni telefoni koji koriste *Android*, najčešće u sebi imaju čip (*SoC System on Chip*) sa *ARM* procesorom i drugim delovima, međutim za upravljanje *GSM/GPRS* komunikacijom postoji poseban čip. Na ovim čipovima, koji se u *Android* svetu obično nazivaju radio-čipovima, pokrenut je poseban operativni sistem koji ostavlja ograničene mogućnosti za rad. O ovim operativnim sistemima i čipovima se ne zna mnogo, podaci o njima su u vlasništvu njihovih proizvođača, a ono na šta smo želeli da skrenemo pažnju, jeste da ne možemo tek tako da se igramo sa *GSM* signalima jer je to odvojen sistem sa jasno definisanim namenom.



Razvoj Androida i licence

Android razvija Google i objavljen je pod ASL (*Android Software Licence*). Kôd do sada objavljenih verzija, javno je dostupan i slobodan za izmene na adresi android.google.com. Kôd trenutne verzije koja je u fazi razvoja, dostupan je samo unutar kompanije ili drugim kompanijama sa kojima je potpisan poseban ugovor. U istom krugu ljudi ostaju i odluke o razvoju Androida. Ovo jeste neobičan princip i nije karakterističan za svet slobodnog softvera, čak mnogi dovode i u pitanje da li je ovo uopšte slobodan softver. Po licenci jeste, a po načinu razvoja nije. Ni zajednica niti bilo ko izvan Googlea ne može ni na kakav način da utiče na razvoj ovog projekta. Opet, lepo od njih što su nam omogućili da zavirimo u kôd i nastavimo tamo gde su oni stali.

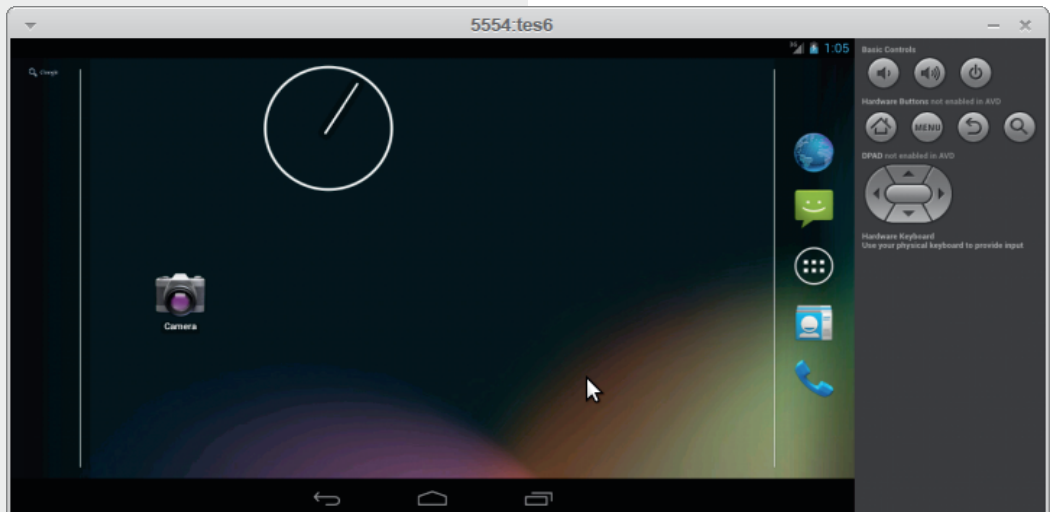
Alternative

Android nije jedini operativni sistem za

prenosive uređaje i mobilne telefone koji je zasnovan na *Linuxu*. U proteklih deset godina imali smo priliku da vidimo projekte koji su se neslavno završili, a u nešto skorijoj istoriji imamo i nove projekte. U prvoj kategoriji su nažalost *OpenMoko*, *MeeGo*, *Maemo* i drugi. Lepa vest je da su *Meego* i *Maemo* nastavili da žive u projektu koji se zove *Sailfish* koji razvija finska kompanija za telefon *Jolla*. Druge dve popularnije alternative su *FirefoxOS* i *Ubuntu* za telefone.

Za kraj

Ovo je bio jedan kratak pregled Androida i retka prilika da pročitate o detaljima sistema koji najverovatnije imate na svom telefonu. Sjajna knjiga na ovu temu je *Embedded Android*, a uvek možete da pitate i redakciju časopisa za više detalja ili pomoć. Na vama je sada ostalo da izvagate da li je *Android Linux*, ili baš i nije.





OpenWay

U petak, 25. travnja (aprila) 2014. godine održaće se Konferencija *open source* rješenja u Novom Mestu u Sloveniji. Novo Mesto postaće na jedan dan središte *open source* znanja, ideja i rješenja s područja *IT*-a, kreativnosti, *3D* printera i još mnogo toga.

Ujutro će se na Fakultetu za informacijske studije održati dvije besplatne radionice namijenjene svima koji su željni tehničkoga znanja. U poslijepodnevним satima predavanja će održati zanimljivi predavači iz svijeta otvorenog koda.

Na kraju događaja očekuje nas ekskluzivni *Firefox party*, a uz proljetne melodije, dobru hranu s roštilja i hladno pivo završit ćemo cjelodnevni put po *open source* rješenjima.

Sudjelovanje na konferenciji je besplatno. Detaljni program, popis predavača te postupak prijave na konferenciju možete pronaći na *web* stranici konferencije: <http://openway.si>



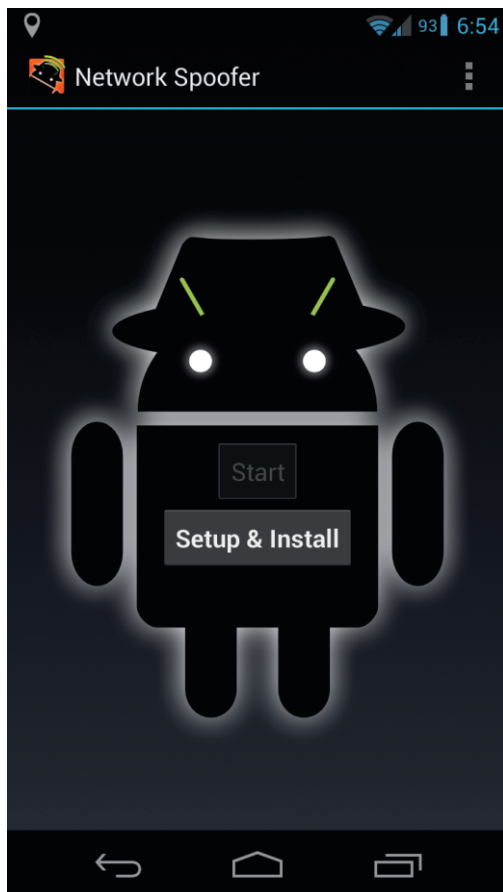


Network Spoofer

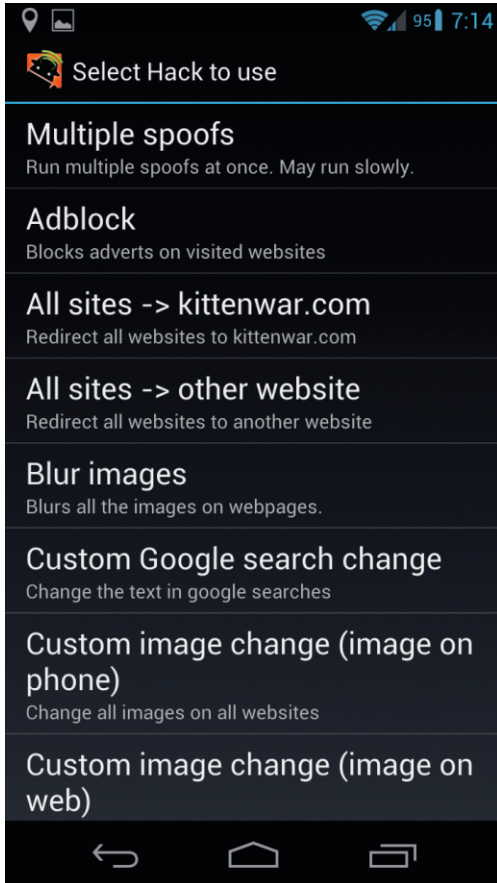
Autor: Gavriilo Prodanović

Kada rutujemo svoj telefon, otvaramo vrata širokih mogućnosti koje nam nudi *Linux*. Možemo da overklokujemo *CPU* telefona, da pišemo po sistemskoj particiji, stavimo neoficijelni *ROM* i još mnogo toga. Na *Androidu* je slabo zastupljen terminal u praksi jer je smanjena *touch* tastatura nezgodna za bilo kakav koristan rad. Zbog toga često koristimo grafičke aplikacije koje će da „kucaju” umjesto nas u terminal i tako olakšaju nam muke. Jednu od takvih predstavimo upravo sada, a ime joj je *Network Spoofer*.

Network Spoofer će pomoći da na zabavan način testiramo bezbjednost mreže i pokažemo kako su male *wifi* kućne mreže ranjive na njuškanje saobraćaja. Nakon instalacije potrebno je podesiti aplikaciju da bi mogla normalno funkcionisati. Podešavanje se zapravo sastoji od izvršavanja jednog preuzimanja oko 100MB, koji je *debian image loop* fajl kompresovan *gzip* kompresijom. Nakon što se preuzimanje završi, *image* će biti otkompresovan na *SD* karticu i u trenutnoj verziji teži 420MB.



Nakon podešavanja možete da startujete okruženje i izaberete željene *spoofove* koje hoćete da koristite. Svi *spoofovi* rade uglavnom isto pod



haubom: *http* saobraćaj se preusmjeri kroz vaš pametni telefon, onda se mjenja *source html* fajla i izmjenjen fajl se dostavlja žrtvi. *Spoofovi* su svi u osnovi bezazleni, kao što je mjenjanje teksta i slika, preusmjeravanje sa jednog sajta na drugi ili preusmjeravanje svakog, na primjer na naš blog, gdje im možemo objasniti kako da se izbave od ove „nesreće”. Za podešavanje *spoofa* potrebno je izabrati ruter iz ponuđene liste, a poslije toga izabrati žrtvu po *IP* adresi ili odlučiti da svi na mreži budu napadnuti. Za prvi *spoof* ćemo pred-

staviti *trollface*. Njegov efekat je jednostavan: zamjenite sve slike sa stranica sa slikom dobro poznatog *trollfacea*. Od ostalih *image spoofova* tu su *blur image* koji muti sliku i *flip image* koji napravi odraz u ogledalu svake slike. Za ova dva *spoofa* koriste se *ImageMagic* paket da bi se slike odradile u letu. Moguće je sve slike zamjeniti sa nekom slikom iz telefona ili sa slikom na vebu. Postoji *youtube spoof* koji omogućuje da svaki otvoreni video zamjenite sa nekim svojim, a ako vam je mrsko da lično izaberete, možete izabrati ponušeni „Rickroll”. Kao klasika na ponudi je zamjena teksta na vebu sa nekim našim rečenicama. Postoji i jedan koristan *spoof* pod imenom *Adblock*, a omogućuje nam da budemo spasilac od dosadnih reklama. Kao šlag na tortu dolazi opcija koja omogućuje korišćenje više *spoofova* u isto vrijeme.

Jedna od stvari što nam se nije svidjela, jeste opcija da se izaberu svi na mreži za žrtve ili samo jedna žrtva. Ne postoji opcija da izaberete dvije ili tri žrtve. Nekada prije spektar sajtova za napad je bio mnogo širi, a danas je onemogućeno da mnoge sajtove napadamo jer koriste *https* protokol, a aplikacija ne posjeduje ništa da se sa ovim izbori. Tako na primjer *spoof* pod imenom *Google search change* više nije funkcionalan jer *Google* podrazumjevano koristi *https* protokol, a *Facebook* je takođe uveo *https* protokol kao podrazumjevan što smanjuje efekat *image spoofova* kao što je *troll face*. Takođe treba biti i oprezan gdje se koristi alatka jer na mrežama sa velikim saobraćajem postoji mogućnost da napravite kolaps i

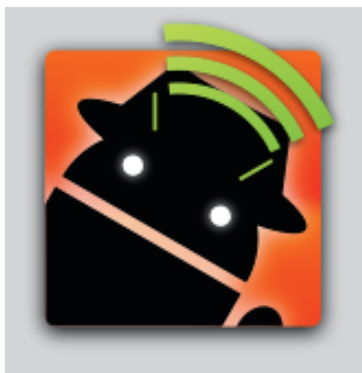


Alatka ne postoji na *Play Storeu*, pa će biti potrebno da je ručno preuzmete sa *SorucaForge.net* sajta, što vjerovatno nije omogućilo popularnost u širokom krugu.

Za kraj dodaćemo nekoliko načina kako da se zaštitite:

- Za početak zaključati svoj *wifi*.
- Najjednostavnije je da eksplicitno kucate **https://** prije linka sajta gdje je to moguće. Protokol šifrovan *SSL*-om ne može lako da se njuši i mjenja kao što smo već pomenuli.
- Dodajte statičku *arp* adresu za vaš ruter na vašem *Linux* uređaju: **arp -s <router_ip> <router_mac>**
- Provjerite da li vaš ruter podržava neki od načina zaštite za ovakav tip ili sličan tip napada, kao što je *DXCP snooping* na primjer.
- Posjedovati zdrav razum je najpouzdanija zaštita. Nemojte koristiti računare koji su vam važni na javnim mrežama.

svima zablokirate pristup internetu. Ispod haube aplikacija mountira *ext2 loop image* u kojoj se nalazi *Debian* i kasnije preko *chroota* pokreće *Squid HTTP Proxy*, koji igra glavnu ulogu u zamjeni sadržaja stranice. Za hvatanja žrtava koristi *arp spoof* alatku iz paketa *dsniff*, a uključen je i *lighttpd*. Za preusmjeravanje saobraćaja kernel *Androida* mora biti kompajliran sa podrškom za *iptables*, što većina danas ima podrazumjevano uključeno u sebi.





open**SUSE**

Linux for open minds

open**SUSE** Conference

2014

The Strength to Change

April 24-28
Dubrovnik, Croatia



5 Days of Awesomeness

conference.opensuse.org