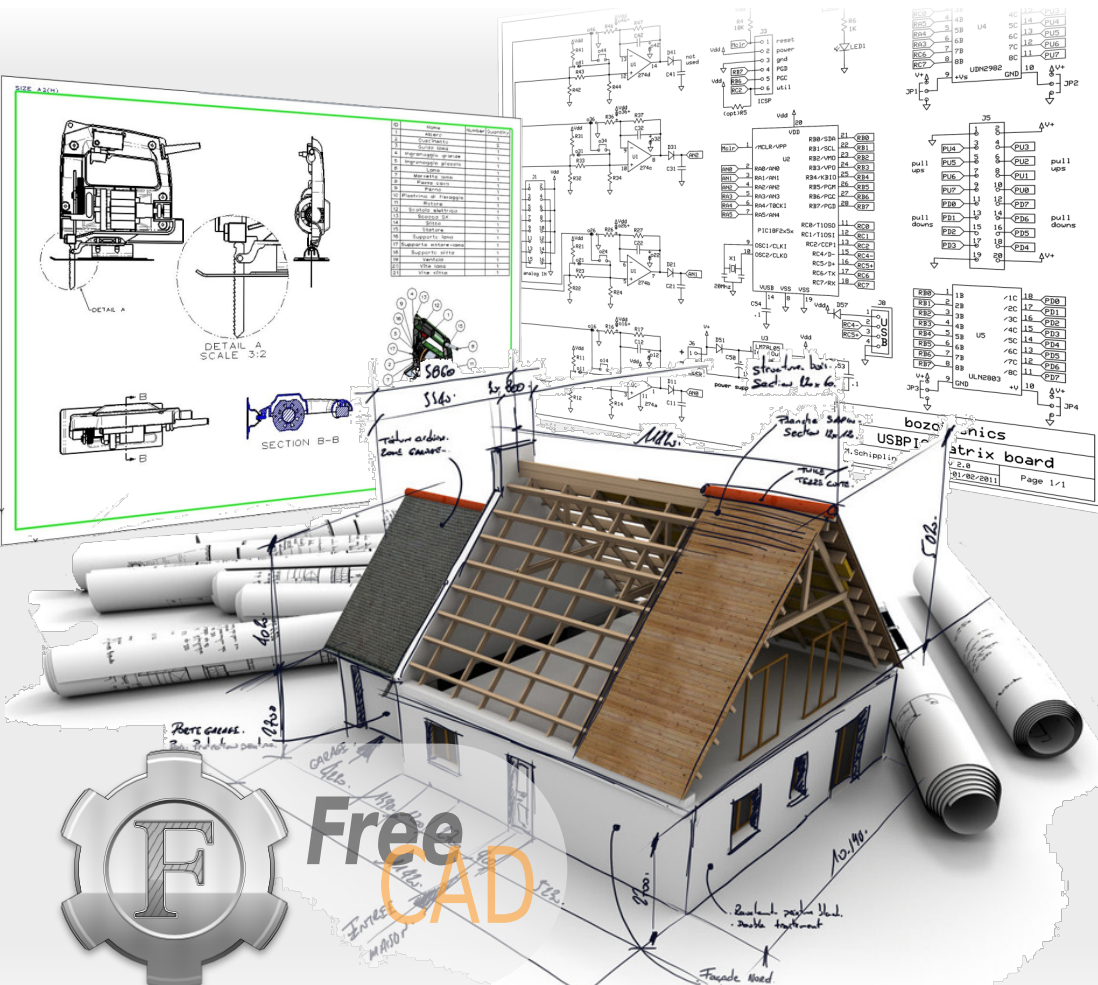




ЛИБРЕ!

Часопис о слободном софтверу



Free CAD

ЈОШ ИЗДВАЈАМО

Ансибл
Гугл - Савршена рекламна агенција



Реч уредника

Где је данас *FLOSS* у свету?

Све више истраживања у свету показују да софтвер отвореног кода преузима доминацију у свим областима. Недавно су компаније Норт Бриц и Блек Дак спровеле истраживање о томе колико се слободни софтвер користи у пословном окружењу (фирмама и компанијама). Донедавно је пословно окружење било неприкосновено царство власничког софтвера. Најновије истраживање сада показује да чак 78% фирми и компанија у свету користи делимично или у потпуности слободан софтвер у свом пословању. Истраживање је показало да је још увек у овом сегменту низак ниво „опен-сорс писмености“, али да се стање знатно поправља у односу на раније године.

Чињеница је да је власнички софтвер старији и да прати развој стоних (десктоп) персоналних рачунара већ више од тридесет пет година. Слободан софтвер је знатно каснио у освајању персоналних рачунара. Тек у последњих десет година са развојем интернета и развој слободног софтвера је приметан и снажан. Захваљујући бржем и снажнијем развоју, приближава се квалитету власничког софтвера и на персоналним рачунарима. Области у којима је слободни софтвер данас супериоран у односу на власнички су технологије које су се развијале упоредо са слободним софтвером: интернет, мобилна телефонија, мобилни рачунари и тако даље.

У овом броју часописа вам представљамо Фри-кед (енг. *FreeCAD*). Овај софтвер је најбољи пример брзог и приметног напретка слободног софтвера у свим областима коришћења рачунара. Пре три године у часопису смо први пут писали о слободним *CAD* програмима. Тада смо констатовали да је стање у тој области јако лоше и да ти програми могу



да служе само за скице. Данас представљамо већ озбиљан пројекат који још има доста својих мана, али у односу на такве програме од пре само три године ово је огроман напредак.

Истраживање које смо поменули каже да је још у много фирми обавештеност о слободном софтверу на ниском нивоу. Истовремено, неке фирме су изградиле већ империју на слободном софтверу. Кад то кажемо, првенствено мислимо на најјачу – Гугл.

Просто је невероватно како је човек у стању од сваког изума да направи оружје. Иако је новац јак мотиватор за рад, у западној култури, пре или касније, онај који заради много новца пожели да освоји свет. О томе како Гугл тумачи реч „Free” из акронима ФЛОСС и шта се десило са „Л” из истог акронима, можете такође да прочитате у овом броју.

Где су Србија и регион у овој области? Као и у свему, каснимо десет година. Све најновије технологије користимо као и сви у свету. Проблем највише лежи у томе што је већина корисника потпуно необавештена и „опен-сорс неписмена”. Због тог незнања све плаћамо више него што је неопходно и угрожена нам је приватност, а да тога нисмо свесни. ЛиБРЕ! зато постоји да би се променила ова статистика и код нас. Крајње време је да се описменимо и у слободном софтверу како не бисмо плаћали данак незнању, и то не само међу „компјутерским стручњацима”, него и међу обичним корисницима. Слободни софтвер већ одавно није „играчка” за штребере, већ технологија која нас окружује. Укључите се и ви у ово „опен-сорс описмењавање” и пишите нам на нашу већ познату адресу електронске поште [libre \[et\] lugons \[dot\] org](mailto:libre[et]lugons[dot]org).

До читања,

ЛиБРЕ! тим

Садржај

Вести

стр. 6

Представљамо

FreeCAD

стр. 9

Како да...?

Увод у програмски језик C (11. део)

Vagrant (2. део)

стр. 17

стр. 20

Ослобађање

Развој слободног софтвера

стр. 24

Слободни професионалац

Ваш посао, *open-source* посао -

Гугл - Савтшена рекламна агенција

Ansible (1. део)

стр. 29

стр. 34

Интернет мреже и комуникације

Шифровани чет (2. део) - *Nitalk*

стр. 38

Сам свој мајстор

Bootstrap - Први кораци

стр. 41

Хардвер

BagleBone Black Rev C: Водич од првог дана (6. део)

Биглбон Блек као Тор егзит

стр. 45

Моћ слободног
софтвера





ЛИБРЕ! пријатељи



REGIONALNI
LINUX PORTAL

linuxzasve.com



LOVČENAC
LINUX USER GROUP



Grupa korisnika GNU/Linux operativnih sistema u Lovčencu

info i tutorijali na srpskom
lubunturs.wordpress.com



Број: 35

Периодика излажења: месечник

Извршни уредник: Стефан Ножинић

Главни лектор:

Адмир Халилкановић

Лектура:

Јелена Мунџан Сашка Спишјак

Милена Беран Милана Војновић

Александар Божиновић

Графичка обрада:

Дејан Маглов Иван Радељић

Дизајн: White Circle Creative Team

Аутори у овом броју:

Ненад Марјановић

Златан Васовић

Никола Харди

Криптопанк

Иван Радељић

Остали сарадници у овом броју:

Марко Новаковић Михајло Богдановић

Почасни чланови редакције:

Жељко Попивода Жељко Шарић

Владимир Попадић

Александар Станисављевић

Контакт:

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

Е-пошта: libre@lugons.org

Вести

19. април 2015.

Педесет година Муровог закона

Пре тачно педесет година је публикован Муров закон, који предвиђа да се број компоненти на чипу дуплира током времена док цена и површина остају исти. Ово чини да број компоненти у чипу расте експоненцијално.

Користан линк: <http://j.mp/1EjJTNg>

28. март 2015.

Европска комисија ће повећати употребу слободног софтвера

Европска комисија је одлучила да ажурира своју стратегију о интерној употреби софтвера, која предвиђа употребу слободних софтверских система као и додатни развој софтверских решења отвореног кода.

Користан линк: <http://j.mp/1EwagHt>

1. април 2015.

Објављен Фајерфокс 37

Мозила је објавила нову верзију свог веб-претраживача отвореног кода.

Користан линк: <http://j.mp/1Pq3Jj5>



3. април 2015.

Гитхаб је био изложен DDoS нападу

Сумња се да је за овај напад одговорна кинеска влада, што се наслуђује из чињенице да су напади стизали из Кине са званичних сервера кинеске владе.

Користан линк: <http://j.mp/1DolKH4>



3. април 2015.

Rajesh De о Сноудену

Генерални саветник америчке Државне безбедносне агенције (НСА) објашњава да је јако тешко за ову америчку агенцију да остане без критика у медијима и да не може да верује да Сноуден, који је седео са њима и бринуо се о сигурности грађана САД-а, могао да уради тако нешто као што је објављивање тајних података ове агенције. Он такође каже да треба наћи баланс између сигурности и приватности грађана.

Користан линк: <http://j.mp/1OLzMI0>



6. април 2015.

ДОРС и КХКЛ од 18. до 20. маја

Дани отворених рачунарских суштава (ДОРС) и Конференција Хрватских корисника Линукса (КХКЛ) одржаће се у Загребу на Факултету електротехнике и рачунарства од 18. до 20. маја.

Користан линк: <http://j.mp/1QpawI0>



Вести

9. април 2015.

Линукс фондација хостује пројекат *Open Encryption*

Ова непрофитна организација ће хостовати пројекат који ће омогућити веб-страницама да добију сигурносне сертификате у року од неколико минута, чиме ће веб учинити глобално безбеднијим.



Користан линк: <http://j.mp/1Ewaw9l>

11. април 2015.

Подршка за енкрипцију за *EXT4* систем датотека

У Гуглу је развијена подршка за енкрипцију у склопу *EXT4* система података. Биће уграђена у нову верзију Андроида М и биће послата главном развојном тиму да се дода у нови Линукс кернел.



Користан линк: <http://j.mp/1FhPJ8j>

22. април 2015.

GCC 5.1 објављен

Објављена је нова верзија Гнуовог компајлера GCC.



Користан линк: <http://j.mp/1Domr38>



FreeCAD



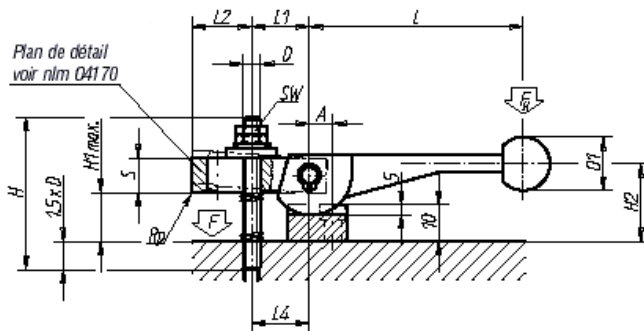
Аутор: Никола Харди

Одрицање од одговорности

Аутор се не бави професионално техником или компјутерским конструисањем. Овај текст је настао из жеље аутора да представи програм који му је помогао да реши проблем пред којим се нашао. Ово неће бити професионални опис програма, али се надамо да ће бити довољно добар опис програма који може многим да буде од користи.

CAD/CAM софтвер

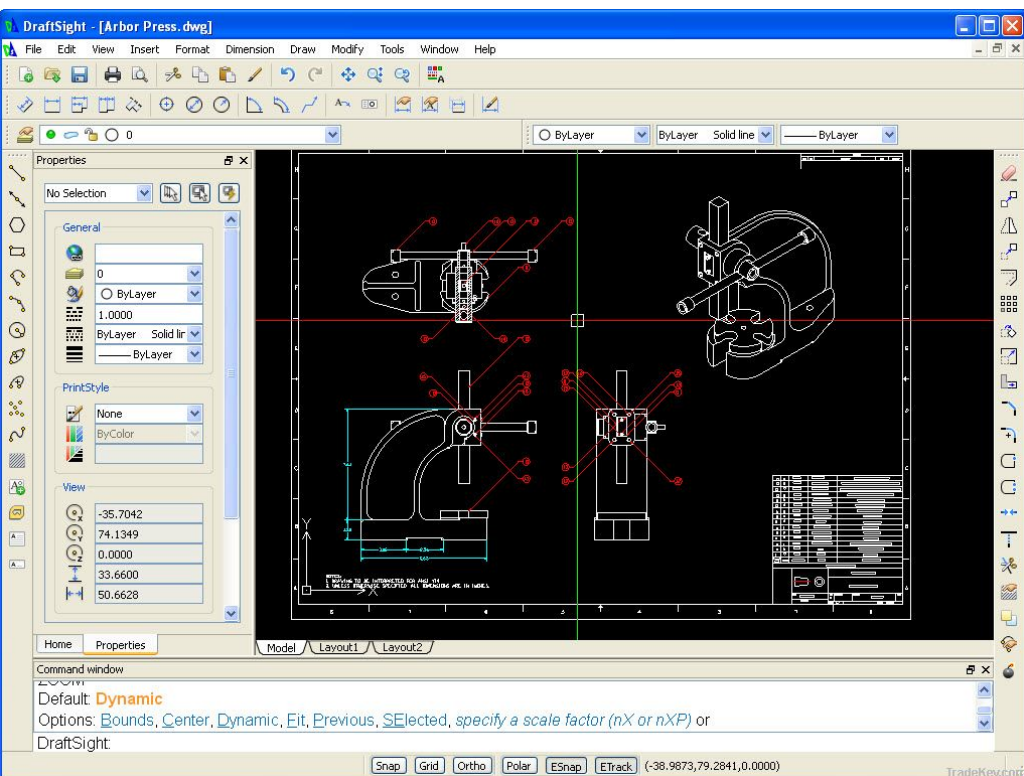
CAD/CAM (енг. *computer-aided design / computer-aided manufacturing*) системи - у преводу: компјутерски системи који помажу при креирању, модификовању, анализи и оптимизацији дизајна са једне стране, и управљању нумерички управљаних машина алатки у производњи, са друге стране. Лаичким речником речено, у овом тексту је представљен софтвер за техничко цртање. Да подсетимо, техничко цртање је прецизно цртање са јасним правилима која свима у ланцу производње омогућавају читљивост цртежа.



Представљамо

CAD/CAM софтвер је, упроштено, софтвер за 2Д и 3Д векторску графику, са нагласком на прецизно цртање алатима за брзо и лако додавање техничких спецификација.

Овакав тип програма је болна тачка у свету слободног софтвера јер, како кажу људи из струке, не постоји алат одговарајућег квалитета за професионалну употребу. Постоје програмска решења која нису отвореног кода, али могу бити покренута на слободним платформама као што су Брикскед (*BricsCAD*) и Драфт-Сајт (*DraftSight*). Нека професионална решења су постала отворена и слободна, као BRL-CAD. О једном потпуно слободном CAD софтверу смо већ писали у овом часопису (*LibreCAD*, ЛиБРЕ! број 1).





Можда је највећи проблем у томе што је Аутодесков Аутокед (*Autodesk AutoCAD*) постао стандард за професионалну употребу, а он је заштитио свој **DWG** формат. Без тог формата, нема комуникације са неким другим *CAD* програмима, па према томе ни сарадње са колегама који користе само Аутокед.

У овом тексту представимо још један слободни *CAD* софтвер - Фрикед (*FreeCAD*).

2Д CAD

Један приступ је цртање крајњих погледа на елемент који се дизајнира. Ово подразумева цртање 2Д пројекција, укључујући финално означавање димензија, шрафирање, попуњавање заглавља итд. Треба направити разлику између *CAD* програма и других програма за 2Д векторску графику, као што су Либреофис дроу (*Libreoffice Draw*) и Инкскејп (*Inkscape*). Иако је помоћу оба ова програма могуће направити цртеж налик на техничку документацију, они имају своје недостатке у односу на *CAD* програме. Пре свега, немају потребну прецизност, уношење измена, постављање димензија и осталих техничких спецификација захтева много више труда, што смањује продуктивност.

3Д моделовање

Ознака „3Д“ се често среће у свету софтвера. Постоје 3Д игре, 3Д филмови, алати за 3Д моделовање и 3Д *CAD/CAM* алати. Пример програма за 3Д моделовање је Блендер. Проблем је што је и у овом програму акценат на изгледу, а не на димензијама. Могуће је нацртати објекат који у простору изгледа како треба, међутим проблем настаје код задавања димензија, означавања димензија и правилног исцртавања пројекција и пресека.

Објекти нацртани оваквим алатом немају запремину, већ имају само омотач. Програми као што је Фрикед, или неки комерцијални, као што су Аутокед, Инвентор, Катиа (*Catia*) или Про инжењер (*PRO/Engineer*) - омогућују кориснику да нацрта 3Д модел елемента који ће заиста имати и запремину, одговарајуће димензије, параметре и моћи ће прописно да се представе пројекцијама у виду техничких цртежа. Све у свему, прави *CAD* софтвери инжењерима пружају више информација о нацртаном телу од обичних 3Д дизајнерских софтвера и такође помажу у даљој обради тих информација.

Представљамо

Пример употребе

Фрикед има донекле стандардни начин пројектовања. Потребно је нацртати скицу елемента. Затим се од скице прави 3Д објекат извлачењем или ротацијом. Поступак се понавља уз примењивање додатних алата као што је обрада ивица, бушење итд. Напоследку, елемент се пројектује у цртеж и врши се његово котирање.

Окружење програма

Фрикед се састоји од више радних окружења (енг. *workbench*). Нека од тих окружења су: почетно окружење, окружење за рад са деловима (енг. *part*), окружење за дизајн делова (енг. *part design*), рад са цртежима (енг. *drawing*) итд. Свако радно окружење има другачију намену и различит сет алата. Први корак је наравно креирање новог пројекта. У левом делу прозора постоји део за приказ структуре тренутног пројекта (енг. *model*) и део за приказ задатака доступних у тренутном окружењу (енг. *tasks*).

Цртање скице

У окружењу за дизајн делова, у делу за задатке, могуће је додати нову скицу у жељеној равни. Цртање скице је први корак у дизајну новог дела. Скица може бити пресек дела или неки од три основна погледа на део. На располагању су алати за цртање геометријских примитива: линија, кругова, лукова, правоугаоника итд. Скице треба цртати слободно, јер је касније свакако неопходно додати ограничења (енг. *constraints*). Ограничења постављају односе између елемената скице. То може да буде угао између две линије, растојање између тачака или дужина линије. Линије могу да буду међусобно паралелне. Тачке могу да буду преклопљене. Са леве стране постоји контрола да ли је скица добро дефинисана или упозорење да нека ограничења не могу да буду испуњена (енг. *solver messages*). Практичан начин којим може да се провери стање скице је насумично повлачење делова скице. Циљ је да скица никада не изгуби свој облик. По завршетку цртања скице може се напустити цртање скице (енг. *exit the editing of the sketch*).

Извлачење елемента

Цртање скице је само почетни корак у дизајну једног дела. Од цртежа у равни

Представљамо

потребно је направити тродимензионална тела. За прелазак из равни у простор могу да послуже алати за истискивање или извлачење скице (енг. *extrude*) или за креирање тела ротирањем скице око неке осе. Осим ових алата могуће је искористити алат за заобљавање ивица или направити удубљење.

Додавање цртежа

Када је део попримио жељени облик, може се прећи на прављење цртежа. Потребно је прећи у окружење за рад са цртежима и у траци са алатима додати нови лист жељеног формата. Следећи корак је додавање дела на цртеж. У левом делу екрана, у приказу структуре пројекта, треба означити жељени део, а у траци са алатима изабрати акцију додавања погледа или ортографске пројекције на цртеж. Уколико у пројекту постоји више листова, при додавању новог погледа је потребно у структури пројекта означити и део и лист у који поглед треба да буде додат.

Додатак за димензионисање цртежа

Један велики недостатак програма је котирање цртежа. Добра ствар је што постоји сјајан додатак за ову намену. Реч је о додатку који додаје окружење за котирање цртежа (енг. *drawing dimensioning*). Упутство за инсталацију и код додатка је могуће преузети са следеће адресе: https://github.com/hamish2014/FreeCAD_drawing_dimensioning. Када је поглед додат на цртеж, преласком у окружење за котирање појављују се алати за котирање. Најављено је да ће овај додатак бити укључен у наредне верзије програма. Иако је основа пројекта написана у програмском језику C++ уз помоћ Qt библиотека, додаци могу да буду написани у Пајтону (*Python*). Пајтон конзола је увек доступна у једном делу прозора и омогућава аутоматизацију многих задатака и истраживање кода програма „наживо“.

Превод програма

Доступан је превод на хрватском језику, што је углавном прихватљиво за наше подручје. Превод на српски је такође започет - преведена је приближно половина садржаја. Међутим, превод из неког разлога није доступан за избор у програму. Уколико постоји довољан број заинтересованих да се овај превод доврши, наш часопис ће радо подржати такву акцију.



Неколико коментара за крај

Фрикед је далеко од савршеног или професионалног решења, али ипак је употребљив. За крај остављамо неколико коментара, добрих и лоших. Будите слободни да поделите са нама и осталим читаоцима ваше цртеже или искуства у раду.

Рад са цртежима

Рад са цртежима је помало неинтуитиван. Највећа замерка одлази на померање елемената на цртежу. Није могуће померање елемената на цртежу курсором миша, већ се позиција елемената мора поставити у прозору за особине тог елемента. Ово може да буде незгодно и заморно. Надамо се да ће овај проблем ускоро бити решен.

Други проблем у окружењу за рад са цртежима је попуњавање заглавља. Слично као и са позицијама, садржај заглавља цртежа може бити измењен у особинама цртежа, не и кликом на текст у приказу самог цртежа. Није страхан проблем али засигурно би клик на текст у заглављу био практичнији.

Стабилност

Иако на сајту програма постоји изричито упозорење да је пројекат у раној фази развоја и да су могући извесни проблеми у реду, такви проблеми нису примећени. И поред нашег доброг искуства, и ми ћемо оставити упозорење да будете пажљиви и редовно чувате измене у вашим пројектима.

Брзо се развија

Пројекат постоји више од 10 година, а у протеклих неколико година приметан је велики напредак у развоју. На пример, у Убунтуу 14.04 је доступна верзија 0.13 која врло оскудева са могућностима у односу на тренутну најновију верзију 0.15. Добра вест је да постоје пакети са новијим верзијама, а за Убунту је доступан и ППА репозиторијум са новим стабилним верзијама и дневним (енг. *daily*) верзијама.

Представљамо

Стање документације

Званична документација није баш сјајна. Међутим, на интернету су доступна многа упутства и савети корисника. Посоји много видео примера на Јутјубу (*YouTube*). Ипак, позивамо вас да допринесете квалитету документације преводом или писањем нових упутстава.

Неистражене могућности

Фрикед нуди много могућности које нису ни поменуте у овом тексту. На пример, доступна су окружења за архитектуру, бродоградњу, рад са склоповима, симулације, пројектовање робота итд. Доступни су многи додаци који још увек нису доспели у стандардно издање. Све у свему, постоји прегршт могућности за истраживање.



Преглед популарности ГНУ/Линукс и БСД дистрибуција за месец април

Distrowatch

1	Mint	3196>
2	Ubuntu	1944>
3	Debian	1533>
4	openSUSE	1241<
5	CentOS	1210<
6	elementary	1174<
7	Fedora	1173>
8	Mageia	921<
9	LXLE	879>
10	Arch	856>
11	Manjaro	855<
12	Android x86	821<
13	Lite	711=
14	MakuluLinux	621>
15	PCLinuxOS	618>
16	Puppy	607>
17	Lubuntu	591>
18	Kali	539<
19	Antergos	539>
20	KaOS	514>
21	Simplicity	512>
22	Zorin	495>
23	Bella	495>
24	SparkyLinux	492>
25	Xubuntu	469>

Пад <
 Пораст >
 Исти рејтинг =
 (Коришћени подаци са Дистровоча)



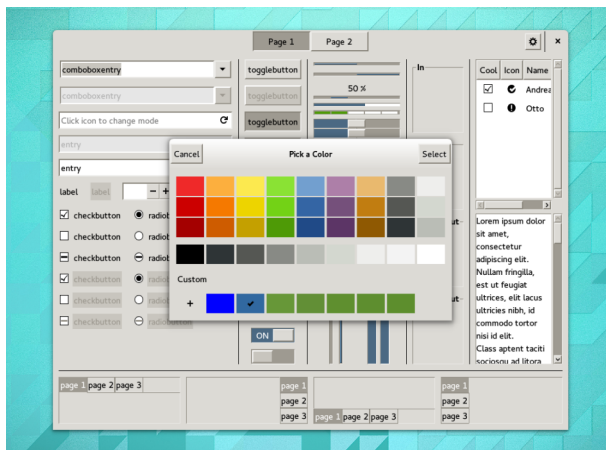
Увод у програмски језик С (11. део)

Аутор: Никола Харди

Шта углавном желимо да направимо?

Почетници у програмирању углавном имају идеју за програм коју желе да реализују па из те идеје проистекне жеља за учењем неког програмског језика. Често је реч о некој апликацији са графичким интерфејсом, о системском програмирању и раду са хардвером или са неком занимљивом математиком. У овом тексту биће представљени примери таквих апликација и биће дате смернице о томе како је такве апликације могуће направити.

Класична *GUI* апликација



Ово је класичан тип десктоп апликација са дугмићима (енг. *buttons*), падајућим листама (енг. *drop down, combo box*), менијима (енг. *menu*) и сличним контролним елементима. Иако је објектно оријентисани приступ данас најзаступљенији у развоју оваквих апликација, постоје и библиотеке за језик С као што је *GTK+*.

Како да...?

Рад са базом података



У претходном делу серијала је споменуто да је датотеке погодно заменити базама података. Рад са базама подразумева много ствари, попут успостављања везе са сервером. Занимљив приступ је и рад са *SQLite* (Ес-кју-лајт) базом која је сачувана у једној датотеци. Цео систем за управљање базом је енкапсулиран у једну датотеку и једну библиотеку - *SQLite* управљачки програм (драјвер).

Комуникација са удаљеним рачунарима

Међурачунарске мреже могу бити врло сложене. Комуникација између два рачунара може да се врши на разним нивоима и разним протоколима. Међутим, корисницима (и програмерима) углавном је довољно да пренесу неки низ бајтова из једног програма у други. Ово се може обавити помоћу транспортног слоја, механизмом „мрежних утичница“ (енг. *network sockets*) и протокола *TCP* или *UDP*. Потребно је свега десетак линија *C* кода да би се успоставила веза између два програма путем мреже, а потом се пренос података састоји од припреме података у форми поруке (string и меморијски блок), слања тих података и евентуалног чекања на одговор. Кључне речи за претрагу на интернету су „*BSD sockets*“ или „*Berkleu sockets*“.

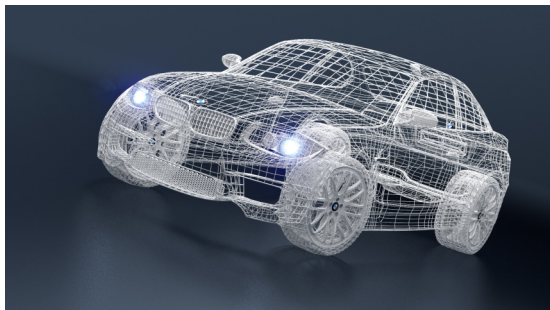
Напредна конзолна апликација



Конзолне апликације могу да имају сложенији интерфејс - могло би се рећи чак и графички интерфејс. Реч је о библиотеци *Ncurses*, која има веома дугу историју. Помоћу ове библиотеке могуће је изградити заиста занимљиве конзолне апликације.

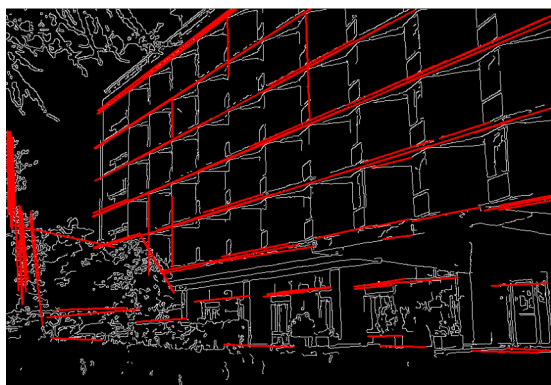


Рад са графиком



рафиком (рецимо, 3Д игром) постоје потпуни покретачи (енг. *game engine*) као што је *OGRE 3D (Object-Oriented Graphics Rendering Engine)*.

Рад са графиком такође може да подразумева много тога. Пре свега, прича се дели на 2Д и 3Д графику. За рад са 2Д графиком може послужити *SDL (Ес-ди-ел)*, а за 3Д *OpenGL (Опен-џи-ел)*. Постоје и друге библиотеке са сличним функционалностима. За напреднији рад са 3Д



пројеката били би препознавање лица, препознавање објеката и аутоматска корекција фотографија. За овакве задатке од непроцењиве важности је библиотека *OpenCV (Опен-си-ви)*.

Компјутерска визија

Иако и ови пројекти на неки начин спадају у рад са графиком, заслужују посебан одељак. Реч је научно-истраживачким пројектима који настају зарад истраживања у области машинског учења и вештачке интелигенције у домену компјутерске визије. Рецимо, примери оваквих

Learn C Programming

Како да...?

VAGRANT

(2. део)

Аутори: Иван Радељић и Стефан Ножинић

У прошлом броју писали смо о предностима коришћења Вагранта при подешавању развојног окружења. Истакли смо зашто сваки од чланова тима има користи. Сада ћемо вам показати како да инсталирате Вагрант и како да користите Вагрант команде.

Инсталација

Најпре је потребно да посетите страницу <http://www.vagrantup.com/downloads> и да преузмете одговарајући пакет за ваш оперативни систем и архитектуру. Инсталер ће аутоматски додати *vagrant* команду у ваш системски *PATH*, тако да ће команда бити доступна глобално у терминалу. По завршетку инсталације у терминалу напишите команду:

```
vagrant -v
```

и програм ће вам исписати верзију која је инсталирана.

Вагрант се такође налази у ризницама пакета многих дистрибуција Линукса, па тиме можете Вагрант инсталирати као и сваки други пакет који инсталирате за вашу дистрибуцију. На пример, ако користите Убунту, процедура је следећа - отворите терминал и Вагрант инсталирате управником пакета на следећи начин:

```
sudo apt-get install vagrant
```



Ако имате било каквих проблема током инсталације, можете нас контактирати на нашу адресу електронске поште *libre [at] lugons [dot] org*.

Иницијализација

Да бисте успешно интегрисали Вагрант у ваш пројекат, потребно је претходно инсталирати Виртуал бокс (*VirtualBox*) или неки други провајдер. Потом треба пронаћи и инсталирати Вагрант кутије (*Vagrant boxes*). У зависности од пројекта, бирамо одговарајућу кутију. На интернет адреси <https://atlas.hashicorp.com/boxes/search> можемо вршити претрагу кутија. За додавање нове кутије користимо:

```
vagrant box add USER/BOX
```

Сада је све спремно за иницијализацију пројекта.

```
vagrant init USER/BOX
```

Ова команда ће креирати Вагрант конфигурациону датотеку (*vagrantfile*) у почетном директоријуму пројекта. Ова датотека је све што нам је потребно да успешно користимо Вагрант даље у пројекту. Наравно, *vagrantfile* датотека садржи све потребне конфигурације и, када желимо да прикључимо новог члана постојећем тиму, довољно је да му проследимо *vagrantfile* датотеку. Ово можемо урадити тако што ћемо ту датотеку поделити на Гит репозиторијуму или неком систему за тимски развој софтвера.

Up and running

У овом стадијуму Вагрант окружење је спремно за покретање. Користимо команду:

```
vagrant up
```

и наше виртуелно развојно окружење ради. Од овог момента све промене у директоријуму Вагрант пројекта на хост машини ће се аутоматски одразити на наше виртуелно окружење. Синхронизација на виртуелној машини се извршава у директоријуму */vagrant*. Ово можемо проверити тако што ћемо приступити виртуелној машини преко *SSH* протокола. Команда за то је:

Како да...?

```
vagrant ssh
```

Након ове команде и приступања виртуелном окружењу, можемо користити све нама добро познате Линукс терминалне команде. По завршетку рада користићемо команду:

```
vagrant halt
```

да бисмо искључили виртуелну машину и сачували све промене. Постоји и команда:

```
vagrant destroy
```

која брише цео пројекат и све зависности које смо инсталирали, тако да ову команду користите тек онда када сте сигурни да је ваш рад на пројекту завршен.

Конфигурација

Као што је већ напоменуто, конфигурација Вагрант машине за конкретан пројекат се обавља у *vagrantfile* конфигурационој датотеци. Ова датотека је Руби (*Ruby*) скрипта која пружа конкретне информације Вагранту о томе која се кутија користи, који се систем аутоматске конфигурације користи (енг. *provisioning*) и слична подешавања.

Ова датотека треба да се постави на систем контрола верзија (енг. *version control system*) и тиме ће сваки члан тима добити исту конфигурацију и, једноставно, командом *vagrant up* добити окружење као и Ви.

Сваки пројекат који користи Вагрант машине треба да има само један *vagrantfile*. Ова конфигурациона датотека не зависи стриктно од платформе коју користите, осим ако се у њој не позивате на функције које су специфичне за неку платформу, што је могуће, али се ретко ради и није препоручљиво.

Када покренете Вагрант, он тражи *vagrantfile* у тренутном директоријуму. Ако ту не пронађе конфигурациону датотеку, он је тражи у родитељском директоријуму и тако даље. Ово је веома корисно јер се можете кретати кроз директоријуме унутар вашег пројекта и увек ћете моћи да позивате Вагрант без потребе да се враћате у главни директоријум вашег пројекта.



Подразумевани *vagrantfile* добијате када у тренутном директоријуму извршите *vagrant init*. Ова конфигурациона датотека користи Руби синтаксу, али знање овог програмског језика није потребно да бисте разумели конфигурацију и подесили исту у складу са вашим потребама и потребама вашег тима и пројекта.

Аутоматска конфигурација

Вагрант подржава такозвани *provisioning*, односно аутоматску конфигурацију кутије приликом првог покретања. Ово је веома корисно јер не морате правити кутију за сваки пројекат, већ, рецимо, можете користити почетну Дебијан конфигурацију, а подесити специфичну конфигурацију у зависности од пројекта. Постоји много система за аутоматску конфигурацију. Најједноставнији систем је *shell*, односно да направите скрипту која ће се покренути приликом првог покретања машине и подесити аутоматски све што сте навели у њој. Овде можете инсталирати своје пакете који су вам потребни, подесити веб сервер и слично. Наравно, датотеке за систем аутоматске конфигурације морате такође поделити са вашим тимом кроз систем контрола верзија.

Поред *shell* приступа могуће је користити и друге системе као што је Ансибл, о којем можете читати у нашем часопису.

Систем аутоматске конфигурације, као и специфични параметри за конкретан систем, подешавају се у *vagrantfile*-у.

У наредним бројевима ћемо детаљније причати о овој тематици, а препоручујемо вам и да консултујете званичну документацију и да се сами опробате у конфигурисању и експериментисању.

За све недоумице и проблеме нас можете контактирати на нашу адресу електронске поште.



VAGRANT

Ослобађање

Развој слободног софтвера



Аутор: Никола Харди

Слободан софтвер отвореног кода постоји, то је чињеница. Често се поставља питање ко и зашто развија овакав софтвер. Друго питање које се често поставља је: „Како могу да се укључим у неки пројекат који ми се допада?“ Овим текстом ћемо покушати да дамо одговор на оба ова питања.

Разлози и начини за укључење у пројекат слободног софтвера

Осећај припадности

Разлога има много и углавном су личне природе. Најважнији, а вероватно и најбоље прикривен, је осећај припадности. Људи су друштвена бића иако програмери можда не изгледају тако. Сви пројекти су пре свега заједнице: имају свој карактер, правила понашања и историју. Неки су престижни, неки су велики, неки су невидљиви. Желимо да будемо део неких од њих.

Дељење кода

Следећи разлог који је врло чест је дељење кода. Људи воле да се похвале својим делом. Програми јесу ауторска дела, имају различиту намену, али и различиту вредност. Програмери воле да се похвале, а углавном то раде ћутке. Један од предлога за оснивање успешне заједнице је да се започне рад самостално, објави јавно што пре и настави са радом. Тек онда ће се други полако прикључивати и настајаће заједница која ће се даље бринути о животу тог пројекта.

Локализација

Програми који су преведени или потпуно локализовани за наше подручје могу да



Развој слободног софтвера

буду у озбиљној предности у односу на друге, можда чак и напредније програме. Постоји велики интерес за рад на слободном софтверу управо због овог разлога.

Подршка

Заједнице формиране око слободног софтвера су заједнице у којима најчешће владају меритократија и знање. Сви ти чланови су у неком тренутку добили подршку од других чланова заједнице и често осећају потребу да ту услугу узврате. Помагати другима значи напредовање у хијерархији заједнице и стицање једног облика угледа и поштовања.

Тестирање

Развој било каквог софтвера подразумева много активности поред самог писања кода. Ту је и писање документације, писање предлога за нове могућности, тестирање итд. Корисници су ти који могу да дају корисну информацију о томе како би пројекат требало да напредује. Поред тога, корисници могу да пруже информације о томе који делови програма се не понашају очекивано. И ово је начин на који неко доприноси развоју пројекта. Следи више конкретних детаља како се укључити у развој једног пројекта.



Ослобађање

Укључивање у пројекат: корак по корак

Комуникација

Први корак би требало да буде упознавање са заједницом. Веб-сајт једног пројекта или заједнице може да ода погрешну слику о њој. Као пример можемо слободно да наведемо почетне странице ЛУГонС или ЛиБРЕ! пројеката, које нису претерано занимљиве, иако су обе заједнице врло активне. Кључ је у томе што заједнице могу да буду најактивније на неким другим каналима за комуникацију. Рецимо, слободно се може рећи да је ЛУГонС заједница најактивнија на свом IRC каналу, а то није толико приметно некоме ко посети сајт ове заједнице. Слично је и са многим другим заједницама. Треба завирити и на IRC канале, форуме, дописне листе (енг. *mailing list*) итд. Ови канали су право место за упознавање заједнице, учење, тражење и пружање подршке и информисање. Помагати другима да реше проблем у вези са једним пројектом уједно значи и пружити подршку том пројекту.

Документација

Проблеми се често понављају и постаје заморно да се знање преноси „усменим“ путем у форми разговора. Због тога постоји документација у разним облицима која захтева како писање, тако и одржавање. Рад на документацији може значити и уређивање хијерархије у документацији, уклањање застарелих информација и усмеравање корисника на читање документације. Документација у OSS (енг. *open source software*) пројектима је често доступна у форми викија (енг. *wiki*). У последњих неколико година, популарна је и форма електронских књига (*epub, PDF, web...*). Још један облик документације који треба поменути је сајт <https://readthedocs.org> који приказује документацију записану у коду у једном пријатнијем облику. Сви ови видови документације захтевају, писање и одржавање, те представљају сјајан начин за почетак рада на неком пројекту.

Локализација

Локализација углавном подразумева превод програма или документације. Део текста који се користи у програмима и може да се преведе смештен је у посебним фајловима. Често је реч о тзв. **.po** фајловима и GNU *gettext* стандарду. Тада се превод програма врши тако што се преузму **.po** фајлови за одговарајући језик, учитају се у уређивач за **.po** датотеке. Тада може бити додат нов садржај или измењен постојећи. Други, модернији начин који се такође ослања на *gettext*



Развој слободног софтвера

су веб-сервиси за уређивање **.po** фајлова. Корисници могу помоћу интернетског прегледача да приступе таквом сервису, додају предлоге превода и оцењују друге преводе. Овакав начин знатно олакшава спајање свих измена у једну целину, контролу превода и комуникацију међу корисницима који се баве преводом. Неки од таквих сајтова су <https://transifex.org> и <https://launchpad.net>.

Пријављивање проблема

Они који су спремни да још више „запрљају руке“ могу да помогну тестирањем најновијих верзија свог омиљеног програма. Многи пројекти имају такозване најтлџ билдове (енг. *nightly build*). У таквој верзији програма се налазе најсвежије измене. Идеја је да се ноћу све измене у коду сакупе и дистрибуирају корисницима који желе да се баве тестирањем. Од корисника се очекује да примећене проблеме пријаве на сајту за праћење проблема (енг. *issue/bug tracker*). Проблеми се не морају односити на најтлџ билд већ може бити у питању било која верзија. Проблеми могу да буду најразличитијег карактера - од погрешног превода, претеране употребе ресурса, до неочекиваног понашања програма. Различити пројекти очекују пријаве о грешкама на различитим местима. Неки пројекти за то користе своје приватне „багзиле“ (енг. *bugzilla*) или користе услуге других сајтова, као што су лончпед (енг. *launchpad*), гитхаб (енг. *github*) или сорс форџ (енг. *source forge*). Најбоље је завирити у документацију или се консултовати са осталима о томе где се налази тај систем. Важно је упамтити да прво треба да се провери да ли је проблем већ пријављен, а потом детаљно описати карактер проблема, верзију програма и система на ком се проблем догодио, и начин на који се може репродуковати такав проблем.

Одржавање пакета

У претходном броју било је речи о начинима како се програми могу дистрибуирати и објашњени су појмови репозиторијума, пакета и њихов значај. Слободан софтвер се дистрибуира у пакетима различитог типа. Чест је случај да много дистрибуција користи исти тип пакета. Међутим, дистрибуције имају своје репозиторијуме. Људи који се баве одржавањем неке дистрибуције се, између осталог, брину и о томе да се софтвер лепо уклопи, да не постоје конфликти и да све међузависности буду испоштоване. Ти људи се у жаргону зову пакетари, а њихов посао је прављење (паковање) и одржавање пакета. Пакетари су често организовани у тимове који се брину о одређеном скупу пакета сличне намене. Паковање софтвера захтева велику одговорност и искуство. Неко би могао да

Ослобађање

каже да је чак једноставније направити измену у коду него правилно спаковати тај код. Због тога пакети поред имена аутора програма садрже и име особе која је направила пакет. Прављењем пакета се подржавају и дистрибуције и програми и ово је сјајан начин да се помогне заједницама, а у питању је и велик изазов.

Писање кода

Под развојем слободног софтвера углавном се подразумева програмирање. Рад на коду захтева проналажење ажурног репозиторијума који садржи код, преузимање кода одговарајућим VCS-ом (енг. *versioning control system* - на пример, гит). Затим код може да буде измењен, а измена послата на ревизију. Измене се традиционално размењују закрпама или печевима (енг. *patch*), а данас се често среће и размена измена у облику пул риквеста (енг. *pull request* - захтев за прихватање измена). При писању измена у коду треба водити рачуна о договорима и стандардима за уређивање кода. Често постоје записане смернице за писање измена (енг. *contributing guide*).

Оснивање пројеката и заједница

Сасвим други ниво подршке неком пројекту би био оснивање локалне заједнице. Локалне заједнице се углавном баве пружањем подршке. Међутим, могу се бавити и локализацијом, и другим стварима. Код оснивања заједница око већ постојећих пројеката треба имати на уму да заједнице често надживе пројекат; квалитет заједнице је од кључне важности за квалитет осталих делова пројекта. Предуслов за долазак нових људи у заједницу је инфраструктура погодна за рад. То подразумева инфраструктуру која је стандардна, лако доступна и редовно одржавана. Надамо се да смо вас охрабрили и да ћете помоћи неком пројекту чије производе често користите. Уколико овај текст није био довољан, слободно нас контактирајте и помоћи ћемо вам да се на прави начин укључите у жељени пројекат.





Ваш посао, опен-сорс посао (3. део)

Гугл - Савршена рекламна агенција

Аутор: Дејан Маглов

Прошло је скоро годину дана откако је овај важан серијал прекинут због спречености изворног аутора серијала Марка Кажиха да га настави. Међутим, после дугог чекања настављамо са новим аутором. Нови аутор ће донети мало другачији концепт и редослед у анализу ове области и надамо се да вам то неће сметати.

У двадесет и четвртом броју ЛиБРЕ! часописа био је објављен увод у овај серијал и најављен је чланак о компанији Ред Хет (објављен већ у следећем, двадесет и петом броју). Такође је био најављен и чланак о Каноникалу (*Canonical*) за неки од наредних у серијалу, који је, нажалост, нагло прекинут па тај чланак никада није објављен.

Нови аутор је одлучио да мало промени редослед и да причу о Каноникалу одложи за неки наредни пут, док ће сада бити речи о најјачој компанији која је своје пословање засновала на слободном софтверу - Гуглу (*Google*)

Нови почетак

На новом почетку серијала морамо да се извинимо читаоцима што неће бити коришћена званична терминологија из света економије. Ови модели финансирања имају своја званичну терминологију у економији, али с обзиром на то да аутор није економиста као ни већина читалаца, она, ради једноставности текста, неће бити коришћена у овом серијалу.

Примарна идеја је да се у овом серијалу идентификују модели и да се покаже да

Слободни професионалац

је могуће развити бизнис у свету слободног софтвера и без директне продаје софтвера као интелектуалне својине.

Слободан софтвер није продукт интелектуалног рада нити појединца нити једне компаније. Он је продукт велике заједнице тако да је немогуће заштитити интелектуална права на тај софтвер и тако створити предуслов за продају. Па ипак, програмери, носиоци пројеката, само су људи који имају своје потребе и не могу живети од идеје и филозофије. За даљи, бржи развој пројеката слободног софтвера потребан је новац. Немогућност директне продаје софтвера захтева више маштовитости у креирању бизниса на основи слободног софтвера.

У тексту о компанији Ред Хет, представили смо савршен модел бизниса који се не заснива на продаји ГНУ/Линукса, него на услугама одржавања, имплементације, едукације, првенствено у пословном окружењу. Доказ да овај модел, барем у Америци, функционише, је чињеница да је још пре 2 године компанија Ред Хет прескочила магичну цифру од милијарду долара профита.

Гугл



Још успешнији модел бизниса је развила компанија Гугл. Беспотребно је доказивати успешност ове компаније. Њени производи као што су интернет-претраживач, интернет-апликације и сервиси, Гугл Кром (*Google Chrome*) интернет-прегледач, Гугл Кром ОС, Андроид ОС за мобилне уређаје (има измењено Линукс језгро) и други - слободни су софтвери или сервиси и притом потпуно бесплатни. Гледајући само

софтвере и сервисе које Гугл развија и подржава могло би се закључити да се ради о правом ФЛОСС пројекту. Нажалост, комерцијализација по Гугловом моделу је потпуно искривила суштину филозофије слободног софтвера.

Када је Гугл у питању, сасвим је јасно одакле потичу његови приходи. Гугл је



Опен-сорс посао

постао савршена глобална рекламна агенција. Популарност својих производа је искористио да наплаћује и пласира рекламни материјал.

Сви знамо да Гугл нуди рекламе сваком кориснику али се мора признати да то раде и са доста дискреције. Још се никада није десило да корисник за претрагу користи Гугл претраживач, а да као резултат добије само плаћене рекламе које немају везе са предметом претраживања. Уметност Гугловог рекламирања је пласирање рекламе на такав начин да корисник није свестан да му је реклама подметнута, већ да има утисак да је сам пронашао решење свог проблема.

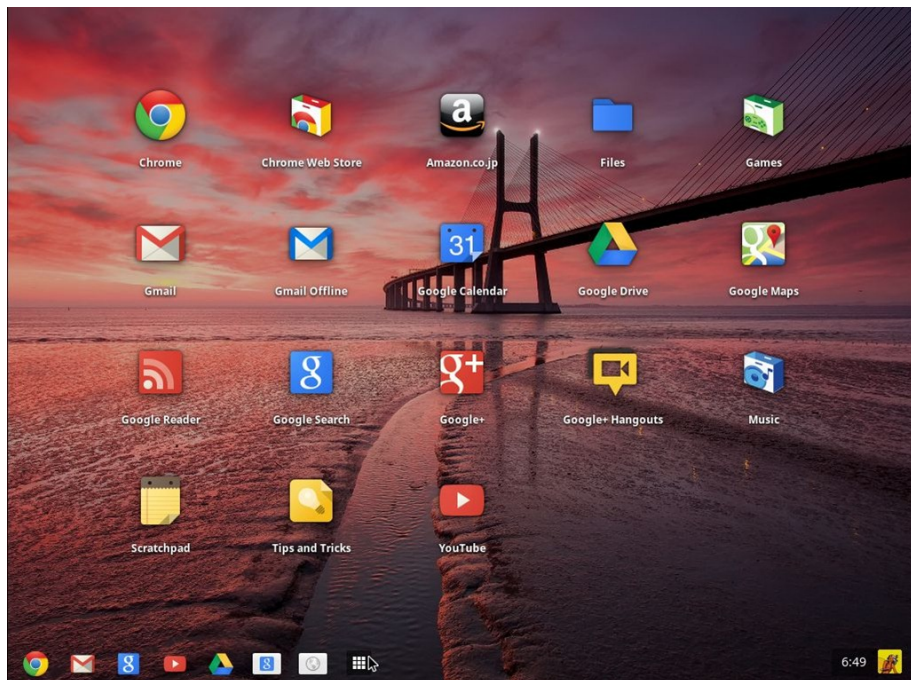
Гугл своју супериорност у претраживању интернет садржаја базира не само на добром познавању предмета претраживања, него и на добром познавању корисника, праћењем његових навика и интересовања. Персонализовање резултата претраге даје квалитетније резултате за корисника јер са мање речи налази оно што га занима. Цена за то је задирање у приватност корисника. Можда нам не би сметало ово задирање у приватност да се не бојимо од злоупотребе тих података и то не само од стране Гугла, него и неких других заинтересованих страна које би могле да преко Гугла дођу до њих.



Гугл је нашао интерес у ФЛОСС-у и претворио га у алат који ће преусмерити кориснике на његов претраживач и сервисе, чиме је тај софтвер постао само бесплатни софтвер отвореног кода, а слободу корисника је изгубио. Према Гулу, будућност десктоп рачунара је у јефтиним машинама без тврдог диска који покреће оперативни систем који има само интернет прегледач и који у потпуности преусмерава корисника на Гуглове већ постојеће веб апликације: Гугл пошту, Гугл драјв, Гугл документе, Гугл пикасу, Гугл мапе, Гугл календар,

Слободни професионалац

Гугл плус, Гугл игре, Гугл интернет продавницу... На тај начин корисник би поприлично уштедео на уређају и софтверу, али би изгубио и оно мало привидне приватности коју има сад.



Међутим, Гугл своје кориснике не тера ни на шта. Нико није присиљен да купи Андроид телефон, користи Кром интернет-прегледач и Гугл претраживач. Корисници сами пристају на то јер добијају квалитетне производе и имају утисак да су сами изабрали најбоље. Слободни софтвер има потенцијал да буде најбољи, али га Гугл злоупотребљава. Уместо потпуне слободе, Гугл је свог корисника стрпао у матрикс свет привида да сам одлучије о свему док је заправо само батерија за Гугл машинерију.

* * *



Опен-сорс посао



Гугл је трошкове развоја софтвера пребацио са корисника на стране заинтересоване за рекламирање, као и на њихове потрошаче. Задовољни су пројектанти софтвера, корисници софтвера, произвођачи робе, јер лако долазе директно до купца, а и купци, јер лако долазе до тражене робе. Гугл не само да тим својим приходима развија свој слободни софтвер, него и преко програма *Google Summer of Code* подржава и независне пројекте слободног кода. Стиче се утисак да је, ипак, и поред злоупотребе, сам слободни софтвер више на добитку него на губитку. Отворени код и антимонополски закони могу да исправе тренутно стање у којем Гугл постаје глобални Велики брат.



Слободни професионалац**A N S I B L E (1. део)**

Аутор: Стефан Ножинић

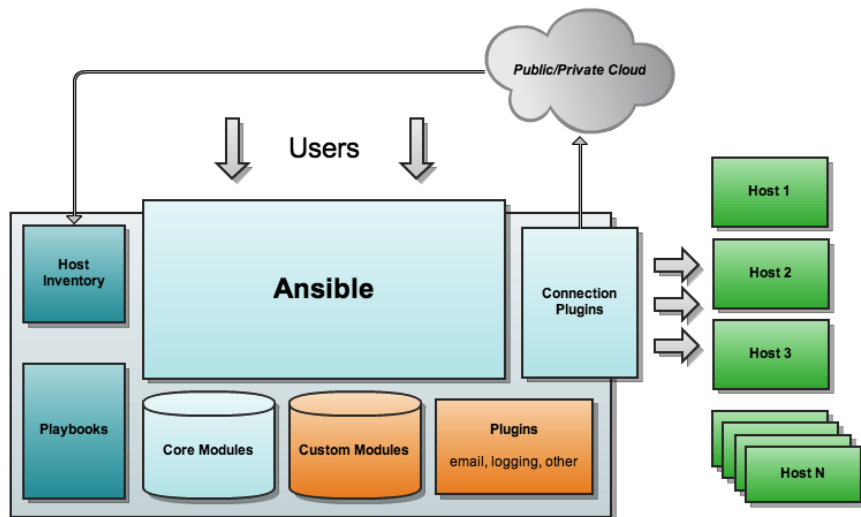
Увод

Администрација кућног система никад није представљала проблем. Све што сте икада требали да урадите јесте по које ажурирање и инсталација потребног софтвера. Када треба радити администрацију озбиљнијих система као што су то бројни сервери, ту се већ прелази у други домен комплексности. Овде је намерно употребљена множина јер се жели нагласити да је реч о више сервера које треба одржавати. Током времена су развијени многи алати за овакав менаџмент. Доста таквих алата је захтевало инсталацију специфичног софтвера на све сервере како би се они могли спојити са главним рачунаром преко којег је администратор управљао читавом групом сервера. Ово је на први поглед изгледало сасвим прихватљиво али се временом показало као погрешно. Софтвер је постајао све комплекснији и имао је све више грешака. Ово је касније администраторима направило више проблема него што су имали пре.

Ансибл (*Ansible*) је један од ретких и квалитетних софтверских решења које је често коришћено и које омогућава лаку аутоматизацију одржавања више сервера. Поред тога, он не захтева знање неког скрипт-језика за неке основне ствари. Такође, не захтева инсталацију на свим серверима јер уместо својих агената на сваком серверу, он повезује главни сервер и остале преко *SSH* (енг. *Secure Shell*) протокола. Ансибл користи језик за конфигурацију који је једноставан и може се савладати без много учења. Ово омогућава



администраторима да се фокусирају на проблеме који се тичу саме стратегије одржавања система уместо на куцање кода.



Због ове једноставности у конфигурацији, Ансилб могу користити програмери, менаџери, системски администратори и људи осталих врста професија. Конфигурацију је лако писати, читати али и одржавати после неког времена.

Дакле, да сумирамо, Ансилб нам омогућава да контролишемо своје сервере са једног рачунара. Ако нам треба инсталација базе података на десет сервера, нема проблема - Ансилб нам то омогућава путем једноставне конфигурације и то можемо урадити са нашег лаптоп рачунара макар били и километрима далеко од тих сервера. Важно је само да имамо *SSH* приступ тим серверима.

Инсталација

Ансилб је подржан у скоро свакој дистрибуцији. Тако постоје пакети за Убунту, Дебиан, Арчлинукс, Центу и остале велике дистрибуције. У случају да не постоји пакет за вашу дистрибуцију, Ансилб можете преузети и инсталирати помоћу алатке *pip*. Препоручујемо, ако имате времена и воље, да у том случају направите пакет за вашу дистрибуцију, проследите то заједници и тиме будете део развојног тима слободног софтвера. Нема боље сатисфакције за једног Линукс

Слободни професионалац

корисника, зар не?

Ако, којим случајем, наиђете на проблем током инсталације, или не знате уопште како да инсталирате Ансибл на ваш рачунар, можете нас контактирати путем наше електронске поште која вам је, надамо се, до сада већ позната - [libre \[et\] lugons \[dot\] org](mailto:libre@lugons.org).

Још мало увода

Пре него што кренемо са основама, треба појаснити како Ансибл комуницира са осталим рачунарима. Ансибл користи *SSH* протокол да комуницира са осталим серверима преко којег им говори које команде извршавају. *SSH*, као што вероватно знате, може се користити да се спојите на удаљени сервер помоћу шифре или помоћу кључа (*RSA / DSA*). Ансибл омогућава оба начина, али такође и, поред употребе *SSH* протокола, конекцију на *chroot*, *lxc*, *jail* или локално.

Ваше удаљене сервере наводите у посебном фајлу тако да Ансибл тачно зна коме треба да приступи.

Први пример

Сада, када имамо мало предзнања о томе шта Ансибл ради и чему конкретно служи, можемо се бавити техничким стварима и погледати како то да конкретно искористимо у пракси.

Претпоставимо да имамо три сервера са следећим адресама: 192.168.1.3, a.home.net, b.home.net и да те сервере желимо да администрирамо одједном.

У фајлу `/etc/ansible/hosts` наводимо адресе тих сервера у сваком реду посебно на следећи начин:

```
192.168.1.3
a.home.net
b.home.net
```

Претпоставља се да имате приступ тим серверима преко *SSH* протокола, односно да имате њихове кључеве (у овом примеру претпостављамо да желимо користити



кључеве уместо шифре).

Сада можемо „пинговати“ све наше сервере наведене у горе описаном фајлу.

```
ansible all -m ping
```

Ансибл ће се повезати на удаљене сервере користећи корисничко име са вашег система одакле покрећете Ансибл. Ово је могуће променити, али то нећемо објашњавати у овом тексту. Ако вас занимају конкретно неки детаљи, можете нас контактирати.

Ако смо установили да све до сада ради како смо очекивали, можемо покренути команду на свим серверима на следећи начин:

```
ansible all -a "/bin/echo hello"
```

Једноставно, зар не?

За крај

Треба нагласити да је Ансибл много више од покретања команди на више рачунара. Праву моћ овог алата тек желимо да вам покажемо, али, као што знате, не можемо толико тога да ставимо у један чланак. Треба оставити простора и за друге теме. Оно што можемо да вам обећамо јесте да ћемо се потрудити да у наредним текстовима о овом алату дамо што више примера и занимљивих информација. Надамо се да ћете и ви убудуће неке преносити сјајне утиске о овом алату као што ми сада преносимо вама.

Ако имате неко питање, предлог, занимљивост или, пак, желите да нам укажете на технички или било какав други пропуст, контактирајте нас на [libre \[et\] lugons \[dot\] org](#) и ми ћемо дати све од себе да будемо још бољи.

Шифровани чет (2. део)

Нилток

Аутор: Криптопанк

Као што смо у прошлом броју писали о новој и фантастичној Суброси, овде ћемо поменути једно новорођенче у свету брзог шифрованог дописивања. Да се не лажемо, није ни близу по практичности и паметним решењима као што је то случај код Субросе, али увек је лепо осетити да се људи труде у овој области стварајући нове и иновативне апликације - такозване апликације-дебитанте.

Кад смо већ код Субросе, њихов сајт је већ неко време недоступан (<http://goo.gl/Nf8neL>), али то не би требало да буде нарочити проблем јер је могуће подићи Суброса сервер из удобности вашег дома (енг. *selfhost*), што су неки већ урадили како би себи приуштили већу сигурност, приватност и, што је најважније, независност од главног удаљеног сервера Субросе.

Иза имена „Нилток“ (*Niltalk*), које веома добро описује ову веб апликацију, крије се веома проста идеја за шифрована дописивања, искована преко мреже у програмском језику Голанг.





Шифровани чет

Ствари функционишу овако: Када посетите њихову веб-страницу (<http://goo.gl/N4wVUO>), видећете поље у које треба да унесете шифру, након чега се прави ваша нова соба за ћаскање и јединствени линк који води њој. Та шифра ће бити коришћена за шифровање текста свих порука и биће потребна као идентификација за приступ соби за ћаскање коју сте управо направили. Сада морате да другима пошаљете овај линк и шифру. Овде се јавља и један проблем који се појављивао и код новијих сигурних и-мејл провајдера попут Протонмејла (*Protonmail*) и Скиптмејла (*SCYPTmail*), а то је да ту шифру сада некако морате јавити или послати другима са којима желите да сигурно ћаскате, али тако да то нико осим њих на интернету не зна.

Логично се намеће решење да у те сврхе користите шифровану пошту, али верујемо да би то деловало мало одбојно и могло би да обесхрабри људе да користе ову апликацију, јер ко зна када и да ли други прегледају своју интернетску пошту. Овде треба обратити пажњу и на то да од тренутка када се нова соба за ћаскање креира, она ће се затворити после сто двадесет минута уколико јој до тада нико не приступи.

 niltalk



Free, instant, private, disposable chat

How does it work?

Niltalk lets you create instant, password protected chat rooms without the need to signup. Simply click the "Create" button, and share the unique chat URL with your peers.

All communication happens over SSL. Niltalk doesn't record or log IP addresses, messages, or peer handles anywhere.

Once a room is created, it has a lifetime of 120 minutes before the first login. Then, the room is automatically deleted after 10 minutes of inactivity (no messages exchanged).

While in a room, you or any of your peers can dispose of the room with the click of a button.

Why can any connected peer dispose of a room?

Niltalk is meant for holding short private conversations between groups of people who have mutually agreed to converse. There is no concept of ownership of a room, and introducing ownership complicates the otherwise simple privacy feature of instant disposal by any participant. This also means that Niltalk isn't really meant for starting conversations by opening up a room to

Интернет мреже и комуникације

Ово, иако добра предострожност, заправо представља пребацивање проблема са себе (Нилтока) на кориснике, који у већини случајева не размишљају о сигурности. Не треба да помињемо да би употреба неког другог вида шифрованог дописивања за размену шифре за Нилток била бесмислена из два разлога. Први, ако је већ тај вид шифрованог ћаскања сигуран, зашто користити Нилток - и други, ако није сигуран, зашто преко њега размењивати тајне?

Наравно, ове собе нису у власништву онога ко их креира и он нема никаквих привилегија у односу на друге саговорнике у истим.

Али, рецимо да сте некако сигурно разменили кључеве и сада се већ лагодно дописујете са пријатељима, приметићете дугме „Dispoze“ за напуштање тренутне собе. Такође, соба ће се аутоматски сама обрисати и тиме постати недоступна након десет минута без икакве активности (порука) унутар исте. Нилток тврди да не чува никакве логове ИН адреса, а сва комуникација са Нилтоком увек иде преко сигурне SSL конекције.

Нилток је само апликација за ћаскање и нема опције попут приватних порука неком саговорнику на листи ван главног канала који сви у соби виде. Нема подршке ни за видео, ни за дељење фајлова, а цела аудио подршка је звучно обавештење о новој поруци унутар собе у којој сте.

Међутим, није све тако лоше. Као прво, Нилток је (наравно) отвореног кода са AGPL3 лиценцом и доступан је за преузимање са Гитхаба (<http://goo.gl/ZeeEM0>), а друга позитивна страна је да нема никаквог регистровања и остављања било каквих личних података. Можете ову апликацију користити и преко Тор анонимне мреже; само треба да смислите добар начин да друштву јавите шифру (ово никако не радите преко друштвених мрежа).





Bootstrap

Први кораци

Аутор: Златан Васовић

Увод

У тридесетом броју ЛиБРЕ! часописа упознали смо вас са Бутстрапом (*Bootstrap*) и рекли смо да је то фрејмворк (енг. *framework*) за веб-дизајн или још једноставније речено, то је скуп предефинисаних алата који олакшавају дизајн веб-презентација и веб-апликација.

Овог пута идемо корак даље и показујемо вам како да почнете да користите Бутстрап у својим пројектима.

Први кораци са Бутстрапом нису тешки. Потребно је преузети спремљене CSS и Џаваскрипт (*JavaScript*) Бутстрап датотеке, одабрати неки од шаблона - или, ипак, „кренути од нуле“ и направити сајт без готовог шаблона.

Преузимање

Све што је потребно за преузимање Бутстрапа је одлазак на <https://github.com/twbs/bootstrap/releases/>, затим налажење потребне верзије (последња стабилна верзија тренутно је 3.3.2) и преузимање „*dist*“ датотеке, која је међу линковима преузимања испод верзије. Актуелни *dist* (унапред спремљен) пакет је <https://github.com/twbs/bootstrap/releases/download/v3.3.1/bootstrap-3.3.1-dist.zip>.

Сам свој мајстор

Након распакивања ZIP архиве у рут (енг. *root*) фолдер вашег сајта и повезивања са ХТМЛ странама сајта наредбама

```
<link href="css/bootstrap.min.css" rel="stylesheet">
<script src="js/bootstrap.js"></script>
```

- Бутстрап је спреман за коришћење.

Бутстрап се може учитати и путем такозваног *CDN*-а (енг. *content delivery network*). Линкове до *CSS* и *JS* датотека, као и одговарајући код за њих, можемо наћи на <http://www.bootstrapcdn.com/>.

Документација

Документација за Бутстрап (<http://getbootstrap.com/getting-started/>) садржи све што је потребно корисницима да почну израду веб-сајтова.

Bootstrap Getting started CSS Components JavaScript Customize Expo Blog

Getting started

An overview of Bootstrap, how to download and use, basic templates and examples, and more.

SOURCEBITS MOBILE MAKEOVER Level up your app with a FREE redesign from the mobile design experts! ads via Carbon

Download

Bootstrap (currently v3.3.4) has a few easy ways to quickly get started, each one appealing to a different skill level and use case. Read through to see what suits your particular needs.

<p>Bootstrap Compiled and minified CSS, JavaScript, and fonts. No docs or original source files are included.</p> <p>Download Bootstrap</p>	<p>Source code Source Less, JavaScript, and font files, along with our docs. Requires a Less compiler and some setup.</p> <p>Download source</p>	<p>Sass Bootstrap ported from Less to Sass for easy inclusion in Rails, Compass, or Sass-only projects.</p> <p>Download Sass</p>
--	--	---

Bootstrap CDN
The folks over at [MaxCDN](#) graciously provide CDN support for Bootstrap's CSS and JavaScript. Just use these [Bootstrap CDN](#) links.

- Download
- What's included
- Compiling CSS and JavaScript
- Basic template
- Examples
- Tools
- Community
- Disabling responsiveness
- Migrating from 2.x to 3.0
- Browser and device support
- Third party support
- Accessibility
- License FAQs
- Translations
- Back to top

Постоји више превода целог сајта документације, али међу њима није српски,



тако да ће вам бити неопходно одређено знање енглеског језика за читање документације.

Подељена је на четири дела - „*Getting started*” (како почети), „*CSS*” (основне *CSS* компоненте), „*Components*” (додатне *CSS* компоненте) и „*JavaScript*” (Џаваскрипт компоненте).

У делу „*Getting started*” се налазе линкови за преузимање, садржај Бутстрапа, листа познатих грешака, поступак преласка са Бутстрапа 2 на верзију 3, као и неколико секција за напредне кориснике Бутстрапа. Остала три дела документације прилично детаљно представљају све компоненте Бутстрапа.

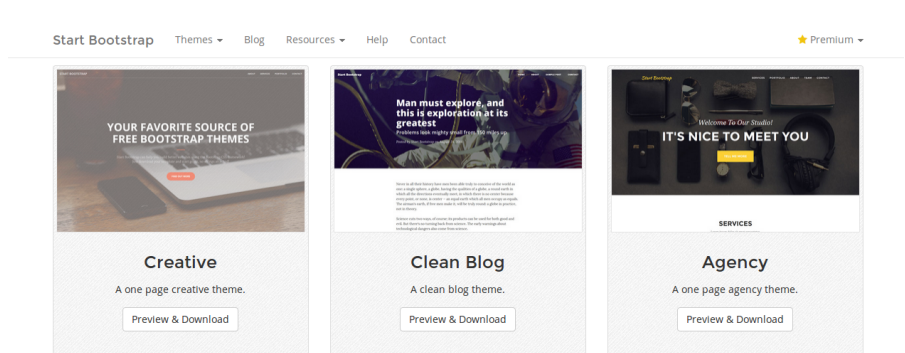
Шаблони

#	Header	Header	Header	Header
1,001	Lorem	ipsum	dolor	sit
1,002	amet	consectetur	adipiscing	elit
1,003	Integer	nec	odio	Praesent
1,003	libero	Sed	cursus	ante
1,004	dapibus	diam	Sed	nisi

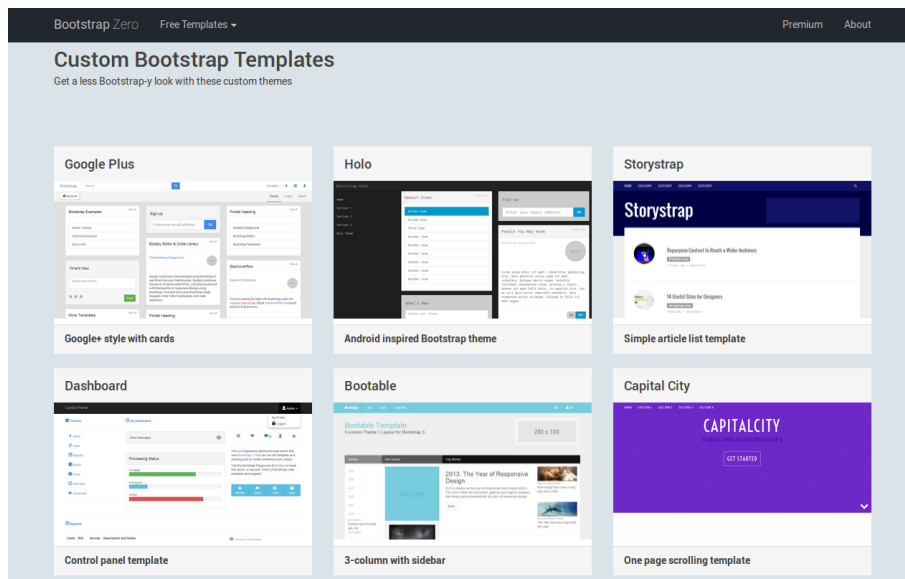
Ипак, није свима лако снаћи се „исправе”. Многима ће при изradi сајтова помоћи велики број шаблона, било званичних (<http://getbootstrap.com/getting-started/#examples>), било корисничких, међу којима има и оних који се плаћају.

Поменућемо неколико најпознатијих сајтова са бесплатним корисничким шаблонима:

Сам свој мајстор



- <http://startbootstrap.com/>
- <https://wrapbootstrap.com/>
- <http://bootswatch.com/>
- <http://bootstrapzero.com/>



Сада смо спремни за следећи корак - израду веб-сајта.



BeagleBone Black Rev C

Водич од првог дана (6. део) - Биглбон Блек као Тор егзит

Аутор: Ненад Марјановић



beaglebone

У неколико претходних бројева писали смо о инсталацији и подешавањима Биглбона ревизије Ц коришћењем Дебијан Линукс дистрибуције. Данас тачку стављамо на сигурност Тор сервера али и самог оперативног система. После инсталације сваког сервера и након само неколико сати, а и раније, провером логова можемо закључити да смо већ на мети хакера. Ово се дешава због присутности великог броја мрежних скенера и алата за брутално пријављивање на *SSH* сервис.

Нажалост, ови напади трају годинама и вероватно ће још дуго бити ту. Анализом пријавних логова, који се у овом случају налазе у **/var/log/auth.log** фајлу, увек можемо утврдити да ли је неко покушавао да нас нападне. Команде које нам могу олакшати процес провере о покушајима напада на наш сервер су следеће:

```
cat /var/log/auth.log | grep failed  
cat /var/log/auth.log | grep failed | wc -l
```

Кратак опис ових команди је следећи:

- cat** приказује задржај фајла
- grep** тражи жељену реч
- wc -l** приказује укупан број

Пример на тек инсталираном серверу након тридесет минута активности:

Хардвер

У претходном броју смо говорили о промени порта за *SSH* конекцију, али данас

```
root@sd-33602:~# cat /var/log/auth.log | grep failed | wc -l
45
root@sd-33602:~#
```

настављамо причу употребом геолокализацијског софтвера. Овај софтвер врши рестрикцију приступа серверу према земљама које одаберемо. За ово решење су нам потребне две **баш** (енг. *bash*) скрипте и два пакета. Цео процес започињемо инсталацијом **geoip** пакета и базе.

```
apt-get install geoip-bin geoip-database
```

Да бисмо проверили успешност инсталације:

```
geoiplookup pentester.iz.rs
```

```
root@sd-33602:~# geoiplookup pentester.iz.rs
GeoIP Country Edition: FR, France
root@sd-33602:~#
```

Прву скрипту креирамо у **/usr/local/bin** досијеу и подешавамо дозволе за њено извршавање.

```
nano /usr/local/bin/sshfilter.sh
```

Затим копирамо следећи садржај:

```
#!/bin/bash
# UPPERCASE space-separated country codes to ACCEPT
ALLOW_COUNTRIES="RS FR" f [ $# -ne 1 ]; then
  echo "Usage: `basename $0` <ip>" 1>&2
  exit 0 # return true in case of config issue
fi
COUNTRY=`/usr/bin/geoiplookup $1 | awk -F " : " '{ print $2 }' | awk
-F " , " '{ print $1 }' | head -n 1`
[[ $COUNTRY = "IP Address not found" || $ALLOW_COUNTRIES =~ $COUNTRY
```



```
]] && RESPONSE="ALLOW" || RESPONSE="DENY" f [ $RESPONSE = "ALLOW" ]
then
    exit 0
else
    logger "$RESPONSE sshd connection from $1 ($COUNTRY)" exit 1
fi
```

Потребно је изменити **ALLOW_COUNTRIES** у зависности из које се земље редовно конектујемо на сервер. За Србију је то **RS**, за Француску **FR** итд. Поштујемо велика слова и размак, а пишемо без зареза уколико уносимо више земаља. Подешавамо дозволу за извршавање скрипте:

```
chmod +x /usr/local/bin/sshfilter.sh
```

Закључавање SSH-а

Да бисмо омогућили употребу геолокализације, морамо изменити два системска конфигурациона фајла.

```
nano /etc/hosts.deny
```

И на крају додајемо:

```
sshd: ALL
```

Затим мењамо садржај у следећем фајлу:

```
nano /etc/hosts.allow
```

И на крају фајла додајемо:

```
sshd: ALL: aclexec /usr/local/bin/sshfilter.sh %a
```

Провера ефикасности система

Уколико смо ауторизацију за SSH подесили само на Србију, добићемо следеће резултате:

Хардвер

```
/usr/local/bin/sshfilter.sh 8.8.8.8
April 13 16:02:37 pi root: DENY sshd connection from 8.8.8.8 (US)
```

Да бисмо проверили ефикасност скрипте након неколико минута, покрећемо:

```
cat /var/log/messages | grep DENY
```

Добићемо следећи резултат:

```
April 13 17:59:54 pi logger: DENY sshd connection from 82.221.102.185 (IS)
April 13 17:47:54 pi logger: DENY sshd connection from 220.227.123.122 (IN)
April 13 17:43:51 pi logger: DENY sshd connection from 221.229.166.252 (CN)
April 13 16:49:04 pi logger: DENY sshd connection from 221.208.245.226 (CN)
```

Ажурирање GeoIP базе

За ово користимо још једну скрипту и **crontab** функцију:

```
nano /usr/local/bin/updategeoip.sh
```

Копирамо следећи садржај:

```
#!/bin/bash
cd /tmp
wget -q
http://geolite.maxmind.com/download/geoip/database/GeoLiteCountry/GeoIP.
dat.gz
if [ -f GeoIP.dat.gz ]
then
    gzip -d GeoIP.dat.gz
    rm -f /usr/share/GeoIP/GeoIP.dat
    mv -f GeoIP.dat /usr/share/GeoIP/GeoIP.dat
else
    echo "The GeoIP library could not be downloaded and updated" i
```

Подешавамо дозволу за извршавања фајла:

```
chmod +x /usr/local/bin/updategeoip.sh
```




Ову базу можемо ажурирати једном месечно и то радимо на следећи начин:

```
crontab -e
```

У фајл уписујемо следећу линију:

```
0 6 15 * * /usr/local/bin/updategeoip.sh
```

Овим подешавањем имаћемо надоградњу сваког петнаестог у месецу у шест сати. И за крај ћемо решити цурење *DNS* информација при посети сајтова од стране корисника нашег Тор сервера. За те потребе мењамо Тор конфигурациони фајл и на крај фајла додајемо следећи садржај:

```
AutomapHostsOnResolve 1
AutomapHostsSuffixes .exit,.onion
```

За крај поново покренемо *SSH* и поново учитамо Тор:

```
service ssh restart && service tor reload
```

За миграцију Тор сервера обавезно је сачувати **/etc/tor/torrc** фајл и садржај **/var/lib/tor/keys** директоријума. У супротном губимо ранг у Тор мрежи и крећемо испочетка.

Овим чланком завршавамо писање о Тор серверу на Биглбон ревизији Ц и прелазимо на поглавље веб сервера и инсталацију најсигурнијег веб сервера данашњице - Хијавата (енг. *Hiawatha*).

До читања.



Congratulations. You are using Tor.



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

