

Mart 2015. Broj 34

LIBRE!

Časopis o slobodnom softveru



SUSE Linux
Enterprise Server

12

SUSE Linux Expert Days
2015 Coming to a city near you!

Frankfurt, 20. januar



JOŠ IZDVAJAMO

Vagrant
LUGoNS BarCamp №4



Creative Commons Autorstvo-Nekomercijalno-Deliti pod istim uslovima

Reč urednika

CopyRight

Odavno nismo „filozofirali“ u reči urednika. Pošto trenutno nema velikih događanja u projektu, iskoristićemo priliku da vas još jednom podsetimo zašto projekat između ostalog postoji.

Odmah na početku naglašavamo da LiBRE! nije fanatično protiv vlasničkog softvera. Nismo ni protiv prava da profesionalni programeri rade i žive od svoje intelektualne svojine. Samo tržište uglavnom ima dovoljno svojih mehanizama da odredi koliko je nečiji proizvod dobar i koristan, a prema tome i kolika je njegova materijalna vrednost. Ono što nama smeta i protiv čega se „borimo“ je kopirajnt (eng. *copyright* - bukvalno, pravo na kopiranje a zapravo vlasničko pravo), koji je proizvod poremećaja tržišta, to jest stvaranja monopola nekog proizvoda. Jedino proizvod koji je stvorio monopol omogućava zloupotrebu kopirajnta. Čovek koji ima potrebu da se sam preveze od mesta A do mesta B ima izbor da automobil kupi, ili, ako mu treba, jednokratno da pozajmi od rođaka. Kad dođe do poremećaja na tržištu, onda korisnik gubi mogućnost izbora, a vlasnik može da kaže: „Ovaj proizvod nije na prodaju, možete samo da ga iznajmite uz odgovarajuću naknadu na određeno vreme. Ne smete da ga otuđite, jer nije vaš i ne smete da ga pozajmljujete i delite.“ Ovo vlasniku omogućava praktično da svoj proizvod prodaje više puta istom kupcu i pri tome ima obavezu samo da izrađuje rezervne delove, ali ne i da održava proizvod u radnom stanju tokom vremena iznajmljivanja.

Nuspojava monopolističkog položaja je mogućnost da nešto što nije u redu u proizvodu proglasiš za standard i da kažeš da to tako treba da radi.

Ovakav nakaradni kopirajnt je proizašao iz zakona o zaštiti intelektualne svojine koji je još u osamnaestom veku osmišljen da zaštiti pisce od neovlašćenog preštampavanja koji je uzeo maha pojavom štamparske prese. Svoju kulminaciju nakaradnosti doživljava kroz muzičku i softversku industriju od sedamdesetih i osamdesetih godina prošlog veka. Nije sporno da treba zaštititi intelektualnu svojinu i da treba omogućiti autorima, umetnicima i naučnicima da žive od svog intelektualnog rada. Nije moralno da pojedini zaslužni umetnici i naučnici umru u bedi, a toliko su zadužili svet svojom intelektualnom zaostavštinom. Takođe nije dobro da se nosioci intelektualne svojine tokom života bore sa nemaštinom, koja ih sprečava da još više doprinesu razvoju



celokupnog društva. Nakaradnost počinje kad kopiraj, umesto toga da bude zaštita autorskih prava, postane zaštita korporacijskih interesa. Samo korporacije su u stanju da smisle višestruke naknade za jedan te isti proizvod istom korisniku i da ih sprovedu u delo. Zakon o autorskim pravima koji je deo američkog ustava bio je teritorijalno orijentisan i važio je samo za teritoriju SAD-a. Pod uticajem korporacija u svaki međunarodni sporazum se ugrađuje i saglasnost potpisnica da prihvataju ovaj zakon i tako on postaje eksteritorijalni. Zakon o autorskim pravima je bio vremenski ograničen, što je omogućilo nosiocima prava da samo određeno vreme naplaćuju intelektualnu svojinu i teralo ih je da dalje nastave da rade. Današnji kopiraj je ograničen na sto godina ili doživotno plus pedeset godina nakon smrti autora. Ovo sigurno nije uvedeno zbog zaštite naslednika autorskih prava nego da bi korporacije izvukle i poslednju kap koristi od intelektualnih prava.

Ovakav nakaradni kopiraj nije podsticajan - a da u softverskoj industriji nije prepoznat kao potpuno pogrešan, bili bismo unazađeni za narednih sto godina. Kao odgovor na ovako nakaradni kopiraj nastao je slobodan softver koji je u većini sektora informacionih tehnologija prevazišao korporacije brže se prilagođavajući novim tehnologijama. Kao rezultat toga imamo dominaciju slobodnog softvera na superkompjuterima, mobilnim uređajima, veb serverima, veb aplikacijama, komunikacijskim uređajima i „pametnim“ kućnim uređajima.

Pričati danas o slobodnom softveru je i dalje pionirski posao. Još uvek se korisnici slobodnog softvera smatraju štreberima a slobodan softver preteškim za običnog korisnika. Prava istina je da je slobodan softver već svuda oko nas i da ga koristimo na mobilnom telefonu, tabletu, na internetu, na kućnim uređajima, u automobilima, pri prognozi vremena... Zato LiBRE! časopis postoji - da bi od drveta svi ugledali šumu.

Za kraj reči urednika moramo da se izvinimo Goranu Stričiću zato što smo u prošlom (februarskom) broju propustili da se javno zahvalimo za članak o LiBRE! časopisu u februarskom broju Sveta kompjutera. Ne bismo voleli da ispadnemo nezahvalni, jer nam podrška Sveta kompjutera mnogo znači.

Svi koji na ma koji način žele da se uključe u projekat časopisa, mogu da se prijave, komentarišu, hvale i kritikuju nas pišući na našu već poznatu adresu elektronske pošte - [libre \[et\] lugons \[dot\] org](mailto:libre[et]lugons[dot]org) .

Do čitanja,

LiBRE! Tim

Sadržaj

Vesti

str. 6

Puls slobode

LUGoNS barcamp №4

str. 10

Predstavljamo

Nomacs - Image lounge

str. 13

Kako da...?

Uvod u programski jezik C (10. deo)

Vagrant (1. deo)

str. 18

str. 25

Oslobađanje

Distribuiranje slobodnog softvera

str. 28

Slobodni profesionalac

SUSE Linux Expert Days 2015 -

Frankfurt, 20. januar

str. 33

Internet mreže i komunikacije

Šifrovani čet (1. deo) - *Subrosa*

str. 40

Sam svoj majstor

Cmft i cmftStudio

str. 44

Hardver

BagleBone Black Rev C: Vodič od prvog dana (5. deo)

Biglbon Blek kao Tor egzit

str. 49

Moć slobodnog
softvera





LIBRE! prijatelji



Grupa korisnika GNU/Linux operativnih sistema u Lovčencu

info i tutorijali na srpskom
lubunturs.wordpress.com



Broj: 34

Periodika izlaženja: mesečnik

Izvršni urednik: Stefan Nožinić

Glavni lektor:

Aleksandar Božinović

Lektura:

Jelena Muncan Saška Spišjak

Milena Beran Milana Vojnović

Admir Halilkanović

Grafička obrada:

Dejan Maglov Ivan Radeljić

Dizajn: White Circle Creative Team

Autori u ovom broju:

Nenad Marjanović

Aleksandar Vesić

Nikola Hardi

Kriptopank

Ivan Radeljić

Dario Manesku

Slobodan Nikolić

Branimir Karadžić

Ostali saradnici u ovom broju:

Marko Novaković

Mihajlo Bogdanović

Počasni članovi redakcije:

Željko Šarić Aleksandar Stanislavljević

Željko Popivoda Vladimir Popadić

Kontakt:

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

E-pošta: libre@lugons.org

Vesti

11. februar 2015.

Gugl pokrenuo svoj novi projekat otvorenog koda

Gugl je pokrenuo Perfkit (eng. *PerfKit*) - svoju novu alatku otvorenog koda za merenje performansi cloud servera.

Koristan link: <http://t.co/JBxFUDXgu0>



15. februar 2015.

CrunchBang se povratio iz kliničke smrti

Nakon što je glavni osnivač ovog projekta odustao od istog, sajt <https://crunchbangplus.org/> obećava nastavak razvoja ove distribucije i već nudi beta verziju.

Koristan link: <http://t.co/gXF3Pwvlx>



16. februar 2015.

Mozila zahteva digitalan potpis dodatka

Mozilla je najavila da će promeniti svoju politiku i zahtevati digitalno potpisivanje svih njenih dodataka za Fajerfoks.

Koristan link: <http://t.co/XmX3RtKn7k>





24. februar 2015.

Direktor NSA traži dozvolu za čitanje enkriptovane komunikacije

Mičel Rodžers, direktor američke bezbednosne agencije (NSA), je izjavio da ne treba dozvoliti enkripciju podataka tako da vlada ne može da ih pročita. On je rekao da vlada treba da ima mogućnost da pročita enkriptovanu komunikaciju kada za tim ima potrebe.

Koristan link: <http://t.co/LvGOHXS1hD>



25. februar 2015.

Fajerfoks 36 izašao

Nova verzija Fajerfoksa pretraživača je izašla sa podrškom za HTTP/2 protokol

Koristan link: <http://t.co/6u33Lacy4B>



25. februar 2015.

Git slavi deseti rođendan

Git u aprilu slavi svoj deseti rođendan za koji će biti organizovan skup u Parizu.

Koristan link: <http://t.co/07GyoDMWmp>



Vesti

25. februar 2015.

Gugl pretvara flash u HTML5

Gugl će automatski pretvarati Fleš (eng. *Flash*) reklame u HTML5.

Koristan link: <http://t.co/jdbzauqBPR>



25. februar 2015.

Rails Girls meetup u Hamburgu

Ovaj događaj je namenjen devojkama koje žele da nauče programiranje u ovom programskom jeziku. U Hamburgu je održan 25. februara ove godine.

Koristan link: <http://t.co/6PHFuJSHVU>



1. mart 2015.

LUGoNS okupljanja

LUGoNS svakog četvrtka organizuje okupljanje u Sting kafeu u Novom Sadu.

Koristan link: <https://t.co/BilW0aHGIn>





2. mart 2015.

Počela prijava predavanja za BalCCon

Počela je prijava radova za ovogodišnji BalCCon2k15.

Koristan link: <https://t.co/9N63albFW>



12. mart 2015.

Gugl najavio gašenje Google Coda

Gugl je najavio da će u januaru naredne godine ugasiti servis Gugle kôd (neg. *Google Code*). Kreiranje novih projekata je već isključeno.

Koristan link: <http://t.co/H4QA4ry5p7>



20. mart 2015.

GNU manifest puni 30 godina

U martu 1985. godine Ričard Stelman je objavio GNU manifest. Ovaj dokument propagira upotrebu računara slobodno bez korišćenja vlasničkog softvera. On je ključan dokument za razvoj slobodnog softvera.

Koristan link: <http://j.mp/1HP8wG2>



Puls slobode

 **LUGoNS**
barcamp №4

Autor: Nikola Hardi



LUGoNS (udruženje korisnika Linuksa u Novom Sadu) je u subotu 4. marta 2015. godine održao četvrti po redu Barkamp (*BarCamp*). Kao i u prethodnih nekoliko puta, i ovaj Barkamp je održan u prostorijama Fakulteta tehničkih nauka. Predavanja i radionice su počele nešto posle 12 sati i trajale su do uveče, nakon čega je druženje tradicionalno nastavljeno u opuštenijoj atmosferi u kafiću.



Kao što smo navikli na prethodnim Barkampovima, predavanja i diskusije su se doticale raznih tema. Bile su zastupljene teme o automatizaciji poslova na računaru, hakerskim igračkama, o bezbednosti i anonimnosti. I ovog puta bila su organizovana predavanja i radionice o slobodnoj karti sveta - Open strit map (eng. *Open Street Map*) i Mapileri (eng. *Mapillary*). Pričalo se o muzičkoj produkciji pomoću slobodnog softvera, osluškivanju radio talasa i stanju bezbednosti domaćih sajtova. Mogli ste da čujete i neke vrlo zanimljive ideje o veštačkoj inteligenciji i uzgajanju veštačkog života pomoću računara. Osim toga, zarad eksperimenta je jedno vreme bila postavljena i ogledna GSM mreža na koju su posetioci mogli da se priključe svojim mobilnim telefonima.



Puls slobode

Predavanja su i ovaj put bila snimana i nadamo se da će video materijal ubrzo biti dostupan na Lugonsovima serverima. Na adresi <ftp://lugons.org> su već dostupna predavanja sa prethodnih događaja koje je priredio LUGoNS, uključujući i Barkampove. Kompletan spisak predavanja dostupan je na sledećoj adresi: <https://events.lugons.org/?p=1704>



Ovaj Barkamp je bio poseban i po tome što je LiBRE! tim pripremio štampano izdanje našeg časopisa. Interesovanje je bilo znatno iznad naših očekivanja. Zainteresovani su mogli dobrovoljnim prilogom da podrže ovu inicijativu i možemo da se pohvalimo da je sto na kojem su primerci časopisa bili postavljeni ubrzo ostao prazan. Za ovaj događaj

pripremili smo dvadeset primeraka, a zahvaljujući dobrovoljnim priložima, dobili smo mogućnost da za sledeći sličan događaj pripremimo dvostruko više primeraka. Nadamo se da će ova akcija zaživeti i da ćemo ubrzo moći da pripremimo koliko god primeraka bude bilo potrebno.

Posećenost je i ovog puta bila vrlo dobra. Velika učionica je bila puna kako dobro poznatih stalnih posetilaca, tako i novih lica. Tražila se još po koja slobodna utičnica za struju.

Najave dešavanja u organizaciji LUGoNS-a možete pratiti na adresi <https://events.lugons.org>

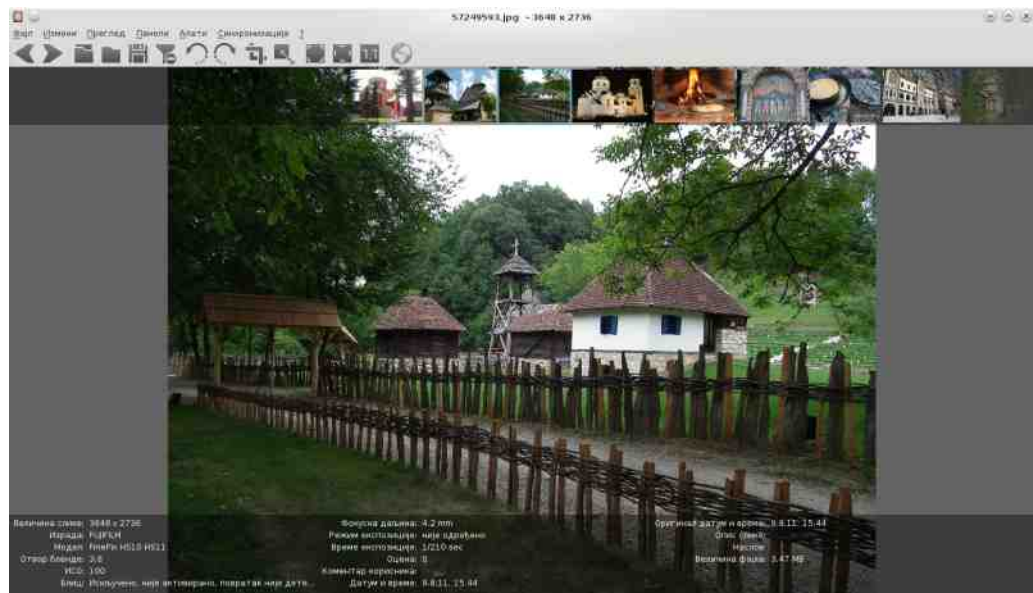


Nomacs - Image lounge



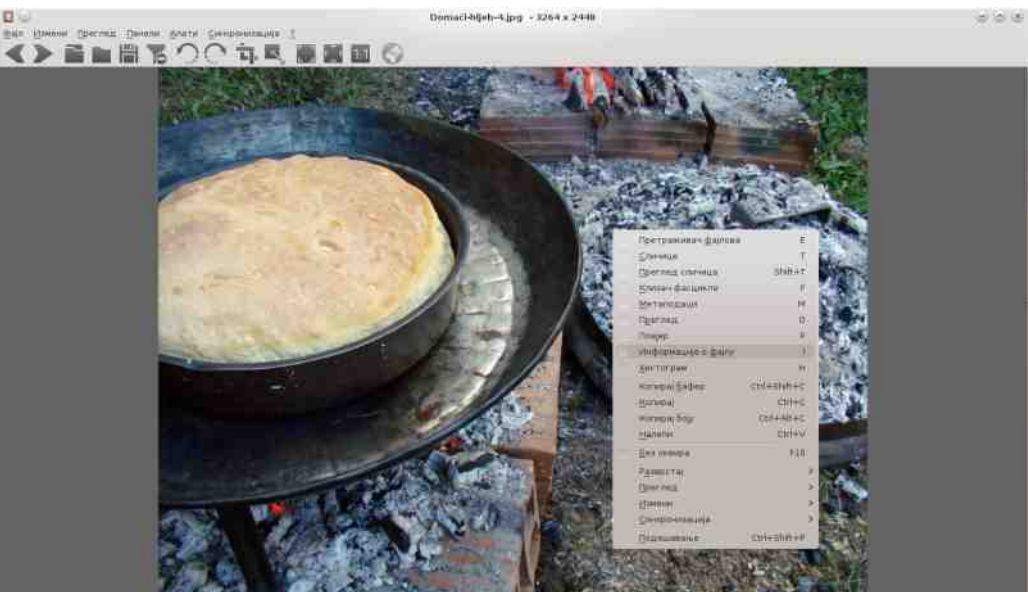
Autor: Slobodan Nikolić

Nomacs (*Nomacs*) je još jedna moguća alternativa kojom možete na svom računaru da obavljate pregled i osnovne izmene fotografija. Aplikacija je dostupna za Vindouz, Linuks, FriBSD, Mek i OS/2, a projekat sa kompletnim imenom Nomacs - Imidž Laundž (*Nomacs - Image Lounge*) vodi se pod GNU GPL licencom. Radi se o programu koji je zasnovan na Qt bibliotekama i koji može, pre svega, poslužiti kao potencijalni izbor za korisnike KDE grafičkog okruženja. Nomacs donosi podršku za dvadeset šest najpoznatijih grafičkih formata, među kojima su i RAW i PSD. U podešavanjima programa trenutno su dostupni prevodi za šesnaest jezika, među kojima je i srpski.

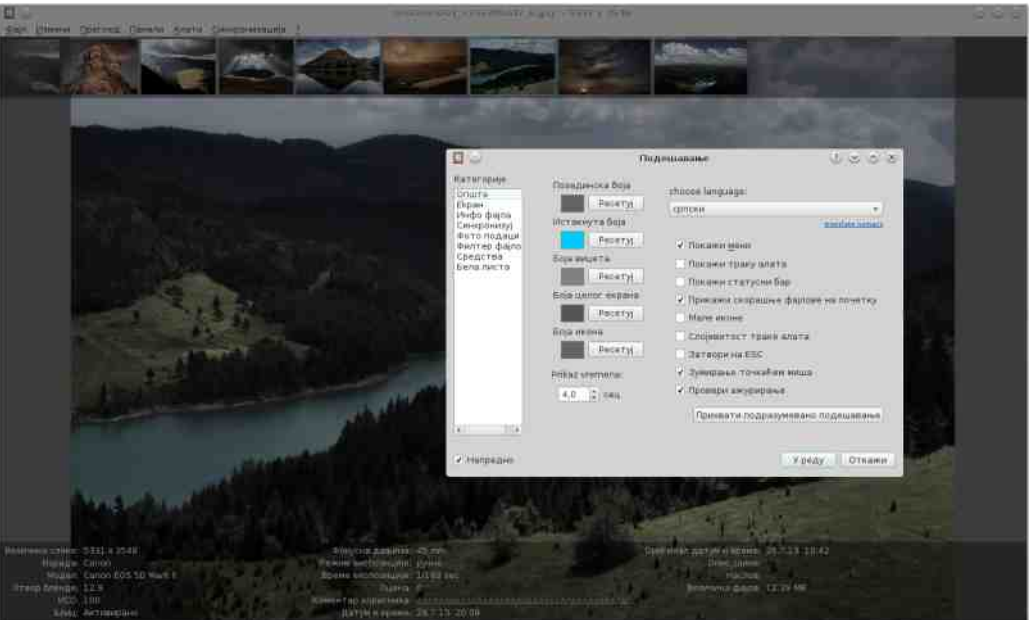
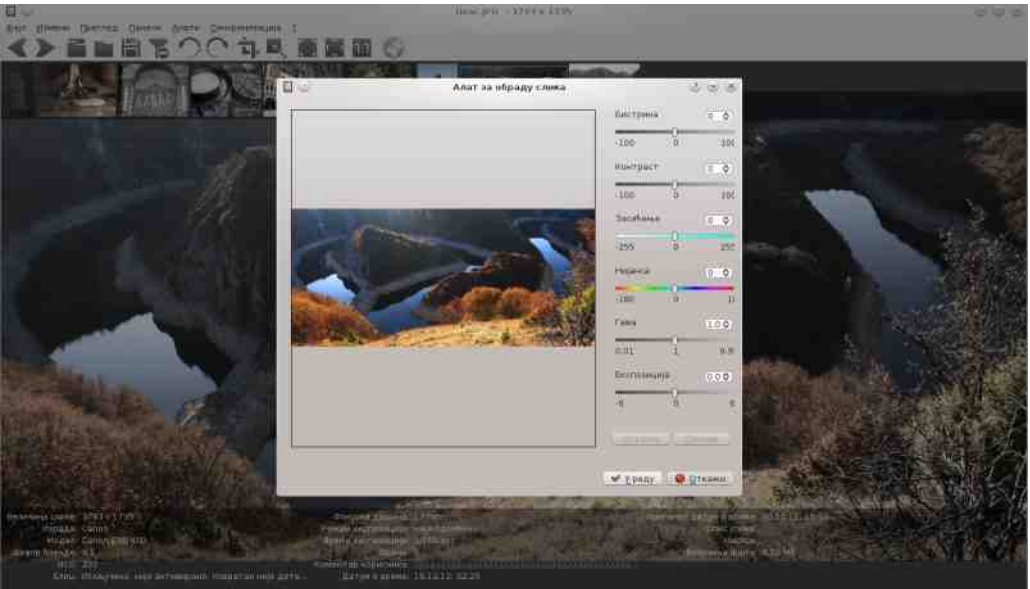


Predstavljamo

Kada se pokrene Nomaks, prvi utisak može biti da se radi o proizvodu koji je siromašan opcijama za kvalitetan pregled fotografija. Stvar se menja odmah ako u meniju Paneli označite opcije kao što su Pretraživač datoteka, Sličice i Metapodaci. To isto može da se dobije klikom na tasterske prečice (E,T,M) i program će dobiti izgled uobičajenog pregledača za slike. Na isti način, mogu se dodavati opcije i kada aktuelnu sliku prebacimo u režim punog ekrana, a on se dobija duplim klikom ili pritiskom na taster *F11*. Na istom mestu (Paneli) određuje se vidljivost menija, trake alata i statusne trake, a dostupne su opcije kao što su plejer, info o trenutnoj datoteci, napomene i histogram. Pregledanje slika koje nudi Nomaks je jednostavno i prirodno, a na korisniku ostaje da se navikne na upotrebu nekoliko tasterskih prečica. Za navigaciju je dovoljno koristiti strelice, dok se zumiranje slika može vršiti na dva načina - strelicama, ali i točkićem miša, prethodno označavajući ovu opciju u podešavanjima programa. Nomaks poseduje opcije za opsecanje (taster *C*), promenu veličine (*Ctrl+T*), kao i alat za obradu slika koji nudi podešavanje elemenata kao što su: bistrina, kontrast, zasićenje, nijanse, gama i ekspozicija. Preko opštih podešavanja program možete vizuelno prilagoditi svojim navikama. Pored određivanja vidljivosti za trake alata i menija, mogu se podesiti i boje za pozadinu, isticanje, pun ekran, vidžete i ikone. Poslednja



Predstavljamo





pogotovo ako smo navikli da se potpomozemo korišćenjem tasterskih prečica. Sam izgled aplikacije je „čist i intuitivan“, a od navika samog korisnika zavisi koji će vidžeti biti prisutni u okviru glavnog prozora. Funkcije osnovnih izmena na fotografijama koje Nomacs poseduje zao-kružuju celinu koja se očekuje od programa ovakve namene, pa verujemo da će mnogim korisnicima predstavljati odgovarajuće rešenje.



Pregled popularnosti *GNU/Linux/BSD* distribucija za mesec mart

Distrowatch

1	Mint	3212<
2	Ubuntu	1752>
3	Debian	1532<
4	openSUSE	1332<
5	Manjaro	1288>
6	Fedora	1229>
7	CentOS	1117>
8	Mageia	1068<
9	elementary	1020>
10	Arch	888<
11	Android x86	861>
12	LXLE	721<
13	Puppy	644<
14	PCLinuxOS	642<
15	Ubuntu MATE	631>
16	Lubuntu	627>
17	Simplicity	570<
18	Xubuntu	564>
19	Kali	563<
20	Lite	561>
21	MakuluLinux	546>
22	Zorin	524<
23	antiX	521>
24	Robolinux	491<
25	Neptune	475>

Pad <
 Porast >
 Isti rejting =
 (Korišćeni podaci sa Distrovoča)

Uvod u programski jezik C

Uvod u rad sa datotekama (10. deo)

Autor: Nikola Hardi

U prethodnom delu serijala, započeto je razmatranje rada sa tekstom u programskom jeziku C. Nadamo se da smo uspeli da predstavimo značaj ove teme. U ovom nastavku serijala biće predstavljen rad sa datotekama. Tok rada klasičnog programa se uglavnom deli na tri faze: učitavanje podataka, obrada, ispis rezultata. U sličnim tekstovima uglavnom je akcenat na samoj obradi. Unos podataka i čuvanje rezultata su jednako važne teme.

Pristupanje datotekama

Prvi korak u radu sa datotekama je „otvaranje” datoteke. Šta to zapravo znači? Bez previše priče o operativnim sistemima, standardnoj biblioteci i sprezi između korisničkih programa i operativnog sistema, cela ideja se svodi na to da program od operativnog sistema traži pristup nekoj datoteci putem mehanizma sistemskih poziva. Operativni sistem potom programu obezbeđuje apstraktnu predstavu te datoteke u vidu toka bajtova, bez potrebe da korisnički program i programer poznaju sistem datoteka (eng. *filesystem*) ili uređaj o kojem je reč.

Datoteci je moguće pristupiti na više nivoa, pozivom funkcije **open()** koja se direktno preslikava u sistemski poziv **open**. Pozivom ove funkcije, operativni sistem korisničkom programu obezbeđuje deskriptor datoteke (eng. *file descriptor*) koji je u suštini evidencioni broj otvorene datoteke na nivou sistema. Dakle, reč je o običnoj celobrojnoj vrednosti. Rad sa datotekama ovim mehanizmom se smatra radom na niskom nivou, a standardna biblioteka jezika



C obezbeđuje i druge mehanizme koji se takođe oslanjaju na sistemski poziv **open**, sa još nekoliko dodataka.

Drugi način, koji i preporučujemo, je pomoću strukture **FILE** i funkcije **fopen()**. Razlog za to je što upotrebom ove ugrađene strukture podataka imamo mogućnost da koristimo i druge standardne funkcije za rad sa datotekama koji će biti opisani.

Datoteke mogu biti otvorene u više režima kao što su režim za čitanje, režim za pisanje, režim za dodavanje sadržaja i kombinacija prethodnih. Ponekad se u literaturi sreću i oznake za rad sa binarnim datotekama, ali one se na većini modernih sistema jednostavno ignorišu.

Sledi primer otvaranja datoteke na oba prethodno opisana načina, a potom i objašnjenje o režimima u kojima je datoteka otvorena.

```
#include <stdio.h> include <fcntl.h> int main()
{
    int fd = open("my_file.txt",
                 O_WRONLY | O_CREAT,
                 S_IRUSR | S_IWUSR );
    printf("%d\n", fd);
    write(fd, "LiBRE!\n", 7);
    close(a);

    return 0;
}
```

Kao što možete da vidite, ovo izgleda malo nezgrapno. Funkcija **open()** ima mnogo namena, pa je zbog toga njena upotreba i složenija. Prvi argument je naziv (tačnije, putanja do datoteke). Sledeći parametar su podešavanja za režim otvaranja datoteke (samo čitanje i kreiranje datoteke ukoliko ona već ne postoji). Treći parametar su podešavanja za dozvole datoteka (eng. *permissions*), koja u o ovom slučaju podrazumeva da samo trenutni korisnik ima pravo pisanja i čitanja. Sledeći izraz je ispisivanje deskriptora datoteka koji bi trebao da bude pozitivan broj ukoliko je otvaranje datoteke bilo uspešno. Nakon toga, vrši se upis teksta **"LiBRE!\n"**, čija je dužina 7 u datoteci sa deskriptorom datoteka **fd**. Po završetku rada sa datotekom, dobro je zatvoriti je

Kako da...?

pozivom funkcije **close()**, jer tek tada možemo biti sigurni da će operativni sistem zapisati sav sadržaj na disk. U biblioteci **<fcntl.h>** su definisani simboli koji su korišćeni za podešavanje režima i kreiranje datoteke.

Više detalja je dostupno u **man** stranicama kojima je moguće pristupiti komandom u terminalu: **man 2 open**. Druge korisne *man* stranice za rad sa datotekama su **write**, **read**, kao i stranice za ostale funkcije koje će biti opisane. Funkcije na višem nivou apstrakcije se nalaze u trećem delu priručnika, dakle: **man 3 fscanf**.

Nešto ugodniji način rada sa datotekama je pomoću funkcija **fopen()**, **fwrite()**, **fread()** itd. Oni pružaju viši nivo apstrakcije i potrebno je manje koda, a i pažnje da bi sve proradilo kako treba. Važno je poznavati i mehanizme na koje se te funkcije naslanjaju, a to su upravo pomenuti sistemski pozivi.

Sledi primer koda u kojem se pristupa datoteci kreiranoj u prethodnom primeru.

```
#include <stdio.h> int main()
{
    char buffer[128];

    FILE *f = fopen("my_file.txt", "r");
    fscanf(f, "%s", buffer);
    fclose(f);

    puts(buffer);

    return 0;
}
```

Ovaj kôd je znatno čitljiviji. Nema kriptičnih parametara ako ništa drugo. Funkcija **fopen()** ima za povratnu vrednost adresu kreirane *FILE* strukture, a ukoliko otvaranje datoteke iz nekog razloga nije bilo uspešno, ta adresa je **0**. Poželjno je proveriti da li je otvaranje bile uspešno. Parametri ove funkcije su putanja do datoteke i režim pristupanja. Postojeći režimi su:

- **r** - samo čitanje



- **r+** - čitanje i pisanje od početka datoteke
- **w** - samo pisanje
- **w+** - pisanje i čitanje, ako datoteka postoji briše se
- **a** - pisanje na kraj datoteke
- **a+** - pisanje na kraj i čitanje

Režimi **w** i **a** će kreirati novu datoteku ukoliko ona već ne postoji.

Opširnija dokumentacija je dostupna u *man* stranicama.

Ostale funkcije

Standardna biblioteka sadrži mnogo funkcija koje rad sa datotekama čine jednostavnijim i ugodnijim. Među njima su čitanje jednog karaktera, čitanje stringa, čitanje linije, čitanje zadatog broja bajtova itd. Sve te funkcije imaju i svoje parove za pisanje podataka. Nazivi ovih funkcija počinju slovom **f**, a vama prepuštamo da pogodite njihove nazive i potražite detalje u *man* stranicama.

Rad sa datotekama ne podrazumeva i rad sa tekstom. Moguće je čuvati i tzv. binarne podatke, kao što su strukture, fotografije i sl. Sve što je potrebno je odrediti koliko bajtova je potrebno učitati ili zapisati, zadati sa kojom datotekom se radi i gde sačuvati rezultat, odnosno odakle pročitati podatke za upis. Veličinu strukture je moguće saznati primenom operatora **sizeof** ili je jednostavno izračunati unapred.

Kod rada sa datotekama, kao i u mnogim drugim situacijama, javljaju se greške: nema dovoljno prostora na disku, program nema dozvolu za pristupanje zadatoj datoteci, u datoteci ne postoje zadati podaci itd. Zbog toga je važno proveravati povratne vrednosti funkcija i protumačiti njihovo značenje na osnovu sadržaja *man* stranica.

Nekoliko reči o separatorima

Kada je reč o radu sa tekstualnom datotekom, javlja se koncept reda, odnosno linije teksta. Pošto su datoteke samo nizovi digitalnih podataka, računari ne poznaju koncept reda teksta. Redovi su apstrakcija poznata ljudima, odnosno korisnicima računara, i zbog toga su veštački uvedeni pojedinim standardima. Standard *ASCII* propisuje vrednosti za više specijalnih znakova, kao što su,

Kako da...?

između ostalog, nov red, povratak glave štampača na početak reda, uključivanje zvona na znakovnom terminalu ili štampaču itd. Kroz istoriju su se javila dva standarda za označavanje kraja reda teksta, među programerima poznata kao *DOS* i Juniks (eng. *Unix*) standardi.

U *DOS* i *Vindouz* svetu se kraj reda označava dvama karakterima - `\r` i `\n`. Ova konvencija je prisutna iz istorijskih razloga i bila je prisutna u doba linijskih štampača kojima je zadavana eksplicitna naredba za prelazak u nov red i vraćanje glave na početak reda. U Juniks svetu je to samo jedan znak, `\n`. Zbog toga su mogući problemi pri razmeni datoteka sa kolegama koje koriste drugačije operativne sisteme. Simptom je jednostavan - sadržaj cele datoteke je u jednom redu ili postoje čudni znakovi na kraju svakog reda. Na Linuksu su dostupna dva jednostavna programa koja rešavaju ovu zbrku, a to su *dos2unix* i *unix2dos*. Programi za uređivanje teksta, takođe, mogu da budu podešeni za rad u jednom ili drugom režimu. Obratite pažnju na ovaj problem jer je prisutan i dan-danas.

Pozicioniranje unutar datoteke

Često postoji potreba da se program pozicionira na drugu lokaciju u datoteci. Ovaj mehanizam je prisutan još iz doba kada su datoteke bile smeštane na magnetnim trakama, pa prema tome i funkcije imaju sličan naziv, odnosno **fseek()** i **rewind()**, što bi na našem jeziku značilo „premotati”. Premotavanje može da bude unapred ili unazad. Referentne tačke mogu da budu od početka datoteke, od trenutne pozicije, ili od kraja datoteke. Postoji i funkcija **ftell()** koja govori trenutnu poziciju unutar datoteke. Funkcija **rewind()** „premotava” datoteku na njen početak.

Ne izmišljajte toplu vodu

Moderni informacioni sistemi, baze podataka, pa i računarska tehnika donekle, potekli su jednim delom iz miljea korporacija koje su se bavile razvojem mašina za rad sa dosijeima. Reč datoteka (eng. *file*) je prvenstveno predstavljala dosije u kartoteci ili sličnoj službi, a posao računara je jednim delom bio da brzo pronađe ili pohrani podatke u odgovarajući dosije. Od koverti sa dosijeima prešli smo na koncept datoteka, a u raznim fazama razvoja računarskih nauka došli smo do koncepta baza podataka. Teorija baza podataka i informacionih sistema nisu tema ni ovog teksta, a ni časopisa; ali važno je napomenuti da ne



treba izmišljati toplu vodu.

Ukoliko postoji potreba za čuvanjem tabelarnih podataka, standardni način je CSV (eng. *Comma Separated Value*), odnosno vrednosti razdvojene zaptetama. U stvarnosti, vrednosti mogu da budu razdvojeni i drugačijim znacima (engl. *delimiterima*), razmacima, zvezdicama, uspravnim crtama itd. Iako je za C česta praksa da se kôd za učitavanje ovakvih podataka piše u nekoliko linija koda, svaki put - mnogi drugi jezici i okruženja za programiranje nude automatizovane alate i biblioteke za ovaj posao. Savetujemo da se taj format ispoštuje.

Ukoliko postoji potreba za pohranjivanjem većeg broja podataka u više datoteka, a podaci su pritom i uvezani, onda je to posao već za jednu bazu podataka. Sistemi za upravljanje bazama podataka sa jedne strane programeru pružaju spregu za pristup podacima na apstraktnom nivou, a sa druge strane sami vode računa kako će to smestiti na disk. Uz to, pružaju i sjajne mehanizme za brzu pretragu, efikasno smeštanje podataka i pribavljanje odgovora na upite. Postoje relacije i nerelacione baze podataka, one za koje su potrebni serveri i one koje rade u memoriji. *SQLite* je biblioteka i koncept baze podataka za koju nije potrebno podešavati server, kreirati korisničke naloge i instancirati pojedinačne baze. Baze se jednostavno čuvaju u jednoj datoteci koja je organizovana na specifičan način. Vredi pogledati, a svakako će o ovoj temi biti reči u nekom od narednih brojeva.

Poliglote i šizofreničari

Ponekad, ipak, postoji potreba za definisanjem novog tipa datoteke. Tako su propisani formati datoteka kao što su **ZIP**, **PDF**, **ELF**, **JPEG** itd. Dobro je pitanje i ogroman problem kako pravilno opisati strukturu jedne datoteke, kako prepoznati tip datoteke i proveriti da li je struktura validna i, još gore, kako to pretočiti u funkcionalan program. Poznati su bezbednosni problemi koji se baziraju upravo na ovim problemima. Zlonamerni programi mogu biti sakriveni unutar druge datoteke, recimo unutar *PDF* dokumenta, ili više programa može biti spojeno unutar jedne izvršne datoteke. Otuda i naziv ovog odeljka - poliglotama se smatraju datoteke koje mogu biti protumačene u više formata, recimo *PDF* i fotografija. Šizofreničari su datoteke koje svoj sadržaj prikazuju drugačije u zavisnosti od različitih programa za pregledanje - naprimer, *Evince* prikazuje *PDF* sa jednim sadržajem, a *Okular PDF* sa drugim sadržajem.

Kako da...?

Ovom temom se bavi *Ange Albertini*. On je na ovogodišnjem CCC-u (eng. *Chaos Communication Congress*) održao predavanje u kojem je opisao ove probleme. Predavanje je dostupno na sajtu <http://media.ccc.de> pod nazivom „*Funky file formats*”. Neke od ovih tehnika su demonstrirane u časopisu *PoC||GTFO*, koji je jednostavno pronaći na internetu. U njima su opisani i detalji o načinu na koji je moguće napraviti ovakve datoteke. Na sajtu <http://www.corkami.com> postoje, između ostalog, i poster koji opisuje strukturu nekih poznatih tipova datoteka. Predlažemo da pogledate o čemu je reč. Oformila se i posebna grupa istraživača koji se donekle bave ovom temom, a reč je o temi pod nazivom *LANGSEC* (<http://langsec.org/>).

U narednom tekstu

Narednim tekstom će ovaj serijal polako biti priveden kraju. Obrađene su najvažnije teme ovog programskog jezika. Vremenom su tekstovi neplanirano porasli, ali ne zamerite. U narednom tekstu (najverovatnije će to biti više od jednog teksta), biće opisane ideje koje prevazilaze sam jezik, smernice šta zanimljivo može da se napravi, korisne biblioteke i drugi važni koncepti. Spremni ste da napravite i svoj prvi program koji ćete moći da koristite svakodnevno. Javite nam se ukoliko želite da obradimo neku temu detaljnije ili da dodamo još koji tekst u serijal.

**Learn C
Programming**



VAGRANT

(1. deo)

Autori: Ivan Radeljić i Stefan Nožinić

Uvod

Kada razvijate aplikacije, veoma je bitno dobro podesiti svoje razvojno okruženje. Svako od nas ima svoje navike, svoje tekstualne uređivače (editore) i ostali softver koji je potreban tokom procesa izrade aplikacija. S druge strane, svaka od aplikacija ima svoje zavisnosti koje takođe treba uskladiti. Ponekad je potrebno razvijati više različitih aplikacija u isto vreme, pa sve to sinhronizovati sa ostatkom razvojnog tima. Ovakve situacije ponekad mogu da se iskomplikuju i da prouzrokuju da aplikacija ili jedan njen deo kod jednog člana tima funkcioniše bez problema, dok se kod drugih pojavljuju određeni problemi. Da bismo sve to izbegli, pravo rešenje je da koristimo Vagrant (engl. *Vagrant*).

Zašto Vagrant?

Vagrant povećava produktivnost razvojnog tima. To je alat kojim kreiramo lagana, prenosiva razvojna okruženja koja se iznova mogu ponovo koristiti. To znači da nas Vagrant oslobađa svih podešavanja i da samo u jednoj konfiguracionoj datoteci biramo koja svojstva ćemo koristiti. Vagrant koristi vodeće provajdere za virtuelizaciju i može biti snabdeven na Virtualboksu (engl. *VirtualBox*), VMVeru (engl. *VMWare*), AWS-u i drugima. Za instalaciju potrebnih zavisnosti i softverskih paketa, kao i za njihovo automatsko

Kako da...?

podešavanje, koristimo šel (*shell*) skripte, Čef (engl. *Chef*), Papet (engl. *Puppet*) ili Ansibl (engl. *Ansible*).

Neke od prednosti ovog softverskog alata su:

- Vagrant čini identično razvojno okruženje svim članovima tima, a da pri tom ne žrtvuje ništa od alata u kojima su programeri navikli da rade (to podrazumeva tekstualne uređivače, IDE¹, brauzere, pa i sam operativni sistem).
- Omogućava programerima da se fokusiraju na sam kod, a ne na konfiguraciju razvojnog okruženja.
- Konfiguracija se obavlja kroz jednu konfiguracionu datoteku i kroz dodatne skripte za automatsko podešavanje paketa i instalaciju istih.
- Konfiguracione datoteke se mogu (i treba) distribuirati u samom repozitorijumu kontrole verzija, kao što je Git (engl. *Git*).



¹ **IDE** (eng. *Integrated Development Environment*) - integrisano razvojno okruženje.



Ako ste jedan od operacionih inženjera, Vagrant vam omogućava jedno lako zamensko okruženje za testiranje, gde možete razvijati i testirati svoje skripte koje posle možete primetiti na serverima u produkciji.

Ako ste dizajner, sa Vagrantom takođe dobijate sve već unapred definisano (podešeno) i vaše je samo da se fokusirate na dizajn. Neće vam više biti potreban programer da vam pomogne da pokrenete aplikaciju na vašem razvojnom okruženju kako biste mogli da izvršite potrebne promene u dizajnu.

Jedna od najvećih prednosti Vagranta - ako vam u bilo kom stepenu razvoja aplikacije zatreba još neka zavisnost, ili ste zaboravili da je na početku dodate, dovoljno je da Vagrantom izmenite konfiguracionu datoteku (*vagrantfile*) i svi u timu će imati identičnu verziju softvera bez obzira na njihovo lokalno okruženje, jer se sva konfiguracija nalazi na sistemu za kontrolu verzija.

VAGRANT [VMWARE INTEGRATION](#) [DOWNLOADS](#) [DOCUMENTATION](#) [BLDG](#) [ABOUT](#)

SET UP

Download and install Vagrant within minutes on Mac OS X, Windows, or a popular distribution of Linux. No complicated setup process, just a simple to use OS-standard installer.

CONFIGURE

Create a single file for your project to describe the type of machine you want, the software that needs to be installed, and the way you want to access the machine. Store this file with your project code.

WORK

Run a single command — "vagrant up" — and sit back as Vagrant puts together your complete development environment. Say goodbye to the "works on my machine" excuse as Vagrant creates identical development environments for everyone on your team.

Distribuiranje slobodnog softvera

Autor: Nikola Hardi

Povod za pisanje članka

Svi smo se mi, kao novi korisnici slobodnog operativnog sistema našli u situaciji da nam je neko na forumu, četu ili nekom sličnom mestu predložio da instaliramo pojedine pakete kako bismo rešili svoje probleme. Još češće smo se našli u situaciji da i sami zaključimo da treba da instaliramo dodatni softver na naš sistem. Neiskusni korisnici, koji su do tada uglavnom koristili Majkrosoft Vindowz operativne sisteme, navikli su se da uputstva za instalaciju i softver traže širom interneta, ali u svetu slobodnog softvera to funkcioniše malo drugačije. Strogo i izričito vam savetujemo da softver instalirate isključivo iz zvaničnih riznica, putem softverskog centra ili paket menadžera. Druge mogućnosti koje se nekada mogu tolerisati su instalacija pomoću odgovarajućih paketa (*.deb*, *.rpm*) ili dodavanjem riznica (recimo *PPA* u slučaju Ubuntu). Poslednja mogućnost kojoj možete da pribegnete je instaliranje iz izvornog koda. Softver instalirajte iz zvaničnih riznica ili repozitorijuma (eng. *software repository*), osim kada imate potrebe za specifičnom verzijom nekog parčeta softvera i znate šta tačno radite i koju odgovornost time preuzimate.

Riznice softvera i ostale čudne reči

Kada se softver instalira pomoću nekog upravljača paketa (*apt*, *yum*, *pacman*, *yaourt* i *emerge*), preuzima se iz zvaničnih riznica. Riznice su mesta na internetu gde su smešteni softverski paketi koji su provereni, testirani i koji će sigurno raditi u vašem sistemu. Dakle, proces instaliranja na primeru Ubuntu izgleda ovako:



Distribuiranje slobodnog softvera

- Softverskim centrom ili drugim alatom se izvršava pretraga paketa koji su dostupni u riznicama;
- Izabrani paket se preuzima iz riznice, uključujući sve dodatne pakete koji su potrebni za njegov rad;
- Vrš se provera da li su svi paketi ispravno preuzeti, kako bi se izbeglo instaliranje neispravnih paketa;
- Vrš se provera autentičnosti preuzetih paketa, proverava se da li je odgovorno lice potpisalo paket koji je preuzet;
- Instaliraju se sve „zavisnosti“ (paketi koji su neophodni za rad) i željeni paket;
- Svi paketi se dodaju na listu instaliranih paketa kako bi upravnik paketima mogao da vodi računa o njihovim verzijama i izvrši kasniju nadogradnju.

U prethodnim redovima je upotrebljeno nekoliko termina kao što su softverski centar, paket, potpis, upravnik paketima i drugi, koje je možda potrebno razjasniti.



Oslobađanje

Softverski centar uglavnom predstavlja grafički alat za pretragu i upravljanje softverom na sistemu, odnosno samo grafički interfejs koji u pozadini poziva upravnika paketima.

Upravnika paketima (eng. *package manager*) je najčešće konzolni alat ili skup alata za pretragu, instaliranje, nadogradnju i uklanjanje paketa.

Paket je skup datoteka neophodnih za instaliranje i ispravan rad softvera.

Potpisi i kontrolne sume su deo paketa za proveru integriteta i autentičnosti preuzetog sadržaja.

Zavisnost je termin koji ukazuje da jedan paket ne može da bude instaliran pre drugog paketa jer se oslanja na njegove funkcionalnosti. Pod terminom zavisnosti podrazumevaju se sve veze koje mogu biti zamršene, ciklične, rekurzivne ili nezgodne na neki drugi način.

Šta su tačno paketi?

Paket je arhiva koja sadrži softver preveden u izvršni oblik koji odgovara vašem sistemu. Pored toga, paketi sadrže informacije o svom integritetu, poput kontrolnih suma i digitalnih potpisa. Unutar paketa se mogu nalaziti i uputstva, informacije o autoru paketa, autoru programa i pomoćni alati i dodatne procedure koji su neophodni za ispravan rad softvera koji taj paket sadrži. Slobodni operativni sistemi se na osnovu distribuiranja softverskih paketa mogu podeliti na „binarne“ (eng. *binary based*) i „bazirane na izvornom kodu“ (eng. *source based*). Kod binarnih distribucija se unutar paketa nalaze već prevedene datoteke u izvršnom obliku, koje se potom samo smeštaju na odgovarajuće mesto u sistemu i odmah mogu da se koriste. Kod distribucija baziranih na izvornom kodu kao što je Džentu (eng. *Gentoo*), u paketu se nalazi lokacija odakle izvorni kod treba da bude preuzet kao i skripta (spisak naredbi) kojim se taj izvorni kod prevodi u izvršni oblik i potom instalira. Upravnika paketa ume da preuzme i proveriti preuzete podatke, a potom i da izvrši sve potrebne korake kako bi softver bio pravilno instaliran.

Jednostavnije instaliranje

U doba kada su pravi hakeri noćima instalirali nov softver i sami pisali



Distribuiranje slobodnog softvera

upravljačke programe (eng. *driver*) za svoje uređaje, softver je gotovo uvek bio instaliran iz izvornog koda, bez upravnika paketima. Gotovo svi paketi i programi se oslanjaju na neke druge biblioteke, programe ili pakete, iz čega sledi da je moguće formirati stablo zavisnosti, a potom i spisak i redosled svih paketa koje je potrebno instalirati. Uveravamo vas da nije tako lako pogoditi redosled instaliranja i odgovarajuće verzije za sve zavisnosti. Zato upravnik paketima to radi za vas.

Automatske nadogradnje

Velika prednost ovakve šeme distribuiranja paketa je u tome što postoje ažurni spiskovi instaliranih paketa. Upravljači paketima redovno preuzimaju spiskove aktuelnih verzija paketa u riznicama i poredе verzije. Na zahtev ili automatski, moguće je proveriti da li postoje nadogradnje za instalirane pakete i preuzeti ih. Sigurni smo da ne želite da u crnoj svesci imate spisak ručno instaliranih paketa i njihovih verzija, a još manje da nekoliko puta mesečno obidete sve sajtove i poredite najnovije verzije sa onima koje imate zavedene u svom spisku.

Bezbednost

Digitalno potpisivanje je procedura koja je prisutna i van konteksta distribuiranja slobodnog softvera. Reč je o procesu u kojem se za određene podatke generiše odgovarajući potpis pomoću ključa. Ono se bazira na konceptu asimetričnog šifrovanja i postoji uvek par koga čine javni i privatni ključ. Vlasnik ključa na osnovu podataka u paketu i svog privatnog ključa generiše odgovarajući potpis. Na osnovu preuzetih podataka i javnog ključa može se proveriti da li preuzeti podaci (u ovom slučaju paket) odgovaraju potpisu, tj. da li je neko usput promenio deo podataka. Kontrolne sume proveravaju da li je sadržaj ispravno preuzet. Potpisi garantuju i autentičnost podataka.

Ustaljeno je mišljenje da su slobodni operativni sistemi generalno bezbedniji od onih drugih. Međutim, ovo nije sasvim tačno jer svaki sistem ima svoje rupice, bubice, gremline i viruse. Razlika je u tome što su slobodni operativni sistemi vrlo neprijatno okruženje za širenje takvih računarskih bolesti. Mehanizmi kojima se softver distribuira i instalira su ključni za to. Pre nego što ovaj članak bude završen i pre nego što ponovimo šta sve loše može da

Oslobađanje

se dogodi ako ne pazite kako i odakle instalirate softver, evo jednog teorijskog primera kada nešto pođe po zlu.

- Neiskusni korisnik želi da instalira program za video razgovore;
- Pretražuje internet u nadi da će pronaći verziju za Linuks i u mračnom ćošku interneta pronalazi nekakvu verziju;
- Instalira je i ubrzo primeti da njegov računar više ne radi kako treba;
- Korisnik zaključuje da je to delo zlonamernog programa i da slobodan softver i nije tako bezbedan.

Evo šta je zapravo moglo da se dogodi.

Kada je korisnik preuzeo paket sa nepoznate lokacije, mogao je da preuzme verziju koja je izmenjena tako da u sebi ima bezbednosne propuste raznih vrsta (udaljeno upravljanje računarom, preuzimanje podataka sa računara i brisanje datoteka). Druga, blaža i više verovatna mogućnost je da je preuzet ispravan paket, ali ručnom instalacijom nije dodan na spisak instaliranih paketa i zbog toga nije bio ažuriran. U međuvremenu je otkriven bezbednosni propust, a nadogradnja koja rešava problem nije instalirana.

Imajte na umu

Pre nego što se opet nađete u situaciji da se pitate kako da instalirate neki paket, program ili bilo kakav drugi deo softvera, imajte na umu da će upravnik paketima uraditi sledeće stvari umesto vas, kao i stvari koje bi trebalo vi da uradite umesto upravnika paketima.

- Proveriti autentičnost lokacija odakle je paket preuzet;
- Kontrolnom sumom proveriti da li je paket ispravno preuzet;
- Proveriti potpis paketa kako bi se isključila mogućnost da je neko presreo preuzimanje i tako podmetnuo zlonamernan kod;
- Razrešiti potrebne zavisnosti (ponoviti prethodne stavke za sve zavisnosti);
 - Ispratiti proceduru za ručno instaliranje;
- Redovno proveravati obaveštenja o otkrivenim bezbednosnim propustima i dostupnim nadogradnjama.

Jednostavnije je pokrenuti softverski centar ili napisati `sudo apt-get install`, zar ne?



Autor: Aleksandar Vesić



zanimljivih predavanja.

U Frankfurtu je 20.01.2015. godine održan jedan od *SUSE Linux Expert Day* koji se od Septembra 2014. godine održavaju širom sveta. Za mesto održavanja izabran je *25Hours Hotel Frankfurt by Levi's* kao jako interesantna lokacija u samom centru Frankfurta. Sam događaj je trajao oko pet sati i sastojao se od niza



Odmah nakon uvodnog govora koji je održao Mihael Jores (*Michael Jores*), regionalni direktor *SUSE Central Europe*, direktor Suse Linuks Enterprajza (*SUSE Linux Enterprise*) Olaf Kirh (*Olaf Kirch*) predstavio je trenutno stanje i planove vezane za dalji razvoj Suseovih proizvoda dajući nam uvid u trenutne trendove razvoja.

Nakon izlaska SLES-a 12 (*SUSE Linux Enterprise Server 12*) u oktobru 2014. godine, ove godine (juni/juli) izlazi i Servis pek (*Service Pack*) za SLES 11. Pored uobičajenih ažuriranja kernela dolazi i podrška za platforme *IBM z13*, *POWER8 BE* i *Intel Haswell EX*. Prema trenutnim planovima, to je zapravo poslednji Servis pek, a SP5 trenutno nije u planu. Napomenuto je da je SAP (*Systems, Applications & Products* - Nemačka korporacija) dao SLES-u 12

Slobodni profesionalac



sertifikat za korišćenje ERP (*Enterprise resource planning*) softverskih komponenti. Zvanična potvrda i odobrenje od strane SAP (*Systems, Applications & Products*) se očekuje u toku ovog kvartala, tako da kupci navedeni proizvod mogu da instaliraju i da im pritom Suse i SAP pružaju punu podršku.



SUSE
OpenStack Cloud

U prvoj polovini ove godine će se pojaviti i Suse Oblak (*SUSE Cloud*) u petoj verziji. Pored vrlo dostupnih virtuelnih mašina biće podržan i Docker-kontejner (*Docker-Container*). Trebalo bi, između ostalog, da bude moguće njima upravljati putem budućih Suse Menadžer (*SUSE Manager*) verzija.

Za treći kvartal 2015. godine je predviđeno ažuriranje Suse Oblaka koji bi trebalo da bude baziran na Open-steku (*OpenStack*), verzija pod imenom „Kilo”. Kod ove verzije bi kontrolni čvorovi (*Control Nodes*), koje koristi Open-stek, mogli da funkcionišu i na SLES-u 12. Za 2016. godinu je planiran Suse Oblak 7 baziran na Open-steku 7.



SUSE
Enterprise Storage

Što se tiče Suse servera za smeštanje podataka (*SUSE Storage Server*), tim proizvodom firma iz Nirnberga želi da dobije svoj deo kolača u biznisu vezanom za softver baziran na skladištenju podataka. Proizvod je namenjen prvenstveno klijentima kojima su potrebni



privatni ili hibridni Oblaci (*hybrid Cloud*), i/ili su u potrazi za alternativama u odnosu na konvencionalna SAN (*Storage Area Network*) rešenja. Osnovna komponenta je Cef (*Ceph*) koji je poznat po sposobnostima vezanim za povećanje kapaciteta (*upscaling*), dobro dupliciranje podataka i mogućnost ispravljanja grešaka. Cef prepoznaje neispravne diskove i prebacuje se na rezervne diskove. Tri osnovna dela koncepta Cef su:

- Smeštanje objekata (*Object Storage*) – pristup putem *C / C ++ - Java*, *Pajtona*, *PHP-a*, ili *RESTful*, implementira *Striping* i *Snapshot* funkcije
- Smeštanje blokova (*Block Storage*) – Smeštanje objekata kao *thin-provisioned Block Storage* (npr. za virtuelne mašine)
- *File System - POSIX* kompatibilan sa direktnim pristupom na *Object Storage*, integrisan u *Linuks kernel* od 2010. (2.6.34), opcionalno je dostupan *FUSE-Client*.

Tokom prošle godine je odabranim korisnicima bio omogućen pristup beta verziji, a prva zvanična verzija proizvoda pojavljuje se u prvom tromesečju ove godine, koja se temelji na verziji Cefa pod nazivom Fajerflaj (*Firefly*). Za treći kvartal 2015. godine planirana je verzija 2.0, koja kao osnovu treba da koristi Cef „Hamer” (*Hammer*). Dok kao server ovde samo SLES 12 dolazi u obzir, po pitanju klijenta je SLES 11 podržan.

Za 2016. godinu je planiran Suse Menadžer 3 kod koga pored podrške za SLES 12 SP1 takođe dolaze i promene funkcija vezane za visoku dostupnost (*High Availability*) i nadgledanje (*Monitoring*). Ostaje nam da vidimo koliko će Suse da preradi *Monitoring* funkciju od *Spejsvoka* (*Spacewalk*) nakon što je *Red Hat* (*Red Hat*) objavio da će se dalje baviti njegovim daljim razvojem. Za 2017. godinu je planiran Suse Menadžer 4.

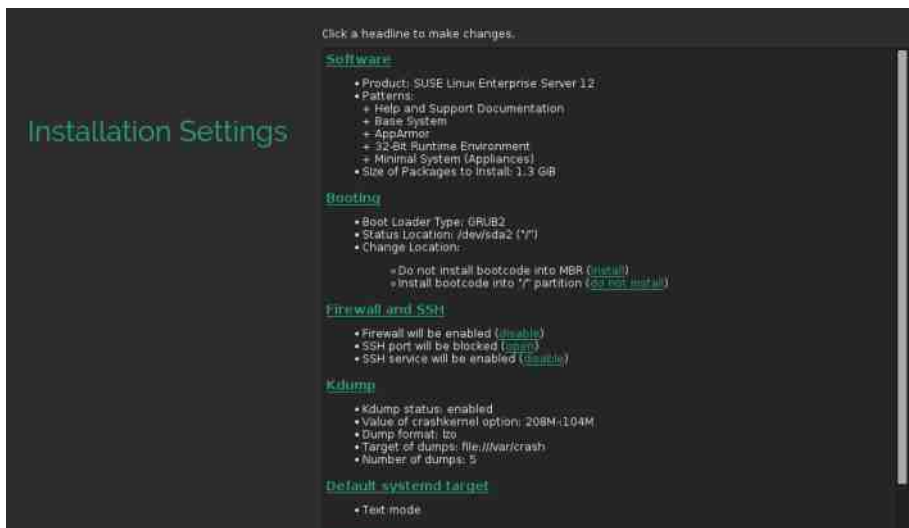
SysVinit vs Systemd

Objavljen je posle pet godina čekanja novo glavno izdanje Suse Linuks Enterprajza. U odnosu na prethodnu verziju izvršene su velike promene, od kojih je *Systemd* najveća i najkontroverznija. Tokom prošle godine je ovo ažuriranje, koje je kod mnogih distribucija zamenilo *SysVinit*, bilo odgovorno za dosta diskusija u zajednici korisnika Linuksa. Tako je bilo i prilikom ovog seminara, gde diskusija „*Systemd vs SysVinit*” nije izostala. Bez namere da komentarišemo i da iznesemo mišljenje, citiraćemo Olafa Kirha:

Slobodni profesionalac

„Svakih deset do petnaest godina ista priča. Kada je SysVinit zamenio RC skripte, svako je bio užasnut, iako nije imao značajne nedostatke. Danas nikome ne nedostaju. Ista priča se ponavlja kada je u pitanju Systemd.“

Systemd je radikalna, ali sasvim savremena promena koja ima mnoge prednosti. Kao i svaka druga tehnička inovacija, *Systemd* nosi sa sobom potrebu da se steknu nove veštine. Distribucije za firme, kao što su Suse Linuxs Enterprajz ili Red Hat Enterprajz Linuxs pružaju klijentima kao opciju da koriste poznate alate (npr. *service*, *chkconfig*, stare konfiguracijske datoteke) kako bi se olakšao prelaz. Ne bi bilo na odmet da se zajednica korisnika Linuxa opходи ka toj temi sa malo više otvorenosti i tolerancije.



* * *

U Suse Linuxs Enterprajzu su napravljene dve radikalne promene: *systemd* i ukidanje podrške za Intelovu *i686* 32-bitnu arhitekturu. Korišćenje čistih 32-bitnih sistema je poslednjih nekoliko godina u padu u odnosu na 64-bitne alternative. SLES 12 takođe neće biti dostupan za sisteme bazirane na Intel Itanijumu (*Intel Itanium - ia64*) i time Suse sam ide u korak sa drugim velikim distribucijama kao što je Red Hat, na primer. Između ostalog, *Xen*, *KVM* (*Kernel-based Virtual Machine*) i *LXC* (*Linuxs kontejneri*) daju nam tri mogućnosti za virtualizaciju. Za tzv. kontejner aplikacije je dostupan Docker.



Btrfs (*ButterFS*) je novi standardni fajl-sistem kod SLES-a 12. On će u potpunosti biti pokriven od strane Suse podrške (pod uslovom da se koriste standardne opcije za konfiguriranje fajl-sistema putem *YaST*-a). On takođe nudi dodatne mogućnosti, poput Snepšatsa (*Snapshots*). Tako Ziper (*Zypper*), na primer, pravi Snepšat (snimak) pre ažuriranja sistema - u slučaju da nakon ažuriranja sistem ne može da se podigne, moguće je pri startu u GRUB-u izabrati prethodni Snepšat za podizanje sistema. Predviđeno je i Suse preporučuje da se *BtrFS* koristi za operativni sistem a *XFS* za korisničke podatke (npr. *MySQL* baza podataka). U SLES-u 12 *ext4* sada ima i podršku za pisanje (*write*) za razliku od SLES-a 11 koji podržava samo čitanje (*read*), jer *ext4* nije bilo moguće dovoljno testirati da bi bio odobren za produktivne sisteme. Dodatno je naglašeno da je *ext4* puno lošiji kod asinhronih ulaznih-izlaznih (*I/O*) poziva u poređenju sa *XFS*-om.

YaST

Centralni alat za konfiguriranje *YaST* je takođe podmlađen, osim estetskih promena on sada koristi *Rubi* (*Ruby*) umesto ranijeg *YCP*-a. Suse nam je obećao i lakše održavanje softvera. Uz ugrađen *Viked* (*Wicked*), *YaST* je dobio novi *Network Backend* koji se ne fokusira samo na *Client*-sisteme i treba da bude koristan u hibridnim Oblacima. Preporučuje se da se mrežna konfiguracija novih sistema implemetira direktno putem *Vikeda* - raniji načini konfiguracije su još uvek podržani. Pojedini programski paketi su sada dostupni u modulima. Oni će biti podržavani samo nekoliko godina, umesto deset do trinaest godina (što je do sada bio običaj - prim.ur.). Trenutno dostupni moduli su:



- *Web and Scripting* : PHP, Pajton, *Ruby on Rails* (3 godine podrške)
- *Legacy - Sendmail*, stare verzije Java, itd. (3 godine podrške)
- *Public Cloud - Public Cloud* - programski paketi (neprekidna integracija)
- *Toolchain - GNU Compiler Collection* (po godišnjem izdanju godinu dana podrške)
- *Advanced System Management - Tools/Frameworks* za administraciju (neprekidna integracija)

Slobodni profesionalac

Suse Mašinerija

Program Suse Mašinerija (*SUSE Machinery*) je tehnička pretpremijera servisa za migraciju postojećih sistema. Mašinerija analizira konfiguraciju sistema, konsoliduje istu i migrira ponudene servise. Osnovna namera je omogućiti migracije sa SLES-a 11 na SLES 12, kao i za hibridne Oblake. Oporavak od katastrofa (*Disaster Recovery*) je poboljšan mogućnostima koje nudi Mašinerija, nažalost primenu ovog programa još uvek ne podržava Suseova podrška.

Izmena jezgra u letu

Moto ovogodišnjeg *SUSE Linux Expert Day* bila je „put prema neprekidnosti rada“ (*Towards Zero Downtime*) kojim je Suse reklamirao funkciju izmene jezgra u letu (*Kernel Live Patching*) kod SLES-a 12. Kao i kod Red Heta ova se funkcija prodaje kao dodatni proizvod u pokušaju da se što više kupaca veže za naprednu podršku (*Premium Support*). Zakrpe za jezgro su dostupne kao RPM datoteke koje instaliraju module i obnavljaju *Initial ramdisks*. Pozive funkcija jezgra preusmeravaju ka novim modulima f-trejs (*ftrace*) i K-graft (*kGraft*) - komponenta koju razvija Suse. Programi, koji su u funkciji, ne moraju ponovo da se pokrenu. Trenutno su samo *x86_64* platforme podržane, a u zavisnosti od reakcija kupaca slediće podrška za ostale arhitekture. K-graftom (*kGraft*) Suse želi da bude konkurentan u odnosu na Red Het i njihov K-peč (*kPatch*) kao i prema Oraklu (*Oracle*) koji je već predstavio K-splajs (*kSplice*), koji takođe koriste module (otvorenog koda) jezgra za implementiranje izmene u toku rada (*Live Patching*). Suse, za razliku od rešenja koje koristi Red Het, vrši detaljne provere povezanosti modula. Red Het je u novembru prošle godine pokrenuo diskusiju o tome da se dođe do ujedinenja srodnih rešenja, a glavnu reč treba da dâ razvojna zajednica.

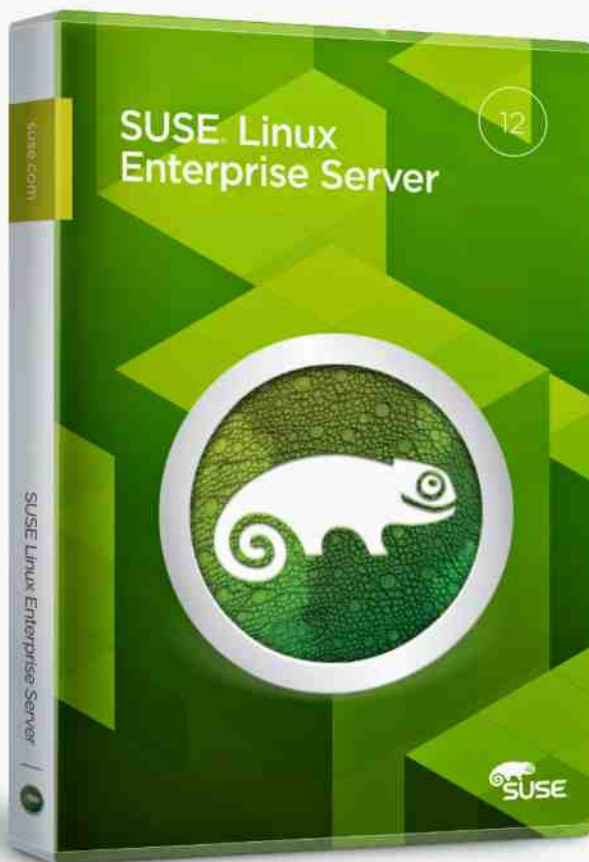
Suse Oblak

U poređenju sa konkurencijom Suse se ponosi sertifikovanim podešavanjima za hardver i softver i podrškom drugih Hipervizorsa (*Hypervisors*) koje ne podržava Open-stek. Suse posebno dobro podržava instalacije *vSphere* na *VMwaru*. Na Suseovoj veb stranici možete preuzeti šezdesetodnevnu test-verziju koja u roku od trideset minuta izvrši potpuno podešavanje vašeg privatno Oblaka. U odnosu na ručnu instalaciju Suse Oblak štedi mnogo rada



SUSE Linux Expert Days

oko konfigurisanja, što je jako interesantno kupcima koji još nemaju iskustva sa Open-stekom (*OpenStack*). Kao i Red Hat, Suse je takođe platinasti član koji jako puno pomaže projektu Open-stek na razne načine a pogotovo finansijski.



Šifrovani čet (1. deo)

Subrosa

Autor: Kriptopank

Danas za međusobno brzo dopisivanje koristimo mnogo servisa i aplikacija, ali da li brinemo o svojoj privatnosti koristeći ih?

Iako smo kroz prethodne brojeve naučili kako da zaštitimo svoju elektronsku poštu, to možda ne pokriva ceo spektar naših komunikacija i svakojakih razmena informacija i podataka preko interneta. Pa ako je tako, hajde da vidimo kako možemo da se dopisujemo sa drugima, brzo, lako i sigurno.

Problem je sledeći: „Hoćemo da se instant dopisujemo, odnosno da četujemo sa jednom ili više osoba u realnom vremenu na siguran način. To znači da želimo sigurni grupni čet. Onda bismo želeli da imamo mogućnost razmene podataka-fajlova ili audio-video komunikaciju. Većinom za ovo koristimo centralizovane servise poput Fejsbuka, i/ili one koje nisu otvorenog koda poput Skajpa (eng. *Skype*) ili Vajbera (*Viber*), i ne pružaju skoro nikakvu privatnost korisnicima od administratora samog servisa, pa možda i državnih bezbednosnih kompanija.”

Međutim, početak nije jednostavan jer smo pogrešili u startu. Ne bi trebalo da koristimo aplikacije i servise koji nisu napravljeni sa idejom o privatnosti samih korisnika niti tome služe. Ako nam je privatnost važna, onda treba da potražimo alternative. Pa, da li one uopšte postoje? Naravno da postoje, a ponekad su čak i bolje.

U ovom (prvom) delu serijala pomenućemo ujedno i najnoviju među njima. Subrosa je zapravo centralizovana aplikacija/platforma otvorenog koda (<http://goo.gl/fAyHev>) koja može da radi u dva moda: klijent i server.



Da biste je koristili, potrebno je da se registrujete, tj. da napravite korisničko ime (eng. *username*) i lozinku (eng. *password*), opciju za mejl možete da preskočite ili da jednostavno upišete lažni mejl da biste zavarali trag. Ni Subrosa ne zna naše tajne ključeve niti čuva lozinke, stoga ako zaboravite lozinku, možete se pozdraviti sa tim nalogom.

Način funkcionisanja ove čet platforme je sledeći:

1. Pri registraciji unosite korisničko ime i šifru. Šifra ne napušta vaš pretraživač.
2. Uneta šifra se potom koristi za generisanje javnog/tajnog para asimetričnih ključeva (*RSA 2048 bits*).
3. Javni ključ se šalje Subrosinim serverima, a tajni ključ se šifrovan čuva na računaru.

```
-----BEGIN RSA PRIVATE KEY-----
MIIBPAIBAAJBAlB0PhLynDZ7SnfLazZwsJT900KcUvKx0ghNHp02IviH8iP4pEz0
o1Lq1K8TpmBA0b4A8VGvtwt/bC8jNbjK87kCAwEAAQJBAAhZ2HHNYARgFKImGW3q
+mRI/k+y0+e/RWWe982wPyilQTvyrrQToVVGkX5wfPQUeAiYE86v1rw2b0Vs4tFb
J00CIQDc8m0cIrwg71o4sGK9UGSRgcH2S1a5Zbcm2UnRQXkCKwIhA3SchY2jkjzc
KU+uLa1VIJXAxN2zg+wQ3UM+lYXt0A0rAiEAs1dkEU5wfhKtLHE39IroIuMIWa7V
ei+B9tqw/FW0r3kCIQCAZL5qAbD9jcTPR4/FGi/90b8EP+0x0DdzsF0z/+ddm4wIg
P1Xbhz/V6KLgLRXvmtUXtbZJ6lg62NZwYA7+xe12n0U=
-----END RSA PRIVATE KEY-----
```

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIBCgKCAQEAsaCbY3Ly52qiW7wFYNss
5Vj7eDCbH/+uu/bIij57dTpUXbsAwQRjM5R+6cSEduhMU/Kk1Et6/LPRC0K7q/Xf
doctrTsyYoXIuug79vXbdcYBiPaHgKE+q20lyfBZ+w7hVkkKPjxsRv+ctZYjX103
lWlajjBmz8oMTlB9byBmyK0zSk0EWxzFJI9DoIAt0uInXIK68hLVhMGkBzLMSowl
furKzV4Bj7QMMwLa4sWCBEVlfMN2busDuTJMRhxgPNrcbCgrcYmoR3zGDTqq9JZy
GXzTXXWhuvKXBoSLCB28ki9+URfDexjy2DU3N/9kc0MLMIAkzL1dtUCtWhWaMIA
dwIDAQAB
-----END PUBLIC KEY-----
```

Pošto su ključevi napravljeni, komunikacija može da počne. Sve što je potrebno je da pronađete osobu sa kojom želite da diskutujete. Ono što se

Internet mreže i komunikacije

potom dešava je da inicijator komunikacije dobije od Subrosinog servera javni ključ osobe sa kojom želi da komunicira. Zatim:

1. pretraživač mu generiše simetrični ključ (*AES-GCM-256*);
2. taj ključ se potom šifruje javnim ključem vašeg budućeg sagovornika i zatim se tako šifrovan šalje njemu;
3. sagovornik dešifruje poruku, i nadalje se taj ključ, koji sada poseduju obe strane, koristi za šifrovanje i dešifrovanje poruka, i uopšte bilo kog tipa podataka između dve strane - tzv. end-tu-end (eng. *end-to-end*) šifrovana komunikacija.

Subrosa pruža mogućnost video komunikacije, audio poziva i dopisivanja (naravno, sve je **end-tu-end** šifrovano). Šifrovanje **end-tu-end** znači da kada osobe A i B komuniciraju, poruka koja je sa jednog kraja (korisnik A) šifrovana, može biti dešifrovana samo na svom drugom kraju (korisnik B).

Subrosa takođe koristi *HSTS* i *PFS*, što ,najprostije rečeno, znači da se za svaku komunikaciju koristi drugi par ključeva. Ovo je samo mera prevencije protiv nekih napada na šifrovanu komunikaciju putem *SSL*-a ([https/ /:](https://)) i odnosi se samo na komunikaciju sa *Subrosa* serverom, koji, kao što smo rekli, mogu podići i korisnici na svojim mašinama.

Izvorni kôd je proveren nezavisno od strane nemačke bezbednosne kompanije *Curve53*.



Subrosa takođe podržava i tzv. *warrant canary*, što je zapravo metod kojim provajder internet-skih komunikacionih usluga obaveštava korisnike da nije primio tajni nalog ili sudski nalog za dostavljanje informacija o svojim korisnicima vlastima u nekoj zemlji (<http://goo.gl/z5nSDG>).



* * *

Dakle, da sumiramo: Subrosa je otvorenog koda, ne zahteva adresu mejla niti bilo kakve lične podatke; možete podići server na svom računaru; ne čuva logove (barem tako tvrde - prim.aut.); pokreće se direktno iz pretraživača i ne zahteva preuzimanje bilo kakvog softvera; podržava **HSTS** (eng. *HTTP Strict Transport Security*), **PFS** (eng. *Perfect Forward Secrecy*) i podržava video, audio i tekstualnu šifrovanu komunikaciju.

Jedan od potencijalnih problema, smatraju neki, je korišćenje Javaskripta i izvršavanje koda sa udaljenog računara, pošto ništa nije potrebno da se instalira na računar. To znači da je platforma potencijalno ranjiva kao većina drugih na XSS, razne vrste indžekšna i MITM od strane servera. Da bi izbegli ove napade, (paranoični) korisnici takođe mogu da hostuju i server koji će koristiti u svojoj komunikaciji.

Moguće je da se anonimnost i privatnost još više unaprede korišćenjem proksija ili povezivanjem na neku virtuelnu privatnu mrežu ili na Tor.

Sam svoj majstor

Cmft i cmftStudio



Autor: Dario Manesku

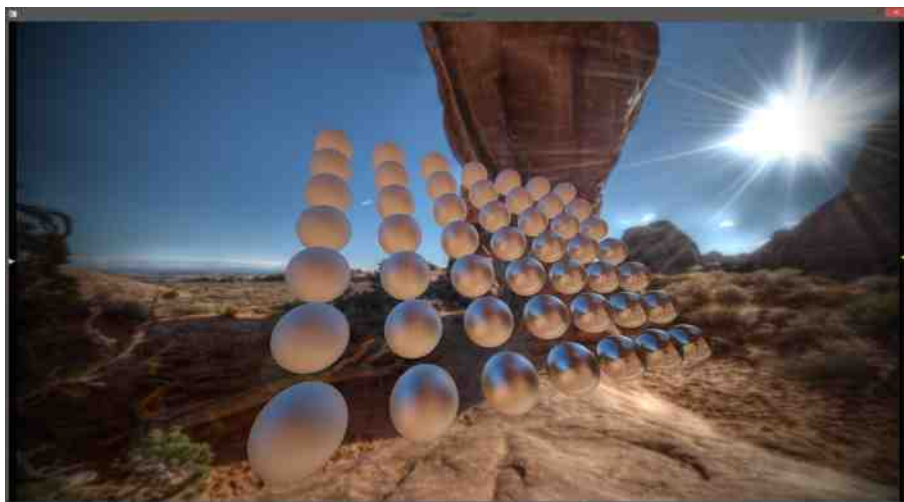
Cmft je alat za filtriranje tzv. kockastih-tekstura (eng. *cubemap texture*) za potrebe računarskih igara i trodimenzionalnih aplikacija, a *cmftStudio* je alat koji pored filtriranja nudi i interaktivnu vizualizaciju i pregled dobijenih rezultata.





Kockaste-teksture su vrste tekstura, koje imaju dosta veliku primenu u računarskoj grafici u realnom vremenu, ili, preciznije rečeno, u računarskim igrama.

Šest strana u obliku kocke čine jednu kockastu-teksturu, a njen uzorak se dobija putem trodimenzionalnih vektora (x,y,z) koji označava pravac od centra kocke. To je jedna suštinska razlika u odnosu na dvodimenzionalne teksture, gde se uzorak dobija putem dvodimenzionalnih koordinata (x,y) .

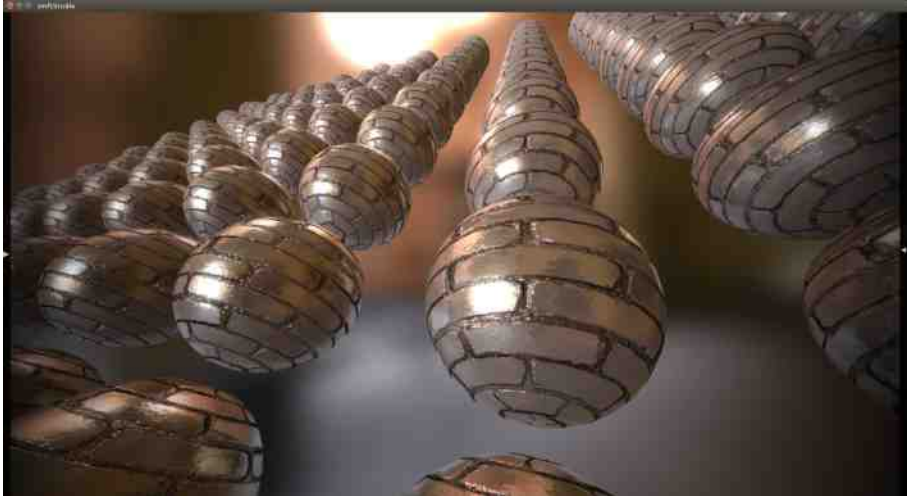


Kao što je lako zaključiti, kockaste-teksture obuhvataju ceo prostor oko svoje centralne tačke i, kao takve, pogodne su za implementaciju raznih tehnika u računarskoj grafici.

Jedna od konkretnih primena kockastih-tekstura je implementacija tzv. *IBL* tehnike (eng. *Image-Based Lighting*) ili osvetljenja zasnovanog na slici. Tehnika obuhvata skladištenje podataka o upadnim svetlosnim zracima sa svih strana za željenu tačku u trodimenzionalnom prostoru i korišćenje tih informacija za izračunavanje, tj. aproksimaciju ambijentalne svetlosti u okruženju te tačke. Podaci o upadnoj svetlosti se najpre moraju obraditi i upakovati u željenom obliku u kockaste-teksture. Taj oblik treba biti pogodan za obradu u realnom vremenu, i upravo za to služe alati *cmft* i *cmftStudio*.

Sam svoj majstor

Ova tehnika je u širokoj upotrebi u, može se reći, skoro svakoj današnjoj modernoj 3D igri, jer su rezultati koji se dobijaju dosta primamljivi, a brzina današnjeg dostupnog hardvera bez problema omogućava primenu ove tehnike u praksi.



Pre nastanka *cmfta* i *cmftStudia*, za ovu namenu ljudi su uglavnom koristili poznati AMD-ov alat Kjubmapdžen (eng. *CubeMapGen*), koji je bio besplatno dostupan na internetu. Septembra 2011. godine, AMD je prekinuo dalji razvoj ovog alata i objavio je javno njegov izvorni kod. Alat je bio dosta dobar - nudio je dosta mogućnosti za podešavanje parametara za filtriranje. Međutim, imao je jednu veliku manu - filtriranje kockastih-tekstura je često dugačak proces, a Kjubmapdžen je vršio obradu isključivo na jednom jezgru procesora. Pojavila se, takođe, na internetu i modifikacija Kjubmapdžena koja je mogla da vrši obradu koristeći tri procesorska jezgra. Međutim, to je i dalje bilo dosta sporo. Filtriranje jedne kockaste-teksture je moglo trajati, u zavisnosti od izabranih parametara, i do šest ili sedam minuta, pa čak i preko petnaest minuta. Mi smo u to vreme koristili navedeni alat i eksperimentisali sa raznim podešavanjima, i taj proces je trajao dosta dugo - na trenutke je čak bio i frustrirajući. Da bismo saznali kakvi će rezultati biti sa pojedinim podešavanjima, bili smo primorani da čekamo dosta dugo, a pritom nam je računar bio opterećen i jedva smo mogli njime da radimo išta drugo.



Tu je nastala ideja da pokušamo da ubrzamo ceo taj proces. Nismo se opredelili za dalju modifikaciju Kjubmapdžena pošto je njegov izvorni kôd bio usko vezan za Dajrekt-iks (eng. *DirectX*), tj. Vindouz platformu, i to nam se nije nimalo dopalo. Odluka je bila da počnem da pravim alat iz početka, sa idejom da:

- bude dostupan za sve aktuelne platforme: Vindouz, Linuks, Mek OS Iks (eng. *Mac OS X*);
- koristi sva dostupna procesorska jezgra i u isto vreme i grafički procesor putem Open-si-el (eng. *OpenCL*) tehnologije, kako bi se postigla što veća brzina obrade podataka;
- ima mogućnost poziva iz komandne linije kako bi se sa lakoćom mogao pozvati iz nekog skript jezika i na taj način kreirao zadatak za serijsko obrađivanje brojnih kockastih-tekstura odjednom;
- bude javno dostupan svima na internetu, zajedno sa izvornim kodom, kako bi što više ljudi bilo u mogućnosti da ga koristi za svoje potrebe.



Nakon nekoliko meseci, upravo smo to i realizovali. *cmft* alat se pojavio na internetu i bio je приметно brži od Kjubmapdžena. Ono što je nekada trajalo sedam minuta sa Kjubmapdženom, *cmft* bi to završio za oko dvadesetak sekundi. Pored toga, imao je i mogućnost učitavanja kao ulaznih podataka

Sam svoj majstor

mного različitih tipova slika okruženja. Napokon, obrada kockastih-tekstura nije više bila toliko naporan proces. Međutim, pošto je u pitanju alat koji se koristi iz komandne linije, potrebno je bilo dokumentovati i objasniti njegovo korišćenje. To je bilo nešto što mi, u to vreme, nismo imali volje i želje da radimo, pa smo došli na bolju ideju: da napravimo dodatni interaktivni alat za vizualizaciju i prikaz rezultata filtriranih kockastih-tekstura, tako da je moguće lakoćom koristiti alat bez potrebe prethodnog čitanja uputstva i smernica. Tako je započeo razvoj *cmftStudia*.

Razvoj *cmftStudia* je trajao приметно duže zbog razloga što je sada trebalo mnogo stvari uraditi i uklopiti na jednom mestu kako bi se postiglo prijatno korisničko iskustvo. Isti ciljevi su prevladali: dostupnost za sve platforme i javno dostupan izvorni kod cele aplikacije. Za razvoj je korišćena *bgfx* biblioteka za iscrtavanje, o kojoj je pisano ranije - u 33. broju ovog časopisa. Ona je takođe otvorenog koda, nudi pristup funkcijama grafičkog procesora za iscrtavanje sa različitih platformi bez skoro ikakve dodatne modifikacije i, kao takva, predstavljala je odličan i očigledan izbor za potrebe razvoja *cmftStudia*.

Sada već dostupan na internetu, *cmftStudio*, pored filtriranja kockastih-tekstura, nudi i još mnoštvo drugih funkcija, kao naprimer: učitavanje 3D modela i njegovih tekstura, kreiranje i podešavanje različitih materijala, pregled iscrtavanja materijala sa raznim osobinama glatkoće i konstitucije pod uticajem direktne i ambijentalne svetlosti, snimanje i učitavanje sesije, itd. Zbog svega ovoga, čak i 3D modelari nalaze upotrebu *cmftStudia* za učitavanje i pregled svojih 3D modela.

U prvih nedelju dana nakon objave, Github stranica *cmftStudio* projekta je ostvarila skoro dve hiljade jedinstvenih pregleda, što pokazuje dosta veliko inicijalno interesovanje za ovakav alat. Kako će dalji razvoj *cmftStudia* teći, u velikoj meri zavisi od utisaka samih korisnika. Za sada alat nudi sve esencijalne opcije za funkciju koju obavlja i može se bez problema koristiti u produkciji kompjuterskih igara ili 3D aplikacija koje imaju potrebu za ovakvim alatom.

Korisni linkovi:

[1] <https://github.com/dariomanesku/cmft>

[2] <https://github.com/dariomanesku/cmftStudio>



BeagleBone Black Rev C

Vodič od prvog dana (5. deo) - Biglbon Blek kao Tor egzit

Autor: Nenad Marjanović



beaglebone

Da bismo bolje razumeli konfiguraciju Tor servera, treba posebno obratiti pažnju na konfiguracioni fajl `/etc/tor/torrc`. Vrednosti u podešavanjima variraju od resursa kojima raspolažemo. Ukoliko želimo da se bavimo administracijom izlaznog (eng. *Exit*) noda, za to je potrebna brza internet konekcija i, naravno, više raspoložive radne memorije (eng. *RAM - Random Access Memory*).

Biglbon Blek rev. C (BBB - eng. *BeagleBone Black Rev C*) raspolaže sa 512MB *RAM*-a, što nam ipak ostavlja određene mogućnosti pri parametrisiranju našeg prvog izlaznog noda. Sve što treba da uradimo je da ograničimo izlaznu politiku na određene portove. U prošlom broju *LiBRE!* časopisa pokazali smo konfigurisanje transfer servera.

ExitPolicy reject *.*

Ovo u prevodu znači da naš server služi isključivo za transfer ka izlaznom serveru. U redakciji smo počeli da testiramo određena podešavanja i, nakon višemesečnog testiranja, došli smo do zaključka da je moguće podržati servise kao što su *SSH, IRC, IRC(s), FTP, SMTP(s), IMAP(s), POP3(s), XMPP, GIT* i *OpenPHP HKP* bez problema na već postojećem transfer nodu. Da bismo dodali ova pravila, potrebno je izmeniti **torrc** fajl, odnosno pre unosa **ExitPolicy reject *.*** treba upisati sledeća pravila.

Hardver

```
ExitPolicy accept *:22 # ssh
ExitPolicy accept *:465 # smtps (SMTP over SSL)
ExitPolicy accept *:993 # imaps (IMAP over SSL)
ExitPolicy accept *:994 # ircs (IRC over SSL)
ExitPolicy accept *:995 # pop3s (POP3 over SSL)
ExitPolicy accept *:5222 # xmpp
ExitPolicy accept *:6660-6669 # IRC
ExitPolicy accept *:6697 # IRC SSL
ExitPolicy accept *:9418 # git
ExitPolicy accept *:11371 # OpenPGP hkp (http keyserver
protocol)
ExitPolicy reject **
```

Prevedeno - prihvati komunikaciju za navedene servise, ostale odbij.

Kao što možete primetiti, nigde nismo podržali portove 443 i 80, ali njih i ne preporučujemo na ovako malim uređajima. Čak i ako ste u situaciji da posedujete adekvatnu brzinu internet konekcije, podržite komunikaciju na portu 443, ali ne i portu 80, ipak, ovo su opcije kojima moramo posvetiti duži period testiranja. I, ukoliko ne želite da vaš server bude korišćen za maliciozni mejl marketing (eng. *SPAM*), ne otvarajte port 25.

Više dostupnih izlaznih pravila možete naći na ovoj lokaciji: <https://trac.torproject.org/projects/tor/wiki/doc/ReducedExitPolicy>. Kako raspoložemo sa više mogućnosti pri podešavanju servera, kada smo u prilici možemo testirati nove portove i limit BBB Rev C uređaja, sa druge strane ipak ne treba preterivati sa brojem izlaznih portova zbog mogućih zakonskih peripetija usled saobraćaja koji prolazi kroz izlazni server.

Tor zastave

Po pravilima Tor zajednice, svaki server dobija zastavice (eng. *Flags*) na osnovu funkcija koje obavlja.

- *Stable*
- *Fast*
- *HSDir*
- *Exit*



- *Valid*
- *Running*

su samo neke od zastavica, ali najbitnije su *Fast* (brz), *Valid* (validan) i *Running* (u funkciji). *Exit* (izlaz) zastavica je rezervisana za servere sa dovoljnim resursima. Međutim, ako vaš server ne dobije ovu zastavicu, a vi ste omogućili port 443, to ne znači da niste učesnik izlaznog saobraćaja, već samo da naš BBB nije dovoljno dugo u mreži ili niste dali dovoljno resursa izlaznom serveru.

Nakon nekoliko sati aktivnosti status našeg servera možemo proveriti na sledećim adresama:

1. <https://globe.torproject.org>
2. <https://atlas.torproject.org>

U polje pretrage unesite nadimak servera i dobićete informacije o Tor serveru nad kojim vršite administraciju.

Sigurnost Tor servera

Biti administrator servera odnosi se na brigu o sigurnosti uređaja koji je dostupan javnosti. Podešavanja se odnose pre svega na *SSH* konekciju i zaštitni zid (eng. *Firewall*). Zlatna pravila su sledeća:

1. ne koristiti standardni port za konekciju na server (uvek podesiti vrednost iznad 20000);
2. dodajte novog korisnika na server i isključite logovanje za rut (eng. *root*) korisnika;
3. kreirajte *SSH* ključ za konekciju na server, zato što korišćenje lozinki nije preporučljivo zbog mogućih brutalnih napada na rut nalog korisnika.

* * *

U narednom broju nastavljamo sa pisanjem o podešavanju izlaznog Tor servera sa akcentom na zaštitu servera i korisnika te pravljenje kopije serverskih konfiguracionih fajlova, kao i o naprednijoj politici zaštite *SSH* servera upotrebom tehnike geolokalizacije administratora servera.



GNU 30th

The GNU Manifesto

What's GNU? Gnu's Not Unix!

GNU, which stands for Gnu's Not Unix, is the name for the complete Unix-compatible software system which I am writing so that I can give it away free to everyone who can use it.

(1) Several other volunteers are helping me. Contributions of time, money, programs and equipment are greatly needed.

So far we have an Emacs text editor with Lisp for writing editor commands, a source level debugger, a yacc-compatible parser generator, a linker, and around 35 utilities. A shell (command interpreter) is nearly completed. A new portable optimizing C compiler has compiled itself and may be released this year. An initial kernel exists but many more features are needed to emulate Unix. When the kernel and compiler are finished, it will be possible to distribute a GNU system suitable for program development. We will use TeX as our text formatter, but an nroff is being worked on. We will use the free, portable X Window System as well. After this we will add a portable Common Lisp, an Empire game, a spreadsheet, and hundreds of other things, plus online documentation. We hope to supply, eventually, everything useful that normally comes with a Unix system, and more.

GNU will be able to run Unix programs, but will not be identical to Unix. We will make all improvements that are convenient, based on our experience with other operating systems. In particular, we plan to have longer file names, file version numbers, a crashproof file system, file name completion perhaps, terminal-independent display support, and perhaps eventually a Lisp-based window system through which several Lisp programs and ordinary Unix programs can share a screen. Both C and Lisp will be available as system programming languages. We will try to support UUCP, MIT Chaosnet, and Internet protocols for communication.

GNU is aimed initially at machines in the 68000/16000 class with virtual memory, because they are the easiest machines to make it run on. The extra effort to make it run on smaller machines will be left to someone who wants to use it on them.

To avoid horrible confusion, please pronounce the g in the word "GNU" when it is the name of this project.

Gnu's Not Unix

