

Април 2014.



ЛИБРЕ!

Часопис о слободном софтверу

број

24



FLOSS на послу



17. април 2014.

Canonical је објавио
Ubuntu 14.04 LTS



26. април 2014.

Пета ревизија *Debian*
7 Wheezy-ја је доступ-
на за преузимање.



Creative Commons Ауторство-Некомерцијално-Делити под истим условима.



Професионална употреба FLOSS-а

Пред вама је двадесет четврти, помало јубиларни број ЛиБРЕ! часописа. Ово значи да се дружимо већ пуне две године. Надамо се да вам нисмо досадили. Овом двадесет четвртом броју нису ишли на руку силни протекли празници, који су утицали да овај број мало касни. Извињавамо се због тога.

Тема овог броја је професионална употреба FLOSS-а. Мишљења смо да је то прави пут за популаризацију FLOSS-а. У данашњем свету новац је највећи мотиватор. За разлику од кућног корисника, фирме не смеју да користе пиратски софтвер. У овом броју показујемо како фирму опремање једног рачунара основним власничким софтвером кошта око 650 евра, што значи да постоји 650 разлога за прелазак на FLOSS решење. Чак и кад власник фирме унајми стручно лице за имплементацију FLOSS-а, то га неће коштати као лиценце за еквивалентни власнички софтвер.

Ми који одавно користимо FLOSS, знамо да FLOSS решења барем у обичним канцеларијским пословима могу успешно да замене власничке софтвере. Кроз серију презентација Привредна комора Београда покушава привреднике да упути у

употребљивост FLOSS-а у пословном окружењу.

Тренд штедње уз помоћ FLOSS-а су усвојиле или усвајају администрације европских земаља. Ако то могу они који су много богатији од нас, требало би да и ми размислимо о уштедама кроз FLOSS. И није само штедња мотив за прелазак на FLOSS, већ и скоро потреба комуникације са развијеним светом који убрзано прелази на отворени документ као стандардни електронски документ.

Прелазак фирми на FLOSS решења отвара могућност покретања бизниса на бази FLOSS-а. О томе говори наш нови серијал „Ваш посао, *open-source* посао”. FLOSS јесте бесплатан, али секундарни домени FLOSS-а (услуге, едукација, имплементација и друго) могу да се наплате јер ће увек бити оних који више воле да плате, него да уче. Ово отвара простор за развој IT индустрије на FLOSS основама, што као резултат даје нови замајак предузетништву и новом запошљавању.

До сада је било покушаја појединаца који су своје рачунаре на послу опремали FLOSS-ом да би ускладили са оним на чему раде код куће. Логичнији је пренос FLOSS-а са

посла на кућне рачунаре који би био условљен завршетком неких послова код куће или просто због вежбања рада на FLOSS-у због постизања веће ефикасности на послу.

Много је разлога зашто популаризација FLOSS-а треба да буде првенствено усмерена према привреди а не према појединцима, зато ће ЛиБРЕ! све више свој садржај усмеравати ка правним лицима и указати им како се FLOSS може употребити у пословне сврхе.

Овога пута, за крај, осим стандардног позива ауторима и графичарима да нам се придруже, упућујемо позив и програмерима да нам се придруже. Формирамо ЛиБРЕ! развојни тим. За почетак ћемо направити ЛиБРЕ! Android апликацију и уредити нашу интернет инфраструктуру, али ко зна - можда тај развојни тим почне да ради и за трећа лица. За контакт, примедбе и предлоге користите нашу, већ познату, интернет адресу [libre \[et\] lugons \[dot\] org](mailto:libre [et] lugons [dot] org).

До читања

ЛиБРЕ! тим

Моћ слободног
софтвера



Број: 24

Периодика излажења: месечник

Извршни уредник: Стефан Ножинић

Главни лектор: Александар Божиновић

Лектура:

Милена Беран

Јелена Мунђан

Маја Панајотовић

Александра Ристовић

Редакција:

Александар Станисављевић

Дејан Чугаљ Сандрина Димитријевић

Горан Мекић Александар Тодоровић

Марко Кажих Михајло Богдановић

Вељко Симић Гаврило Продановић

Никола Харди Александар Брковић

Жељко Шарић Владимир Цицовић

Данило Ђокић Јоаким Јањатовић

Петар Симовић Стефан Стојановић

Ромео Млинар Марко Новаковић

Дејан Петровић Жељко Попивоца

Златан Васовић

Сарадници:

Велимир Бакса Милован Кривокапић

Иван Булатовић Ладислав Урошевић

Тамара Ђорђевић Бојан Богдановић

Владимир Попадић

Графичка обрада:

Дејан Маглов

Иван Радељић

Дизајн:

Младен Шћекић

Зоран Лојпур

Контакт:

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

Е-пошта: libre@lugons.org

<http://libre.lugons.org>



ЛИБРЕ! вести стр. 6

Вести



Пулс слободе стр. 8

Уговор Републике Србије
са *Microsoft*-ом (6. део) стр. 8

Презентација:
Linux у пракси стр. 14



Представљамо стр. 19

Сербиан 2014 (2. део) стр. 19



VI text editor стр. 25

OpenStreetMap стр. 28

Како да...? стр. 32

Увод у програмски
језик *C* (2. део) стр. 32

Learn C Programming

Ослобађање стр. 37

Неколико филозофских
мисли о софтверу:
Развој софтвера стр. 26



Слободни професионалац стр. 42





Ваш посао,
open-source посао (1. део) стр. 42

Интернет, мреже
комуникације стр. 44

Tor мрежа:
Online анонимност и
сигурност (3. део) стр. 44



Сам свој мајстор стр. 48

Underscore стр. 48



ЛИБРЕ! пријатељи





Unreal Engine 4 поддржава Linux и SteamOS

4. април 2014.



Unreal је објавио да ће Unreal Engine 4.1, који ће бити објављен за неколико недеља, у потпуности поддржавати SteamOS и Linux.

Користан линк: <http://bit.ly/1s3G7VJ>

Snowden користио Linux дистрибуцију за анонимност

15. април 2014.



Snowden је користио Tails Linux дистрибуцију како би остао анониман на интернету.

Користан линк: <http://bit.ly/1fGKYdA>

Ubuntu ново издање 14.04

17. април 2014.



Canonical је објавио ново издање најпопуларније Linux дистрибуције.

Користан линк: <http://bit.ly/1oe1Rfz>

Микропроцесор Елбрус-4С спреман за серијску производњу

19. април, 2014.



Микропроцесор Елбрус-4С спреман је за серијску производњу, а као основну платформу за развој

руски инжењери користе Linux.

Користан линк: <http://bit.ly/1fZ59OA>

OpenBSD форковао OpenSSL као LibreSSL

22. април, 2014.



LibreSSL је форк OpenSSL API-а који развија OpenBSD развојни тим.

Користан линк: <http://www.libressl.org/>

Покренути пројекат Core Infrastructure initiative

26. април 2014.



Поучена искуством са Heartbleed сигурносном грешком у OpenSSL-у, Open-Source заједница је окупила највеће стручњаке из области информационалних технологија да би се ове афере у будућности спречиле.

Користан линк: <http://bit.ly/1oe1X6F>

Објављен Debian 7.5

26. април 2014.



Пета ревизија Debian 7 Wheezy-ја је доступна. Debian 7.5 је последње стабилно издање популарне дистрибуције са безбедносним закрпама и другим ажурирањима.

Користан линк: <http://bit.ly/1nZnpME>



Firefox 29 је доспућан за преузимање
29. април, 2014.



Нова стабилна верзија доноси многе промене и ново корисничко сучеље под именом *Australis*.

Користан линк: <http://bit.ly/SIGC17>

Објављена KaOS 2014.04 Linux
дистрибуција
30. април 2014.



Објављена је дистрибуција по узору на *ArchLinux* са фокусом на *KDE* графичко окружење и на *Qt*.

Користан линк: <http://bit.ly/1i6gOPY>

Canonical прекинуо развој пројекта
Ubuntu for Android
1. мај 2014.



Разлог за ово, како наводи *Canonical*, јесу већи приоритети као што је то *Ubuntu* за мобилне уређаје.

Користан линк: <http://bit.ly/Rc3NdB>

Објављен је OpenBSD 5.5
1. мај, 2014.



Промене су примећене у сигурности система и инсталеру, а проширена је подршка за хардвере.

Користан линк: <http://j.mp/1lMpmwr>

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



Европски фестивал старих рачунара, петнаести по реду, одржао се 3. и 4. маја 2014. године у

Минхену.

Користан линк: <http://bit.ly/1s3CmQg>

ЛИБРЕ! пријатељи

LUTHERUS

Et in Arcadia ego!

free
OPEN.net

Think about this

 ICT časopis
ICTCASOPIS.ICT.EDU.TS

 **LOVĆENAC**
LINUX USER GROUP 
Grupa korisnika GNU/Linux operativnih sistema u Lovćencu

info i tutorijali na srpskom
lubunturs.wordpress.com

lubuntu 



Уговор Републике Србије са *Microsoft*-ом

(6. део)

Информатика у основним школама и *FLOSS*

Аутор: Дејан Маглов

Ово је једна од кључних тема овог серијала. Као никада до сада у овом серијалу, налазимо се на врло клизавом терену. Ово је врло обимна тема и може да се сагледа из много углова. Одмах на почетку да кажемо да критика није упућена на рачун учитеља, наставника и професора. Закључци и предлози наведени у овом чланку су управо засновани на предлозима самих наставника.

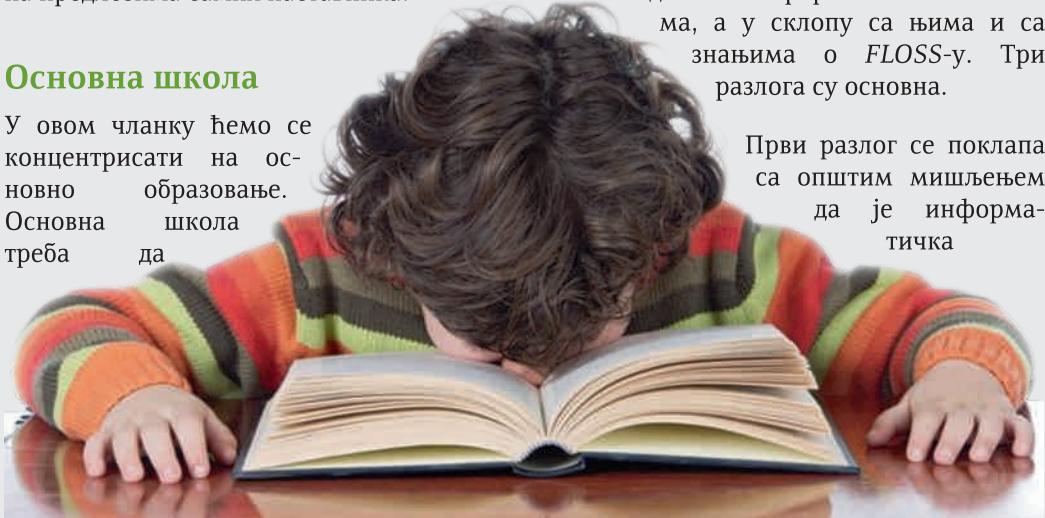
Основна школа

У овом чланку ћемо се концентрисати на основно образовање. Основна школа треба да

обезбеди широка темељна знања која могу лако да се надограђују у било ком смеру у наставку школовања. Кад смо у прошлом броју помињали медијску блокаду *FLOSS*-а, рекли смо да комерцијални медији не морају да причају о *FLOSS*-у, али да школа мора.

Има више разлога зашто мислимо да је основно школство право време за сва темељна знања, па и за озбиљније овладавање информатичким знањима, а у склопу са њима и са знањима о *FLOSS*-у. Три разлога су основна.

Први разлог се поклапа са општим мишљењем да је информатичка





писменост за даљи живот одмах иза функционалне писмености на матерњем језику и функционалне писмености на неком страном језику. Кад се запошљаваш, више нико не пита да ли си писмен на матерњем језику, то се подразумева. Прво питање је да ли знаш неки страни језик, а након тога да ли си компјутерски писмен. Ако је информатика у животу толико важна, онда и треба да се проучава упоредо са осталим темељним предметима као што су матерњи језик, страни језик и математика.

Други разлог је плод вишедеценијске транзиције и општег осиромашења. Статистике показују да све више деце не наставља даље школовање после обавезне основне школе. Ово су поразне чињенице. Средњошколско образовање није обавезно, зато овладавање основном информатичком писменошћу не треба одлагати на период средње школе, под изговором да деца у основној школи нису стасала за информатику. Основци су мали сунђери знања и предодређени су да лакше од старијих усвајају све врсте новотарија. Не стоје тврдње да информатика не треба деци која не желе или не могу да наставе школовање након основне школе јер то је део опште писмености.

Трећи разлог се односи директно на *FLOSS* у основној школи. Чињенице да данашња деца, ако имају услове у родитељском дому, у основну школу доносе неко предзнање о информатици. Наставници се углавном слажу да деца већ доста знају о рачунарима, али су та знања несистематична и нефункционална. Та знања су плод визуелног

памћења. Исправљање нечега што је погрешно научено, понекад је теже него учење од почетка. Овде *FLOSS* може да помогне. Релативно мала заступљеност *FLOSS*-а код нас омогућава да се изједначе деца која никад нису видела компјутер са оном која су виђала само *Windows*. *Linux* је довољно различит да поништи већ стечена погрешна знања, а довољно једнак да се преко њега могу надоградити знања из било које области информатике, али сада на добрим основама.

Тренутно стање

Иако се сви слажу да је информатичка писменост део опште писмености становништва, код нас информатика није изборила ни приближни статус матерњег језика, страног језика или математике. Информатика још нема свој обавезни званични предмет у основној школи. Информатика у основним школама се изучава кроз три предмета:

1. Од играчке до рачунара (изборни предмет - нижи разреди) ,
2. Техничко и информатичко образовање (обавезни предмет - виши разреди) и
3. Информатика и рачунарство (изборни предмет - виши разреди).

Оваквим стањем су збуњени и ученици и наставници. Наставни планови ових предмета се преклапају и понављају. Изборни предмети се не оцењују па ни ученици не сматрају да су нешто много важни. Изостанци са изборних предмета нису санкционисани ни на који начин, чак ни кроз смањење оцене из владања.



Изучавање информатике кроз изборне предмете се показало само као чисто губљење времена.

Наше основне школе углавном немају информатичке кабинете, што своди учење на папир и оловку. Деца са предзнањем је то досадно, а онима који нису ни видели рачунар, то су „шпанска села”. Онда и није чудно што деца беже са часова изборног предмета.

Хардвер

Без рачунара нема озбиљног учења информатике и не само информатике него и свих других предмета. Чисто учење теорије не оставља трајан траг у меморији. Деца многе ствари памте визуелно. Без практичног рада можда ће имати добре оцене, али ће и даље падати на *PISA* тестовима јер је такво знање површно и неповезано са праксом.

Многе основне школе код нас немају информатички кабинет, па чак ни рачунар који би им служио као наставно средство. Сиромаштво доводи до тога да и они који имају рачунарски кабинет, имају застареле рачунаре. Чињеница је да се информатика брзо развија и да најмодернији рачунари за пет до шест година застаревају. Наша земља нема толика средства да све школе снабдева новим рачунарима, а камоли да их сваких пет до шест година обнавља.

Шта је решење? Прво треба утврдити шта ће деца на тим рачунарима радити. Сигурно је да неће радити нешто што захтева превише процесорске снаге и велику количину радне меморије. Основци ће учити основне информатичке појмове: хардвер, оперативни систем, апликативни софтвер, мреже, интернет, комуникацију и основе програмирања. Све то је боље учити на





старом рачунару него помоћу папира и оловке. Ако је хардвер стар, софтвер не мора бити. Једини нови софтвер који може да покреће старе рачунаре, јесте слободни софтвер.

Кад смо утврдили да се може одвијати настава на старом хардверу али без губитка актуелности захваљујући новом, актуелном слободном софтверу, онда се намеће и једино решење за недостајећи хардвер у основним школама. Државни органи и јавна предузећа имају рачунаре који застаревају за озбиљни рад. Уместо да буду рециклирани, много боље би било да буду донирани школама које немају информатичке кабинете. Са малим улагањем и уз помоћ локалне заједнице слободног софтвера тај хардвер може да се опреми са новим софтвером и припреми за своју нову образовну улогу.

Софтвер

Ситуација са хардвером намеће изучавање основа информатике на *FLOSS* решењима. Да ли ће тиме наша деца бити оштећена у односу на децу из богатијег дела света? Ми мислимо да неће. Штавише, изучавање основа информатике је темељније на софтверима отвореног кода. Са друге стране, читав богатији свет се све више окреће ка *FLOSS* решењима, тако да ће познавање *FLOSS*-а бити све битније. Са треће стране, кад деца кроз *FLOSS* схвате базично функционисање хардвера и начин на који корисник преко софтвера комуницира са тим хардвером, неће им бити тешко да после у вишим нивоима школовања овладају специфичним новим хардвером и софтвером затвореног кода.

FLOSS не само што је, захваљујући отвореном коду, транспарентан и





погодан за изучавање основа информатике, већ поседује и велику количину едукативног занимљивог софтвера примереног основцима. Ово омогућава да преко информатике нека друга знања која су стечена у школи, деца примене и вежбају.

Нативно вишекориснички систем, какав је *FLOSS*, омогућује безбедну употребу рачунара без „тренирања строгоће”. Сваки узраст ученика може да добије своја специфична овлашћења која им омогућују да слободно раде оно што је примерено за њихов ниво, а остало што није препоручљиво за њих, забраниће им сам систем. Ово учитељима и наставницима омогућава да се опусте и престану да изигравају „полицајце” који стриктно контролишу шта деца раде. Чак и у условима када најстарија генерација има најшира могућа овлашћења над системом, наставници имају, са једног рачунара, одличан преглед и контролу

над свим умреженим рачунарима.

FLOSS је мање рањив на инфекцију рачунарским вирусима, што олакшава одржавање система. Не само да се олакшава одржавање система, него се и штеди новац за специфичне антивирусне софтвере без којег се власнички софтвер не може замислити.

Закључак

1. Информатика је део основне писмености човека и као таква мора добити статус осталих темељних предмета као што су матерњи језик, страни језик и математика.
2. Без рачунара нема озбиљне наставе информатике, а и у осталим предметима рачунар би морао бити један од важнијих наставних средстава.
3. Рачунари који су завршили свој продуктивни рад, могу да наставе свој рад у образовном процесу.



4. Ако не можемо обезбедити нови хардвер, нови софтвер можемо. С обзиром на то, *FLOSS* је логично решење које се само намеће старом хардверу.
5. *FLOSS* је јефтин, безбедан и погодан за проучавање основа информатике, довољно различит од власничког софтвера да сва деца, без обзира на предзнање, почињу од нуле, а довољно је сличан да се знање усвојено на *FLOSS*-у лако може надограђивати на било којој другој платформи без понављања.
6. Вишекориснички систем који обезбеђује *FLOSS* омогућује да на истим рачунарима уче и виши и нижи разреди, али са различитим овлашћењима који су примерени сваком нивоу обуке понаособ.

Много је разлога зашто је информатика неопходна и то од самог почетка школовања. Оно што сада имамо у тој области, неодговарајуће је и води у заостајање за развијеним светом. Развијени свет кроз *FLOSS* намерно штеди и масовно прелази на та решења. Ми смо у прилици да немамо куд. За скуп хардвер и софтвер немамо средстава. Иако идемо са различитим мотивима ка *FLOSS*-у, ми и развијени свет баш уз помоћ *FLOSS*-а можемо да се нађемо у заједничкој тачки.

Ову реформу није лако спровести, али ако се сви уозбиљимо и кренемо заједно ка неком циљу, изводљиво је. Потребна је политичка воља и боља организација *FLOSS* заједница, али о томе ћемо у неком другом наставку.

Наставиће се.

Преглед популарности *GNU/Linux* /*BSD* дистрибуција за месец април

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>

Пад <
Пораст >
Исти рејтинг =
(Коришћени подаци са *Distrowatch*-а)



Презентација:

PRIVREDNA KOMORA BEOGRADA
www.kombeg.org.rs

ubuntu SRBIJA

LINUX U PRAKSI

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

Аутор: Дејан Маглов

Место: Привредна комора Београда,
Кнеза Милоша 12, Београд
Време: 17.04.2014.
Организатори: Привредна комора
Београда и *Ubuntu* заједница Србије

У Привредној комори Београда, 17. априла 2014. године, одржана је презентација „*Linux* у пракси”. Суорганизатор овог скупа поред Привредне коморе Београда била је *Ubuntu* заједница Србије. Циљ скупа је да се привредницима Београда представе *FLOSS* алтернативе најчешће коришћених софтвера у пословном окружењу.

Презентацијом су биле обухваћене следеће теме:

- Увод у *FLOSS*
- *Linux* као серверски систем
- *Linux* као носећи ОС у предузећима
- *FLOSS* софтвер у свакодневной пословној употреби



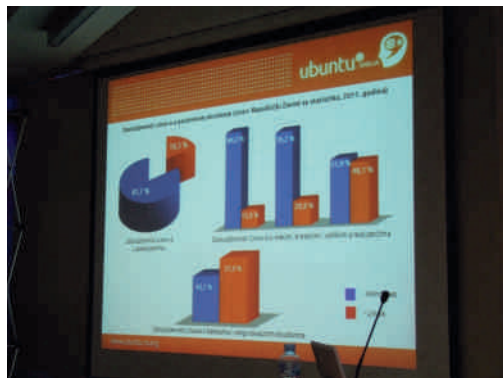


Презентација

Након поздравног говора који је у име домаћина, Привредне коморе Београда, одржао Душан Ракић, уводну реч је имао Ладислав Урошевић, координатор *Ubuntu* заједнице Србије.



великим предузећима.



Статистика заступљености *FLOSS*-а по областима примене показује доминацију власничког софтвера на пољу *desktop* рачунара и пословних сервера, док на *web*-серверима, *web*-у и супер-компјутерима *FLOSS* решења су доминантна. Зато је други део презентације коју је водио Бојан Богдановић, главни координатор *Ubuntu* заједнице Србије, била посвећена *FLOSS* алтернативи за пословни сервер.

Он је присутнима представио *FLOSS* као алтернативу постојећим власничким софтверима и презентовао тренутно стање заступљености *FLOSS* решења у Србији. По статистикама које смо могли да чујемо од Ладислава Урошевића, заступљеност *FLOSS*-а у нашим предузећима је пропорционална са величином предузећа. У великим предузећима заступљеност *FLOSS* решења је знатно већа него код малих и средњих предузећа. С обзиром да је учешће малих и средњих предузећа у нашој привреди на нивоу 50% укупних прихода привреде и преко 90% од укупног броја предузећа у Србији, отвара се велики простор за ширење *FLOSS*-а у пословном окружењу. Ако велика предузећа виде интерес и употребљивост *FLOSS*-а у пословном окружењу, поставља се питање зашто се то не би прелило и на мања и средња предузећа, којима уштеде на трошковима софтвера значе, можда више него



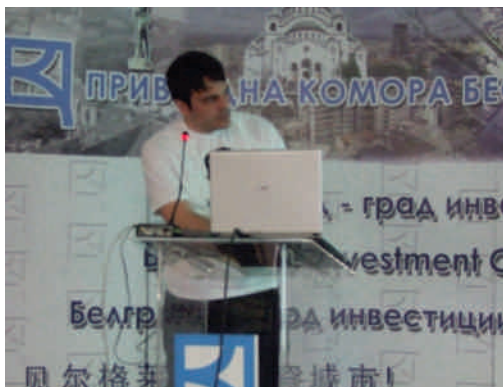
Бојан Богдановић је присутнима укратко представио *Zentyal*, *FLOSS* сервер базиран на *Ubuntu*. Поента овог дела презентације је да *Zentyal* има пуну примењивост попут власничког сервера али



уз знатно мање трошкове. Једино што се, евентуално, плаћа применом *Zentyal*-а јесте подршка и одржавање од стране неког стручног лица, све остало је бесплатно, али је суштина да оваква уштеда не иде на уштрб квалитета оваквог серверског решења.



За илустрацију колико *FLOSS* може да буде моћан, био је задужен Марко Новаковић, сарадник *Ubuntu* заједнице Србије.

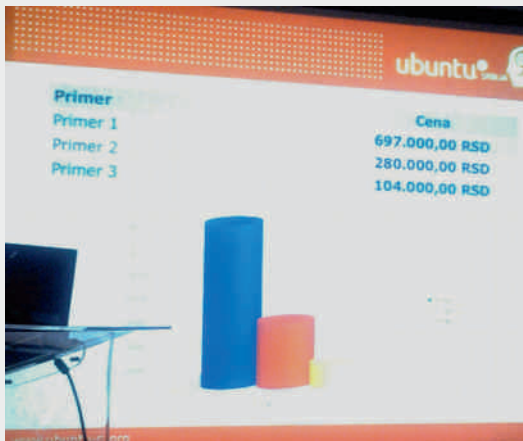
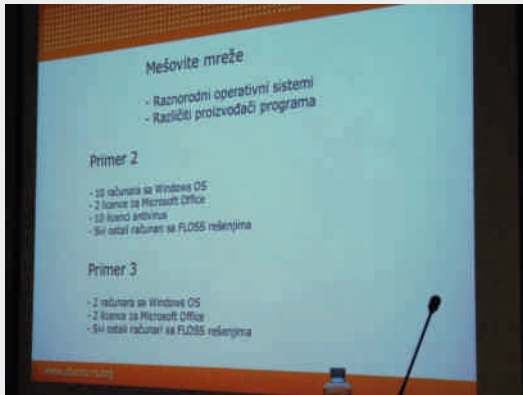


Марко је присутним гостима представио *Ubuntu studio*, специјализовану *Ubuntu* верзију за развој мултимедијалних пројеката, као и неколико готових мултимедијалних пројеката који су развијени само уз помоћ *FLOSS*-а. Нарочито је

била ефективна презентација делова анимираног филма који је у потпуности био израђен помоћу *Blender*-а, слободног софтвера за 3D моделовање и анимацију.



У наставку презентације Бојан Богдановић је дао и један пример могуће уштеде на софтверу за мало предузеће са десет рачунара. Рачуница је показала да то предузеће, да би опремило тих десет рачунара са легалним власничким оперативним системом (нпр. *Windows*, јефтинија варијанта у односу на *MAC OS X*), власничким *Office* пакетима (*MS. Office*), антивирусним софтвером (нпр. *AVG*), софтвером за компримовање фајлова (*zip*), софтвером за нарезивање дискова (*Nero*) и само једним сервером (*SBS/ Essentials*), мора да издвоји око 697 000 РСД. Ово је, морамо признати, приличан издатак за мало предузеће. Уштеда коју Бојан препоручује, јесте замена неких власничких софтвера *FLOSS* алтернативама и то у две варијанте. Прва варијанта је узета уз претпоставку да је власнички *OS* дошао прединсталиран са хардвером и да се, као такав, његов трошак не може избећи. Уз власнички *OS* морају се додати још



десет лиценци за антивирусни софтвер и само две лиценце за власнички *Office* пакет, због специфичних потреба предузећа. Ако се остали софтвери замене са *FLOSS* алтернативама (*Zentyal server*, *LibreOffice*, *7-zip*, *InfraRecorder* и други), трошкови ове варијанте се смањује на 280 000 РСД. Трећа варијанта је уједно и комбиновани систем који подразумева само два рачунара са власничким *OS*-ом, два антивирусна пакета и две лиценце за власнички *Office* пакет, који могу да обезбеде несметано извршавање неких специфичних потреба предузећа за власничким софтвером, а сви остали програми на та два рачунара, као и остали рачунари, били би опремљени *FLOSS* решењима, плус *FLOSS* сервер. Овакав модел трошкове своди на само 104 000 РСД. Уштеда је значајна, а при томе није угрожено функционисање предузећа.

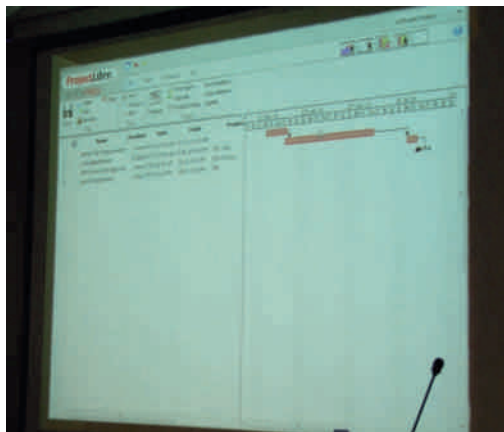
На крају презентације Милан Влаховић из Инфо центра, сектора за информатику, представио је више *FLOSS* алтернатива које су јако примењиве у пословном окружењу.



Прво је представио алтернативе за *MS Project manager* као што су *ProjectLibre*,



GanttProject и *GNOME Planner*. Затим је представио програме за визуелизацију идеја као што су *FreePlane*, *FreeMind* и *VYM*. Такође, занимљива област за пословну примену је управљање са пословима и временом. Милан Влаховић је представио *Korganizer*, *Getting Things Gnome*, *Gnome Time Tracker* и *GTD-free*. Статистичка обрада и графички приказ података је такође занимљива област за пословну примену, а Влаховић је у тој области представио *RStudio* и *GNU PSPP* (замена за *IBM SPSS*).



Закључак

Презентација „*Linux* у пракси” по нашем мишљењу је успела. Привукла је пажњу четрдесетак гостију који су могли да чују доста занимљивих информација о примени *FLOSS*-а у пословном окружењу. Презентација није била усмерена према промовисању *Ubuntu*-а, како би неко могао да помисли с обзиром да је суорганизатор била *Ubuntu* заједница Србије. Циљ презентера је првенствено била усмерена ка промовисању *FLOSS*-а уопште, као јефтине и употребљиве

алтернативе власничком софтверу у пословном окружењу.

Ово неће бити једина презентација која ће промовисати *FLOSS* у Привредној комори Београда. За наредни период су најављене презентације:

- „*LibreOffice* на Вашем радном месту” – два дана, 15-16. мај 2014.
- „Мапе ума за боље резултате у Вашем пословању” – 6. јун 2014.
- „Користите *R!* – Интерактивна анализа Ваших података” – два дана, 17-18. јун 2014.
- „Основе управљања пројектима – *ProjectLibre*” – два дана, 26-27. јун 2014.

ЛиБРЕ! поздравља ову акцију Привредне коморе Београда и *Ubuntu* заједнице Србије са циљем промовисања *FLOSS* решења у пословном окружењу као и најаву наставка ове акције. Сматрамо да је ово, можда, најбољи пут за популаризацију *FLOSS*-а. За разлику од приватних корисника који не могу баш у потпуности да схвате концепт легалности и уштеде на *FLOSS*-у, привредници ће бити итекако заинтересовани за овај начин уштеде. Једном када се *FLOSS* устоличи у фирмама, лако ће и из фирми прећи и на кућне рачунаре.





Сербиан 2014 (2. део)

Аутор: Дејан Маглов

Увод

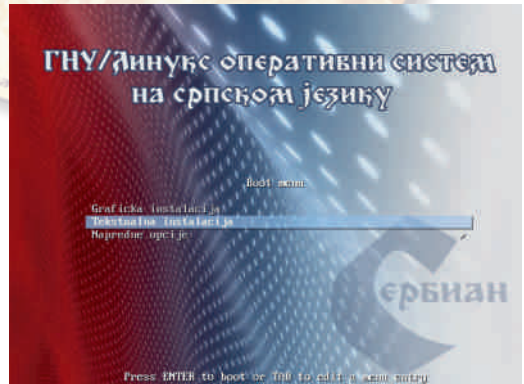
Прошлог пута коментарисали смо концепт Сербиана. Овог пута ћемо се концентрисати на његову употребљивост. И даље ћемо се држати искључиво пројектног задатка који, да подсетимо, претпоставља да је Сербиан српска дистрибуција, потпуно локализована на српски језик и то српском ћирилицом, намењен првенствено мање искусним корисницима *Linux*-а, који кориснику нуди принцип „инсталирај и одмах користи” и на крају, можда и најспорнији задатак, да Сербиан пружи кориснику употребљивост *Mint*-а или *Kubuntu*-а.

Инсталација

Сама чињеница да је Сербиан компонован на бази стабилног *Debian*-а, уноси мало сумње да ће Сербиан испунити задатак да ће бити дистрибуција за почетнике.

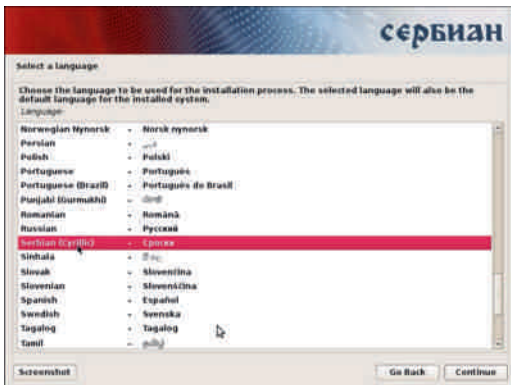
Већ при првом покретању *Serbian2014-1.iso*-а види се да Сербиан нема живи диск. По нашем мишљењу то је мана. Данас је већ стандардно да дистрибуције које флертују са почетницима, имају

живи диск. Живи диск је излог за дистрибуцију који будућем кориснику презентује, угрубо, своје могућности и прилагођеност хардверу на којем ће радити. Заморно је инсталирати дистрибуцију због пробе, а чист губитак је времена ако инсталација не проради на нашем хардверу.



Занемаримо сад чињеницу да живог диска нема и пређимо на инсталацију. У понуди нам је графички и *CLI* инсталер. Почетник ће изабрати углавном графички инсталер, што и ми чинимо. Дочекује нас *Debian*-ов графички инсталер. Поуздан је, али знатно компликованији од инсталера *Kubuntu*-а или *Mint*-а. Заправо, није толико у питању компликованост него чињеница да овај графички инсталер изгледа као мало

нашминкани *CLI* инсталер са приличним бројем корака, при чему сваки корак одрађује само по једну функцију. За разлику од *Debian*-овог графичког инсталера, модерни графички инсталери, попут инсталера *Kubuntu*-а или *Mint*-а, имају мањи број корака, али унутар њих је груписано више подешавања, као на пример подешавање времена и локације или подешавање *root* и корисничких налога, или сва подешавања партиционисања на једној страни. Ово није примедба на рачун Србиана, барем не директно, него више на конзервативност *Debian*-а.



Ако се апсолутни почетник снађе у овом *Debian*-овом графичком инсталеру, инсталација ће протећи глатко и брзо, са аутоматским рестартом на крају.

Први поглед на Србиан

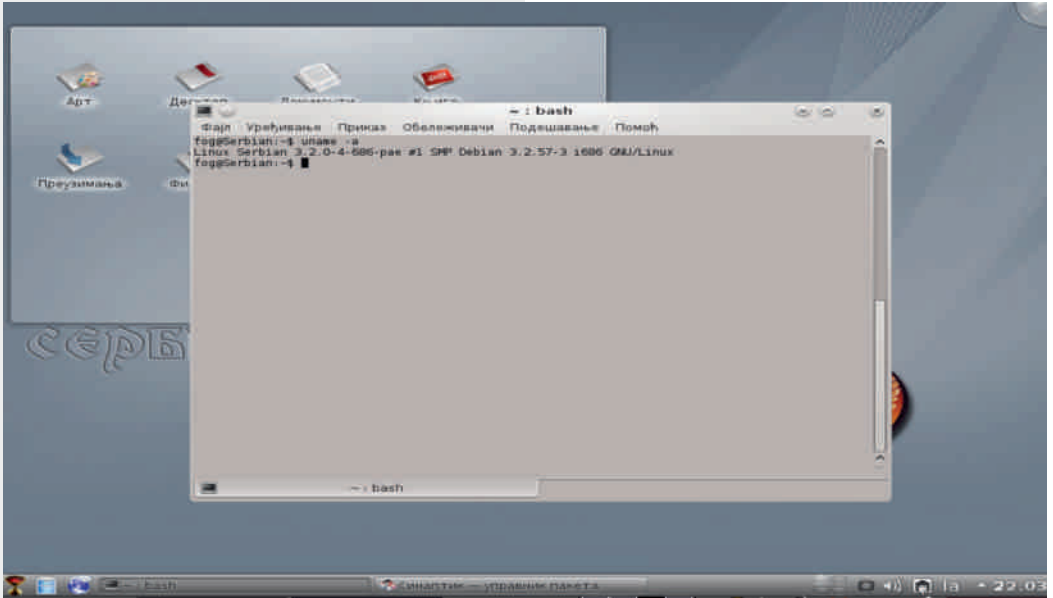
Једна од озбиљнијих примедби на форумима, а на рачун Србиана јесте његов подразумевани изглед. Колико смо ми схватили, није примедба упућена на „ружноћу” Србиана. Руку на срце, треба бити мајстор и толико упропастити *KDE*,

па да он буде ружан. Ствар је у томе да многи нису разумели да је ово **српска дистрибуција**, па самим тим су и напали стилизована национална обележја која Србиан има у подразумеваном изгледу. Не видимо никаквог разлога да се ми бунимо против наших обележја нити видимо разлога да правимо дистрибуцију за читав свет. Нама треба српска дистрибуција и није лоше да корисник на први поглед види да је на правом месту.

С обзиром да је могуће и на неки други начин локализовати Србиан, на пример у босанску латиницу, исто тако се може врло лако променити и спорни *wallpaper*, тако да не видимо препреку да и они који нису Срби, користе ову првенствено српску дистрибуцију, уколико то желе.

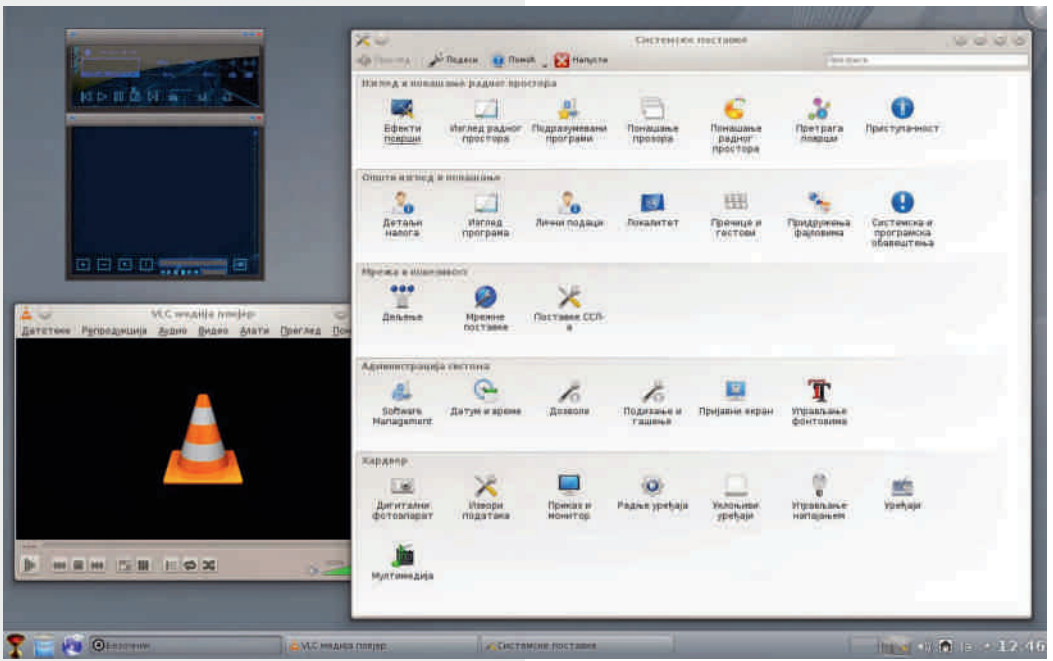
Пошто не постоји живи диск Србиана, једини начин да видите његов изглед, јесте да га инсталирате и ако имате среће са хардвером (као ми на тестној машини), након рестарта добићете Србиан *KDE* у пуном сјају. Први утисак након рестарта јесте да је композитор Србиана са укусом уредио графички интерфејс, без вишкова и кичастих детаља. Монотонију сивих нијанси лепо разбијају дискови са стилизованим српским грбом у виду Феникса. Уз подразумевани *wallpaper* лепо се уклопила *oxigen KDE* тема, тако да је општи утисак пријатан за око.

Избор *KDE*-а за подразумевано окружење радне површи главног издања Србиана увек ће бити контроверзан. *Linux* корисници се бинарно деле у два табора – на оне који обожавају *KDE* због његовог префињеног изгледа и на остале који га не воле због „компликованог” поде-



шавања и оптерећења графичке картице, која због обима KDE ефеката може да

почне да „штуца“ ако је стара и лоша. Средине нема. Композитори Србијана





су, чини нам се, били свесни ове чињенице, па су развили и *Openbox* Србиан за слабије и старије рачунаре, али и за оне који не воле *KDE*.

Објективно, комбинација стабилног *Debian*-а са старим али поузданим *LTS kernel*-ом 3.2 и *KDE* 4.8 није баш идеални спој. *Kernel* 3.2 добро ради на старијим рачунарима. *KDE* 4.8 може да ради на тим старијим рачунарима, али они морају да буду мало јачи да *KDE* не би „штучао“. За одређивање максималне и минималне конфигурације хардвера који подржава Србиан, треба мало више времена и више тестера тако да вам не можемо сада рећи која је оптимална конфигурација хардвера за Србиан.

Локализација

Пријатан је осећај када те на рачунару дочека матерњи језик исписан ћирилицом. Трудили смо се да нађемо неки програм или упутство које није преведено, али нисмо успели да га нађемо. Све је локализовано, барем у *KDE* издању. У *Openbox* издању смо нашли неколико нелокализованих делова или делимично локализованих програма. Најучљивија је делимична локализација *Geany*-а. Са те стране корисници *KDE* издања могу да буду задовољни.

Оно што најчешће није локализовано, јесте систем помоћи. *KDE* систем помоћи је код нас био подразумевано искључен, тако да нисмо успели да видимо да ли је ту све локализовано. За наредни наставак рецензије Србиана имамо у плану интервју са аутором, па



ћемо га питати да ли је намерно искључен овај систем помоћи или ми нисмо нешто добро подесили. Ако је намерно искључен, питаћемо га зашто је искључен и да ли ће бити у некој другој верзији укључен.

Персонална подешавања окружења и система

По једноставности персонализације система можемо да одредимо за кога је заправо намењен тај дистри. Када кажемо да је дистри за почетнике, треба разликовати апсолутне рачунарске почетнике и искусне кориснике рачунара, али почетнике на *Linux*-у. По нашем мишљењу, идеални дистри за апсолутне почетнике је *Slackware*. Како?! Једноставно, апсолутни почетници свакако не знају да инсталирају ни један *OS*, па онда морају да траже помоћ од искуснијих рачунарских корисника да им инсталирају систем. Након инсталације, *Slackware* корисник има све што почетнику треба, а нема лако доступне графичке помоћнице где почетник може, чепркајући по системским подешавањима, лако направити погрешку и наудити свом систему.

Већи проблем су ови „искусни почетници”. Они су захтевнији у погледу изгледа, функционалности свог система, а размажени су власничким софтвером који им нуди графичко сучеље за свако подешавање система. *Mint* и *Kubuntu* су се прилично приближили овим „искусним почетницима”, нудећи сва подешавања на клик. Ово је двосекли мач јер омогућава почетницима да својим акцијама науде систему.

Србиан је ту негде на пола пута између *Slackware*-а и *Kubuntu*-а, што је стандардно за *Debian*. Оно што олакшава, доступна подешавања на клик имају све опције на српском језику, па ће се у тим подешавањима снаћи и почетници који не познају довољно енглески језик.

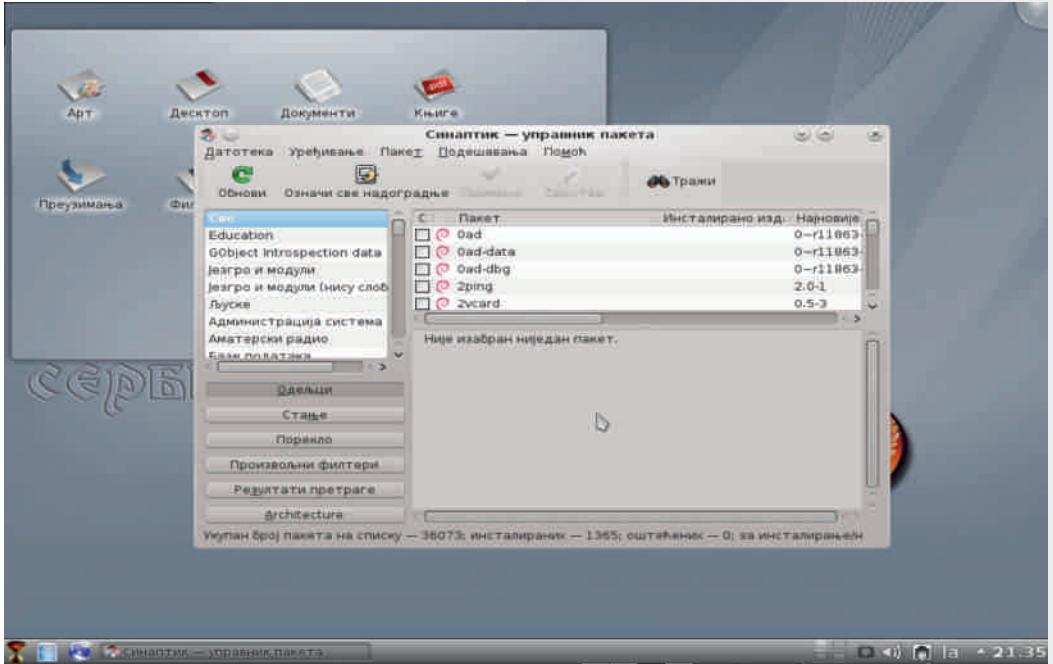
Програми

Највећа замерка коју истичу *Linux* корисници, који не воле *KDE*, јесте то да уз *KDE* долази и гомила не баш тако популарног *KDE* софтвера. Уз *KDE* обично долази *Koffice* уместо *LibreOffice*-а или *Krita* уместо *GIMP*-а или *Konqueror* уместо *Firefox*-а и слично. *KDE* оријентисане дистрибуције углавном промовишу ове *KDE* пројекте који нису лоши, али ипак нису најпопуларнији *Linux* софтвери и драго нам је да Србиан није кренуо тим смером.

Србиан је задржао најважније, најбоље и битне *KDE* апликације (*Dolphin*, *Kate*, *K3b*, *Gwenview*, *Konversation* и *Okular*), али се није либио да дода и *GTK* апликације (*GIMP*, *VLC*, *LibreOffice*, *IceWeasel* (*Debian* форк *Firefox*-а), *Chromium*, *Skype* и *Clementine*). Закључак је да Србиан има добро избалансиран избор најпопуларнијих *FLOSS* програма без много дуплирања.

Србиан нема фенси пакет менаџер попут *Kubuntu*-а (Софтверски центар). Он се углавном ослања на поуздани *Synaptic*. *Synaptic* можда није најлепши пакет менаџер за почетнике, али је од софтверског центра и бржи и поузданији.

По завршеној инсталацији додали смо препоручене ризнице. Искористили смо



прилику да пробамо *Synaptic* за *update* (освежење) система и за пробу додали још неколико програма (*Inkscape* и *Scribus*). Освежавање система и инсталација је протекла глатко, а и програми су задржали подразумевану локализацију, углавном. *Inkscape*, на пример, био је потпуно локализован на српску ћирилицу, али *ImageMagick* који је повучен као његова међузависност, није.

Закључак

Закључак је да је подразумевана поставка Србиана лепо испоштовала задату локализацију, али ће се током коришћења појавити у новоинсталираним програмима и програми који нису локализовани, и не испуњавају пројектни задатак Србиана. Да би се ово избегло, једини начин је формирати сопствене

ризнице са искључиво потпуно локализованим софтверским пакетима. Ово је вероватно превелики задатак за малу заједницу која одржава овај систем. Можда једног дана у будућности.

Мада смо имали намеру да више критикујемо изведбу Србиана у овом броју, ипак смо се ограничили само на утврђивање стања. Покушаћемо до следећег броја да ступимо у контакт са аутором и да му поставимо питања која ће нам разјаснити мистерију, зашто Србиан изгледа баш овако и који су планови за будућност овог пројекта.





VI text editor

Аутор: Александар Брковић

У четвртном броју смо вам писали о такозваном „рату уређивача текста“ где смо радили поређење неких уређивача. У овом чланку вам представљамо конкретно један од њих.

Linux дистрибуције су погодне за обраду текста исто као и други оперативни системи. Постоји више добрих програма за обраду текста на *Linux* платформи, као што су *AbiWord*, *Koffice* пакет, *OpenOffice* и *LibreOffice*. Поред наведених програма, текст се може обрађивати и у терминал

моду. Један од најпознатијих уређивача текста у терминал моду јесте *Vi* уређивач текста. *Vi* је програм за обраду текста без много функција. Његова главна одлика је писање текста без форматирања, тако да га други програм може прочитати. *Vi* је програм који долази из *UNIX*-а. Постоји побољшана верзија која се назива *Vim*. Проблем са овим уређивачем је што га многи људи не воле. Корисник је принуђен да запамти много комбинација тастера да би био у могућности да уради ствари које би неки други уређивач текстова урадио много лакше.

```
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.alioth.debian.org
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 уређивач текстова је практичан у ситуацији када је потребно брзо унети промене у текстуалну датотеку или написати кратку белешку. Са друге стране, постоје корисници који уз помоћ

датотека која се уређује. То је видљив део текста који може да се измени уз помоћ командне секвенце. Након извршења командне секвенце, промене које је иницирао корисник,

```

File Edit View Bookmarks Settings Help
~$ cat /etc/default/grub
# If 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=s0
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] 341, 1238C 1, 1 All
  
```

ekranu nakon zavrhretka procesa. Kada se pozove Vi uređivac teksta, privremena radna površina koja se otvara, naziva se *buffer*. Ova *buffer* ili radni prostor je na raspolaganju sve vreme dok se koristi tekstualni uređivac. Korisnik može u nje da stavi datoteku koja se veñ nalazi u rachu-

araru, ili da kreira novu datoteku. Nakon što korisnik napusti uređivac, privremeni *buffer* ðe biti odbacen. ovog uređivaca rade komplikovane stvari, čak i dizajniranje *web* strana. Ova *editor* je takođe veoma efikasna i za poređanje šablona, konstruisanje makroa i izvršavanje *shell* komandi iz uređivaca. Navedeni uređivac je *full-screen* uređivac, koji funkcioniše nad čitavom datotekom odjednom, koristeći pri tome ceo ekran terminalnog prozora. Dovoljava kretanje kroz datoteku i izvođenje promena na različitim lokacijama za vreme iste sesije.

Organizacija VI uređivaca teksta

Organizacija VI uređivaca teksta

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

Display je orijentisan a program interaktivna. Vi tekstualni uređivac omogućava korisniku da stvori, modifikuje i memorise datoteke na rachu, a sve preko terminala. Pri likom korišćenja ovog programa, terminal deluje kao prozor u kome se vidi

Komandni mod je početno stanje na koje se nailazi kada se Vi uređivac poziva iz UNIX luskе. To je stanje koje je spremno za prijem uputstava o tome, kakvu akciju je potrebno izvršiti. Najčешće su to



скроловање, претрага текста, померање курсора, прелазак на режим уноса текста, уметање текста, брисање линије, речи или знакова, мењање линије речи или знакова и слично.

Режим уноса текста је друга страна *Vi* уређивача. Режиму уноса текста се може приступити само из командног режима. Корисници који умеју да раздвоје ова два режима рада, врло брзо ће савладати контролу овог уређивача текстуалних датотека. Након што из командног мода корисник уђе у режим уноса текста, све накнадно откуцано у терминалу ће се директно одразити на датотеку.

Приликом уређивања неке датотеке, корисник долази у ситуацију да се креће напред и назад између командног режима и режима за унос текста. На пример, корисник може да одлучи да промени реч у једној реченици (мод уноса текста), па затим да се помери на крај датотеке (командни режим), дода још једну страницу текста (режим за унос текста), па онда исправи неке грешке у куцању (режим уноса текста), помери се на средину текста (командни режим), па ту онда убаци нову реченицу (режим уноса текста). Многим корисницима ће ово звучати компликовано, међутим није тако. Ово је много лакше урадити, него читати.

Најчешћа грешка коју корисници направе приликом рада са *Vi* текстуалним уређивачем, јесте да забораве у коме режиму рада се тренутно налазе. Ова заборавност доводи до тога да корисници покушају да издају команду док се налазе у моду за унос текста. Ово резултира чудном комбинацијом знакова који се

појаве у терминалу, што натера корисника да упути низ негативних речи на рачун овог текстуалног уређивача. Уколико се појави недоумица у коме режиму рада се тренутно налазите, притисните *ESC* типку која се налази у горњем левом углу тастатуре. На овај начин се враћате у командни режим и знаћете где се налазите.

Vi уређивач текста поседује напредну функцију која омогућава кориснику да измени више датотека одједном. Могућност приступа ка више датотека без напуштања текстуалног едитора дозвољава кориснику да потражи информације у другој датотеци, без напуштања уређивача. Датотеке се отварају у истом уређивачу и могу да деле исте именоване *buffer*-е, чиме се омогућава дељење текста и његов пренос између датотека.

Уколико користите неки од једноставних уређивача за унос текста, као што су *Gedit*, *Mousepad*, *Leafpad*, *Kate*, препоручујемо вам да испробате *Vi* уређивач текста. Рад са овим уређивачем није толико тежак, као што изгледа на први поглед. Надамо се да ће овај текст заинтересовати наше читаоце да учине такав корак.

Корисни линкови:

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



OpenStreetMap



Аутор: Јоаким Јањатовић

Шта је OSM и чему служи?

OpenStreetMap је картографски еко-систем несагледивих размера. Да бисмо једноставније објаснили од чега се састоји, упоредићемо га са два свима позната система, Wikipedia-ом и Linux-ом. Попут Wikipedia-е, OSM има централну базу података <http://osm.org> коју може да допуни свако ко има отворен налог е-поште и жељу да подели своје знање са другима. Подаци послати централној бази одмах постају доступни свим регистрованим и нерегистрованим корисницима. Ако загребемо мало дубље, долазимо до паралеле са Linux-ом. База података је као kernel тј. језгро оперативног система које представља ослонац за све остале програме. Зашто баш поређење са Linux-ом? Нико на свету не може да испрати колико тачно дистрибуција Linux-а постоји, па чак ни то колико програма има за само једну од тих дистрибуција. OSM је основа за екосистем у ком уствари нико не зна где су границе тог система. Тачан одговор на питање чему служи OSM, аутор овог текста не може дати.

Први наш додир са OSM-ом, био је преко главног портала и базе података

<http://osm.org>. Ако је неко већ користио *Google Map Maker*, може претпоставити како би OSM могао да функционише. Међутим, OSM је много више од *web* алата за цртање улица. С обзиром да се у Србији за тражење нечега на карти најчешће користи *Google Maps*, крећемо од тога да OSM почетнику личи на GM, али временом се стиче сасвим другачији утисак.

Лиценца

OSM база података доступна је под ODBL лиценцом. У преводу, то значи да сви подаци могу да се користе у комерцијалне и некомерцијалне сврхе, а једино је обавезно нагласити да подаци долазе управо од OSM и његових „доприносиоца”. Наравно, с циљем опстанка пројекта и поштовања ауторских права, забрањено је прецртавање из других извора као што су *Google Maps* или штампане карте, осим ако немате изричиту дозволу власника тих података.

Како то ради?

Ако сте само посматрач, ради тако што отворите *web* страницу или неки програм који не мора да буде повезан са интернетом, а затим претражујете шта



вас занима на учитаној *OSM* карти. Ако путујете негде и треба вам *GPS* навигација, преузмете неки од већ доступних програма за платформу вашег мобилног уређаја, учитате карту за државу или део света који вам треба, и унесете одредиште. Чар је у томе што је и већина програма која користи *OSM* карте доступна бесплатно. Тако, рецимо уз *OsmAnd*, *Android* телефон постаје ауто-навигација којој је довољно једнократно повезивање на интернет ради преузимања карте.

За оне који се питају зашто нека улица, продавница или фирма није учртана на карти, одговор је да могу и сами да је учртају. Управо то је *OSM* мото. Када би свако учртао бар једну улицу, имали бисмо потпуну и најдетаљнију карту

света. *OSM* се заснива на знању мештана који желе то знање да поделе. У неким тренуцима, то подсећа на игрице у којима не видите одмах целу карту, већ како напредујете, откривате део по део (прим.аут.). Цртачи-картографи имају на располагању више врста алата, од којих се најозбиљнијим сматра *JOSM*. То је уређивач направљен у *Java*-и и зато ради на више врста оперативних система. Доступне су разне врсте подлога укључујући и *Bing* сателитске снимке, сопствене и *GPS* трагове других корисника, као и мноштво додатака од којих бисмо издвојили и дело домаћих добровољаца *SerbianTransliterator*. Он омогућава уношење назива на једном писму, рецимо ћирилице, и једноставно пресловљавање на латиницу.



- 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

slippy map covering [Belgrade](#).'"/>

Где је почело и докле се стигло?

OSM пројекат је започет у Енглеској и то из потребе за картом која нема лиценцна ограничења употребе. Тренутно је најразвијенији у Немачкој, где се често као пример запањујуће детаљне карте наводи зоолошки врт у Берлину. OSM тренутно броји преко милион и петсто хиљада регистрованих корисника, од којих је више од двадесет хиљада месечно активних картографа. У базу је могуће унети скоро било који податак, а на кориснику, односно пружаоцу услуга приказа је да одлучи шта од тога жели да види на карти. У Србији OSM заједница се шири полако, али сигурно. Основним подацима покривени су сви већи градови. Мања места још увек зависе од

тога да ли је неки мештанин чуо за OSM, или да ли је неко од активних корисника пролазио кроз то место. Издвојили бисмо и чињеницу да је стандардизован унос назива улица на српском, што омогућава избор приказаног писма као и вишејезични приказ. У Новом Саду уз помоћ OSM добровољаца, LUGoNS-а и ФТН-а већ су одржани предавања и радионица на ову тему.

Ко све користи OSM?

Друштвена мрежа *Foursquare* користи OSM базу података као основу за оцењивање места, затим *web* презентације које имају велик број прегледа дневно, користе OSM уместо *Google Maps*-а јер је бесплатан без обзира на посећеност, а приметили смо да је користе и полицајци



у познатој немачкој ТВ серији „Кобра 11“ (прим.аут.). У случају елементарних непогода, добровољци се скупљају и уцртавају најновије податке, тако да се зна који путеви су проходни а који не. Посебне карте које користе ову базу података, доступне су и за морепловце, планинаре, бицикliste, туристе и људе са посебним потребама. Постоје чак и штампане OSM карте. Моруће је бесплатно преузети комплетну векторску базу података света (~35GB компресовано) и направити сопствени приказ, анализирати податке, а уз добру идеју и зарадити.



Како да се придружим и одакле да почнем?

Препоручујемо да прво нешто више прочитате на OSM wiki страници https://wiki.openstreetmap.org/wiki/Sr:Main_Page, на којој се објављују договори и смернице везане за картографе у Србији. Позивамо вас да нам се придружите и на форуму

<http://forum.openstreetmap.org/viewforum.php?id=62> где ћете наћи одговоре на већ постављена питања, а можете и сами поставити питање. Ту се скупљају корисни линкови везани за OSM. Доступно је доста материјала за обуку почетника од којих су најбољи написани на енглеском и немачком језику. Нажалост, мало садржаја је преведено на српски, па су преводиоци добровољци као и картографи свакако добродошли. Сви заинтересовани нека се јаве на форуму!

Будућност OSM-а?

OSM је одличан вид сарадње отвореног и власничког, односно јавног и комерцијалног, у којем базу података попуњавају углавном добровољци, а онда свако ко има идеју како да је употреби и заради, може то слободно да уради без обавезе плаћања астрономских износа монополистичким компанијама. Баш те фирме које користе OSM податке, спонзоришу и одржавају доста корисних OSM алата. То је рецепт за најжурнију, свима доступну карту света. Не гледамо у стаклену куглу већ у екран, али ипак видимо светлу будућност OSM-а.



Увод у програмски језик C

(2. део)

Аутор: Стефан Ножинић

У претходном броју смо дали оквирну слику о томе како изгледа један једноставан C програм који смо назвали *Hello world* и који је имао задатак да на стандардни излаз испише реченицу *Hello, world*. Такође смо објаснили једноставан процес превођења изворног C кода у извршни коришћењем *GNU Compiler-a (gcc)*. У овом делу вам објашњавамо шта су то типови података, променљиве и како да им доделите вредности у програму или да их учитате са стандардног улаза. Такође вам представљамо и неке основне, али и честе грешке које праве не само почетници него и искусни програмери. Ове грешке се називају синтаксне грешке и настају обично када се јави грешка у куцању (тзв. *typo*).

Типови података

У радној меморији рачунара чувају се разни подаци, као што су бројеви, текст, имена и друго. На пример, када отворите овај часопис, ваш *PDF* читач (или *ePUB*, ако читате то издање) вероватно је учитани фајл сместио у *RAM (Random Access Memory)*, одатле чита потребне податке и приказује их на вашем екрану. Ти подаци се пре приказивања обично

обрађују, па тако добијате лепу и читљиву страницу овог документа. Код *PDF* читача ово се зове рендеровање и служи да векторски формат прикаже као слику на екрану.

Као што смо рекли, постоје разни подаци који се могу чувати у меморији. Они су подељени по њиховом садржају и за сваки тип података је одређено тачно колико меморије заузима. Заузеће меморије зависи од програмског језика, али и од архитектуре на којој се покреће програм, односно за коју је програм преведен (компајлиран).

Сваки тип, поред своје величине, има свој скуп могућих вредности (домен), скуп операција које се могу извршити над овим типом података и скуп релација које важе за овај тип података. Овде вам представљамо неке основне типове података, њихове кључне речи у C програмском језику које ће нам бити корисне код дефинисања променљивих и дајемо вам увид у неке операције, односно релације за сваки од наведених типова.

Први тип који ћемо да поменемо, јесте *integer*. Ово није најједноставнији тип, али се обично наводи као први јер је најближи човековом схватању рачунара.



Овај тип описује целе бројеве (... , -2, -1, 0, 1, 2, ...) и кључна реч у C-у му је *int*. Домен овог типа су цели бројеви који улазе у опсег $[-2^{31}, 2^{31} - 1]$. Сигурно сте се запитали зашто баш овакав опсег и како смо дошли до њега. Одговор лежи у чињеници да на већини архитектура *integer* заузима 4B односно 32 бита. Ако сте у добрим односима са комбинаториком, онда знате да је то укупно 2^{32} комбинација, па је домен подељен на негативне вредности, позитивне вредности и нулу. Операције које важе за овај тип података, исте су оне које важе и за целе бројеве у математици. То су операције доделе (оператор =), сабирање, одузимање, множење, дељење и модуло (% оператор). Ништа није другачије ни са

релацијама, па ту имамо релације једнакости (== оператор), веће, мање, веће или једнако, мање или једнако и *није једнако*. Потребно је обратити пажњу да се оператор за доделу вредности (=) обележава са једним знаком једнакости, док се релација за једнакост обележава са два.

Следећи тип података је *char*. Он представља један карактер. Заузима 1B у меморији, па је тиме могуће описати 256 вредности. Ово је сасвим довољно да опише енглески алфабет, бројеве и још доста других знакова. Најчешћа операција је операција доделе, а најчешћа релација је релација еквиваленције (= - додела, == - релација еквиваленције). Свако слово има своју бројчану вредност





која се може видети у ASCII табели.

Тип података који служи за опис неких реалних бројева је *double*. За њега важе све релације и операције као и за *integer*, изузев модуло операције (она нема овде сврхе, зар не? - прим.аут.).

Последњи тип података који ћемо поменути у овом тексту, јесте *bool* (*boolean* у неким другим језицима). За његов опис је такође потребан 1B и може имати само две вредности, а то су *true* или *false*, односно тачно и нетачно. Операције које се могу извршити на овим типовима, јесу операција доделе, логичко „и” (&&), логичко „или” (||) и логичко „не” (!). Овај тип података враћа све релације на другим типовима података, па се овај тип, иако не на први поглед, доста користи у условном гранању, о којем ће бити више речи у наредном тексту.

Променљиве

Сада када знамо оквирно чему служе неки основни типови података, можемо да их искористимо и сачувамо нешто у меморију.

Свака променљива мора да има свој тип података. Променљиве би обично требало да имају вредност, али ће се ваш програм успешно превести чак и кад нека променљива не буде имала додељену вредност. Наравно, треба напоменути да ово искључиво зависи од подешавања вашег преводаоца. Свака променљива се дефинише на следећа два начина:

```
tip_podataka ime_promenljive =
vrednost;
```

```
tip_podataka ime_promenljive;
```

У првом случају је променљивој додељена одмах вредност, док у другом случају она нема вредност (има неку насумичну вредност која нам није од неке користи). Што се тиче другог случаја, променљивој можемо доделити касније вредност, коришћењем оператора доделе:

```
ime_promenljive = vrednost;
```

Следе примери дефинисања неких променљивих:

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

char d = 'a';

bool e = true;

double f = 0.0015;
```

Учитавање променљивих са стандардног улаза

Сви они који редовно користе конзолне програме, знају да они могу да питају корисника за неки податак током свог извршавања и да чекају да корисник тај податак унесе. Овај податак се потом смешта у меморију и касније се користи и обрађује. После обраде резултат се исписује на стандардни излаз, коришћењем нама свима добро познате функције *printf* коју смо користили у нашем првом програму да бисмо исписали фразу *Hello, world!* на стандардни излаз.

Неко је рекао да се уз пример најбоље учи, а ми кажемо да се уз пример најбоље објашњава.

Програм који учитава два броја и потом исписује њихов збир:

```
#include <stdio.h> 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 0;
}
```

Као што можемо видети, користили смо исто *main* функцију као прошли пут, учитали смо исту библиотеку, али нам је код унутар наше *main* функције мало другачији. Прво смо дефинисали две променљиве типа *int* које нам служе да у њих учитамо задате бројеве. Потом, помоћу нове функције *scanf* учитавамо те бројеве. Ова функција захтева *string* који говори како да парсира улаз, па тако функција зна да ми очекујемо два цела броја (*%d*) размакнута једним размаком. Потребно је приметити да је размак направљен и на почетку *string*-а, што функцији говори да не обраћа пажњу на специјалне знакове, као прелазак у нови ред (ентер). Ово значи да ви можете унети први број, притиснути тастер ентер, а затим унети други број и опет притиснути ентер. Функција тражи и имена променљивих у које ће бити читане вредности. Потребно је приметити знак *&* на почетку имена обе променљиве. Овај знак говори да се

податак упише на локацију адреса тих променљивих у меморији. Ако вам ово изгледа збуњујуће, верујте - нисте једини. О адресама у локацијама у меморији ћемо писати у будућим бројевима када будемо дискутовали о показивачима. Као што сте могли да приметите, обе функције (*scanf* и *printf*) користе специјалне изразе да би се навео тип података. Тако *%d* говори да се на улазу очекује *int* тип података, или да се на излазу исписује вредност променљиве тог типа. Неки остали знакови за форматирање су:

%c - карактер типа *char* *%lf* - реални број, тип података *double* *%d* - цео број, *int* тип података

Честе синтаксне грешке

Оставићемо за крај променљиве и типове мало по страни и враћамо се поново на почетак. Хајде да видимо, које све грешке могу да се направе и како преводилац реагује на њих.

Посматрајмо следећи код:

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

Ако ово покушамо да преведемо са коришћењем *gcc* преводиоца као у тексту који је објављен у прошлом броју, добићемо следећу грешку:

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



Као што видимо, негде смо изоставили „;”, тј. тамо где је требало да се налази. Потребно је погледати у ред где стоји `return 0` без потребног знака на крају.

Погледајмо сада овај код:

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

Шта каже преводац?

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

Видите ли где је грешка? Омашили смо назив библиотеке!

А ово?

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

излаз:

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

Као што видимо, специјалан карактер „\1” не постоји. Потребно је приметити да ће се по почетним подешавањима овако нешто превести уз упозорење.

Завршетак

У овом тексту смо вам дали основне информације о типовима података, променљивама и стандардном улазу и излазу. Исто тако смо показали неке стандардне грешке које праве почетници, али и они искусни, како бисмо вам указали на грешке које ваш преводац избацује и које вам могу помоћи да исте грешке уклоните. Овде треба напоменути да преводац не упозорава на грешке које нису стриктно везане за сам језик, па вас тако неће упозорити ако сте направили део програма који може да се закуца, или ако сте направили сигурносни пропуст.

Learn C Programming



Неколико филозофских мисли о софтверу:

Развој софтвера

Аутор: Дејан Маглов

Увод

Ма колико се човек, као врста, трудио да буде креативан, увек се то своди на опонашање природе. Човеков идеал је вештачки створити нешто што је природа одавно већ створила. У информатичком свету, идеал је креирати вештачку интелигенцију по узору на нас саме, односно наш мозак. Шездесетак година развоја информатике је довољан период да можемо да подвучемо неке прве црте и анализирамо развој из филозофске перспективе.

Креационизам или еволуција

Проучавајући природу, човек је развио две водеће теорије настанка и развоја живота (постанка), креационистичку и теорију еволуције. Креационистичка теорија је старија и заснована је на догми. Креационистичку теорију подржава већина водећих религија света без веће жеље за доказивањем исправности исте.

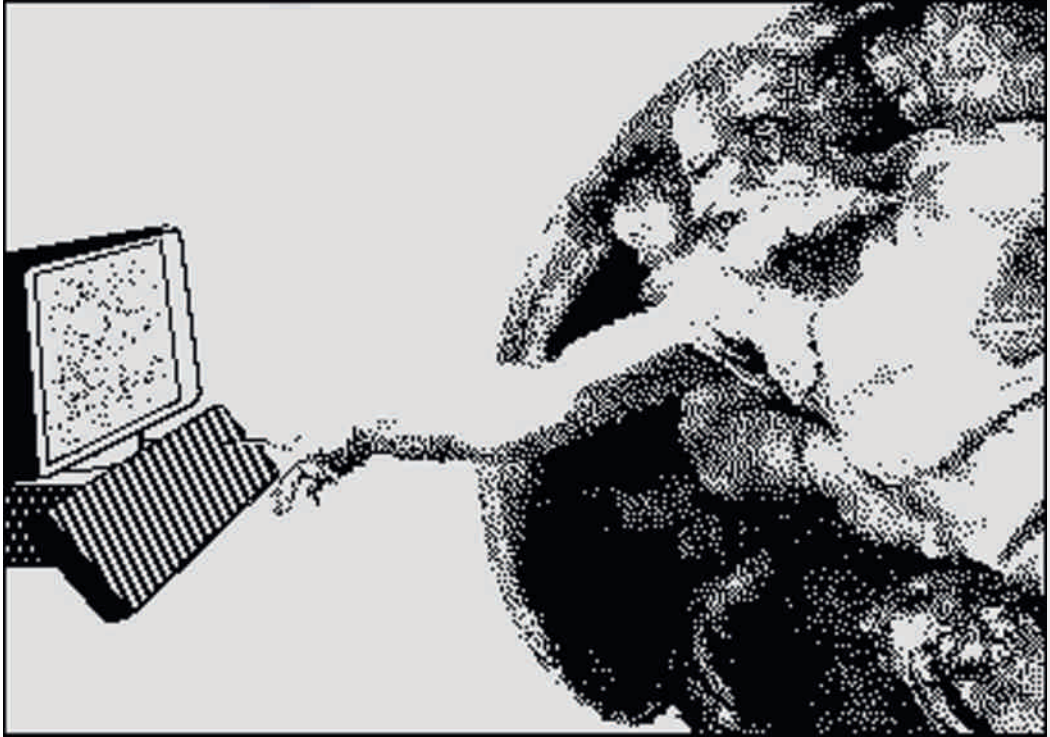
Током историје човечанства је, чак, било опасно по живот доводити у питање постојање Створитеља живота (Бога). Створитељу није само додељена улога креатора живота, него се тврди да Створитељ управља копирањем, модификовањем, комбиновањем и уништавањем својих креација.

За разлику од креационистичке теорије, еволуционистичка теорија није баш јасно и недвосмислено објаснила сам почетак настанка живота (постоје теорије усмерене у том правцу). По еволуцији, даљи развој живота је препуштен стихији по неким јасним правилима.

Правила су:

- Јединка има одређен век трајања
- Јединке се саме размножавају (копирају)
- Полно размножавање (комбиновање) омогућава модификовање оригинала
- Најјаче и најприлагодљивије јединке опстају

Еволуцијска теорија није никад 100%



негирала постојање Створитеља живота (можда под утицајем религије), али га је свела само на Креатора и некога ко је успоставио правила по којима се еволуција даље одвија без његове директне интервенције (једна од теорија).

Софтвер и теорије постанка

Просто је невероватно колико софтвер подсећа на живи организам.

Софтвер има:

- свој животни век
- има своје екосистеме (хардвер)
- размножава се копирањем
- може се комбиновати са другим

софтвером (комбиновање и модификација)

- може бити једноћелијски (у једној датотеци) или вишећелијски (користи више различитих библиотека)
- може бити сложен организам са различитим функцијама (OS са апликативним софтвером)

Смрт софтвера настаје:

- нестанком његовог екосистема (хардвера)
- застаревањем
- бива „поједен” од стране неког већег софтвера
- изумире зато што постоји бољи софтвер за исту функцију



Ово су природни услови за „живот” софтвера. Власнички софтвер има још неколико услова који зависе од његовог власника.

Нисмо сигурни да ли је неко на овај начин размишљао о софтверу. У сваком случају, човек је крајем двадесетог века добио прилику да преко софтвера експериментално провери теорије постанка.

Креационизам

Креационистичку теорију проверава власнички софтвер. Закон о заштити интелектуалних права је створио услове да овај софтвер увек буде у власништву свог створитеља, тако да створитељ власничког софтвера поред креирања има моћ да контролише копирање, модификовање, комбиновање и смрт своје креације.

Власнички софтвер нема толико видљиве карактеристике „живог” организма. Он је само роба са својом тржишном вредношћу која је еквивалентна употребној вредности (сврси).

Карактеристике:

- Сваки власнички софтвер је настао као потреба да задовољи неке потребе његовог створитеља (функционалне, материјалне, духовне, и друге потребе)
- Његов живот је ограничен углавном на период док служи свом власнику
- У току свог живота може бити контролисано копиран, модификован и комбинован
- Понеки може у току свог живота и да

се „одметне” у „дивљину” и помоћу пирата буде копиран независно од свог створитеља

- Живот му престаје кад више не задовољава потребе свог власника
- Ретки срећни власнички софтвери након свог комерцијалног живота буду пуштени у „дивљину” са бесплатном лиценцом да умру „природном” смрћу без могућности да се модификују и да продуже свој живот.

Са друге стране, ако се живот на Земљи посматра кроз призму креационизма, онда може да се постави филозофско питање да ли су и жива бића заиста жива, или се све одвија према неком Божијем плану без сопствене воље Божијих креација. Са тим у вези, онда и вечно питање сврхе постојања човека, добија на важности. Ако се човек одметне и престане да обавља функцију због које је креиран, да ли ће га Бог (Створитељ) укинути као нешто непотребно?

Постоји још један проблем везан за креационистички приступ софтверу. Човек није савршен и не може да игра улогу Бога. Да ли је неко чуо да је Бог укинуо неку своју креацију зато што је креирао неку савршенију, па због тога нема времена да одржава претходну верзију? Или, да је укинуо неку своју креацију зато што се оженио и има преча посла од одржавања своје креације?

Еволуција

У свету софтвера није могуће да он настане спонтано. Слободни софтвер такође има креатора као и власнички софтвер, међутим његов развој више



личи на еволуцију независну од креатора.

Карактеристике:

- Креатор слободног софтвера ствара производ са одређеном функцијом и успоставља правила даљег развоја. Након тога, слободан софтвер се пушта у „дивљину“ (није обавезно, али без пуштања софтвера у дивљину, он нема све карактеристике слободног софтвера)
- На даљи развој слободног софтвера креатор може да има пресудни утицај, али и не мора
- Софтвер у овом случају „живи“ свој слободан живот по принципу – најјачи и најприлагодљивији опстају
- Најјачи пројекти окупљају заједнице који се даље брину о развоју
- Неки еволуирају у потпуно непредвидљивим правцима из угла креатора

Када на овакав начин посматрамо софтвер, он није само слободан за копирање, модификовање и комбинавање, него испада да је суштински слободан по принципу „живе“ јединке.

Овај модел развоја софтвера се показао као врло ефикасан. Иако је настао знатно касније него модел власничког софтвера, и поред свих препрека је по развоју стигао власнички софтвер, а у неким елементима и престагао.

Мане и предности власничког и слободног софтвера

Власнички софтвер је за сада економски јачи. Из ове тврдње произилазе све предности овог модела. Новац овом моделу обезбеђује тимове најбољих, најквалитетнијих креативаца са зацртаним роковима, што омогућава да мањи тимови обављају више посла у краће време. По потреби тај новац прилагођава и екосистеме (читајте



хардвер) по мери креираног софтвера. Новац омогућава бољи маркетинг за овај софтвер и боље, прецизније скенирање потреба корисника софтвера.

Слободни софтвер се са друге стране помало понаша као дивља звер, која уколико није строго под контролом, може да се развија у непредвиђеним правцима. Овакво поређење нисте вероватно нигде могли да чујете. Ако сте дугогодишњи корисник слободног софтвера, сетићете се бар неколико примера да сте налетели на идеалан софтвер за вас, који је временом са својим развојем мало „подивљао” и постао неприступачан за вас.

Слободни софтвер је сада у дивљини и мале су шансе да ће га више ико зауставити у његовом развоју. Док је у почетку требало креирати прве слободне софтвере, сад је веће умеће похватати их у дивљини и „припитомити” их да обављају задате радње. Ипак, не треба паничити – сви најбољи пројекти слободног софтвера окупљају око себе *FLOSS* заједнице, које стриктно воде рачуна о свом пројекту и не дозвољавају да он „подивља”. *FLOSS* заједнице јесу прави кротитељи дивљих звери и воде рачуна не само о томе да он не „подивља”, него и да дугорочно ради посао за своје кориснике на најбољи и предвидив начин. Управо *FLOSS* заједнице су један од кључних фактора успеха слободног софтвера.

Уместо закључка

Шта је поента ове приче? Превасходно упоређење тренутног стања и односа

између власничког и слободног софтвера кроз теорије које су свима јасне или бар мислимо да су нам јасне. Циљ није био да се оцрни једна или друга страна и један концепт постави као једино исправан. Ови концепти ће још дуго коегзистирати без могућности да један коначно победи. Вероватније је да ће у будућности доћи до приближавања ових концепата.

Са растом употребљивости слободног софтвера повећаваће се и интересовање за њега. То ће довести до дела комерцијализације слободног софтвера, али не у смислу да ће се софтвер продавати, него ће се продавати одржавање и подршка јер увек ће бити корисника који ће радије да плате одржавање него да то сами уче. Са друге стране, падаће цене власничког софтвера, што ће вероватно довести до ослобађања дела тог софтвера и препуштања *FLOSS* заједници развој и одржавање мање битних делова власничког софтвера са циљем уштеде и концентрације власничких развојних тимова на, за њих, битне делове. Ово су наше претпоставке на основу већ уочених тенденција. Да ли ће то баш тако бити? Видећемо у будућности.

Ваш посао, *open-source* посао

(1. део)

Аутор: Марко Кажих

Чак и данас, нажалост, појмови „посао” и „*open-source*” у истој реченици изазивају неверицу, чуђење и лавину питања. Ипак, информационе технологије пролазе кроз период фундаменталне промене филозофије и начина на који се нове технологије развијају, конзумирају, рекламирају и продају.

Шта је *open-source* у мору технологије коју користимо свакодневно? Наш *Android* телефон, или можда језгро нашег *iOS*-а? Наш прегледач и у њему *Facebook* ћаскање засновано на *XMPP*-у, *SQLite* у *Skype*-у, *MySQL* у вашем, исто толико *open-source* *WordPress* блогу који покреће *open-source* *PHP*, са *open-source* *Apache*-ом или *nginx*-ом. Причамо наравно о случају у којем се још нисте ослободили власничких технологија као што је *Windows*. Дакле, ово јесте ера *open-source*-а.

Поменути софтвер углавном има једну заједничку особину: То су комерцијални производи засновани на *open-source*-у. Компаније као што

су *Google*, *Amazon*, *Apple*, па чак и *Microsoft*, зависе од *open-source*-а и уграђују га у своје комерцијалне производе. Оно што је далеко битније за њихово пословање, јесте да су њихови целокупни модели пословања засновани око *open-source*-а. Њихова зарада и будућност је у *open-source*-у управо због тога што се *open-source* поставио на трон сувереног владара иновације и покретача целокупне информационе индустрије данашњице (*а и сушрашњице*, прим. ауш.), а то је чињеница која се не сме занемарити.

Open-source сам по себи није и никад неће бити модел пословања. *Open-source* је покрет, филозофија и бољи пут. Како смо већ поменули, комерцијални ентитети постају свесни позиције *open-source*-а у *IT*-у данашњице и сходно томе налазе нове начине да терет прихода пребаце са самог суштинског производа на све секундарне домене производа. То могу бити додатне функције, одржавање, едукација, помоћ, имплементација као и упаривање једног бесплатног и *open-source* производа са



другим комерцијалним. Зато њихови модели пословања данас циљају како пословне кориснике тако и општу популацију. Рецимо, *Google* вам неће продати *Chrome*, али ће зато своје сервисе и рекламе, на којима има највећи профит, лакше и боље серверати крајњем кориснику. Контрола коју *Google* има над начином на који приступа конзументима, јесте контрола коју би свако желео. Управо *open-source* је омогућио *Google*-у да направи солидну платформу на којој ће брже и боље серверати своје комерцијалне производе.

Можда ћете рећи: „Добро, али ми нисмо *Google*. На којим основама може фирма у повоју да изгради профитабилан посао који се базира на *open-source*-у?” Питање је свакако на

месту, а циљ овог чланка и делова који следе, јесте да вас кроз примере упознају са скоро свим моделима пословања на *open-source*-у, да вам покажемо успехе али и неуспехе компанија и њихових пројеката, и надамо се да ћете када серијал приведемо крају, имати јасан одговор на ваше питање. У следећем делу представимо модел пословања зачетника целе ове приче, *Red Hat Inc.*-а – продају професионалних услуга, што је један од најранијих пословних модела заснован на *open-source*-у, затим ћемо упознати *Canonical* и тако дати део одговора на питање које сигурно мучи свакога ко жели да заради, не заборављајући притом *open-source* заједницу.



Tor мрежа:

Online анонимност и сигурност

(3. део)

Аутор: Петар Симиовић

Ако сте просечан корисник Linux-а и желите мало приватности у узбурканим водама интернета, Tor ће урадити и много више од тога за вас. Међутим, ако сте пак мало више неповерљиви и скептични по питању сигурности или параноични по питању прислушкивања, постоје још неки кораци који се могу предузети с циљем побољшања сигурности сваког корисника. Разлога за неповерење увек има, чак и у најсигурнијим системима и добро је да се што чешће контролишу и унапређују.

Сигурно сте у протеклих десетак дана чули или прочитали за сигурносни пропуст у *OpenSSL* протоколу, илустративно названог *Heartbleed*, који је угрозио две трећине сајтова на интернету и потегао многа питања и подигао велику прашину. Ако се питате да ли је *Tor* остао поштеђен у овој сајбер чистки - одговор је негативан. Наиме, потенцијално угрожени корисници су они који користе програме у којима је имплементирана рањива верзија *OpenSSL* протокола тј. верзија 0.1.0, све остале верзије су

сигурне бар што се тиче *Tor* корисника. Такође, *Tor* је почео да ставља на црну листу све излазне чворове који нису ажурирали верзије *Tor* софтвера и *OpenSSL*-а.





Као што знамо, ништа није сто посто сигурно, али то не значи да треба одустати и одрећи се права на приватност. Багова ће увек бити, зато је златно правило увек користити најновије верзије софтвера, по могућству чим се појаве.

Осим основног софтвера за анонимно сурфовање, треба отићи још неколико корака даље и преиспитати сигурност система које свакодневно користимо.

Први такав корак у хијерархијској организацији, логично, био би сам оперативни систем. *OS* је платформа за покретање свих програма и стога мора бити такав да можемо да верујемо да ће он бити сигуран и да неће својим радом угрозити сигурност програма који се ослањају на његов рад. У прошлим деловима смо већ поменули да је основна мера предострожности да се бирају само оперативни системи и програми отвореног кода (*open-source*), а пожељно је да буду и под *GPL* лиценцом. За екстремне случајеве, оперативни системи који пружају максималан ниво безбедности су *Tails*, *Whonix* и један новији *OS* лепог имена *Snowden tribute* (<http://goo.gl/Dvsu1x>) за који је потребно издвојити 4.9€ у виду донације. Нешто више о програмима за бољу *online* анонимност и сигурност за разне платформе можете погледати на интересантном сајту *PRISM* \times *Break* на адреси (<http://goo.gl/z63xPn>).

Следећи корак би био, рецимо *BIOS* рачунара, где препоручујемо корисницима избегавање *UEFI BIOS*-а или његово онемогућавање тамо где је већ присутан. Уколико је могуће,

препоручујемо инсталацију *Coreboot*-а (<http://goo.gl/fjH2t3>) некадашњег *linuxbios*-а, пошто постоји подршка само за веома ограничен број матичних плоча и њихових верзија. Кад смо већ код *BIOS*-а, скренули бисмо пажњу на *firmware* рутер као на још једно место где постоји алтернатива власничком софтверу (*propriety software*, у виду следећих пет *open-source firmware*-а: *OpenWrt* (<http://goo.gl/KKBmm8>), *DD-WRT* (<http://goo.gl/aVCHDQ>), *Tomato Firmware* (<http://goo.gl/Z3ZuFb>), *Gargoyle* (<http://goo.gl/TVa6T1>) и *PAPARoute* (<http://goo.gl/O9y7kk>).



И последњи, али не и најмање важан, јесте сам хардвер. Ту постоје неке алтернативе данас најпознатијим произвођачима корисничког хардвера у виду разних *arduino*, готових хардверских решења по принципу *plug-and-play* попут немачког *enigmaxbox* (<http://goo.gl/a3Lq2P>) и *freedombox* (<http://goo.gl/zd4RAf>), а ту су и *raspberrry Pi* и *Project Novena* као потпуно отворен лаптоп. О последња два је било речи у 21. броју ЛиБРЕ! часописа.

У оквиру *Tor* мреже постоје многи скривени сервиси који чекају да буду откривени. У њиховом откривању нам могу помоћи специјализовани претраживачи попут већ поменутог *TorSearch*-а (<http://goo.gl/FnNe4B>), а ове недеље се појавио и најновији под називом *Grams* (<http://goo.gl/68hKa7>) и почео да индексира и оне најопасније сајтове по

којима је ова мрежа и позната. Овакви специјализовани претраживачи умногоме помажу *Tor* корисницима јер су сајтови често принуђени да мењају своје адресе и локације што отежава корисницима њихово проналажење. Овде ћемо скренути пажњу на неколико програма који нису онако светлећи и шљаштећи као већина оних на који смо навикли уживајући у удобношћу интернета и повољностима нових технологија.

Напоменули бисмо да су ово само неки, егзотични *commandline* програмчићи за ентузијасте *geek*-ове и оне најугроженије. За проверу интернет поште имамо *mutt* (лепо кратко упутство имате на: <http://goo.gl/b7Qxfy>), за слушање музике пројекат са *github*-а *cmus* (<http://goo.gl/UNw0We>), за твитовање *ttytter* (<http://goo.gl/rbUoRf>), за *IM* имамо *Pidgin*-ов *Finch* (<http://goo.gl/bWbFuI>), за *IRC chat* ту је *Irssi* (<http://goo.gl/aE5AKR>) и за сурфовање нетом ту су *Lynx*, *Links* и *ELinks* (<http://goo.gl/W1u0cP>).

Иако ово нису најудобнији програми за просечног корисника, могу у екстремним случајевима (ако мислите да сте из неког разлога под сталном присмотром, или ако сте тајни дописник) да вашим

непријатељима знатно отежају посао. Ту су и они стандардни програми и апликације за које није лоше знати, попут алтернативних претраживача интернета који су препоручљиви и ван *Tor* мреже за свакодневну употребу, а у духу избегавања глобалног надзора *Google*-а: *ixquick* (<http://goo.gl/OgdtDX>), *Startpage* (<http://goo.gl/BS1RLZ>) или све популарнијег *DuckDuckGo* (<http://goo.gl/FUIcPI>). Поред претраживача за *online chat*, ту је *Cryptocat* (<http://goo.gl/qdJ4wq>) и за отпремање видеа без регистрације ту је *vidd* (<http://goo.gl/5oPf2K>), а за инстант дељење мултимедије уз *chat* ту је *Volafile* (<http://goo.gl/UuRpdK>), нови сервис попут старог доброг *4chan*-а. У оквиру *Tor*-а, навели бисмо само пројекте са *github*-а као *TorBirdy* и *TorChat*.

Споменућемо овде три платформе за дописнике медијима, рецимо *WikiLeaks*-у, који су попут *Snowdena* сигурно међу најугроженијима, што се приватности и живота тиче. *Strongbox* од *The New Yorker*-а (<http://goo.gl/03ziZp>), *SecureDrop* организације *Pressfreedomfoundation* (<http://goo.gl/10CTna>) могу да се обезбеде путем *RISEUP* организације која пружа *VPN*, *email* и *chat* услуге другим групама људима и организацијама које





се боре за социјалне слободе слободе говора на интернету, а све се додатно рутира кроз *Tor* (<http://goo.gl/mnRoki>).

Додаћемо још да постоји и упутство за инсталацију сопственог излазног чвора уз помоћ *Raspberry Pi*-а на сигуран начин, а да се не упадне у невоље са евентуалним протоком недозвољеног садржаја кроз исти, али га аутор овог текста још није испробао јер је помало скептичан (прим.аут.), а ви како вам воља али пажљиво: (<http://goo.gl/DGzqU6>).

Није тешко увидети да су најкоришћеније и најраспрострањеније



апликације и програми, а у последње време и сам хардвер (<http://goo.gl/ABEuIu>) потенцијално најопаснији јер ономе ко пронађе рањивост у њима и злоупотреби је, доноси велику моћ и утицај на највећи део интернет популације. Многе организације се труде баш то, да пронађу рањивост и злоупотребе је. Међутим, има једна светла тачка, а то је да се свест о угроженој приватности, безбедности и бескрупулозној софтверској и хардверској шпијунажи све више шири у

јавности, а са њом и слободне алтернативе власничком софтверу, а сада и хардверу. Остаје да видимо какву легуру све могу да направе и шта ће нам донети у скорој будућности.





_.underscore

Аутор: Златан Васовић

Underscore је JavaScript библиотекa која садржи велики број корисних алатки, али без проширивања основе програмског језика. То је библиотекa која тежи да буде минималистична и брза, како не би (превише) успорила ваш сајт.

Underscore садржи и имитацију нативних JavaScript метода, уколико оне не постоје. Ипак, те имитације нису проширење основе програмског језика, већ обичне функције библиотеке.

Статистика

Underscore је најпопуларнији пакет на *npm* регистру. Он је веома популаран и на другим регистрима JavaScript пакета.

- 5400+ *npm* пакета који користе Underscore
- 70+ star-ова на *npm* регистру
- 10000+ star-ова на GitHub-у
- 2000+ fork-ова на GitHub-у

Инсталација

Underscore ради и у прегледачима и у *node.js* окружењу, па га можете користити на више начина. Можете га преузети директно са сајта, или додати неким од JavaScript менаџера пакета (нпр. *Require.js*, *npm* и *Bower*).

node.js (*npm*)

```
npm install underscore
```

Require.js

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

Bower

```
bower install underscore
```

Component

```
component install  
jashkenas/underscore
```



API

Underscore се најчешће користи са променљивом `_` (доња црта). Тада је синтакса `_.метода(аргументи)`.

Најчешће коришћене методе

- `_.each` (`_.forEach`) — имитација нативне цикличне петље за низове и објекте
- `_.map` — имитација нативног трансформатора чланова низа или објекта
- `_.invoke` — позива методу и примењује је на низ или објекат
- `_.reduce` (`_.inject`, `_.foldl`) — функцијом спаја све вредности низа у једну
- `_.filter` (`_.select`) — имитација нативног филтрирања низа
- `_.every` — имитација нативне провере да ли сви чланови низа пролазе тест
- `_.some` — имитација нативне провере да ли било који члан низа пролази тест
- `_.indexOf` — имитација нативне методе за проналажење места члана низа

Underscore-contrib

За све оне којима нешто недостаје у *Underscore-y*, ту је *Underscore-contrib*, проширење *Underscore-a*. Ту се налазе све функције које би или превише увећале библиотеку, или би превише успориле читавање странице. Више информација можете наћи на

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

Алтернативе

Постоји више алтернатива *Underscore* библиотеци. Најпопуларније су *Lo-Dash*

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

Прелазак на алтернативе је вредан, ако бринете о перформансама.

Линкови

- [1] Сајт: <http://underscorejs.org/>
- [2] Изворни код: <https://github.com/jashkenas/underscore>
- [3] Објашњење изворног кода: <http://underscorejs.org/docs/underscore.html>



JavaScript
UNDERScore.js

ЛИБРЕ!

Часопис о слободном софтверу



2 године
са Вама.

Хвала што нам верујете!