

April 2014.

LIBRE!



Časopis o slobodnom softveru

broj

24



FLOSS na poslu



17. april 2014.

Canonical je objavio
Ubuntu 14.04 LTS



26. april 2014.

Peta revizija *Debian 7*
Wheezy-ja je dostupna za
preuzimanje.



Creative Commons Autorstvo-Nekomercijalno-Deliti pod istim uslovima.



Profesionalna upotreba FLOSS-a

Pred vama je dvadeset četvrti, po malo jubilarni broj LiBRE! časopisa. Ovo znači da se družimo već pune dve godine. Nadamo se da vam nismo dosadili. Ovom dvadeset četvrtom broju nisu išli na ruku silni protekli praznici, koji su uticali da ovaj broj malo kasni. Izvinjavamo se zbog toga.

Tema ovog broja je profesionalna upotreba FLOSS-a. Mišljenja smo da je to pravi put za popularizaciju FLOSS-a. U današnjem svetu novac je najveći motivator. Za razliku od kućnog korisnika, firme ne smeju da koriste piratski softver. U ovom broju pokazujemo kako firmu opremanje jednog računara osnovnim vlasničkim softverom košta oko 650 evra, što znači da postoji 650 razloga za prelazak na FLOSS rešenje. Čak i kad vlasnik firme unajmi stručno lice za implementaciju FLOSS-a, to ga neće koštati kao licence za ekvivalentni vlasnički softver.

Mi koji odavno koristimo FLOSS, znamo da FLOSS rešenja barem u običnim kancelarijskim poslovima mogu uspešno da zamene vlasničke softvere. Kroz seriju prezentacija Privredna komora Beograda pokušava privrednike da uputi u upotrebljivost FLOSS-a u poslovnom okruženju.

Trend štednje uz pomoć FLOSS-a su usvojile ili usvajaju administracije evropskih zemalja. Ako to mogu oni koji su mnogo bogatiji od nas, trebalo bi da i mi razmislimo o uštedama kroz FLOSS. I nije samo štednja motiv za prelazak na FLOSS, već i skora potreba komunikacije sa razvijenim svetom koji ubrzano prelazi na otvoreni dokument kao standardni elektronski dokument.

Prelazak firmi na FLOSS rešenja otvara mogućnost pokretanja biznisa na bazi FLOSS-a. O tome govori naš novi serijal „Vaš posao, *open-source* posao”. FLOSS jeste besplatan, ali sekundarni domeni FLOSS-a (usluge, edukacija, implementacija i drugo) mogu da se naplate jer će uvek biti onih koji više vole da plate, nego da uče. Ovo otvara prostor za razvoj IT industrije na FLOSS osnovama, što kao rezultat daje novi zamajac preduzetništvu i novom zapošljavanju.

Do sada je bilo pokušaja pojedinaca koji su svoje računare na poslu opremali FLOSS-om da bi uskladili sa onim na čemu rade kod kuće. Logičniji je prenos FLOSS-a sa posla na kućne računare koji bi bio uslovljen završetkom nekih poslova kod kuće ili



prosto zbog vežbanja rada na *FLOSS*-u zbog postizanja veće efikasnosti na poslu.

Mnogo je razloga zašto popularizacija *FLOSS*-a treba da bude prvenstveno usmerena prema privredi a ne prema pojedincima, zato će **LiBRE!** sve više svoj sadržaj usmeravati ka pravnim licima i ukazati im kako se *FLOSS* može upotrebiti u poslovne svrhe.

Ovoga puta, za kraj, osim standardnog poziva autorima i grafičarima da nam se pridruže, upućujemo poziv i programerima da nam se pridruže. Formiramo **LiBRE!** razvojni tim. Za početak ćemo napraviti **LiBRE! Android** aplikaciju i urediti našu internet infrastrukturu, ali ko zna - možda taj razvojni tim počne da radi i za treća lica. Za kontakt, primedbe i predloge koristite našu, već poznatu, internet adresu `libre [et] lugons [dot] org`.

Do čitanja

LiBRE! tim

Moć slobodnog
softvera



Broj: 24

Periodika izlaženja: mesečnik

Izvršni urednik: Stefan Nožinić

Glavni lektor: Aleksandar Božinović

Lektura:

Milena Beran

Jelena Munčan

Maja Panajotović

Aleksandra Ristović

Redakcija:

Aleksandar Stanisavljević

Željko Šarić

Marko Kažić

Veljko Simić

Nikola Hardi

Dejan Čugalj

Goran Mekić

Danilo Đokić

Petar Simović

Romeo Mlinar

Dejan Petrović

Zlatan Vasović

Sandrina Dimitrijević

Aleksandar Todorović

Mihajlo Bogdanović

Gavrilo Prodanović

Aleksandar Brković

Joakim Janjatović

Stefan Stojanović

Marko Novaković

Vladimir Cicović

Željko Popivoda

Saradnici:

Velimir Baksa

Ivan Bulatović

Tamara Đorđević

Vladimir Popadić

Milovan Krivokapić

Bojan Bogdanović

Ladislav Urošević

Grafička obrada:

Dejan Maglov

Ivan Radeljić

Dizajn:

Mladen Ščekić

Zoran Lojpur

Kontakt:

IRC: #floss-magazin na irc.freenode.net

E-pošta: libre@lugons.org

<http://libre.lugons.org>



LiBRE! vesti str. 6

Vesti



Puls slobode str. 8

Ugovor Republike Srbije
sa *Microsoftom* (6. deo) str. 8

Prezentacija:
Linux u praksi str. 14



Predstavljamo str. 19

Serbian 2014 (2. deo) str. 19



VI text editor str. 25

OpenStreetMap str. 28

Kako da...? str. 32

Uvod u programski
jezik C (2. deo) str. 32

Learn C Programming

Oslobađanje str. 37

Nekoliko filozofskih
misli o softveru:
Razvoj softvera str. 26



Slobodni profesionalac str. 42





Vaš posao,
open-source posao (1. deo) str. 42

Internet mreže i
komunikacije str. 44

Tor mreža:
Online anonimnost i
sigurnost (3. deo) str. 44



Sam svoj majstor str. 48

Underscore str. 48



LIBRE! prijatelji





Unreal Engine 4 podržava Linux i SteamOS

4. april 2014.



Unreal je objavio da će Unreal Engine 4.1, koji će biti objavljen za nekoliko nedelja, u potpunosti podržavati SteamOS i Linux.

Koristan link: <http://bit.ly/1s3G7VJ>

Snowden koristio Linux distribuciju za anonimnost

15. april 2014.



Tails

Snowden je koristio Tails Linux distribuciju kako bi ostao anonim na internetu.

Koristan link: <http://bit.ly/1fGKYdA>

Ubuntu novo izdanje 14.04

17. april 2014.

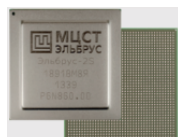


Canonical je izdao novo izdanje najpopularnije Linux distribucije.

Koristan link: <http://bit.ly/1oe1Rfz>

Mikroprocesor Elbrus-4S spreman za serijsku proizvodnju

19. april, 2014.



Mikroprocesor Elbrus-4S spreman je za serijsku proizvodnju, a kao osnovnu platformu za

razvoj ruski inženjeri koriste Linux.

Koristan link: <http://bit.ly/1fZ59OA>

OpenBSD forkovao OpenSSL kao LibreSSL

22. april, 2014.



LibreSSL je fork OpenSSL API-a koji razvija OpenBSD razvojni tim.

Koristan link: <http://www.libressl.org/>

Pokrenut projekat Core Infrastructure initiative

26. april 2014.



Poučena iskustvom sa Heartbleed sigurnosnom greškom u OpenSSL-u, Open-Source zajednica je okupila najveće stručnjake iz oblasti informacionih tehnologija da bi se ove afere u budućnosti sprečile.

Koristan link: <http://bit.ly/1oe1X6F>

Objavljen Debian 7.5

26. april 2014.



Peta revizija Debian 7 Wheezyja je dostupna. Debian 7.5 je poslednje stabilno izdanje popularne distribucije sa bezbednosnim zakrpama i drugim ažuriranjima.

Koristan link: <http://bit.ly/1nZnpME>

Firefox 29 je dostupan za preuzimanje
29. april, 2014.



Nova stabilna verzija donosi mnoge promene i novo korisničko sučelje pod imenom *Australis*.

Koristan link: <http://bit.ly/SlGC17>

Objavljena KaOS 2014.04 Linux distribucija
30. april 2014.



Objavljena je distribucija po uzoru na *ArchLinux* sa fokusom na *KDE* grafičko okruženje i na *Qt*.

Koristan link: <http://bit.ly/1i6gOPY>

Canonical prekinuo razvoj projekta Ubuntu for Android
1. maj 2014.



Razlog za ovo, kako navodi *Canonical*, jesu veći prioriteti kao što je to *Ubuntu* za mobilne uređaje.

Koristan link: <http://bit.ly/Rc3NdB>

Objavljen je OpenBSD 5.5
1. maj, 2014.



Promene su primećene u sigurnosti sistema i instaleru, a proširena je podrška za hardvere.

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

VCFe 15 (Vintage Computer Festival Europa)
2. maj 2014.



Evropski festival starih računara, petnaesti po redu, održao se 3. i 4. maja 2014. godine u

Minhenu.

Koristan link: <http://bit.ly/1s3CmQg>

LIBRE! prijatelji

LUTHERUS

Et in Arcadia ego!

free
OPEN.net

Think about this



ICT časopis

ictcasopis.ict.edu.rs



LOVĆENAC
LINUX USER GROUP



Grupa korisnika GNU/Linux operativnih sistema u Lovćencu

info i tutorijali na srpskom
lubunturs.wordpress.com

lubuntu



Ugovor Republike Srbije sa Microsoftom

(6. deo)

Informatika u osnovnim školama i FLOSS

Autor: Dejan Maglov

Ovo je jedna od ključnih tema ovog serijala. Kao nikada do sada u ovom serijalu, nalazimo se na vrlo klizavom terenu. Ovo je vrlo obimna tema i može da se sagleda iz mnogo uglova. Odmah na početku da kažemo da kritika nije upućena na račun učitelja, nastavnika i profesora. Zaključci i predlozi navedeni u ovom članku su upravo zasnovani na predlozima samih nastavnika.

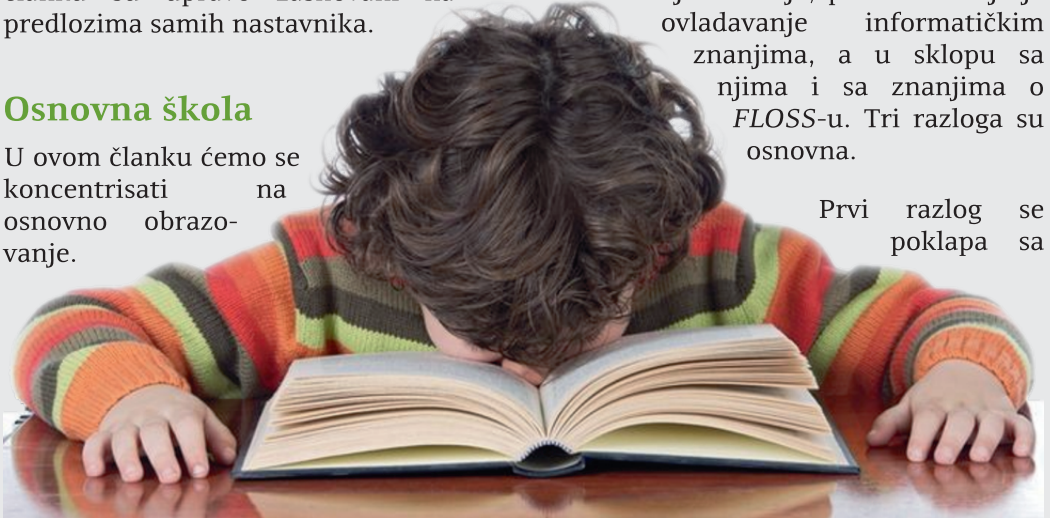
Osnovna škola

U ovom članku ćemo se koncentrisati na osnovno obrazovanje.

Osnovna škola treba da obezbedi široka temeljna znanja koja mogu lako da se nadograđuju u bilo kom smeru u nastavku školovanja. Kad smo u prošlom broju pominjali medijsku blokadu FLOSS-a, rekli smo da komercijalni mediji ne moraju da pričaju o FLOSS-u, ali da škola mora.

Ima više razloga zašto mislimo da je osnovno školstvo pravo vreme za sva temeljna znanja, pa i za ozbiljnije ovladavanje informatičkim znanjima, a u sklopu sa njima i sa znanjima o FLOSS-u. Tri razloga su osnovna.

Prvi razlog se poklapa sa





opštim mišljenjem da je informatička pismenost za dalji život odmah iza funkcionalne pismenosti na maternjem jeziku i funkcionalne pismenosti na nekom stranom jeziku. Kad se zapošljavaš, više niko ne pita da li si pismen na maternjem jeziku, to se podrazumeva. Prvo pitanje je da li znaš neki strani jezik, a nakon toga da li si kompjuterski pismen. Ako je informatika u životu toliko važna, onda i treba da se proučava uporedo sa ostalim temeljnim predmetima kao što su maternji jezik, strani jezik i matematika.

Drugi razlog je plod višedecenijske tranzicije i opšteg osiromašenja. Statistike pokazuju da sve više dece ne nastavlja dalje školovanje posle obavezne osnovne škole. Ovo su porazne činjenice. Srednjoškolsko obrazovanje nije obavezno, zato ovladavanje osnovnom informatičkom pismenošću ne treba odlagati na period srednje škole, pod izgovorom da deca u osnovnoj školi nisu stasala za informatiku. Osnovci su mali suđeri znanja i predodređeni su da lakše od starijih usvajaju sve vrste novotarija. Ne stoje tvrdnje da informatika ne treba deci koja ne žele ili ne mogu da nastave školovanje nakon osnovne škole jer to je deo opšte pismenosti.

Treći razlog se odnosi direktno na FLOSS u osnovnoj školi. Činjenice da današnja deca, ako imaju uslove u roditeljskom domu, u osnovnu školu donose neko predznanje o informatici. Nastavnici se uglavnom slažu da deca već dosta znaju o računarima, ali su ta znanja nesistematična i nefunkcionalna. Ta znanja su plod vizuelnog

pamćenja. Ispravljanje nečega što je pogrešno naučeno, ponekad je teže nego učenje od početka. Ovde FLOSS može da pomogne. Relativno mala zastupljenost FLOSS-a kod nas omogućava da se izjednače deca koja nikad nisu videla kompjuter sa onom koja su viđala samo *Windows*. *Linux* je dovoljno različit da poništi već stečena pogrešna znanja, a dovoljno jednak da se preko njega mogu nadograditi znanja iz bilo koje oblasti informatike, ali sada na dobrim osnovama.

Trenutno stanje

Iako se svi slažu da je informatička pismenost deo opšte pismenosti stanovništva, kod nas informatika nije izborila ni približni status maternjeg jezika, stranog jezika ili matematike. Informatika još nema svoj obavezni zvanični predmet u osnovnoj školi. Informatika u osnovnim školama se izučava kroz tri predmeta:

- Od igračke do računara (izborni predmet - niži razredi) ,
- Tehničko i informatičko obrazovanje (obavezni predmet - viši razredi) i
- Informatika i računarstvo (izborni predmet - viši razredi).

Ovakvim stanjem su zbunjeni i učenici i nastavnici. Nastavni planovi ovih predmeta se preklapaju i ponavljaju. Izborni predmeti se ne ocenjuju pa ni učenici ne smatraju da su nešto mnogo važni. Izostanci sa izbornih predmeta nisu sankcionisani ni na koji način, čak ni kroz smanjenje ocene iz vladanja. Izučavanje informatike kroz izborne



predmete se pokazalo samo kao čisto gubljenje vremena.

Naše osnovne škole uglavnom nemaju informatičke kabinete, što svodi učenje na papir i olovku. Deci sa predznanjem je to dosadno, a onima koji nisu ni videli računar, to su „španska sela”. Onda i nije čudno što deca beže sa časova izbornog predmeta.

Hardver

Bez računara nema ozbiljnog učenja informatike i ne samo informatike nego i svih drugih predmeta. Čisto učenje teorije ne ostavlja trajan trag u memoriji. Deca mnoge stvari pamte vizuelno. Bez praktičnog rada možda će imati dobre ocene, ali će i dalje padati na PISA testovima jer je takvo znanje površno i nepovezano sa praksom.

Mnoge osnovne škole kod nas nemaju informatički kabinet, pa čak ni računar koji bi im služio kao nastavno sredstvo. Siromaštvo dovodi do toga da i oni koji imaju računarski kabinet, imaju zastarele računare. Činjenica je da se informatika brzo razvija i da najmoderniji računari za pet do šest godina zastarevaju. Naša zemlja nema tolika sredstva da sve škole snabdeva novim računarima, a kamoli da ih svakih pet do šest godina obnavlja.

Šta je rešenje? Prvo treba utvrditi šta će deca na tim računarima raditi. Sigurno je da neće raditi nešto što zahteva previše procesorske snage i veliku količinu radne memorije. Osnovci će učiti osnovne informatičke pojmove: hardver, operativni sistem, aplikativni softver, mreže, internet, komunikaciju i osnove programiranja. Sve to je bolje učiti na starom računaru nego pomoću





papira i olovke. Ako je hardver star, softver ne mora biti. Jedini novi softver koji može da pokreće stare računare, jeste slobodni softver.

Kad smo utvrdili da se može odvijati nastava na starom hardveru ali bez gubitka aktuelnosti zahvaljujući novom, aktuelnom slobodnom softveru, onda se nameće i jedino rešenje za nedostajeći hardver u osnovnim školama. Državni organi i javna preduzeća imaju računare koji zastarevaju za ozbiljni rad. Umesto da budu reciklirani, mnogo bolje bi bilo da budu donirani školama koje nemaju informatičke kabinete. Sa malim ulaganjem i uz pomoć lokalne zajednice slobodnog softvera taj hardver može da se opremi sa novim softverom i pripremi za svoju novu obrazovnu ulogu.

Softver

Situacija sa hardverom nameće izučavanje osnova informatike na *FLOSS* rešenjima. Da li će time naša deca biti oštećena u odnosu na decu iz bogatijeg dela sveta? Mi mislimo da neće. Štaviše, izučavanje osnova informatike je temeljnije na softverima otvorenog koda. Sa druge strane, čitav bogatiji svet se sve više okreće ka *FLOSS* rešenjima, tako da će poznavanje *FLOSS*-a biti sve bitnije. Sa treće strane, kad deca kroz *FLOSS* shvate bazično funkcionisanje hardvera i način na koji korisnik preko softvera komunicira sa tim hardverom, neće im biti teško da posle u višim nivoima školovanja ovladaju specifičnim novim hardverom i softverom zatvorenog koda.

FLOSS ne samo što je, zahvaljujući otvorenom kodu, transparentan i pogodan za izučavanje osnova informatike,





već poseduje i veliku količinu edukativnog zanimljivog softvera primerenog osnovcima. Ovo omogućava da preko informatike neka druga znanja koja su stečena u školi, deca primene i vežbaju.

Nativno višekorisnički sistem, kakav je *FLOSS*, omogućuje bezbednu upotrebu računara bez „treniranja strogoće”. Svaki uzrast učenika može da dobije svoja specifična ovlašćenja koja im omogućuju da slobodno rade ono što je primerno za njihov nivo, a ostalo što nije preporučljivo za njih, zabraniće im sam sistem. Ovo učiteljima i nastavnicima omogućava da se opuste i prestanu da izigravaju „policajce” koji striktno kontrolišu šta deca rade. Čak i u uslovima kada najstarija generacija ima najšira moguća ovlašćenja nad sistemom, nastavnici imaju, sa jednog računara, odličan pregled i kontrolu nad svim umreženim računarima.

FLOSS je manje ranjiv na infekciju računarskim virusima, što olakšava održavanje sistema. Ne samo da se olakšava održavanje sistema, nego se i štedi novac za specifične antivirusne softvere bez kojeg se vlasnički softver ne može zamisliti.

Zaključak

- Informatika je deo osnovne pismenosti čoveka i kao takva mora dobiti status ostalih temeljnih predmeta kao što su maternji jezik, strani jezik i matematika.
- Bez računara nema ozbiljne nastave informatike, a i u ostalim predmetima računar bi morao biti jedan od važnijih nastavnih sredstava.
- Računari koji su završili svoj produktivni rad, mogu da nastave svoj rad u obrazovnom procesu.
- Ako ne možemo obezbediti novi



hardver, novi softver možemo. S obzirom na to, *FLOSS* je logično rešenje koje se samo nameće starom hardveru.

- *FLOSS* je jeftin, bezbedan i pogodan za proučavanje osnova informatike, dovoljno različit od vlasničkog softvera da sva deca, bez obzira na predznanje, počinju od nule, a dovoljno je sličan da se znanje usvojeno na *FLOSS*-u lako može nadograđivati na bilo kojoj drugoj platformi bez ponavljanja.
- Višekorisnički sistem koji obezbeđuje *FLOSS* omogućuje da na istim računarima uče i viši i niži razredi, ali sa različitim ovlašćenjima koji su primereni svakom nivou obuke ponaosob.

Mnogo je razloga zašto je informatika neophodna i to od samog početka školovanja. Ono što sada imamo u toj oblasti, neodgovarajuće je i vodi u zaostajanje za razvijenim svetom. Razvijeni svet kroz *FLOSS* namerno štedi i masovno prelazi na ta rešenja. Mi smo u prilici da nemamo kud. Za skup hardver i softver nemamo sredstava. Iako idemo sa različitim motivima ka *FLOSS*-u, mi i razvijeni svet baš uz pomoć *FLOSS*-a možemo da se nađemo u zajedničkoj tački.

Ovu reformu nije lako sprovesti, ali ako se svi uozbiljimo i krenemo zajedno ka nekom cilju, izvodljivo je. Potrebna je politička volja i bolja organizacija *FLOSS* zajednica, ali o tome ćemo u nekom drugom nastavku.

Nastaviće se...

Pregled popularnosti GNU/Linux /BSD distribucija za mesec april

Distrowatch

1	Mint	3701<
2	Ubuntu	3345<
3	Debian	1952>
4	elementary	1529>
5	Arch	1324>
6	Fedora	1309>
7	openSUSE	1227<
8	Mageia	1209>
9	Zorin	1207>
10	Lubuntu	1126<
11	Ultimate	1051>
12	Xubuntu	987>
13	Puppy	893>
14	Ubuntu GNOME	826<
15	CentOS	813>
16	Bodhi	779>
17	PCLinuxOS	769=
18	Manjaro	737<
19	Simplicity	724<
20	Kubuntu	700<
21	CrunchBang	685>
22	Tails	676>
23	Kali	644>
24	Red Hat	635>
25	Robolinux	626>

Pad <

Porast >

Isti rejting =

(Korišćeni podaci sa *Distrowatcha*)



Prezentacija:





LINUX U PRAKSI

17. aprila, četvrtak, u 14 časova
Velika sala Privredne komore Beograda
ulica Kneza Miloša 12

Autor: Dejan Maglov

Mesto: Privredna komora Beograda,
Kneza Miloša 12, Beograd

Vreme: 17.04.2014.

Organizatori: Privredna komora
Beograda i *Ubuntu* zajednica Srbije

U Privrednoj komori Beograda, 17. aprila 2014. godine, održana je prezentacija „Linux u praksi”. Suorganizator ovog skupa pored Privredne komore Beograda bila je *Ubuntu* zajednica Srbije. Cilj skupa je da se privrednicima Beograda predstave *FLOSS* alternative najčešće korišćenih softvera u poslovnom okruženju.

Prezentacijom su bile obuhvaćene sledeće teme:

- Uvod u *FLOSS*
- *Linux* kao serverski sistem
- *Linux* kao noseći OS u preduzećima
- *FLOSS* softver u svakodnevnoj poslovnoj upotrebi





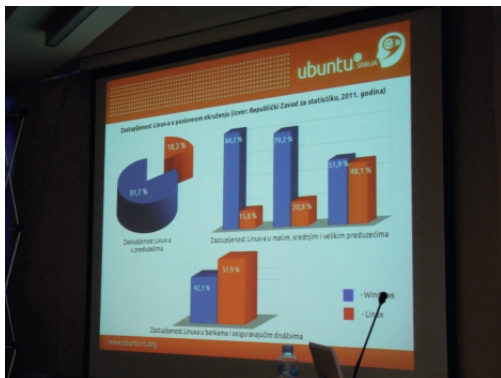
Prezentacija

Nakon pozdravnog govora koji je u ime domaćina, Privredne komore Beograda, održao Dušan Rakić, uvodnu reč je imao Ladislav Urošević, koordinator *Ubuntu* zajednice Srbije.



On je prisutnima predstavio *FLOSS* kao alternativu postojećim vlasničkim softverima i prezentovao trenutno stanje zastupljenosti *FLOSS* rešenja u Srbiji. Po statistikama koje smo mogli da čujemo od Ladislava Uroševića, zastupljenost *FLOSS*-a u našim preduzećima je proporcionalna sa veličinom preduzeća. U velikim preduzećima zastupljenost *FLOSS* rešenja je znatno veća nego kod malih i srednjih preduzeća. S obzirom da je učešće malih i srednjih preduzeća u našoj privredi na nivou 50% ukupnih prihoda privrede i preko 90% od ukupnog broja preduzeća u Srbiji, otvara se veliki prostor za širenje *FLOSS*-a u poslovnom okruženju. Ako velika preduzeća vide interes i upotrebljivost *FLOSS*-a u poslovnom okruženju, postavlja se pitanje zašto se to ne bi prelilo i na manja i srednja preduzeća, kojima uštede na troškovima softvera znače, možda više nego velikim

preduzećima.



Statistika zastupljenosti *FLOSS*-a po oblastima primene pokazuje dominaciju vlasničkog softvera na polju *desktop* računara i poslovnih servera, dok na *web*-serverima, *webu* i *super*-kompjuterima *FLOSS* rešenja su dominantna. Zato je drugi deo prezentacije koju je vodio Bojan Bogdanović, glavni koordinator *Ubuntu* zajednice Srbije, bila posvećena *FLOSS* alternativni za poslovni server.



Bojan Bogdanović je prisutnima ukratko predstavio *Zentyal*, *FLOSS* server baziran na *Ubuntu*. Poenta ovog dela prezentacije je da *Zentyal* ima punu



primenjivost poput vlasničkog servera ali uz znatno manje troškove. Jedino što se, eventualno, plaća primenom *Zentyala* jeste podrška i održavanje od strane nekog stručnog lica, sve ostalo je besplatno, ali je suština da ovakva ušteda ne ide na uštrb kvaliteta ovakvog serverskog rešenja.



Za ilustraciju koliko *FLOSS* može da bude moćan, bio je zadužen Marko Novaković, saradnik *Ubuntu* zajednice Srbije.



Marko je prisutnim gostima predstavio *Ubuntu studio*, specijalizovanu *Ubuntu* verziju za razvoj multimedijalnih projekata, kao i nekoliko gotovih multimedijalnih projekata koji su razvijeni samo

uz pomoć *FLOSS*-a. Naročito je bila efektivna prezentacija delova animiranog filma koji je u potpunosti bio izrađen pomoću *Blendera*, slobodnog softvera za 3D modelovanje i animaciju.



U nastavku prezentacije Bojan Bogdanović je dao i jedan primer moguće uštede na softveru za malo preduzeće sa deset računara. Računica je pokazala da to preduzeće, da bi opremilo tih deset računara sa legalnim vlasničkim operativnim sistemom (npr. *Windows*, jeftinija varijanta u odnosu na *MAC OS X*), vlasničkim *Office* paketima (*MSoftware*), antivirusnim softverom (npr. *AVG*), softverom za komprimovanje fajlova (*zip*), softverom za narezivanje diskova (*Nero*) i samo jednim serverom (*SBS Essentials*), mora da izdvoji oko 697 000 RSD. Ovo je, moramo priznati, priličan izdatak za malo preduzeće. Ušteda koju Bojan preporučuje, jeste zamena nekih vlasničkih softvera *FLOSS* alternativama i to u dve varijante. Prva varijanta je uzeta uz pretpostavku da je vlasnički OS došao predinstaliran sa hardverom i da se, kao takav, njegov trošak ne može izbeći. Uz vlasnički OS moraju se dodati

Pojedinačna licenca	Linux	Windows
Operativni sistem	-	20.000,00 RSD
Server (SBS/Essentials)	-	47.000,00 RSD
Office	-	30.000,00 RSD
Antivirus	-	2.000,00 RSD
Arhiviranje (zip, rar)	-	2.000,00 RSD
Programi za narezivanje	-	11.000,00 RSD
Ukupno (10 računara + server)	-	697.000,00 RSD

Mešovite mreže

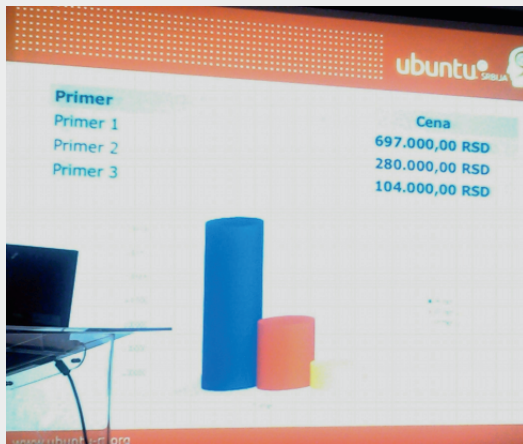
- Raznorodni operativni sistemi
- Različiti proizvođači programa

Primer 2

- 10 računara sa Windows OS
- 2 licence za Microsoft Office
- 10 licenci antivirus
- Svi ostali računari sa FLOSS rešenjima

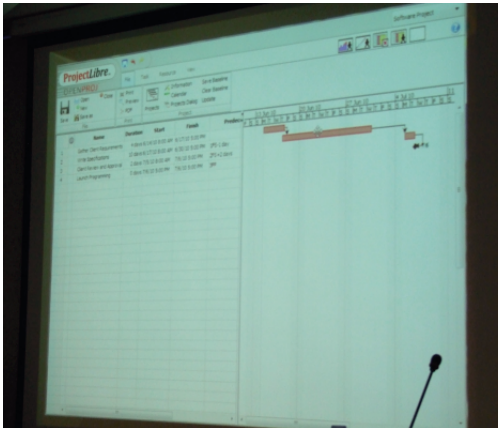
Primer 3

- 2 računara sa Windows OS
- 2 licence za Microsoft Office
- Svi ostali računari sa FLOSS rešenjima



još deset licenci za antivirusni softver i samo dve licence za vlasnički *Office* paket, zbog specifičnih potreba preduzeća. Ako se ostali softveri zamene sa *FLOSS* alternativama (*Zentyal server*, *LibreOffice*, *7-zip*, *InfraRecorder* i drugi), troškovi ove varijante se smanjuje na 280 000 RSD. Treća varijanta je ujedno i kombinovani sistem koji podrazumeva samo dva računara sa vlasničkim OS-om, dva antivirusna paketa i dve licence za vlasnički *Office* paket, koji mogu da obezbede nesmetano izvršavanje nekih specifičnih potreba preduzeća za vlasničkim softverom, a svi ostali programi na ta dva računara, kao i ostali računari, bili bi opremljeni *FLOSS* rešenjima, plus *FLOSS* server. Ovakav model troškove svodi na samo 104 000 RSD. Ušteda je značajna, a pri tome nije ugroženo funkcionisanje preduzeća.

Na kraju prezentacije Milan Vlahović iz Info centra, sektora za informatiku, predstavio je više *FLOSS* alternativa koje su jako primenjive u poslovnom okruženju. Prvo je predstavio alternative za *MS Project manager* kao što su *ProjectLibre*, *GanttProject* i *GNOME Planner*. Zatim je predstavio programe za vizuelizaciju ideja kao što su *FreePlane*, *FreeMind* i *VYM*. Takođe, zanimljiva oblast za poslovnu primenu je upravljanje sa poslovima i vremenom. Milan Vlahović je predstavio *Korganizer*, *Getting Things Gnome*, *Gnome Time Tracker* i *GTD-free*. Statistička obrada i grafički prikaz podataka je takođe zanimljiva oblast za poslovnu primenu, a Vlahović je u toj oblasti predstavio *RStudio* i *GNU PSP*



(zamena za IBM SPSS).

Zaključak

Prezentacija „Linux u praksi” po našem mišljenju je uspela. Privukla je pažnju četrdesetak gostiju koji su mogli da čuju dosta zanimljivih informacija o primeni FLOSS-a u poslovnom okruženju. Prezentacija nije bila usmerena prema promovisanju *Ubuntu*a, kako bi neko mogao da pomisli s obzirom da je suorganizator bila *Ubuntu* zajednica Srbije. Cilj presentera je prvenstveno bila usmerena ka promovisanju FLOSS-

a uopšte, kao jeftine i upotrebljive alternative vlasničkom softveru u poslovnom okruženju.

Ovo neće biti jedina prezentacija koja će promovisati FLOSS u Privrednoj komori Beograda. Za naredni period su najavljene prezentacije:

- „LibreOffice na Vašem radnom mestu” – dva dana, 15-16. maj 2014.
- „Mape uma za bolje rezultate u Vašem poslovanju” – 6. jun 2014.
- „Koristite R! – Interaktivna analiza Vaših podataka” – dva dana, 17-18. jun 2014.
- „Osnove upravljanja projektima – ProjectLibre” – dva dana, 26-27. jun 2014.

LiBRE! pozdravlja ovu akciju Privredne komore Beograda i *Ubuntu* zajednice Srbije sa ciljem promovisanja FLOSS rešenja u poslovnom okruženju kao i najavu nastavka ove akcije. Smatramo da je ovo, možda, najbolji put za popularizaciju FLOSS-a. Za razliku od privatnih korisnika koji ne mogu baš u potpunosti da shvate koncept legalnosti i uštede na FLOSS-u, privrednici će biti itekako zainteresovani za ovaj način uštede. Jednom kada se FLOSS ustoliči u firmama, lako će i iz firmi preći i na kućne računare.





Serbian 2014

(2. deo)

Autor: Dejan Maglov

Uvod

Prošlog puta komentarisali smo koncept Serbiana. Ovog puta ćemo se koncentrisati na njegovu upotrebljivost. I dalje ćemo se držati isključivo projektnog zadatka koji, da podsetimo, pretpostavlja da je Serbian srpska distribucija, potpuno lokalizovana na srpski jezik i to srpskom ćirilicom, namenjen prvenstveno manje iskusnim korisnicima *Linuxa*, koji korisniku nudi princip „instaliraj i odmah koristi” i na kraju, možda i najsporniji zadatak, da Serbian pruži korisniku upotrebljivost *Minta* ili *Kubuntua*.

Instalacija

Sama činjenica da je Serbian komponovan na bazi stabilnog *Debiana*, unosi malo sumnje da će Serbian ispuniti zadatak da će biti distribucija za početnike.

Već pri prvom pokretanju *Serbian2014-1.iso*-a vidi se da Serbian nema živi disk. Po našem mišljenju to je mana. Danas je već standardno da distribucije koje flertuju sa početnicima, imaju živi disk.

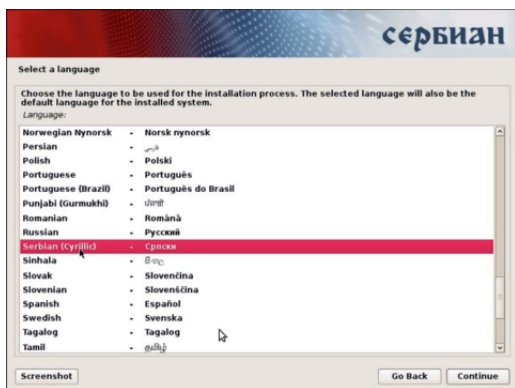
Živi disk je izlog za distribuciju koji budućem korisniku prezentuje, ugrubo, svoje mogućnosti i prilagođenost hardveru na kojem će raditi. Zamorno je instalirati distribuciju zbog probe, a čist gubitak je vremena ako instalacija ne proradi na našem hardveru.



Zanemarimo sad činjenicu da živog diska nema i pređimo na instalaciju. U ponudi nam je grafički i *CLI* instaler. Početnik će izabrati uglavnom grafički instaler, što i mi činimo. Dočekuje nas *Debianov* grafički instaler. Pouzdan je, ali znatno komplikovaniji od instalera *Kubuntua* ili *Minta*. Zapravo, nije toliko u pitanju komplikovanost nego činjenica da ovaj grafički instaler izgleda kao malo našminkani *CLI* instaler sa



priličnim brojem koraka, pri čemu svaki korak odrađuje samo po jednu funkciju. Za razliku od *Debianovog* grafičkog instalera, moderni grafički instaleri, poput instalera *Kubuntua* ili *Minta*, imaju manji broj koraka, ali unutar njih je grupisano više podešavanja, kao na primer podešavanje vremena i lokacije ili podešavanje *root* i korisničkih naloga, ili sva podešavanja particionisanja na jednoj strani. Ovo nije primedba na račun *Serbiana*, barem ne direktno, nego više na konzervativnost *Debiana*.



Ako se apsolutni početnik snađe u ovom *Debianovom* grafičkom instaleru, instalacija će proteći glatko i brzo, sa automatskim restartom na kraju.

Prvi pogled na Serbian

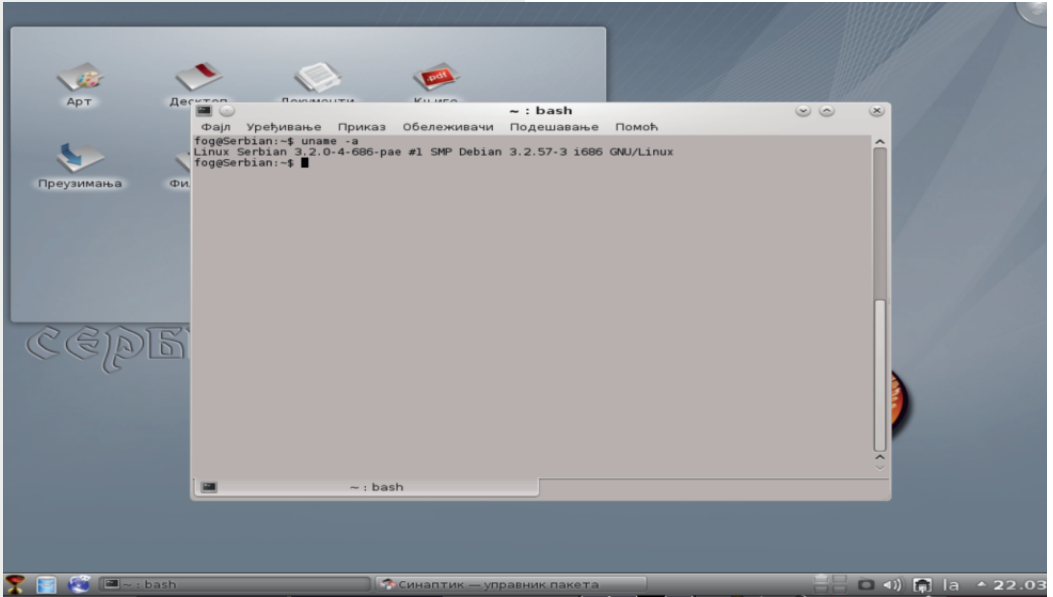
Jedna od ozbiljnijih primedbi na forumima, a na račun *Serbiana* jeste njegov podrazumevani izgled. Koliko smo mi shvatili, nije primedba upućena na „ružnoću” *Serbiana*. Ruku na srce, treba biti majstor i toliko upropastiti

KDE, pa da on bude ružan. Stvar je u tome da mnogi nisu razumeli da je ovo **srpska distribucija**, pa samim tim su i napali stilizovana nacionalna obeležja koja *Serbian* ima u podrazumevanom izgledu. Ne vidimo nikakvog razloga da se mi bunimo protiv naših obeležja niti vidimo razloga da pravimo distribuciju za čitav svet. Nama treba srpska distribucija i nije loše da korisnik na prvi pogled vidi da je na pravom mestu.

S obzirom da je moguće i na neki drugi način lokalizovati *Serbian*, na primer u bosansku latinicu, isto tako se može vrlo lako promeniti i sporni *wallpaper*, tako da ne vidimo prepreku da i oni koji nisu Srbi, koriste ovu prvenstveno srpsku distribuciju, ukoliko to žele.

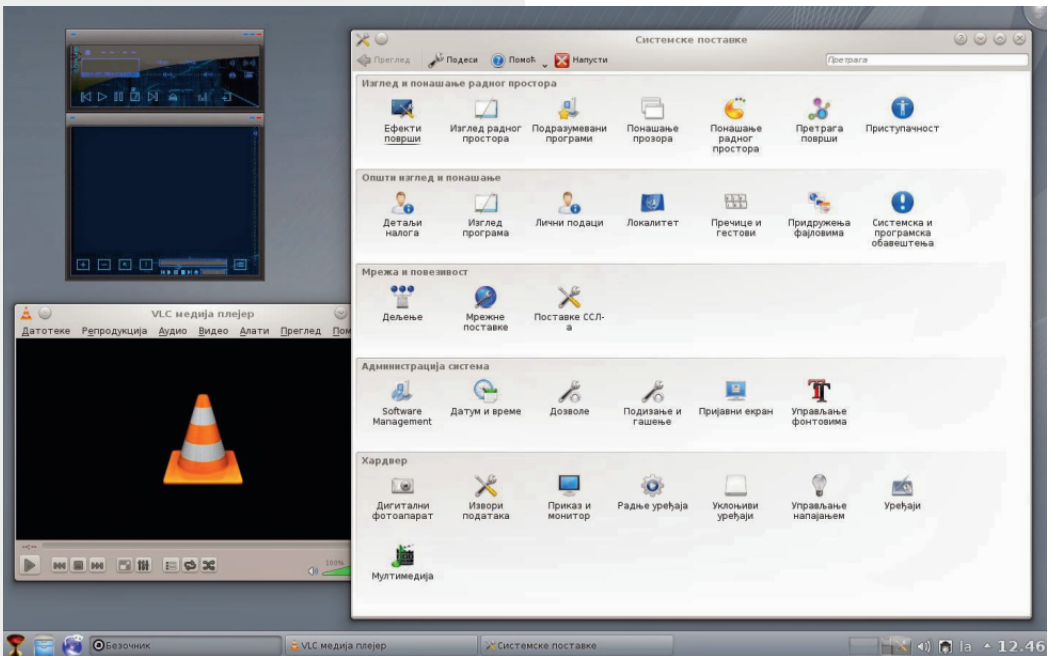
Pošto ne postoji živi disk *Serbiana*, jedini način da vidite njegov izgled, jeste da ga instalirate i ako imate sreće sa hardverom (kao mi na testnoj mašini), nakon restarta dobićete *Serbian KDE* u punom sjaju. Prvi utisak nakon restarta jeste da je kompozitor *Serbiana* sa ukusom uredio grafički interfejs, bez viškova i kičastih detalja. Monotoniju sivih nijansi lepo razbijaju diskovi sa stilizovanim srpskim grbom u vidu Feniksa. Uz podrazumevani *wallpaper* lepo se uklopila *oxigen KDE* tema, tako da je opšti utisak prijatan za oko.

Izbor *KDE*-a za podrazumevano okruženje radne površi glavnog izdanja *Serbiana* uvek će biti kontroverzan. *Linux* korisnici se binarno dele u dva tabora – na one koji obožavaju *KDE* zbog njegovog prefinjenog izgleda i na ostale koji ga ne vole zbog „komplikovanog” podešavanja i opterećenja



grafičke kartice, koja zbog obima KDE efekata može da počne da „štuca“ ako je

stara i loša. Sredine nema. Kompozitori Serbia su, čini nam se, bili svesni ove





činjenice, pa su razvili i *Openbox* Serbian za slabije i starije računare, ali i za one koji ne vole *KDE*.

Objektivno, kombinacija stabilnog *Debiana* sa starim ali pouzdanim *LTS kernelom* 3.2 i *KDE* 4.8 nije baš idealni spoj. *Kernel* 3.2 dobro radi na starijim računarima. *KDE* 4.8 može da radi na tim starijim računarima, ali oni moraju da budu malo jači da *KDE* ne bi „štucuo“. Za određivanje maksimalne i minimalne konfiguracije hardvera koji podržava Serbian, treba malo više vremena i više testera tako da vam ne možemo sada reći koja je optimalna konfiguracija hardvera za Serbian.

Lokalizacija

Prijatan je osećaj kada te na računaru

dočeka maternji jezik ispisan ćirilicom. Trudili smo se da nađemo neki program ili uputstvo koje nije prevedeno, ali nismo uspeli da ga nađemo. Sve je lokalizovano, barem u *KDE* izdanju. U *Openbox* izdanju smo našli nekoliko nelokalizovanih delova ili delimično lokalizovanih programa. Najuočljivija je delimična lokalizacija *Geanya*. Sa te strane korisnici *KDE* izdanja mogu da budu zadovoljni.

Ono što najčešće nije lokalizovano, jeste sistem pomoći. *KDE* sistem pomoći je kod nas bio podrazumevano isključen, tako da nismo uspeli da vidimo da li je tu sve lokalizovano. Za naredni nastavak recenzije Serbia imamo u planu intervju sa autorom, pa ćemo ga pitati da li je namerno isključen ovaj sistem pomoći ili mi nismo nešto dobro



podesili. Ako je namerno isključen, pitaćemo ga zašto je isključen i da li će biti u nekoj drugoj verziji uključen.

Personalna podešavanja okruženja i sistema

Po jednostavnosti personalizacije sistema možemo da odredimo za koga je zapravo namenjen taj distro. Kada kažemo da je distro za početnike, treba razlikovati apsolutne računarske početnike i iskusne korisnike računara, ali početnike na *Linuxu*. Po našem mišljenju, idealni distro za apsolutne početnike je *Slackware*. Kako?! Jednostavno, apsolutni početnici svakako ne znaju da instaliraju ni jedan OS, pa onda moraju da traže pomoć od iskusnijih računarskih korisnika da im instaliraju sistem. Nakon instalacije, *Slackware* korisnik ima sve što početniku treba, a nema lako dostupne grafičke pomoćnice gde početnik može, čeprkajući po sistemskim podešavanjima, lako napraviti pogrešku i nađati svom sistemu.

Veći problem su ovi „iskusni početnici”. Oni su zahtevniji u pogledu izgleda, funkcionalnosti svog sistema, a razmaženi su vlasničkim softverom koji im nudi grafičko sučelje za svako podešavanje sistema. *Mint* i *Kubuntu* su se prilično približili ovim „iskusnim početnicima”, nudeći sva podešavanja na klik. Ovo je dvosekli mač jer omogućava početnicima da svojim akcijama nađu sistemu.

Serbian je tu negde na pola puta između *Slackwarea* i *Kubuntua*, što je standardno za *Debian*. Ono što olakša-

va, dostupna podešavanja na klik imaju sve opcije na srpskom jeziku, pa će se u tim podešavanjima snaći i početnici koji ne poznaju dovoljno engleski jezik.

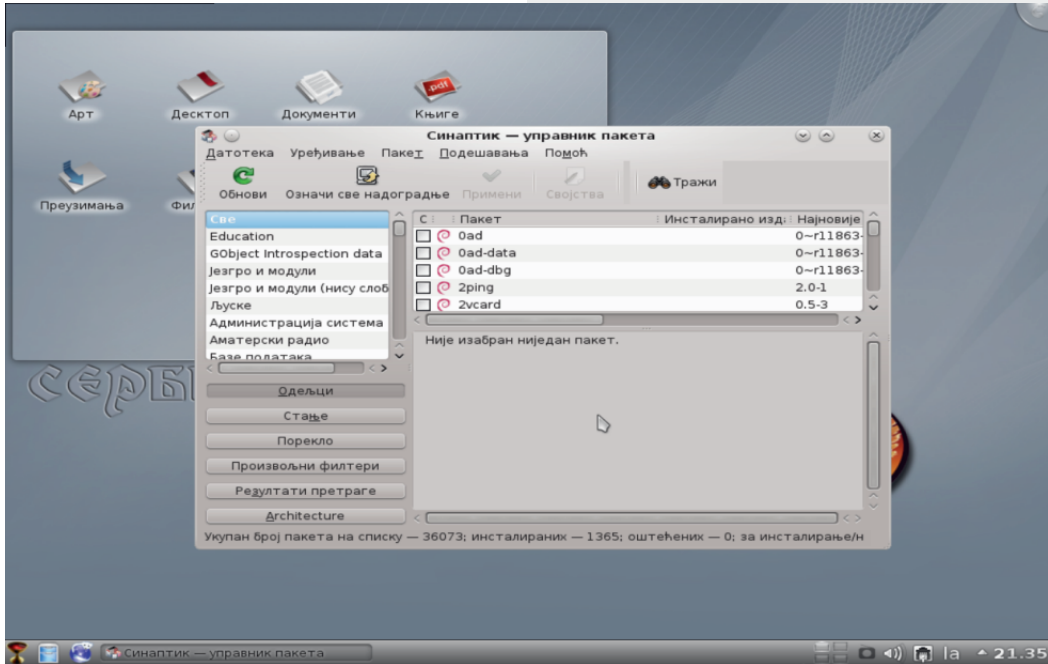
Programi

Najveća zamerka koju ističu *Linux* korisnici, koji ne vole *KDE*, jeste to da uz *KDE* dolazi i gomila ne baš tako popularnog *KDE* softvera. Uz *KDE* obično dolazi *Koffice* umesto *LibreOfficea* ili *Krita* umesto *GIMP*-a ili *Konqueror* umesto *Firefox*a i slično. *KDE* orijentisane distribucije uglavnom promovisu ove *KDE* projekte koji nisu loši, ali ipak nisu najpopularniji *Linux* softveri i drago nam je da Serbian nije krenuo tim smerom.

Serbian je zadržao najvažnije, najbolje i bitne *KDE* aplikacije (*Dolphin*, *Kate*, *K3b*, *Gwenview*, *Konversation* i *Okular*), ali se nije libio da doda i *GTK* aplikacije (*GIMP*, *VLC*, *LibreOffice*, *IceWeasel* (*Debian* fork *Firefox*a), *Chromium*, *Skype* i *Clementine*). Zaključak je da Serbian ima dobro izbalansiran izbor najpopularnijih *FLOSS* programa bez mnogo dupliranja.

Serbian nema fensi paket menadžer poput *Kubuntua* (Softverski centar). On se uglavnom oslanja na pouzdani *Synaptic*. *Synaptic* možda nije najlepši paket menadžer za početnike, ali je od softverskog centra i brži i pouzdaniji.

Po završenoj instalaciji dodali smo preporučene riznice. Iskoristili smo priliku da probamo *Synaptic* za *update* (osveženje) sistema i za probu dodali još nekoliko programa (*Inkscape* i



Scribus). Osvežavanje sistema i instalacija je protekla glatko, a i programi su zadržali podrazumevanu lokalizaciju, uglavnom. *Inkscape*, na primer, bio je potpuno lokalizovan na srpsku cirilicu, ali *ImageMagic* koji je povučen kao njegova međuzavisnost, nije.

Zaključak

Zaključak je da je podrazumevana postavka *Serbiana* lepo ispoštovala zadataku lokalizaciju, ali će se tokom korišćenja pojaviti u novoinstaliranim programima i programi koji nisu lokalizovani, i ne ispunjavaju projektni zadatak *Serbiana*. Da bi se ovo izbeglo, jedini način je formirati sopstvene riznice sa isključivo potpuno lokalizovanim softverskim paketima. Ovo je verovatno preveliki zadatak za malu

zajednicu koja održava ovaj sistem. Možda jednog dana u budućnosti.

Mada smo imali nameru da više kritikujemo izvedbu *Serbiana* u ovom broju, ipak smo se ograničili samo na utvrđivanje stanja. Pokušaćemo do sledećeg broja da stupimo u kontakt sa autorom i da mu postavimo pitanja koja će nam razjasniti misteriju, zašto *Serbian* izgleda baš ovako i koji su planovi za budućnost ovog projekta.





VI text editor

Autor: Aleksandar Brković

U četvrtom broju smo vam pisali o takozvanoj „ratu uređivača teksta” gde smo radili poređenje nekih uređivača. U ovom članku vam predstavljamo konkretno jedan od njih.

Linux distribucije su pogodne za obradu teksta isto kao i drugi operativni sistemi. Postoji više dobrih programa za obradu teksta na Linux platformi, kao što su *AbiWord*, *Koffice* paket, *OpenOffice* i *LibreOffice*. Pored navedenih programa, tekst se može obrađivati i u

terminal modu. Jedan od najpoznatijih uređivača teksta u terminal modu jeste *Vi* uređivač teksta. *Vi* je program za obradu teksta bez mnogo funkcija. Njegova glavna odlika je pisanje teksta bez formatiranja, tako da ga drugi program može pročitati. *Vi* je program koji dolazi iz *UNIX*-a. Postoji poboljšana verzija koja se naziva *Vim*. Problem sa ovim uređivačem je što ga mnogi ljudi ne vole. Korisnik je prinuđen da zapamti mnogo kombinacija tastera da bi bio u mogućnosti da uradi stvari koje bi neki drugi uređivač tekstova uradio mnogo lakše.

```
File Edit View Bookmarks Settings Help      -:vi
VIM - Vi IMproved
      version 7.3.429
      by Bram Moolenaar et al.
Modified by pkg-vim-maintainers@lists.ubuntu.com
Vim is open source and freely distributable

  Become a registered Vim user!
type  :help register<Enter>  for information
type  :q<Enter>               to exit
type  :help<Enter> or <F1>    for on-line help
type  :help version7<Enter>  for version info

0,0-1 All
```



Vi uređivač tekstova je praktičan u situaciji kada je potrebno brzo uneti promene u tekstualnu datoteku ili napisati kratku belešku. Sa druge strane, postoje korisnici koji uz pomoć

uređuje. To je vidljiv deo teksta koji može da se izmeni uz pomoć komandne sekvence. Nakon izvršenja komandne sekvence, promene koje je inicirao korisnik, biće vidljive na ekranu nakon

```

File Edit View Bookmarks Settings Help
*** you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
# 'info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
#GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR="lsb_release -i -s 2> /dev/null || echo Debian"
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command 'vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"

# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"

/etc/default/grub* [readonly] 34L, 1238C
  
```

završetka procesa. Kada se pozove Vi uređivač teksta, privremena radna površina koja se otvara, naziva se *buffer*. Ovaj *buffer* ili radni prostor je na raspolaganju sve vreme dok se koristi tekstualni uređivač. Korisnik može u njega da stavi datoteku koja se već nalazi u računaru, ili da

ovog uređivača rade komplikovane stvari, čak i dizajniranje *web* strana. Ovaj editor je takođe veoma efikasan i za poređenje šablona, konstruisanje makroa i izvršavanje *shell* komandi iz uređivača. Navedeni uređivač je *full-screen* uređivač, koji funkcioniše nad čitavom datotekom odjednom, koristeći pri tome ceo ekran terminalnog prozora. Dozvoljava kretanje kroz datoteku i izvođenje promena na različitim lokacijama za vreme iste sesije.

kreira novu datoteku. Nakon što korisnik napusti uređivač, privremeni *buffer* će biti odbačen.

Organizacija Vi uređivača teksta

Vi je sofisticiran uređivač sa širokim spektrom komandi. Ipak, osnova struktura je jednostavna. Postoje dva osnovna moda koja se koriste u radu sa uređivačem. Prvi mod je komandni mod ili komandni režim, a drugi mod je režim unosa teksta.

Display je orijentisan a program interaktivan. Vi tekstualni uređivač omogućava korisniku da stvori, modifikuje i memoriše datoteke na računaru, a sve preko terminala. Prilikom korišćenja ovog programa, terminal deluje kao prozor u kome se vidi datoteka koja se

Komandni mod je početno stanje na koje se nailazi kada se Vi uređivač poziva iz *UNIX* ljuske. To je stanje koje je spremno za prijem uputstava o tome, kakvu akciju je potrebno izvršiti.



Najčešće su to skrolovanje, pretraga teksta, pomeranje kursora, prelazak na režim unosa teksta, umetanje teksta, brisanje linije, reči ili znakova, menjanje linije reči ili znakova i slično.

Režim unosa teksta je druga strana *Vi* uređivača. Režimu unosa teksta se može pristupiti samo iz komandnog režima. Korisnici koji umeju da razdvoje ova dva režima rada, vrlo brzo će savladati kontrolu ovog uređivača tekstualnih datoteka. Nakon što iz komandnog moda korisnik uđe u režim unosa teksta, sve naknadno otkucano u terminalu će se direktno odraziti na datoteku.

Prilikom uređivanja neke datoteke, korisnik dolazi u situaciju da se kreće napred i nazad između komandnog režima i režima za unos teksta. Na primer, korisnik može da odluči da promeni reč u jednoj rečenici (mod unosa teksta), pa zatim da se pomeri na kraj datoteke (komandni režim), doda još jednu stranicu teksta (režim za unos teksta), pa onda ispravi neke greške u kucanju (režim unosa teksta), pomeri se na sredinu teksta (komandni režim), pa tu onda ubaci novu rečenicu (režim unosa teksta). Mnogim korisnicima će ovo zvučati komplikovano, međutim nije tako. Ovo je mnogo lakše uraditi, nego čitati.

Najčešća greška koju korisnici naprave prilikom rada sa *Vi* tekstualnim uređivačem, jeste da zaborave u kome režimu rada se trenutno nalaze. Ova zaboravnost dovodi do toga da korisnici pokušaju da izdaju komandu dok se nalaze u modu za unos teksta. Ovo

rezultira čudnom kombinacijom znakova koji se pojave u terminalu, što natera korisnika da uputi niz negativnih reči na račun ovog tekstualnog uređivača. Ukoliko se pojavi nedoumica u kome režimu rada se trenutno nalazite, pritisnite *ESC* tipku koja se nalazi u gornjem levom uglu tastature. Na ovaj način se vraćate u komandni režim i znaćete gde se nalazite.

Vi uređivač teksta poseduje naprednu funkciju koja omogućava korisniku da izmeni više datoteka odjednom. Mogućnost pristupa ka više datoteka bez napuštanja tekstualnog editora dozvoljava korisniku da potraži informacije u drugoj datoteci, bez napuštanja uređivača. Datoteke se otvaraju u istom uređivaču i mogu da dele iste imenovane *buffere*, čime se omogućava deljenje teksta i njegov prenos između datoteka.

Ukoliko koristite neki od jednostavnih uređivača za unos teksta, kao što su *Gedit*, *Mousepad*, *Leafpad*, *Kate*, preporučujemo vam da isprobate *Vi* uređivač teksta. Rad sa ovim uređivačem nije toliko težak, kao što izgleda na prvi pogled. Nadamo se da će ovaj tekst zainteresovati naše čitaoce da učine takav korak.

Korisni linkovi:

- [1] <http://en.wikipedia.org/wiki/Vi>
- [2] <https://engineering.purdue.edu/ECN/Support/KB/Docs/ViTextEditorTutorial/printerfriendly>



OpenStreetMap



Autor: Joakim Janjatović

Šta je OSM i čemu služi?

OpenStreetMap je kartografski ekosistem nesagledivih razmera. Da bismo jednostavnije objasnili od čega se sastoji, upoređićemo ga sa dva svima poznata sistema, *Wikipedia*-om i *Linux*-om. Poput *Wikipedia*-e, OSM ima centralnu bazu podataka <http://osm.org> koju može da dopuni svako ko ima otvoren nalog e-pošte i želju da podeli svoje znanje sa drugima. Podaci poslani centralnoj bazi odmah postaju dostupni svim registrovanim i neregistrovanim korisnicima. Ako zagrebemo malo dublje, dolazimo do paralele sa *Linux*om. Baza podataka je kao *kernel* tj. jezgro operativnog sistema koje predstavlja oslonac za sve ostale programe. Zašto baš poređenje sa *Linux*om? Niko na svetu ne može da isprati koliko tačno distribucija *Linux*a postoji, pa čak ni to koliko programa ima za samo jednu od tih distribucija. OSM je osnova za ekosistem u kom ustvari niko ne zna gde su granice tog sistema. Tačan odgovor na pitanje čemu služi OSM, autor ovog teksta ne može dati.

Prvi naš dodir sa OSM-om, bio je preko glavnog portala i baze podataka

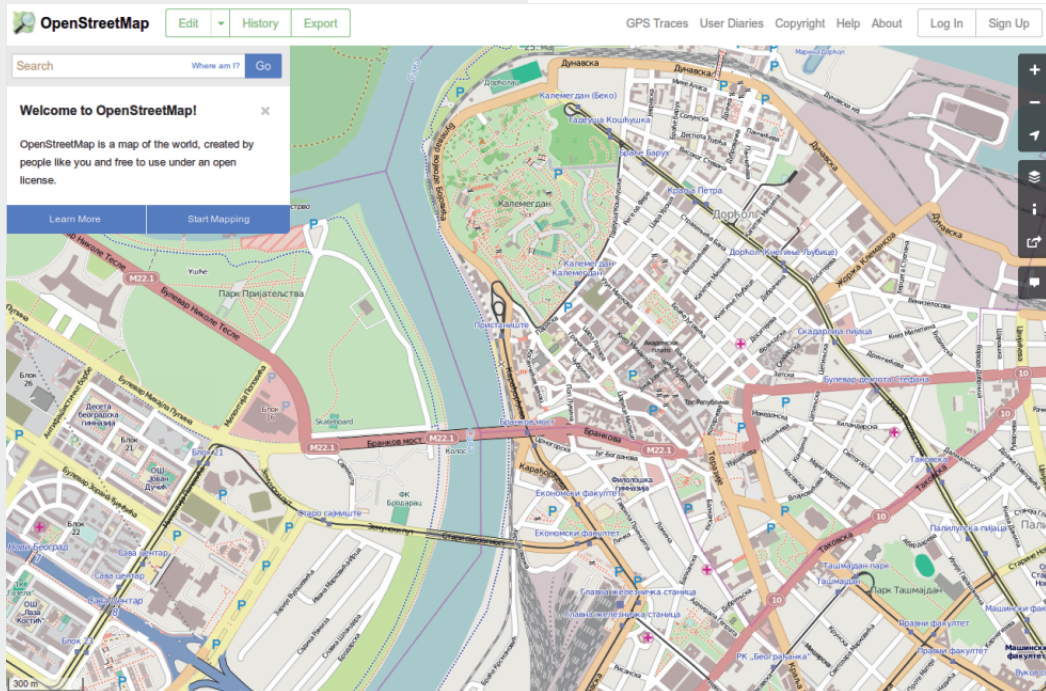
<http://osm.org>. Ako je neko već koristio *Google Map Maker*, može pretpostaviti kako bi OSM mogao da funkcioniše. Međutim, OSM je mnogo više od *web* alata za crtanje ulica. S obzirom da se u Srbiji za traženje nečega na karti najčešće koristi *Google Maps*, krećemo od toga da OSM početniku liči na GM, ali vremenom se stiče sasvim drugačiji utisak.

Licenca

OSM baza podataka dostupna je pod *ODBL* licencom. U prevodu, to znači da svi podaci mogu da se koriste u komercijalne i nekomercijalne svrhe, a jedino je obavezno naglasiti da podaci dolaze upravo od OSM i njegovih „doprinosioca”. Naravno, s ciljem opstanka projekta i poštovanja autorskih prava, zabranjeno je precrtavanje iz drugih izvora kao što su *Google Maps* ili štampane karte, osim ako nemate izričitu dozvolu vlasnika tih podataka.

Kako to radi?

Ako ste samo posmatrač, radi tako što otvorite *web* stranicu ili neki program koji ne mora da bude povezan sa internetom, a zatim pretražujete šta vas



zanima na učitanj OSM karti. Ako putujete negde i treba vam GPS navigacija, preuzmete neki od već dostupnih programa za platformu vašeg mobilnog uređaja, učitate kartu za državu ili deo sveta koji vam treba, i unesete odredište. Čar je u tome što je i većina programa koja koristi OSM karte dostupna besplatno. Tako, recimo uz *OsmAnd*, *Android* telefon postaje auto-navigacija kojoj je dovoljno jednokratno povezivanje na internet radi preuzimanja karte.

Za one koji se pitaju zašto neka ulica, prodavnica ili firma nije ucrtana na karti, odgovor je da mogu i sami da je ucrtaju. Upravo to je OSM moto. Kada bi svako ucrtao bar jednu ulicu, imali bismo potpunu i najdetaljniju kartu

sveta. OSM se zasniva na znanju meštana koji žele to znanje da podele. U nekim trenucima, to podseća na igrice u kojima ne vidite odmah celu kartu, već kako napredujete, otkrivete deo po deo (prim.aut.). Crtači-kartografi imaju na raspolaganju više vrsta alata, od kojih se najozbiljnijim smatra *JOSM*. To je uređivač napravljen u *Java*-i i zato radi na više vrsta operativnih sistema. Dostupne su razne vrste podloga uključujući i *Bing* satelitske snimke, sopstvene i *GPS* tragove drugih korisnika, kao i mnoštvo dodataka od kojih bismo izdvojili i delo domaćih dobrovoljaca *SerbianTransliterator*. On omogućava unošenje naziva na jednom pismu, recimo ćirilici, i jednostavno preslovljavanje na latinicu.



Main Page
The map
Map Features
Contributors
Help
Blogs
Shop
Donations
Recent changes

Tools
What links here
Related changes
Special pages
Printable version
Permanent link
Page information
Cite this page

English Create account Log in

Page Discussion Read View source View history Search

Sr:Main Page

Доступни језици Help

العربية • azərbaycanca • български • bosanski • català • český • dansk • Deutsch • Eesti • Ελληνικά • English • español • فارسی • français • 한국어 • Հայերեն • हिन्दी • Bahasa Indonesia • Ispania • Italiano • ייִדיש • latviešu • lietuvių • македонски • Bahasa Melayu • Nederlands • 日本語 • norsk bokmål • polski • português • português do Brasil • română • русский • shqip • slovenčina • slovenščina • српски / srpski • suomi • svenska • ไทย • Türkçe • українська • Tiếng Việt • 中文 (简体) • 中文 (繁體)

Остали језици — Помозите нам превести ову Викију show

*Добро дошли на **ОтенСтритМап**, пројекат који ствара и шири **слободне** географске податке читавог света. Започели смо га јер већина карата за које мислите да су слободне заправо садрже правна или техничка ограничења за њихово коришћење, онемогућавајући луѓе да их користе на домаћим, стваралачке, па и неочекиване начине.*

Карта 🔗	Користи главну карту 🔗 или неки од бројних других сервиса.
Вики Србија	Званична вики страница Србије.
Форум 🔗	Придружите нам се на форуму 🔗 .
Употреба ОСМ 🔗	За шта се све употребљава ОтенСтритМап и на који начин?
Водич за почетнике	Желим да сазнам више. Како да се прикључим?
Картографски пројекти	Више о врстама картографских пројеката.
Објекти на	Шта све може да се учрта на карту?.

Слика недеље 👍

Although this map looks distinctly hand-drawn, it is a data-driven rendering created using **TileMill** and **GIMP**, and available as a **slippy map** covering **Toulouse** 👍

Other featured images

Gde je počelo i dokle se stiglo?

OSM projekat je započet u Engleskoj i to iz potrebe za kartom koja nema licencna ograničenja upotrebe. Trenutno je najrazvijeniji u Nemačkoj, gde se često kao primer zapanjujuće detaljne karte navodi zoološki vrt u Berlinu. OSM trenutno broji preko milion i petsto hiljada registrovanih korisnika, od kojih je više od dvadeset hiljada mesečno aktivnih kartografa. U bazu je moguće uneti skoro bilo koji podatak, a na korisniku, odnosno pružaocu usluga prikaza je da odluči šta od toga želi da vidi na karti. U Srbiji OSM zajednica se širi polako, ali sigurno. Osnovnim podacima pokriveni su svi veći gradovi. Manja mesta još uvek zavise od toga da

li je neki meštанин čuo za OSM, ili da li je neko od aktivnih korisnika prolazio kroz to mesto. Izdvojili bismo i činjenicu da je standardizovan unos naziva ulica na srpskom, što omogućava izbor prikazanog pisma kao i višejezični prikaz. U Novom Sadu uz pomoć OSM dobrovoljaca, LUGoNSa i FTN-a već su održani predavanja i radionica na ovu temu.

Ko sve koristi OSM?

Društvena mreža *Foursquare* koristi OSM bazu podataka kao osnovu za ocenjivanje mesta, zatim *web* prezentacije koje imaju velik broj pregleda dnevno, koriste OSM umesto *Google Mapsa* jer je besplatan bez obzira na posećenost, a primetili smo da je koriste



i policajci u poznatoj nemačkoj TV seriji „Kobra 11” (prim.aut.). U slučaju elementarnih nepogoda, dobrovoljci se skupljaju i ucrtavaju najnovije podatke, tako da se zna koji putevi su prohodni a koji ne. Posebne karte koje koriste ovu bazu podataka, dostupne su i za moreplovce, planinare, bicikliste, turiste i ljude sa posebnim potrebama. Postoje čak i štampane OSM karte. Moguće je besplatno preuzeti kompletnu vektorsku bazu podataka sveta (~35GB kompresovano) i napraviti sopstveni prikaz, analizirati podatke, a uz dobru ideju i zaraditi.



Kako da se pridružim i odakle da počnem?

Preporučujemo da prvo nešto više pročitate na *OSM wiki* stranici https://wiki.openstreetmap.org/wiki/Sr:Main_Page, na kojoj se objavljuju dogovori i smernice vezane za kartografe u Srbiji. Pozivamo vas da nam se pridružite i na forumu

<http://forum.openstreetmap.org/viewforum.php?id=62> gde ćete naći odgovore na već postavljena pitanja, a možete i sami postaviti pitanje. Tu se skupljaju korisni linkovi vezani za OSM. Dostupno je dosta materijala za obuku početnika od kojih su najbolji napisani na engleskom i nemačkom jeziku. Nažalost, malo sadržaja je prevedeno na srpski, pa su prevodioci dobrovoljci kao i kartografi svakako dobrodošli. Svi zainteresovani neka se jave na forumu!

Budućnost OSM-a?

OSM je odličan vid saradnje otvorenog i vlasničkog, odnosno javnog i komercijalnog, u kojem bazu podataka popunjavaju uglavnom dobrovoljci, a onda svako ko ima ideju kako da je upotrebi i zaradi, može to slobodno da uradi bez obaveze plaćanja astronomskih iznosa monopolističkim kompanijama. Baš te firme koje koriste OSM podatke, sponzorišu i održavaju dosta korisnih OSM alata. To je recept za najažurniju, svima dostupnu kartu sveta. Ne gledamo u staklenu kuglu već u ekran, ali ipak vidimo svetlu budućnost OSM-a.



Uvod u programski jezik C

(2. deo)

Autor: Stefan Nožinić

U prethodnom broju smo dali okvirnu sliku o tome kako izgleda jedan jednostavan C program koji smo nazvali *Hello world* i koji je imao zadatak da na standardni izlaz ispisuje rečenicu *Hello, world*. Takođe smo objasnili jednostavan proces prevođenja izvornog C koda u izvršni korišćenjem *GNU Compilera (gcc)*. U ovom delu vam objašnjavamo šta su to tipovi podataka, promenljive i kako da im dodelite vrednosti u programu ili da ih učitate sa standardnog ulaza. Takođe vam predstavljamo i neke osnovne, ali i česte greške koje prave ne samo početnici nego i iskusni programeri. Ove greške se nazivaju sintaksne greške i nastaju obično kada se javi greška u kucanju (tzv. *typo*).

Tipovi podataka

U radnoj memoriji računara čuvaju se razni podaci, kao što su brojevi, tekst, imena i drugo. Na primer, kada otvorite ovaj časopis, vaš *PDF* čitač (ili *ePUB*, ako čitate to izdanje) verovatno je učitan fajl smestio u *RAM (Random Access Memory)*, odatle čita potrebne podatke i prikazuje ih na vašem ekranu. Ti podaci se pre prikazivanja obično obrađuju, pa tako dobijate lepu i čitljivu

stranicu ovog dokumenta. Kod *PDF* čitača ovo se zove renderovanje i služi da vektorski format prikaže kao sliku na ekranu.

Kao što smo rekli, postoje razni podaci koji se mogu čuvati u memoriji. Oni su podeljeni po njihovom sadržaju i za svaki tip podataka je određeno tačno koliko memorije zauzima. Zauzeće memorije zavisi od programskog jezika, ali i od arhitekture na kojoj se pokreće program, odnosno za koju je program preveden (kompajliran).

Svaki tip, pored svoje veličine, ima svoj skup mogućih vrednosti (domen), skup operacija koje se mogu izvršiti nad ovim tipom podataka i skup relacija koje važe za ovaj tip podataka. Ovde vam predstavljamo neke osnovne tipove podataka, njihove ključne reči u C programskom jeziku koje će nam biti korisne kod definisanja promenljivih i dajemo vam uvid u neke operacije, odnosno relacije za svaki od navedenih tipova.

Prvi tip koji ćemo da pomenemo, jeste *integer*. Ovo nije najjednostavniji tip, ali se obično navodi kao prvi jer je najbliži čovekovom shvatanju računara. Ovaj tip opisuje cele brojeve (... , -2, -1, 0, 1, 2, ...) i ključna reč u C-u mu je *int*.



Domen ovog tipa su celi brojevi koji ulaze u opseg -2^{31} , $2^{31} - 1$]. Sigurno ste se zapitali zašto baš ovakav opseg i kako smo došli do njega. Odgovor leži u činjenici da na većini arhitektura *integer* zauzima 4B odnosno 32 bita. Ako ste u dobrim odnosima sa kombinatorikom, onda znate da je to ukupno 2^{32} kombinacija, pa je domen podeljen na negativne vrednosti, pozitivne vrednosti i nulu. Operacije koje važe za ovaj tip podataka, iste su one koje važe i za cele brojeve u matematici. To su operacije dodele (operator =), sabiranje, oduzimanje, množenje, deljenje i modulo (% operator). Ništa nije drugačije ni sa relacijama, pa tu imamo relacije jednakosti (== operator), veće, manje, veće ili jednako, manje ili

jednako i *nije jednako*. Potrebno je obratiti pažnju da se operator za dodelu vrednosti (=) obeležava sa jednim znakom jednakosti, dok se relacija za jednakost obeležava sa dva.

Sledeći tip podataka je *char*. On predstavlja jedan karakter. Zauzima 1B u memoriji, pa je time moguće opisati 256 vrednosti. Ovo je sasvim dovoljno da opiše engleski alfabet, brojeve i još dosta drugih znakova. Najčešća operacija je operacija dodele, a najčešća relacija je relacija ekvivalencije (= - dodela, == - relacija ekvivalencije). Svako slovo ima svoju brojčanu vrednost koja se može videti u *ASCII* tabeli.





Tip podataka koji služi za opis nekih realnih brojeva je *double*. Za njega važe sve relacije i operacije kao i za *integer*, izuzev modulo operacije (ona nema ovde svrhe, zar ne? - prim.aut.).

Poslednji tip podataka koji ćemo pomenuti u ovom tekstu, jeste *bool* (*boolean* u nekim drugim jezicima). Za njegov opis je takođe potreban *1B* i može imati samo dve vrednosti, a to su *true* ili *false*, odnosno tačno i netačno. Operacije koje se mogu izvršiti na ovim tipovima, jesu operacija dodele, logičko „i” (&&), logičko „ili” (||) i logičko „ne” (!). Ovaj tip podataka vraća sve relacije na drugim tipovima podataka, pa se ovaj tip, iako ne na prvi pogled, dosta koristi u uslovnom grananju, o kojem će biti više reči u narednom tekstu.

Promenljive

Sada kada znamo okvirno čemu služe neki osnovni tipovi podataka, možemo da ih iskoristimo i sačuvamo nešto u memoriju.

Svaka promenljiva mora da ima svoj tip podataka. Promenljive bi obično trebalo da imaju vrednost, ali će se vaš program uspešno prevesti čak i kad neka promenljiva ne bude imala dodeljenu vrednost. Naravno, treba napomenuti da ovo isključivo zavisi od podešavanja vašeg prevodioca. Svaka promenljiva se definiše na sledeća dva načina:

```
tip_podataka ime_promenljive =
vrednost;
tip_podataka ime_promenljive;
```

U prvom slučaju je promenljivoj

dodeljena odmah vrednost, dok u drugom slučaju ona nema vrednost (ima neku nasumičnu vrednost koja nam nije od neke koristi). Što se tiče drugog slučaja, promenljivoj možemo dodeliti kasnije vrednost, korišćenjem operatora dodele:

```
ime_promenljive = vrednost;
```

Slede primeri definisanja nekih promenljivih:

```
int a = 1;
int b = 1000;
int c = -5645;

char d = 'a';

bool e = true;

double f = 0.0015;
```

Učitavanje promenljivih sa standardnog ulaza

Svi oni koji redovno koriste konzolne programe, znaju da oni mogu da pitaju korisnika za neki podatak tokom svog izvršavanja i da čekaju da korisnik taj podatak unese. Ovaj podatak se potom smešta u memoriju i kasnije se koristi i obrađuje. Posle obrade rezultat se ispisuje na standardni izlaz, korišćenjem nama svima dobro poznate funkcije *printf* koju smo koristili u našem prvom programu da bismo ispisali frazu *Hello, world!* na standardni izlaz.

Neko je rekao da se uz primer najbolje uči, a mi kažemo da se uz primer

najbolje objašnjava.

Program koji učitava dva broja i potom ispisuje njihov zbir:

```
#include int main()
{
    int a;
    int b;
    printf("Unesite dva broja:
");
    scanf(" %d %d", &a, &b);
    int zbir = a+b;
    printf("Zbir je: %d",
zbir);
    return 01
}
```

Kao što možemo videti, koristili smo isto *main* funkciju kao prošli put, učitali smo istu biblioteku, ali nam je kod unutar naše *main* funkcije malo drugačiji. Prvo smo definisali dve promenljive tipa *int* koje nam služe da u njih učitamo zadate brojeve. Potom, pomoću nove funkcije *scanf* učitavamo te brojeve. Ova funkcija zahteva *string* koji govori kako da parsira ulaz, pa tako funkcija zna da mi očekujemo dva cela broja (*%d*) razmaknuta jednim razmakom. Potrebno je primetiti da je razmak napravljen i na početku *stringa*, što funkciji govori da ne obraća pažnju na specijalne znakove, kao prelazak u novi red (enter). Ovo znači da vi možete uneti prvi broj, pritisnuti taster enter, a zatim uneti drugi broj i opet pritisnuti enter. Funkcija traži i imena promenljivih u koje će biti učitane vrednosti. Potrebno je primetiti znak & na početku imena obe promenljive. Ovaj znak govori da se podatak upiše na lokaciju adresa tih promenljivih u

memoriji. Ako vam ovo izgleda zbudujuće, verujte - niste jedini. O adresama u lokacijama u memoriji ćemo pisati u budućim brojevima kada budemo diskutovali o pokazivačima. Kao što ste mogli da primetite, obe funkcije (*scanf* i *printf*) koriste specijalne izraze da bi se naveo tip podataka. Tako *%d* govori da se na ulazu očekuje *int* tip podataka, ili da se na izlazu ispisuje vrednost promenljive tog tipa. Neki ostali znakovi za formatiranje su:

%c - karakter tipa *char* *%lf* - realni broj, tip podataka *double* *%d* - ceo broj, *int* tip podataka

Česte sintaksne greške

Ostavićemo za kraj promenljive i tipove malo po strani i vraćamo se ponovo na početak. Hajde da vidimo, koje sve greške mogu da se naprave i kako prevodilac reaguje na njih.

Posmatrajmo sledeći kod:

```
#include int main()
{
    printf("Hello, world!\n");
    return 0
}
```

Ako ovo pokušamo da prevedemo sa korišćenjem *gcc* prevodioca kao u tekstu koji je objavljen u prošlom broju, dobićemo sledeću grešku:

```
error: expected ';' before '{'
token
```

Kao što vidimo, negde smo izostavili „;“,



tj. tamo gde je trebalo da se nalazi. Potrebno je pogledati u red gde stoji `return 0` bez potrebnog znaka na kraju.

Pogledajmo sada ovaj kod:

```
#include int main()
{
    printf("Hello,
world!\n");
    return 0;
}
```

Šta kaže prevodilac?

```
fatal error: studio.h: No
such file or directory
```

Vidite li gde je greška? Omašili smo naziv biblioteke!

A ovo?

```
#include int main()
{
    printf("Hello,
world!\1");
    return 0;
}
```

izlaz:

```
warning: unknown escape
sequence: '\1' [enabled by
default]
```

Kao što vidimo, specijalan karakter „\1” ne postoji. Potrebno je primetiti da će se po početnim podešavanjima ovako nešto prevesti uz upozorenje.

Završetak

U ovom tekstu smo vam dali osnovne informacije o tipovima podataka, promenljivama i standardnom ulazu i izlazu. Isto tako smo pokazali neke standardne greške koje prave početnici, ali i oni iskusni, kako bismo vam ukazali na greške koje vaš prevodilac izbacuje i koje vam mogu pomoći da iste greške uklonite. Ovde treba napomenuti da prevodilac ne upozorava na greške koje nisu striktno vezane za sam jezik, pa vas tako neće upozoriti ako ste napravili deo programa koji može da se zakuca, ili ako ste napravili sigurnosni propust.

Learn C Programming



Nekoliko filozofskih misli o softveru:

Razvoj softvera

Autor: Dejan Maglov

Uvod

Ma koliko se čovek, kao vrsta, trudio da bude kreativan, uvek se to svodi na oponašanje prirode. Čovekov ideal je veštački stvoriti nešto što je priroda odavno već stvorila. U informatičkom svetu, ideal je kreirati veštačku inteligenciju po uzoru na nas same, odnosno naš mozak. Šezdesetak godina razvoja informatike je dovoljan period da možemo da podvučemo neke prve crte i analiziramo razvoj iz filozofske perspektive.

Kreacionizam ili evolucija

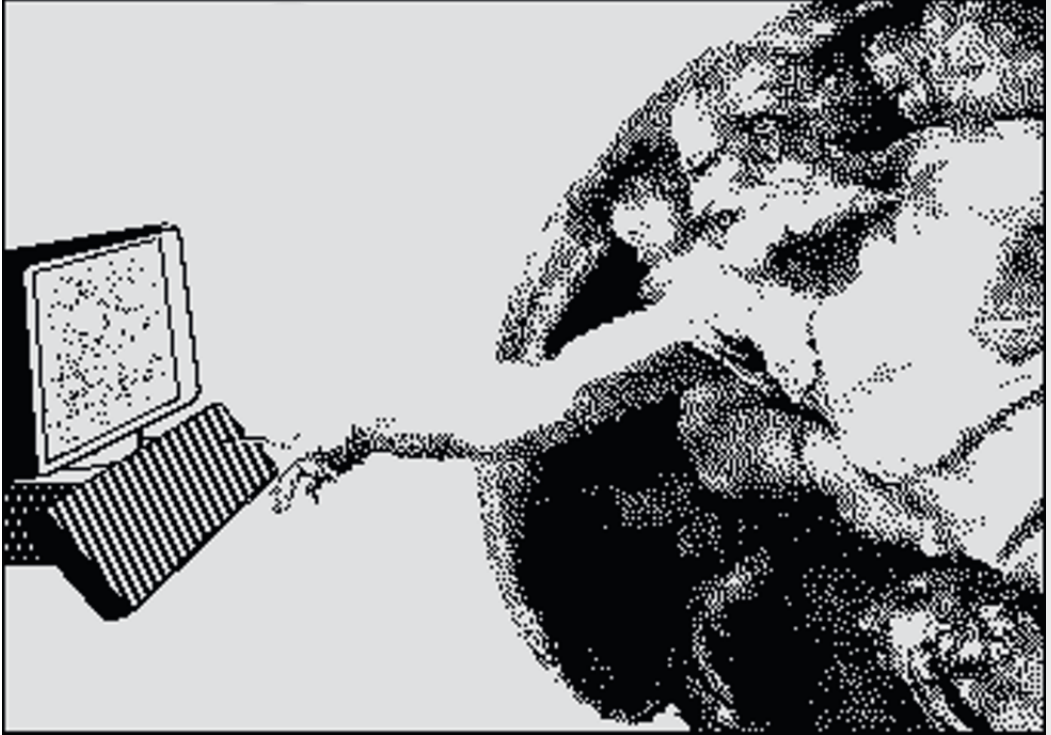
Proučavajući prirodu, čovek je razvio dve vodeće teorije nastanka i razvoja života (postanka), kreacionističku i teoriju evolucije. Kreacionistička teorija je starija i zasnovana je na dogmi. Kreacionističku teoriju podržava većina vodećih religija

sveta bez veće želje za dokazivanjem ispravnosti iste. Tokom istorije čovečanstva je, čak, bilo opasno po život dovesti u pitanje postojanje Stvoritelja života (Boga). Stvoritelju nije samo dodeljena uloga kreatora života, nego se tvrdi da Stvoritelj upravlja kopiranjem, modifikovanjem, kombinovanjem i uništavanjem svojih kreacija.

Za razliku od kreacionističke teorije, evolucionistička teorija nije baš jasno i nedvosmisleno objasnila sam početak nastanka života (postoje teorije usmerene u tom pravcu). Po evoluciji, dalji razvoj života je prepušten stihiji po nekim jasnim pravilima.

Pravila su:

- Jedinica ima određen vek trajanja
- Jedinke se same razmnožavaju (kopiraju)
- Polno razmnožavanje (kombinovanje) omogućava modifikovanje originala
- Najjače i najprilagodljivije jedinice



opstaju

Evolucijska teorija nije nikad 100% negirala postojanje Stvoritelja života (možda pod uticajem religije), ali ga je svela samo na Kreatora i nekoga ko je uspostavio pravila po kojima se evolucija dalje odvija bez njegove direktne intervencije (jedna od teorija).

Softver i teorije postanka

Prosto je neverovatno koliko softver podseća na živi organizam.

Softver ima:

- svoj životni vek

- ima svoje ekosisteme (hardver)
- razmnožava se kopiranjem
- može se kombinovati sa drugim softverom (kombinovanje i modifikacija)
- može biti jednoćelijski (u jednoj datoteci) ili višećelijski (koristi više različitih biblioteka)
- može biti složen organizam sa različitim funkcijama (//OS// sa aplikativnim softverom)

Smrt softvera nastaje:

- nestankom njegovog ekosistema (hardvera)
- zastarevanjem
- biva „pojeden” od strane nekog većeg



softvera

- izumire zato što postoji bolji softver za istu funkciju

Ovo su prirodni uslovi za „život” softvera. Vlasnički softver ima još nekoliko uslova koji zavise od njegovog vlasnika.

Nismo sigurni da li je neko na ovaj način razmišljao o softveru. U svakom slučaju, čovek je krajem dvadesetog veka dobio priliku da preko softvera eksperimentalno proveri teorije postanka.

Kreacionizam

Kreacionističku teoriju proverava vlasnički softver. Zakon o zaštiti intelektualnih prava je stvorio uslove da ovaj softver uvek bude u vlasništvu svog stvoritelja, tako da stvoritelj vlasničkog softvera pored kreiranja ima moć da kontroliše kopiranje, modifikovanje, kombinovanje i smrt svoje kreacije.

Vlasnički softver nema toliko vidljive karakteristike „živog” organizma. On je samo roba sa svojom tržišnom vrednošću koja je ekvivalentna upotrebnosti vrednosti (svrsi).

Karakteristike:

- Svaki vlasnički softver je nastao kao potreba da zadovolji neke potrebe njegovog stvoritelja (funktionalne, materijalne, duhovne, i druge potrebe)
- Njegov život je ograničen uglavnom na period dok služi svom vlasniku
- U toku svog života može biti

kontrolisano kopiran, modifikovan i kombinovan

- Poneki može u toku svog života i da se „odmetne” u „divljinu” i pomoću pirata bude kopiran nezavisno od svog stvoritelja
- Život mu prestaje kad više ne zadovoljava potrebe svog vlasnika
- Retki srećni vlasnički softveri nakon svog komercijalnog života budu pušteni u „divljinu” sa besplatnom licencom da umru „prirodnom” smrću bez mogućnosti da se modifikuju i da produže svoj život.

Sa druge strane, ako se život na Zemlji posmatra kroz prizmu kreacionizma, onda može da se postavi filozofsko pitanje da li su i živa bića zaista živa, ili se sve odvija prema nekom Božijem planu bez sopstvene volje Božijih kreacija. Sa tim u vezi, onda i večno pitanje svrhe postojanja čoveka, dobija na važnosti. Ako se čovek odmetne i prestane da obavlja funkciju zbog koje je kreiran, da li će ga Bog (Stvoritelj) ukinuti kao nešto nepotrebno?

Postoji još jedan problem vezan za kreacionistički pristup softveru. Čovek nije savršen i ne može da igra ulogu Boga. Da li je neko čuo da je Bog ukinuo neku svoju kreaciju zato što je kreirao neku savršeniju, pa zbog toga nema vremena da održava prethodnu verziju? Ili, da je ukinuo neku svoju kreaciju zato što se oženio i ima preča posla od održavanja svoje kreacije?



Evolucija

U svetu softvera nije moguće da on nastane spontano. Slobodni softver takođe ima kreatora kao i vlasnički softver, međutim njegov razvoj više liči na evoluciju nezavisnu od kreatora.

Karakteristike:

- Kreator slobodnog softvera stvara proizvod sa određenom funkcijom i uspostavlja pravila daljeg razvoja. Nakon toga, slobodan softver se pušta u „divljinu” (nije obavezno, ali bez puštanja softvera u divljinu, on nema sve karakteristike slobodnog softvera)
- Na dalji razvoj slobodnog softvera kreator može da ima presudni uticaj, ali i ne mora
- Softver u ovom slučaju „živi” svoj slobodan život po principu – najjači i najprilagodljiviji opstaju
- Najjači projekti okupljaju zajednice

koji se dalje brinu o razvoju

- Neki evoluiraju u potpuno nepredvidljivim pravcima iz ugla kreatora

Kada na ovakav način posmatramo softver, on nije samo slobodan za kopiranje, modifikovanje i kombinovanje, nego ispada da je suštinski slobodan po principu „žive” jedinke.

Ovaj model razvoja softvera se pokazao kao vrlo efikasan. Iako je nastao znatno kasnije nego model vlasničkog softvera, i pored svih prepreka je po razvoju stigao vlasnički softver, a u nekim elementima i prestigao.

Mane i prednosti vlasničkog i slobodnog softvera

Vlasnički softver je za sada ekonomski jači. Iz ove tvrdnje proizilaze sve



prednosti ovog modela. Novac ovom modelu obezbeđuje timove najboljih, najkvalitetnijih kreativaca sa zacrtanim rokovima, što omogućava da manji timovi obavljaju više posla u kraće vreme. Po potrebi taj novac prilagođava i ekosisteme (čitajte hardver) po meri kreiranog softvera. Novac omogućava bolji marketing za ovaj softver i bolje, preciznije skeniranje potreba korisnika softvera.

Slobodni softver se sa druge strane pomalo ponaša kao divlja zver, koja ukoliko nije strogo pod kontrolom, može da se razvija u nepredviđenim pravcima. Ovakvo poređenje niste verovatno nigde mogli da čujete. Ako ste dugogodišnji korisnik slobodnog softvera, setićete se bar nekoliko primera da ste naleteli na idealan softver za vas, koji je vremenom sa svojim razvojem malo „podivljao” i postao nepristupačan za vas.

Slobodni softver je sada u divljini i male su šanse da će ga više iko zaustaviti u njegovom razvoju. Dok je u početku trebalo kreirati prve slobodne softvere, sad je veće umeće pohvatati ih u divljini i „pripitomiti” ih da obavljaju zadate radnje. Ipak, ne treba paničiti – svi najbolji projekti slobodnog softvera okupljaju oko sebe *FLOSS* zajednice, koje striktno vode računa o svom projektu i ne dozvoljavaju da on „podivlja”. *FLOSS* zajednice jesu pravi krotitelji divljih zveri i vode računa ne samo o tome da on ne „podivlja”, nego i da dugoročno radi posao za svoje korisnike na najbolji i

predvidiv način. Upravo *FLOSS* zajednice su jedan od ključnih faktora uspeha slobodnog softvera.

Umesto zaključka

Šta je poenta ove priče? Prevažodno upoređenje trenutnog stanja i odnosa između vlasničkog i slobodnog softvera kroz teorije koje su svima jasne ili bar mislimo da su nam jasne. Cilj nije bio da se ocrni jedna ili druga strana i jedan koncept postavi kao jedino ispravan. Ovi koncepti će još dugo koegzistirati bez mogućnosti da jedan konačno pobedi. Verovatnije je da će u budućnosti doći do približavanja ovih koncepata.

Sa rastom upotrebljivosti slobodnog softvera povećavaće se i interesovanje za njega. To će dovesti do dela komercijalizacije slobodnog softvera, ali ne u smislu da će se softver prodavati, nego će se prodavati održavanje i podrška jer uvek će biti korisnika koji će radije da plate održavanje nego da to sami uče. Sa druge strane, padaće cene vlasničkog softvera, što će verovatno dovesti do oslobađanja dela tog softvera i prepuštanja *FLOSS* zajednici razvoj i održavanje manje bitnih delova vlasničkog softvera sa ciljem uštede i koncentracije vlasničkih razvojnih timova na, za njih, bitne delove. Ovo su naše pretpostavke na osnovu već uočenih tendencija. Da li će to baš tako biti? Videćemo u budućnosti.

Vaš posao, *open-source* posao

(1. deo)

Autor: Marko Kažić

Čak i danas, nažalost, pojmovi „posao” i „*open-source*” u istoj rečenici izazivaju nevericu, čuđenje i lavinu pitanja. Ipak, informacione tehnologije prolaze kroz period fundamentalne promene filozofije i načina na koji se nove tehnologije razvijaju, konzmiraju, reklamiraju i prodaju.

Šta je *open-source* u moru tehnologije koju koristimo svakodnevno? Naš *Android* telefon, ili možda jezgro našeg *iOS*-a? Naš pregledač i u njemu *Facebook* časnikanje zasnovano na *XMPP*-u, *SQLite* u *Skypeu*, *MySQL* u vašem, isto toliko *open-source Wordpress* blogu koji pokreće *open-source PHP*, sa *open-source Apacheom* ili *nginxom*. Pričamo naravno o slučaju u kojem se još niste oslobodili vlasničkih tehnologija kao što je *Windows*. Dakle, ovo jeste era *open-sourcea*.

Pomenuti softver uglavnom ima jednu zajedničku osobinu: To su komercijalni proizvodi zasnovani na *open-sourceu*. Kompanije kao što su *Google*, *Amazon*,

Apple, pa čak i *Microsoft*, zavise od *open-sourcea* i ugrađuju ga u svoje komercijalne proizvode. Ono što je daleko bitnije za njihovo poslovanje, jeste da su njihovi celokupni modeli poslovanja zasnovani oko *open-sourcea*. Njihova zarada i budućnost je u *open-sourceu* upravo zbog toga što se *open-source* postavio na tron suverenog vladara inovacije i pokretača celokupne informacione industrije današnjice (a i *sutrašnjice*, *prim. aut.*), a to je činjenica koja se ne sme zanemariti.

Open-source sam po sebi nije i nikad neće biti model poslovanja. *Open-source* je pokret, filozofija i bolji put. Kako smo već pomenuli, komercijalni entiteti postaju svesni pozicije *open-sourcea* u *IT*-u današnjice i shodno tome nalaze nove načine da teret prihoda prebace sa samog suštinskog proizvoda na sve sekundarne domene proizvoda. To mogu biti dodatne funkcije, održavanje, edukacija, pomoć, implementacija kao i uparivanje jednog besplatnog i *open-source* proizvoda sa drugim



komercijalnim. Zato njihovi modeli poslovanja danas ciljaju kako poslovne korisnike tako i opštu populaciju. Recimo, *Google* vam neće prodati *Chrome*, ali će zato svoje servise i reklame, na kojima ima najveći profit, lakše i bolje servirati krajnjem korisniku. Kontrola koju *Google* ima nad načinom na koji pristupa konzumentima, jeste kontrola koju bi svako želeo. Upravo *open-source* je omogućio *Googleu* da napravi solidnu platformu na kojoj će brže i bolje servirati svoje komercijalne proizvode.

Možda ćete reći: „Dobro, ali mi nismo *Google*. Na kojim osnovama može firma u povelju da izgradi profitabilan posao koji se bazira na *open-sourceu*?” Pitanje je

svakako na mestu, a cilj ovog članka i delova koji slede, jeste da vas kroz primere upoznaju sa skoro svim modelima poslovanja na *open-sourceu*, da vam pokažemo uspehe ali i neuspehe kompanija i njihovih projekata, i nadamo se da ćete kada serijal privedemo kraju, imati jasan odgovor na vaše pitanje. U sledećem delu predstavimo model poslovanja začetnika cele ove priče, *Red Hat Inc.*-a – prodaju profesionalnih usluga, što je jedan od najranijih poslovnih modela zasnovan na *open-sourceu*, zatim ćemo upoznati *Canonical* i tako dati deo odgovora na pitanje koje sigurno muči svakoga ko želi da zaradi, ne zaboravljajući pritom *open-source* zajednicu.



Tor mreža:

Online anonimnost i sigurnost

(3. deo)

Autor: Petar Simović

Ako ste prosečan korisnik *Linuxa* i želite malo privatnosti u uzburkanim vodama interneta, *Tor* će uraditi i mnogo više od toga za vas. Međutim, ako ste pak malo više nepoverljivi i skeptični po pitanju sigurnosti ili paranoični po pitanju prisluškivanja, postoje još neki koraci koji se mogu preduzeti s ciljem poboljšanja sigurnosti svakog korisnika. Razloga za nepoverenje uvek ima, čak i u najsigurnijim sistemima i dobro je da se što češće kontrolišu i unapređuju.

Sigurno ste u proteklih desetak dana čuli ili pročitali za sigurnosni propust u *OpenSSL* protokolu, ilustrativno nazvanog *Heartbleed*, koji je ugrozio dve trećine sajtova na internetu i potegao mnoga pitanja i podigao veliku prašinu. Ako se pitate da li je *Tor* ostao pošteđen u ovoj sajber čistki - odgovor je negativan. Naime, potencijalno ugroženi korisnici su oni koji koriste programe u kojima je implementirana ranjiva verzija *OpenSSL* protokola tj. verzija 0.1.0, sve ostale verzije su sigurne bar

što se tiče *Tor* korisnika. Takođe, *Tor* je počeo da stavlja na crnu listu sve izlazne čvorove koji nisu ažurirali verzije *Tor* softvera i *OpenSSL*-a.





Kao što znamo, ništa nije sto posto sigurno, ali to ne znači da treba odustati i odreći se prava na privatnost. Bagova će uvek biti, zato je zlatno pravilo uvek koristiti najnovije verzije softvera, po mogućstvu čim se pojave.

Osim osnovnog softvera za anonimno surfovanje, treba otići još nekoliko koraka dalje i preispitati sigurnost sistema koje svakodnevno koristimo.

Prvi takav korak u hijerarhijskoj organizaciji, logično, bio bi sam operativni sistem. OS je platforma za pokretanje svih programa i stoga mora biti takav da možemo da verujemo da će on biti siguran i da neće svojim radom ugroziti sigurnost programa koji se oslanjaju na njegov rad. U prošlim delovima smo već pomenuli da je osnovna mera predostrožnosti da se biraju samo operativni sistemi i programi otvorenog koda (*open-source*), a poželjno je da budu i pod *GPL* licencom. Za ekstremne slučajeve, operativni sistemi koji pružaju maksimalan nivo bezbednosti su *Tails*, *Whonix* i jedan noviji OS lepog imena *Snowden tribute* (<http://goo.gl/Dvsu1x>) za koji je potrebno izdvojiti 4.9€ u vidu donacije. Nešto više o programima za bolju *online* anonimnost i sigurnost za razne platforme možete pogledati na interesantnom sajtu *PRISM Break* na adresi (<http://goo.gl/z63xPn>).

Sledeći korak bi bio, recimo *BIOS* računara, gde preporučujemo korisnicima izbegavanje *UEFI BIOS*-a ili njegovu onemogućavanje tamo gde je već prisutan. Ukoliko je moguće, preporučujemo instalaciju *Coreboota*

(<http://goo.gl/fjH2t3>) nekadašnjeg *linux-biosa*, pošto postoji podrška samo za veoma ograničen broj matičnih ploča i njihovih verzija. Kad smo već kod *BIOS*-a, skrenuli bismo pažnju na *firmware* ruter kao na još jedno mesto gde postoji alternativa vlasničkom softveru (*propriety software*, u vidu sledećih pet *open-source firmwarea*: *OpenWrt* (<http://goo.gl/KKBmm8>), *DD-WRT* (<http://goo.gl/aVCHDQ>), *Tomato Firmware* (<http://goo.gl/Z3ZuFb>), *Gargoyle* (<http://goo.gl/TVa6T1>) i *PAPARoute* (<http://goo.gl/O9y7kk>).



PAPAROUTER
Plug And Play Anonymity Router

I poslednji, ali ne i najmanje važan, jeste sam hardver. Tu postoje neke alternative danas najpoznatijim proizvođačima korisničkog hardvera u vidu raznih *arduino*, gotovih hardverskih rešenja po principu *plug-and-play* poput nemačkog *enigmax* (<http://goo.gl/a3Lq2P>) i *freedombox* (<http://goo.gl/zd4RAF>), a tu su i *raspberry Pi* i *Project Novena* kao potpuno otvoren laptop. O poslednja dva je bilo reči u 21. broju *LiBRE!* časopisa.

U okviru *Tor* mreže postoje mnogi skriveni servisi koji čekaju da budu otkriveni. U njihovom otkrivanju nam mogu pomoći specijalizovani pretraživači poput već pomenutog *TorSearcha* (<http://goo.gl/FnNe4B>), a ove nedelje se pojavio i najnoviji pod nazivom *Grams* (<http://goo.gl/68hKa7>) i počeo da indeksira i one najopasnije sajtove po kojima



je ova mreža i poznata. Ovakvi specijalizovani pretraživači umnogome pomažu *Tor* korisnicima jer su sajтови često prinuđeni da menjaju svoje adrese i lokacije što otežava korisnicima njihovo pronalaženje. Ovde ćemo skrenuti pažnju na nekoliko programa koji nisu onako svetleći i šljašteći kao većina onih na koji smo navikli uživajući u udobnošću interneta i povoljnostima novih tehnologija.

Napomenuli bismo da su ovo samo neki, egzotični *commandline* programčići za entuzijaste *geove* i one najugroženije. Za proveru internet pošte imamo *mutt* (lepo kratko uputstvo imate na: <http://goo.gl/b7Qxfy>), za slušanje muzike projekat sa *githuba* *cmus* (<http://goo.gl/UNw0We>), za tvitovanje *ttytter* (<http://goo.gl/rbUoRf>), za IM imamo *Pidginov Finch* (<http://goo.gl/bWbFuI>), za IRC chat tu je *Irssi* (<http://goo.gl/aE5AKR>) i za surfovanje netom tu su *Lynx*, *Links* i *ELinks* (<http://goo.gl/W1u0cP>).

Iako ovo nisu najudobniji programi za prosečnog korisnika, mogu u ekstremnim slučajevima (ako mislite da ste iz nekog razloga pod stalnom

prismotrom, ili ako ste tajni dopisnik) da vašim neprijateljima znatno otežaju posao. Tu su i oni standardni programi i aplikacije za koje nije loše znati, poput alternativnih pretraživača interneta koji su preporučljivi i van *Tor* mreže za svakodnevnu upotrebu, a u duhu izbegavanja globalnog nadzora *Googlea*: *ixquick* (<http://goo.gl/OgdtDX>), *Startpage* (<http://goo.gl/BS1RLZ>) ili sve popularnijeg *DuckDuckGo* (<http://goo.gl/FUIcPI>). Pored pretraživača za *online chat*, tu je *Cryptocat* (<http://goo.gl/qdJ4wq>) i za otpremanje videa bez registracije tu je *vidd* (<http://goo.gl/5oPf2K>), a za instant deljenje multimedije uz *chat* tu je *Volafile* (<http://goo.gl/UuRpdK>), novi servis poput starog dobrog *4chana*. U okviru *Tora*, naveli bismo samo projekte sa *githuba* kao *TorBirdy* i *TorChat*.

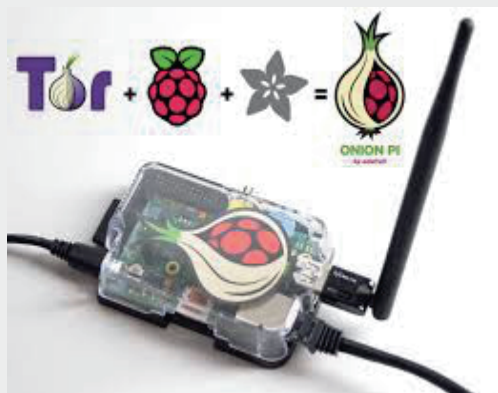
Spomenućemo ovde tri platforme za dopisnike medijima, recimo *WikiLeaksu*, koji su poput *Snowdena* sigurno među najugroženijima, što se privatnosti i života tiče. *Strongbox* od *The New Yorkera* (<http://goo.gl/03ziZp>), *Secure-Drop* organizacije *Pressfreedom-foundation* (<http://goo.gl/10CTna>) mogu da se obezbede putem *RISEUP* oragnizacije koja pruža *VPN*, *email* i





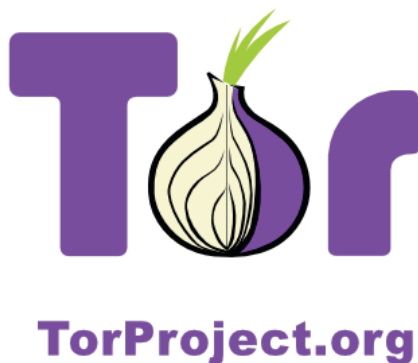
chat usluge drugim grupama ljudima i organizacijama koje se bore za socijalne slobode slobode govora na internetu, a sve se dodatno rutira kroz Tor (<http://goo.gl/mnRoki>).

Dodaćemo još da postoji i uputstvo za instalaciju sopstvenog izlaznog čvora uz pomoć *Raspberry Pia* na siguran način, a da se ne upadne u nevolje sa eventualnim protokom nedozvoljenog sadržaja kroz isti, ali ga autor ovog teksta još nije isprobao jer je pomalo skeptičan (prim.aut.), a vi kako vam volja ali pažljivo: (<http://goo.gl/DGzqU6>).



Nije teško uvideti da su najkorišćenije i najrasprostranjenije aplikacije i programi, a u poslednje vreme i sam hardver (<http://goo.gl/ABEu1u>) potencijalno najopasniji jer onome ko pronade ranjivost u njima i zloupotrebi je, donosi veliku moć i uticaj na najveći deo internet populacije. Mnoge organizacije se trude baš to, da pronadu ranjivost i zloupotrebe je. Međutim, ima jedna svetla tačka, a to je da se svest o ugroženoj privatnosti, bezbednosti i beskrupuloznoj softverskoj i hardverskoj špijunaži sve više širi u

javnosti, a sa njom i slobodne alternative vlasničkom softveru, a sada i hardveru. Ostaje da vidimo kakvu leguru sve mogu da naprave i šta će nam doneti u skoroj budućnosti.





_.underscore

Autor: Zlatan Vasović

Underscore je *JavaScript* biblioteka koja sadrži veliki broj korisnih alatki, ali bez proširivanja osnove programskog jezika. To je biblioteka koja teži da bude minimalistična i brza, kako ne bi (previše) usporila vaš sajt.

Underscore sadrži i imitaciju nativnih *JavaScript* metoda, ukoliko one ne postoje. Ipak, te imitacije nisu proširenje osnove programskog jezika, već obične funkcije biblioteke.

Statistika

Underscore je najpopularniji paket na *npm* registru. On je veoma popularan i na drugim registrima *JavaScript* paketa.

- 5400+ *npm* paketa koji koriste *Underscore*
- 70+ starova na *npm* registru
- 10000+ starova na *GitHubu*
- 2000+ forkova na *GitHubu*

Instalacija

Underscore radi i u pregledačima i u

node.js okruženju, pa ga možete koristiti na više načina. Možete ga preuzeti direktno sa sajta, ili dobiti nekim od *JavaScript* menadžera paketa (npr. *Require.js*, *npm* i *Bower*).

node.js (*npm*)

```
npm install underscore
```

Require.js

```
require(["underscore"] // ...
```

Bower

```
bower install underscore
```

Component

```
component install  
jashkenas/underscore
```

API

Underscore se najčešće koristi sa promenljivom `_` (donja crta). Tada je



sintaksa `_.metoda(argumenti)`.

Najčešće korišćene metode

- `_.each` (`_.forEach`) — imitacija native ciklične petlje za nizove i objekte
- `_.map` — imitacija native transformatora članova niza ili objekta
- `_.invoke` — poziva metodu i primenjuje je na niz ili objekat
- `_.reduce` (`_.inject`, `_.foldl`) — funkcijom spaja sve vrednosti niza u jednu
- `_.filter` (`_.select`) — imitacija native filtriranja niza
- `_.every` — imitacija native provere da li svi članovi niza prolaze test
- `_.some` — imitacija native provere da li bilo koji član niza prolazi test
- `_.indexOf` — imitacija native metode za pronalaženje mesta člana niza

Underscore-contrib

Za sve one kojima nešto nedostaje u *Underscoreu*, tu je *Underscore-contrib*, proširenje *Underscorea*. Tu se nalaze sve funkcije koje bi ili previše uvećale biblioteku, ili bi previše usporile učitavanje stranice. Više informacija možete naći na

<http://documentcloud.github.io/underscore-contrib/>.

Alternative

Postoji više alternativa *Underscore* biblioteci. Najpopularnije su *Lo-Dash*

(<http://lodash.com/>) i *Lazy.js* (<http://danieltao.com/lazy.js/>).

Prelazak na alternative je vredan, ako brinete o performansama.

Linkovi:

- [1] Sajt: <http://underscorejs.org/>
- [2] Izvorni kod: <https://github.com/jashkenas/underscore>
- [3] Objašnjenje izvornog koda: <http://underscorejs.org/docs/underscore.html>



JavaScript

UNDERSCORE.JS

LIBRE!

Časopis o slobodnom softveru



2 godine sa
Vama.

Hvala što nam verujete!