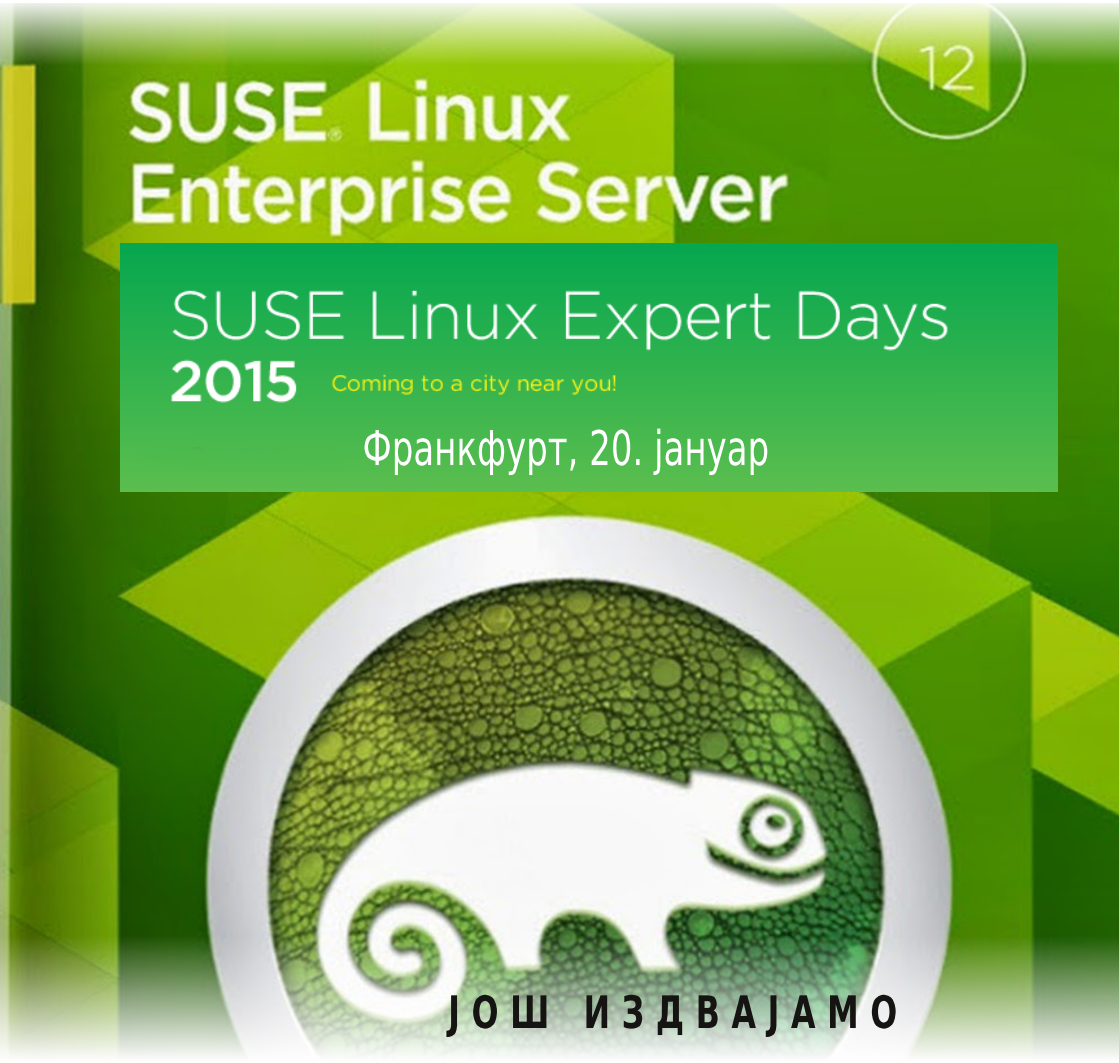
Март 2015. **Број 34**

ЛиБРЕ!

Часопис о слободном софтверу



ЈОШ ИЗДВАЈАМО

Vagrant

LUGoNS BarCamp №4

Реч уредника

CopyRight

Одавно нисмо „филозофирали" у речи уредника. Пошто тренутно нема неких великих догађања у пројекту искористићемо прилику да још једном подсетимо зашто пројекат између осталог постоји.

Одмах на почетку да нагласимо да ЛиБРЕ! није фанатично против власничког софтвера. Нисмо ни против права да професионални програмери раде и живе од своје интелектуалне својине. Само тржиште углавном има довољно својих механизама да одреди колико је нечији производ добар и користан а па према томе и колика је његова материјална вредност. Оно што нама смета и против чега се „боримо" је копирајт (енг. *copyright* - буквално, право на копирање а заправо власничко право) који је производ поремећаја тржишта то јесте стварања монопола неког производа. Једино производ који је створио монопол омогућава злопупотребу копирајта. Човек који има потребу да се сам превезе од А до Б има избор да аутомобил купи или ако му треба једнократно да позајми од рођака. Кад дође до поремећаја на тржишту онда корисник губи могућност избора а власник може да каже: „Овај производ није на продају, можете само да га изнајмите уз одговарајућу накнаду на одређено време, не смете да га отуђите јер није ваш и не смете да га позајмљујете и делите." Ово власнку омогућава практично да свој производ продаје више пута истом купцу и при томе има обавезу само да израђује резевне делове али не и да одржава производ у радном стању током времена изнамљивања.

Нус појава монополистичког положаја је могућност да нешто што није баш у реду у производу прогласиш за стандард и да то тако треба да ради.

Овакав накарадни копирајт је произашао из закона о заштити интелектуалне својине који је још у осамнестом веку осмишљен да заштити писце од неовлашћеног прештампавања који је узео маха појавом штампарске пресе. Своју кулминацију накарадности доживљава кроз музичку и софтверску индустрију од седамдесетих и осамдесетих година прошлог века. Није спорно да треба заштитити интелектауалну својину и омогућити ауторима, уметницима, научницима да живе од свог интелектуалног рада. Није морално да поједини заслужни уметници и научници умру у беди а толико су задужили свет својом интелектуалном заоставштином. Такође није добро да се носиоци интелектуалне својине током живота боре са немаштоном која их спречава да још више допринесу развоју целокупног друштва. Накарадност почиње кад

Реч уредника

копирајт уместо тога да буде заштита ауторских права постане заштита корпорациских интереса. Само корпорације су у стању да смисле вишеструке надокнаде за један те исти производ истом кориснику и да их спроведу у дело. Закон о ауторским правима који је део америчког устава је био територијално орјентисан и важио је само за територију САД. Под утицајем корпорација у сваки међународни споразум се уграђује и сагласност потписница да прихватају овај закон и тако он постаје ехтериторијални. Закон о ауторским правима је био временски ограничен што је омогучило носиоцима права да само одређено време наплаћују интелекуалну својину и терало их је да даље наставе да раде. Данашњи копирајт је ограничен на сто година или доживотно плус педесет година након смрти аутора. Ово сигурно није уведено због заштите наследника ауторских права него да корпорације извуку и последњу кап користи од интелектуалних права.

Овакав накарадни копирајт није подстицајан и да у софтверској индустрији није препознат као потпуно погрешан били би уназађени за наредних сто година. Као одговор на овако накарадни копирајт настао је слободан софтвер који је у већини ИТ сектора превазишао корпорације брже се прилагођавајући новим технологијама. Као резултат тога је доминација слободног софтвера на суперкомпјутерима, мобилним уређајима, веб серверима, веб апликацијама, комуникациским уређајима, „паметним" кућним уређајима...

Причати данас о слободном софтверу је још увек пионирски посао. Још увек се корисници слободног софтвера сматрају за штребере а слободан софтвер за претежак за обичног корисника. Права истина је да је слободан софтвер већ свуда око нас и користимо га на мобином телефону, таблету, на интернету, на кућним уређајима, аутомобилима, при прогнози времена... Зато ЛиБРЕ! часопис постоји да би од дрвета сви угледали шуму.

За крај речи уденика морамо да се извинимо Горану Стричићу зато што смо у прошлом (фебруарском) броју пропустили да се јавно захвалимо за чланак о ЛиБРЕ! часопису у фебруарском броју Света компјутера. Не би волели да испаднемо незахвални јер нам подршка Света компјутера много значи.

Сви који на било који начин желе да се укључе у пројекат часописа могу да се пријаве, коментаришу, хвале и критикују пишући на нашу већ познату адресу електронске поште - libre [et] lugons [dot] org .

До читања

ЛиБРЕ! Тим

Садржај

Моћ слободног софтвера

ubuntu SRBIJA

REGIONALNI LINUX PORTAL
linuxzasve.com

Slackware Srbija

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

WHITE CIRCLE & CREATIVE TEAM
GRAPHIC DESIGN & WEB DEVELOPMENT

info i tutorijali na srpskom
lubunturs.wordpress.com
lubuntu

ЈУГоНС
УДРУЖЕЊЕ ЛИНУКС КОРИСНИКА НОВОГ САДА

archlinux.rs

ЛИНог
Линукс блог | вести и мишљења

ubuntu LoCo Montenegro

Број: 34
Периодика излажења: месечник

Извршни уредник: Стефан Ножинић
Главни лектор:
Александар Божиновић

Лектура:
Јелена Мунћан Сашка Спишјак
Милена Беран Милана Војновић
Адмир Халилкановић

Графичка обрада:
Дејан Маглов Иван Радељић

Дизајн: White Circle Creative Team

Аутори у овом броју:
Ненад Марјановић Иван Радељић
Александар Весић Дарио Манеску
Никола Харди Слободан Николић
Криптопанк Бранимир Караџић

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

11. фебруар 2015.

Гугл покренуо свој нови пројекат отвореног кода

Гугл је покренуо Перфкит (енг. *PerfKit*) - своју нову алатку отвореног кода за мерење перформанси cloud сервера.

Користан линк: http://t.co/JBxFUDXgu0

---

15. фебруар 2015.

CrunchBang се повратио из клиничке смрти

Након што је главни оснивач овог пројекта одустао од истог, сајт https://crunchbangplusplus.org/ обећава наставак развоја ове дистрибуције и већ нуди бета верзију.

Користан линк: http://t.co/gXF3PwvIix

---

16. фебруар 2015.

Мозила захтева дигиталан потпис додатака

Мозила је најавила да ће променити своју политику и захтевати дигитално потписивање свих њених додатака за Фајерфокс.

Користан линк: http://t.co/XmX3RtKn7k

---

24. фебруар 2015.

Директор *NSA* тражи дозволу за читање енкриптоване комуникације

Мичел Роџерс, директор америчке безбедносне агенције (*NSA*), је изјавио да не треба дозволити енкрипцију података тако да влада не може да их прочита. Он је рекао да влада треба да има могућност да прочита енкриптовану комуникацију када за тим има потребе.

Користан линк: http://t.co/LvGOHXS1hD

---

25. фебруар 2015.

Фајерфокс 36 изашао

Нова верзија Фајерфокса претраживача је изашла са подршком за *HTTP/2* протокол

Користан линк: http://t.co/6u33Lacy4B

---

25. фебруар 2015.

Гит слави десети рођендан

Гит у априлу слави свој десети рођендан за који ће бити организован скуп у Паризу.

Користан линк: http://t.co/07GyoDMWMp

Вести

25. фебруар 2015.

Гугл претвара flash у *HTML5*

Гугл ће аутоматски претварати Флеш (енг. *Flash*) рекламе у *HTML5*.

Користан линк: http://t.co/JdbzauqBPR

25. фебруар 2015.

Rails Girls meetup у Хамбургу

Овај догађај је намењен девојкама које желе да науче програмирање у овом програмском језику. У Хамбургу је одржан 25. фебруара ове године.

Користан линк: http://t.co/6PHFuJSHVu

1. март 2015.

ЛУГоНС окупљања

ЛУГоНС сваког четвртка организује окупљање у Стинг кафеу у Новом Саду.

Користан линк: https://t.co/BilW0aHGIn

2. март 2015.

Почела пријава предавања за *BalCCon*

Почела је пријава радова за овогодишњ BalCCon2k15.

Користан линк: https://t.co/9N63aIbIFW

---

12. март 2015.

Гугл најавио гашење *Google Code*-а

Гугл је најавио да ће у јануару наредне године угасити сервис Гугле кôд (нег. *Google Code*). Креирање нових пројеката је већ искључено.

Користан линк: http://t.co/H4QA4ry5p7

---

20. март 2015.

ГНУ манифест пуни 30 година

У марту 1985. године Ричард Стелман је објавио ГНУ манифест. Овај документ пропагира употребу рачунара слободно без коришћења власничког софтвера. Он је кључан документ за развој слободног софтвера.

Користан линк: http://j.mp/1HP8wG2

Пулс слободе

LUGoNS barcamp №4

Аутор: Никола Харди

ЛУГоНС (удружење корисника Линукса у Новом Саду) је у суботу 4. марта 2015. године одржао четврти по реду Баркамп (*BarCamp*). Као и у претходних неколико пута, и овај Баркамп је одржан у просторијама Факултета техничких наука. Предавања и радионице су почеле нешто после 12 сати и трајале су до увече, након чега је дружење традиционално настављено у опуштенијој атмосфери у кафићу.

barcamp №4

Као што смо навикли на претходним Баркамповима, предавања и дискусије су се дотицале разних тема. Биле су заступљене теме о аутоматизацији послова на рачунару, хакерским играчкама, о безбедности и анонимности. И овог пута била су организована предавања и радионице о слободној карти света - Опен стрит мап (енг. *Open Street Map*) и Мапилери (енг. *Mapillary*). Причало се о музичкој продукцији помоћу слободног софтвера, ослушкивању радио таласа и стању безбедности домаћих сајтова. Могли сте да чујете и неке врло занимљиве идеје о вештачкој интелигенцији и узгајању вештачког живота помоћу рачунара. Осим тога, зарад експеримента је једно време била постављена и огледна *GSM* мрежа на коју су посетиоци могли да се прикључе својим мобилним телефонима.

Пулс слободе

Предавања су и овај пут била снимана и надамо се да ће видео материјал убрзо бити доступан на Лугонсовим серверима. На адреси ftp.lugons.org су већ доступна предавања са претходних догађаја које је приредио ЛУГоНС, укључујући и Баркампове. Комплетан списак предавања доступан је на следећој адреси: https://events.lugons.org/?p=170 .

Овај Баркамп је био посебан и по томе што је ЛиБРЕ! тим припремио штампано издање нашег часописа. Интересовање је било знатно изнад наших очекивања. Заинтересовани су могли добровољним прилогом да подрже ову иницијативу и можемо да се похвалимо да је сто на којем су примерци часописа били постављени убрзо остао празан. За овај догађај припремили смо двадесет примерака, а захваљујући добровољним прилозима, добили смо могућност да за следећи сличан догађај припремимо двоструко више примерака. Надамо се да ће ова акција заживети и да ћемо убрзо моћи да припремимо колико год примерака буде било потребно.

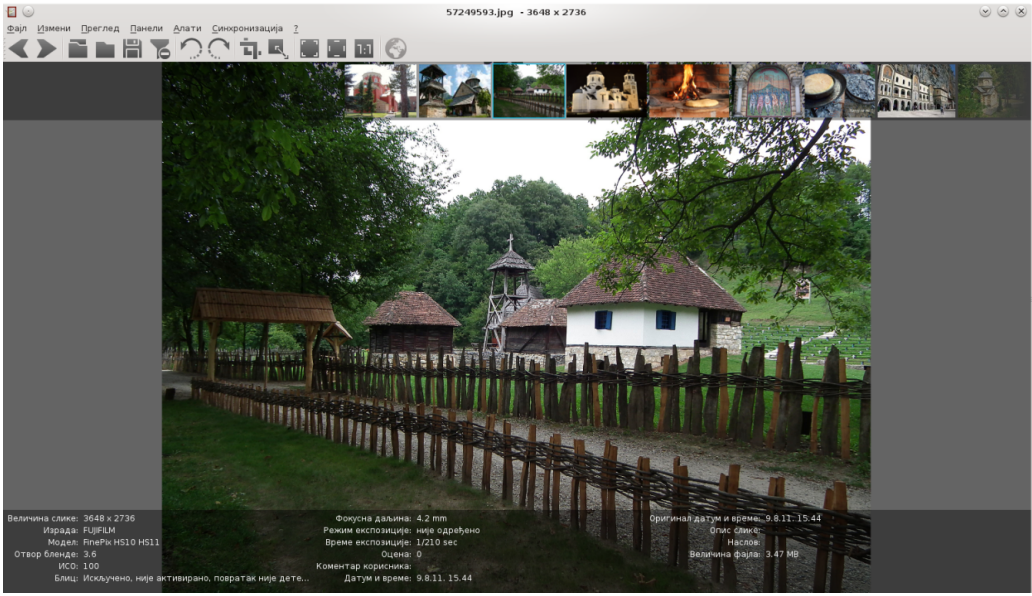
Посећеност је и овог пута била врло добра. Велика учионица је била пуна како добро познатих сталних посетилаца, тако и нових лица. Тражила се још по која слободна утичница за струју.

Најаве дешавања у организацији ЛУГоНС-а можете пратити на адреси https://events.lugons.org

Nomacs - Image lounge

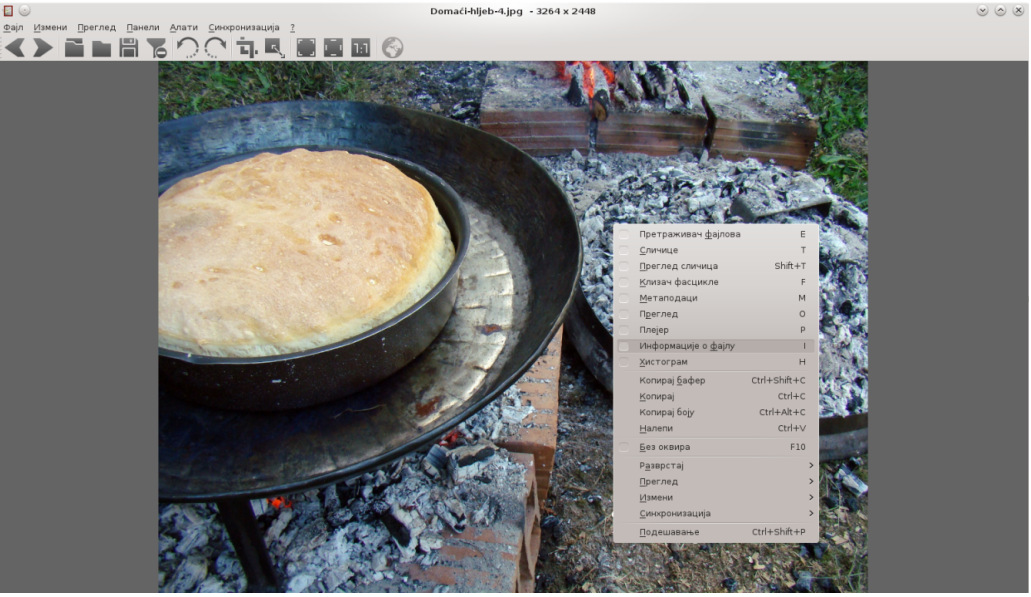
Аутор: Слободан Николић

Номакс (*Nomacs*) је још једна могућа алтернатива којом можете на свом рачунару да обављате преглед и основне измене фотографија. Апликација је доступна за Виндоуз, Линукс, ФриБСД, Мек и ОС/2, а пројекат са комплетним именом Номакс - Имиџ Лаунџ (*Nomacs - Image Lounge*) води се под *GNU GPL* лиценцом. Ради се о програму који је заснован на *Qt* библиотекама и који може, пре свега, послужити као потенцијални избор за кориснике КДЕ графичког окружења. Номакс доноси подршку за двадесет шест најпознатијих графичких формата, међу којима су и *PAW* и ПСД. У подешавањима програма тренутно су доступни преводи за шеснаест језика, међу којима је и српски.



Представљамо

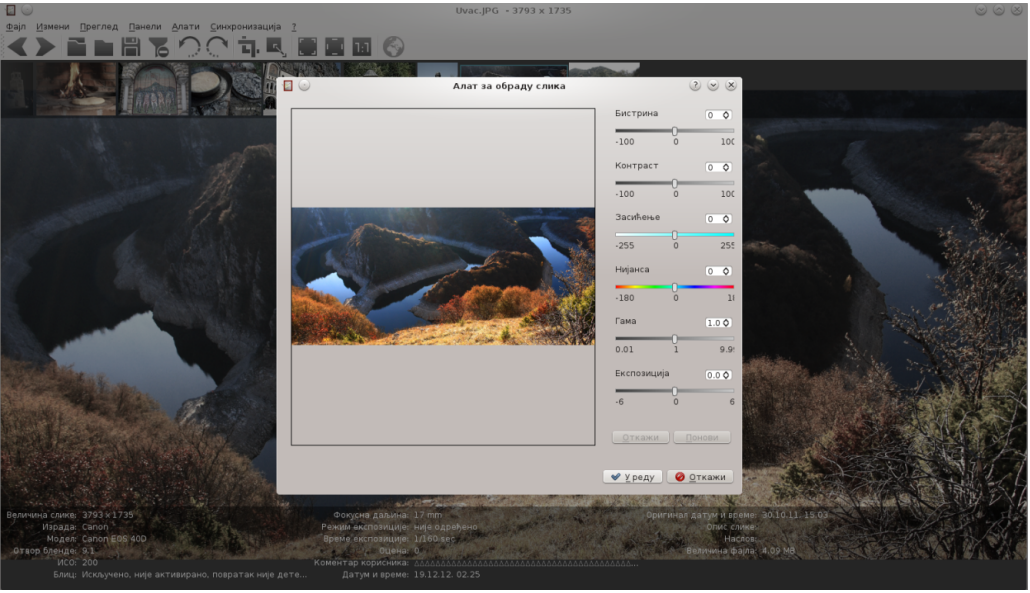
Када се покрене Номакс, први утисак може бити да се ради о производу који је сиромашан опцијама за квалитетан преглед фотографија. Ствар се мења одмах ако у менију Панели означите опције као што су Претраживач датотека, Сличице и Метаподаци. То исто може да се добије кликом на тастерске пречице (E,T,M) и програм ће добити изглед уобичајеног прегледача за слике. На исти начин, могу се додавати опције и када актуелну слику пребацимо у режим пуног екрана, а он се добија дуплим кликом или притиском на тастер *F11*. На истом месту (Панели) одређује се видљивост менија, траке алата и статусне траке, а доступне су опције као што су плејер, инфо о тренутној датотеци, напомене и хистограм. Прегледање слика које нуди Номакс је једноставно и природно, а на кориснику остаје да се навикне на употребу неколико тастерских пречица. За навигацију је довољно користити стрелице, док се зумирање слика може вршити на два начина - стрелицама, али и точкићем миша, претходно означавајући ову опцију у подешавањима програма. Номакс поседује опције за опсецање (тастер *C*), промену величине (*Ctrl+T*), као и алат за обраду слика који нуди подешавање елемената као што су: бистрина, контраст, засићење, нијансе, гама и експозиција. Преко општих подешавања програм можете визуелно прилагодити својим навикама. Поред одређивања видљивости за траке алата и менија, могу се подесити и боје за позадину, истицање, пун екран, виџете и иконе. Последња

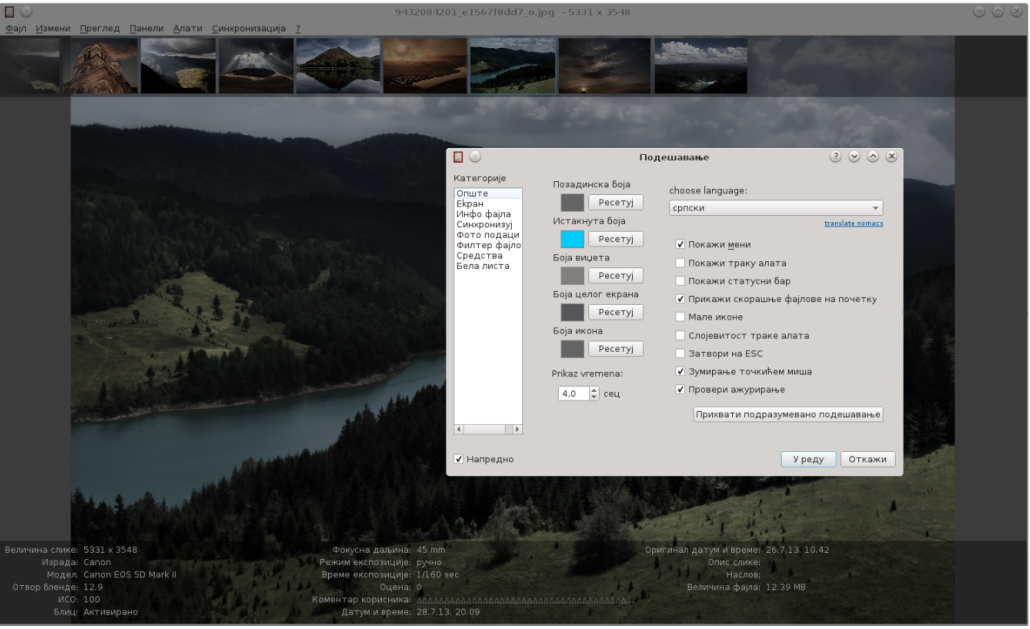


верзија програма (са ознаком 2.2.) донела је нове функције као што је читање слика које су спаковане као *zip* архива, или се налазе у канцеларијском документу (dosx, xlsx, pptx). Од новости ту су још могућност постављања панела са сличицама на сва четири положаја, приватни режим који не чува недавне датотеке, као и могућност да се уз слику дода напомена. Номакс је доступан за преузимање у званичним складиштима дистрибуција Федора, Арч Линукс и опенСУСЕ, док се за инсталацију на Убунтуу и Минту користи ризница: https://launchpad.net/~nomacs/ . За инсталацију на Дебијану, корисници могу употребити пакете дистрибуције *Siduction*. http://packages.siduction.org/extra/pool/main/n/nomacs/

Пошто немамо много програма за преглед фотографија који су засновани на *Qt* библиотекама, КДЕ корисници не треба да пропусте прилику да провере како изгледа тренутна алтернативна понуда. Корисници којима одговарају брзе и компактне апликације могу слободно да испробају Номакс. Ако треба да се издвоји нека предност коју Номакс поседује у односу на Гвенвју (*Gwenview*), онда су то свакако брже покретање али и добра опција зумирања точкићем миша. Преглед и навигација слика обављају се лако и једноставно, поготово ако смо

Представљамо





навикли да се потпомажемо коришћењем тастерских пречица. Сам изглед апликације је „чист и интуитиван”, а од навика самог корисника зависи који ће виџети бити присутни у оквиру главног прозора. Функције основних измена на фотографијама које Номакс поседује заокружују целину која се очекује од програма овакве намене, па верујемо да ће многим корисницима представљати одговарајуће решење.

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

Distrowatch

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

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

Како да...?

Увод у програмски језик *C*

Увод у рад са датотекама (10. део)

Аутор: Никола Харди

У претходном делу серијала, започето је разматрање рада са текстом у програмском језику *C*. Надамо се да смо успели да представимо значај ове теме. У овом наставку серијала биће представљен рад са датотекама. Ток рада класичног програма се углавном дели на три фазе: учитавање података, обрада, испис резултата. У сличним текстовима углавном је акценат на самој обради. Унос података и чување резултата су једнако важне теме.

Приступање датотекама

Први корак у раду са датотекама је „отварање" датотеке. Шта то заправо значи? Без превише приче о оперативним системима, стандардној библиотеци и спрези између корисничких програма и оперативног система, цела идеја се своди на то да програм од оперативног система тражи приступ некој датотеци путем механизма системских позива. Оперативни систем потом програму обезбеђује апстрактну представу те датотеке у виду тока бајтова, без потребе да кориснички програм и програмер познају систем датотека (енг. *filesystem*) или уређај о којем је реч.

Датотеци је могуће приступити на више нивоа, позивом функције **open()** која се директно пресликава у системски позив **open**. Позивом ове функције, оперативни систем корисничком програму обезбеђује дескриптор датотеке (енг. *file descriptor*) који је у суштини евиденциони број отворене датотеке на нивоу система. Дакле, реч је о обичној целобројној вредности. Рад са датотекама овим механизмом се сматра радом на ниском нивоу, а стандардна библиотека језика *C* обезбеђује и друге механизме који се такође ослањају на системски позив **open**, са још неколико додатака.

Други начин, који и препоручујемо, је помоћу структуре **FILE** и функције **fopen()**. Разлог за то је што употребом ове уграђене структуре података имамо могућност да користимо и друге стандардне функције за рад са датотекама који ће бити описани.

Датотеке могу бити отворене у више режима као што су режим за читање, режим за писање, режим за додавање садржаја и комбинација претходних. Понекад се у литератури срећу и ознаке за рад са бинарним датотекама, али оне се на већини модерних система једноставно игноришу.

Следи пример отварања датотеке на оба претходно описана начина, а потом и објашњење о режимима у којима је датотека отворена.

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

   return 0;
}
```

Као што можете да видите, ово изгледа мало незграпно. Функција **open()** има много намена, па је због тога њена употреба и сложенија. Први аргумент је назив (тачније, путања до датотеке). Следећи параметар су подешавања за режим отварања датотеке (само читање и креирање датотеке уколико она већ не постоји). Трећи параметар су подешавања за дозволе датотека (енг. *permissions*), која у о овом случају подразумева да само тренутни корисник има право писања и читања. Следећи израз је исписивање дескриптора датотека који би требао да буде позитиван број уколико је отварање датотеке било успешно. Након тога, врши се упис текста **“LiBRE!\n”**, чија је дужина 7 у датотеци са дескриптором датотека **fd**. По завршетку рада са датотеком, добро је затворити је позивом функције **close()**, јер тек тада можемо бити сигурни да ће оперативни систем записати сав садржај на диск. У библиотеци **<fcntl.h>** су дефинисани симболи

који су коришћени за подешавање режима и креирање датотеке.

Више детаља је доступно у **man** страницама којима је могуће приступити командом у терминалу: **man 2 open**. Друге корисне *man* странице за рад са датотекама су **write**, **read**, као и странице за остале функције које ће бити описане. Функције на вишем нивоу апстракције се налазе у трећем делу приручника, дакле: **man 3 fscanf**.

Нешто угоднији начин рада са датотекама је помоћу функција **fopen()**, **fwrite()**, **fread()** итд. Они пружају виши ниво апстракције и потребно је мање кода, а и пажње да би све прорадило како треба. Важно је познавати и механизме на које се те функције наслањају, а то су управо поменути системски позиви.

Следи пример кода у којем се приступа датотеци креираној у претходном примеру.

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

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

    puts(buffer);

    return 0;
}
```

Овај кôд је знатно читљивији. Нема криптичних параметара ако ништа друго. Функција **fopen()** има за повратну вредност адресу креиране *FILE* структуре, а уколико отварање датотеке из неког разлога није било успешно, та адреса је **0**. Пожељно је проверити да ли је отварање биле успешно. Параметри ове функције су путања до датотеке и режим приступања. Постојећи режими су:

- **r** - само читање
- **r+** - читање и писање од почетка датотеке
- **w** - само писање

- **w+** - писање и читање, ако датотека постоји брише се
- **a** - писање на крај датотеке
- **a+** - писање на крај и читање

Режими **w** и **a** ће креирати нову датотеку уколико она већ не постоји.

Опширнија документација је доступна у *man* страницама.

Остале функције

Стандардна библиотека садржи много функција које рад са датотекама чине једноставнијим и угоднијим. Међу њима су читање једног карактера, читање стринга, читање линије, читање задатог броја бајтова итд. Све те функције имају и своје парове за писање података. Називи ових функција почињу словом **f**, а вама препуштамо да погодите њихове називе и потражите детаље у *man* страницама.

Рад са датотекама не подразумева и рад са текстом. Могуће је чувати и тзв. бинарне податке, као што су структуре, фотографије и сл. Све што је потребно је одредити колико бајтова је потребно учитати или записати, задати са којом датотеком се ради и где сачувати резултат, односно одакле прочитати податке за упис. Величину структуре је могуће сазнати применом оператора **sizeof** или је једноставно израчунати унапред.

Код рада са датотекама, као и у многим другим ситуацијама, јављају се грешке: нема довољно простора на диску, програм нема дозволу за приступање задатој датотеци, у датотеци не постоје задати подаци итд. Због тога је важно проверавати повратне вредности функција и протумачити њихово значење на основу садржаја *man* страница.

Неколико речи о сепараторима

Када је реч о раду са текстуалном датотеком, јавља се концепт реда, односно линије текста. Пошто су датотеке само низови дигиталних података, рачунари не познају концепт реда текста. Редови су апстракција позната људима, односно корисницима рачунара, и због тога су вештачки уведени појединим стандардима. Стандард *ASCII* прописује вредности за више специјалних знакова, као што су, између осталог, нов ред, повратак главе штампача на почетак реда, укључивање

звона на знаковном терминалу или штампачу итд. Кроз историју су се јавила два стандарда за означавање краја реда текста, међу програмерима позната као *DOS* и Јуникс (енг. *Unix*) стандарди.

У *DOS* и Виндоуз свету се крај реда означава двама карактерима - **\r** и **\n**. Ова конвенција је присутна из историјских разлога и била је присутна у доба линијских штампача којима је задавана експлицитна наредба за прелазак у нов ред и враћање главе на почетак реда. У Јуникс свету је то само један знак, **\n**. Због тога су могући проблеми при размени датотека са колегама које користе другачије оперативне системе. Симптом је једноставан - садржај целе датотеке је у једном реду или постоје чудни знакови на крају сваког реда. На Линуксу су доступна два једноставна програма која решавају ову збрку, а то су *dos2unix* и *unix2dos*. Програми за уређивање текста, такође, могу да буду подешени за рад у једном или другом режиму. Обратите пажњу на овај проблем јер је присутан и дан-данас.

Позиционирање унутар датотеке

Често постоји потреба да се програм позиционира на другу локацију у датотеци. Овај механизам је присутан још из доба када су датотеке биле смештане на магнетним тракама, па према томе и функције имају сличан назив, односно **fseek()** и **rewind()**, што би на нашем језику значило „премотати”. Премотавање може да буде унапред или уназад. Референтне тачке могу да буду од почетка датотеке, од тренутне позиције, или од краја датотеке. Постоји и функција **ftell()** која говори тренутну позицију унутар датотеке. Функција **rewind()** „премотава” датотеку на њен почетак.

Не измишљајте топлу воду

Модерни информациони системи, базе података, па и рачунарска техника донекле, потекли су једним делом из миљеа корпорација које су се бавиле развојем машина за рад са досијеима. Реч датотека (енг. *file*) је првенствено представљала досије у картотеци или сличној служби, а посао рачунара је једним делом био да брзо пронађе или похрани податке у одговарајући досије. Од коверти са досијеима прешли смо на концепт датотека, а у разним фазама развоја рачунарских наука дошли смо до концепта база података. Теорија база података и информационих система нису тема ни овог текста, а ни часописа; али важно је напоменути да не треба измишљати топлу воду.

Уколико постоји потреба за чувањем табеларних података, стандардни начин је *CSV* (енг. *Comma Separated Value*), односно вредности раздвојене запетама. У стварности, вредности могу да буду раздвојени и другачијим знацима (енгл. *delimiterima*), размацима, звездицама, усправним цртама итд. Иако је за *C* честа пракса да се кôд за учитавање оваквих података пише у неколико линија кода, сваки пут - многи други језици и окружења за програмирање нуде аутоматизоване алате и библиотеке за овај посао. Саветујемо да се тај формат испоштује.

Уколико постоји потреба за похрањивањем већег броја података у више датотека, а подаци су притом и увезани, онда је то посао већ за једну базу података. Системи за управљање базама података са једне стране програмеру пружају спрегу за приступ подацима на апстрактном нивоу, а са друге стране сами воде рачуна како ће то сместити на диск. Уз то, пружају и сјајне механизме за брзу претрагу, ефикасно смештање података и прибављање одговора на упите. Постоје релационе и нерелационе базе података, оне за које су потребни сервери и оне које раде у меморији. *SQLite* је библиотека и концепт базе података за коју није потребно подешавати сервер, креирати корисничке налоге и инстанцирати појединачне базе. Базе се једноставно чувају у једној датотеци која је организована на специфичан начин. Вреди погледати, а свакако ће о овој теми бити речи у неком од наредних бројева.

Полиглоте и шизофреничари

Понекад, ипак, постоји потреба за дефинисањем новог типа датотеке. Тако су прописани формати датотека као што су **ZIP**, **PDF**, **ELF**, **JPEG** итд. Добро је питање и огроман проблем како правилно описати структуру једне датотеке, како препознати тип датотеке и проверити да ли је структура валидна и, још горе, како то преточити у функционалан програм. Познати су безбедносни проблеми који се базирају управо на овим проблемима. Злонамерни програми могу бити сакривени унутар друге датотеке, рецимо унутар *PDF* документа, или више програма може бити спојено унутар једне извршне датотеке. Отуда и назив овог одељка - полиглотама се сматрају датотеке које могу бити протумачене у више формата, рецимо *PDF* и фотографија. Шизофреничари су датотеке које свој садржај приказују другачије у зависности од различитих програма за прегледање - например, *Evince* приказује *PDF* са једним садржајем, а Окулар *PDF* са другим садржајем.

Овом темом се бави *Анге Албертини*. Он је на овогодишњем *CCC*-у (енг. *Chaos Communication Congress*) одржао предавање у којем је описао ове проблеме. Предавање је доступно на сајту http://media.ccc.de под називом „*Funky file formats*". Неке од ових техника су демонстриране у часопису *PoC||GTFO*, који је једноставно пронаћи на интернету. У њима су описани и детаљи о начину на који је могуће направити овакве датотеке. На сајту http://www.corkami.com постоје, између осталог, и постери који описују структуру неких познатих типова датотека. Предлажемо да погледате о чему је реч. Оформила се и посебна група истраживача који се донекле баве овом темом, а реч је о теми под називом *LANGSEC* (http://langsec.org/).

У наредном тексту

Наредним текстом ће овај серијал полако бити приведен крају. Обрађене су најважније теме овог програмског језика. Временом су текстови непланирано порасли, али не замерите. У наредном тексту (највероватније ће то бити више од једног текста), биће описане идеје које превазилазе сам језик, смернице шта занимљиво може да се направи, корисне библиотеке и други важни концепти. Спремни сте да направите и свој први програм који ћете моћи да користите свакодневно. Јавите нам се уколико желите да обрадимо неку тему детаљније или да додамо још који текст у серијал.

VAGRANT

(1. део)

Аутор: Иван Радељић и Стефан Ножинић

Увод

Када развијате апликације, веома је битно добро подесити своје развојно окружење. Свако од нас има своје навике, своје текстуалне уређиваче (едиторе) и остали софтвер који је потребан током процеса израде апликација. С друге стране, свака од апликација има своје зависности које такође треба ускладити. Понекад је потребно развијати више различитих апликација у исто време, па све то синхронизовати са остатком развојног тима. Овакве ситуације понекад могу да се искомпликују и да проузрокују да апликација или један њен део код једног члана тима функционише без проблема, док се код других појављују одређени проблеми. Да бисмо све то избегли, право решење је да користимо Вагрант (енгл. *Vagrant*).

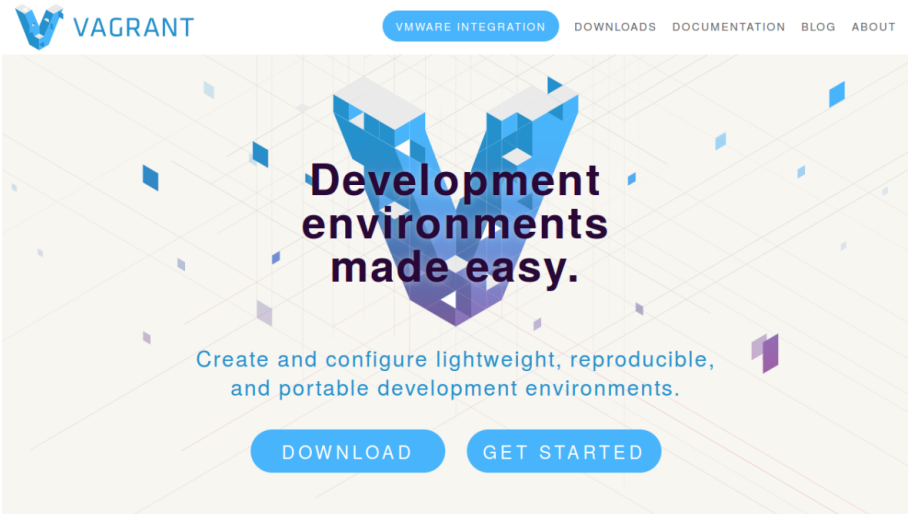
Зашто Вагрант?

Вагрант повећава продуктивност развојног тима. То је алат којим креирамо лагана, преносива развојна окружења која се изнова могу поново користити. То значи да нас Вагрант ослобађа свих подешавања и да само у једној конфигурационој датотеци бирамо која својства ћемо користити. Вагрант користи водеће провајдере за виртуелизацију и може бити снабдевен на Виртуалбоксу (енгл. *VirtualBox*), ВМВеру (енгл. *VMWare*), *AWS*-у и другима. За инсталацију потребних зависности и софтверских пакета, као и за њихово аутоматско

подешавање, користимо шел (*shell*) скрипте, Чеф (енгл. *Chef*), Папет (енгл. *Puppet*) или Ансибл (енгл. *Ansible*).

Неке од предности овог софтверског алата су:

- Вагрант чини идентично развојно окружење свим члановима тима, а да при том не жртвује ништа од алата у којима су програмери навикли да раде (то подразумева текстуалне уређиваче, ИДЕ[1], браузере, па и сам оперативни систем).
- Омогућава програмерима да се фокусирају на сам код, а не на конфигурацију развојног окружења.
- Конфигурација се обавља кроз једну конфигурациону датотеку и кроз додатне скрипте за аутоматско подешавање пакета и инсталацију истих.
- Конфигурационе датотеке се могу (и треба) дистрибуирати у самом репозиторијуму контроле верзија, као што је Гит (енгл. *Git*).



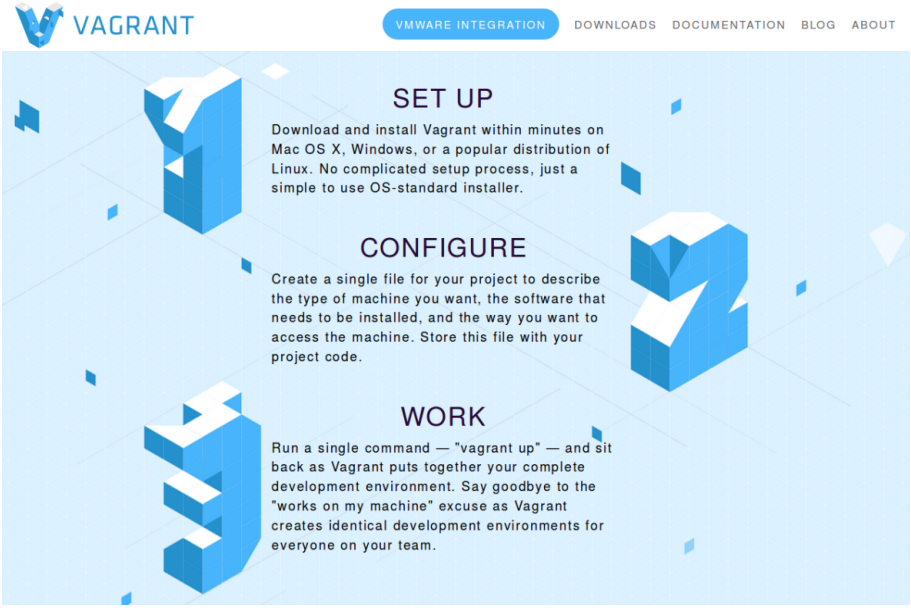
[1] ИДЕ (енг. *Integrated Development Environment*) - интегрисано развојно окружење.

Vagrant

Ако сте један од операционих инжењера, Вагрант вам омогућава једно лако заменско окружење за тестирање, где можете развијати и тестирати своје скрипте које после можете приметити на серверима у продукцији.

Ако сте дизајнер, са Вагрантом такође добијате све већ унапред дефинисано (подешено) и ваше је само да се фокусирате на дизајн. Неће вам више бити потребан програмер да вам помогне да покренете апликацију на вашем развојном окружењу како бисте могли да извршите потребне промене у дизајну.

Једна од највећих предности Ваграната - ако вам у било ком степену развоја апликације затреба још нека зависност, или сте заборавили да је на почетку додате, довољно је да Ваграном измените конфигурациону датотеку (*vagrantfile*) и сви у тиму ће имати идентичну верзију софтвера без обзира на њихово локално окружење, јер се сва конфигурација налази на систему за контролу верзија.



Ослобађање

Дистрибуирање слободног софтвера

Аутор: Никола Харди

Повод за писање чланка

Сви смо се ми, као нови корисници слободног оперативног система нашли у ситуацији да нам је неко на форуму, чету или неком сличном месту предложио да инсталирамо поједине пакете како бисмо решили своје проблеме. Још чешће смо се нашли у ситуацији да и сами закључимо да треба да инсталирамо додатни софтвер на наш систем. Неискусни корисници, који су до тада углавном користили Мајкрософт Виндоуз оперативне системе, навикли су се да упутства за инсталацију и софтвер траже широм интернета, али у свету слободног софтвера то функционише мало другачије. Строго и изричито вам саветујемо да софтвер инсталирате искључиво из званичних ризница, путем софтверског центра или пакет менаџера. Друге могућности које се некада могу толерисати су инсталација помоћу одговарајућих пакета (*.deb, .rpm*) или додавањем ризница (рецимо *PPA* у случају Убунтуа). Последња могућност којој можете да прибегнете је инсталирање из изворног кода. Софтвер инсталирајте из званичних ризница или репозиторијума (енг. *software repository*), осим када имате потребе за специфичном верзијом неког парчета софтвера и знате шта тачно радите и коју одговорност тиме преузимате.

Ризнице софтвера и остале чудне речи

Када се софтвер инсталира помоћу неког управљача пакета (*apt, yum, pacman, yaourt* и *emerge*), преузима се из званичних ризница. Ризнице су места на интернету где су смештени софтверски пакети који су проверени, тестирани и који ће сигурно радити у вашем систему. Дакле, процес инсталирања на примеру Убунтуа изгледа овако:

- Софтверским центром или другим алатом се извршава претрага пакета који су доступни у ризницама;
- Изабрани пакет се преузима из ризнице, укључујући све додатне пакете који су потребни за његов рад;
- Врши се провера да ли су сви пакети исправно преузети, како би се избегло инсталирање неисправних пакета;
- Врши се провера аутентичности преузетих пакета, проверава се да ли је одговорно лице потписало пакет који је преузет;
- Инсталирају се све „зависности” (пакети који су неопходни за рад) и жељени пакет;
- Сви пакети се додају на листу инсталираних пакета како би управник пакетима могао да води рачуна о њиховим верзијама и изврши каснију надоградњу.

У претходним редовима је употребљено неколико термина као што су софтверски центар, пакет, потпис, управник пакетима и други, које је можда потребно разјаснити.

Ослобађање

Софтверски центар углавном представља графички алат за претрагу и управљање софтвером на систему, односно само графички интерфејс који у позадини позива управник пакетима.

Управник пакетима (енг. *package manager*) је најчешће конзолни алат или скуп алата за претрагу, инсталирање, надоградњу и уклањање пакета.

Пакет је скуп датотека неопходних за инсталирање и исправан рад софтвера.

Потписи и контролне суме су део пакета за проверу интегритета и аутентичности преузетог садржаја.

Зависност је термин који указује да један пакет не може да буде инсталиран пре другог пакета јер се ослања на његове функционалности. Под термином зависности подразумевају се све везе које могу бити замршене, цикличне, рекурзивне или незгодне на неки други начин.

Шта су тачно пакети?

Пакет је архива која садржи софтвер преведен у извршни облик који одговара вашем систему. Поред тога, пакети садрже информације о свом интегритету, попут контролних сума и дигиталних потписа. Унутар пакета се могу налазити и упутства, информације о аутору пакета, аутору програма и помоћни алати и додатне процедуре који су неопходни за исправан рад софтвера који тај пакет садржи. Слободни оперативни системи се на основу дистрибуирања софтверских пакета могу поделити на „бинарне" (енг. *binary based*) и „базиране на изворном коду" (енг. *source based*). Код бинарних дистрибуција се унутар пакета налазе већ преведене датотеке у извршном облику, које се потом само смештају на одговарајуће место у систему и одмах могу да се користе. Код дистрибуција базираних на изворном коду као што је Џенту (енг. *Gentoo*), у пакету се налази локација одакле изворни код треба да буде преузет као и скрипта (списак наредби) којим се тај изворни код преводи у извршни облик и потом инсталира. Управник пакета уме да преузме и провери преузете податке, а потом и да изврши све потребне кораке како би софтвер био правилно инсталиран.

Једноставније инсталирање

У доба када су прави хакери ноћима инсталирали нов софтвер и сами писали

управљачке програме (енг. *driver*) за своје уређаје, софтвер је готово увек био инсталиран из изворног кода, без управника пакетима. Готово сви пакети и програми се ослањају на неке друге библиотеке, програме или пакете, из чега следи да је могуће формирати стабло зависности, а потом и списак и редослед свих пакета које је потребно инсталирати. Уверавамо вас да није тако лако погодити редослед инсталирања и одговарајуће верзије за све зависности. Зато управник пакетима то ради за вас.

Аутоматске надоградње

Велика предност овакве шеме дистрибуирања пакета је у томе што постоје ажурни спискови инсталираних пакета. Управљачи пакетима редовно преузимају спискове актуелних верзија пакета у ризницама и пореде верзије. На захтев или аутоматски, могуће је проверити да ли постоје надоградње за инсталиране пакете и преузети их. Сигурни смо да не желите да у црној свесци имате списак ручно инсталираних пакета и њихових верзија, а још мање да неколико пута месечно обиђете све сајтове и поредите најновије верзије са онима које имате заведене у свом списку.

Безбедност

Дигитално потписивање је процедура која је присутна и ван контекста дистрибуирања слободног софтвера. Реч је о процесу у којем се за одређене податке генерише одговарајући потпис помоћу кључа. Оно се базира на концепту асиметричног шифровања и постоји увек пар кога чине јавни и приватни кључ. Власник кључа на основу података у пакету и свог приватног кључа генерише одговарајући потпис. На основу преузетих података и јавног кључа може се проверити да ли преузети подаци (у овом случају пакет) одговарају потпису, тј. да ли је неко успут променио део података. Контролне суме проверавају да ли је садржај исправно преузет. Потписи гарантују и аутентичност података.

Устаљено је мишљење да су слободни оперативни системи генерално безбеднији од оних других. Међутим, ово није сасвим тачно јер сваки систем има своје рупице, бубице, гремлине и вирусе. Разлика је у томе што су слободни оперативни системи врло непријатно окружење за ширење таквих рачунарских болести. Механизми којима се софтвер дистрибуира и инсталира су кључни за то. Пре него што овај чланак буде завршен и пре него што поновимо шта све лоше може да се догоди ако не пазите како и одакле инсталирате софтвер, ево једног

теоријског примера када нешто пође по злу.

- Неискусни корисник жели да инсталира програм за видео разговоре;
- Претражује интернет у нади да ће пронаћи верзију за Линукс и у мрачном ћошку интернета проналази некакву верзију;
- Инсталира је и убрзо примети да његов рачунар више не ради како треба;
- Корисник закључује да је то дело злонамерног програма и да слободан софтвер и није тако безбедан.

Ево шта је заправо могло да се догоди.

Када је корисник преузео пакет са непознате локације, могао је да преузме верзију која је измењена тако да у себи има безбедносне пропусте разних врста (удаљено управљање рачунаром, преузимање података са рачунара и брисање датотека). Друга, блажа и више вероватна могућност је да је преузет исправан пакет, али ручном инсталацијом није додат на списак инсталираних пакета и због тога није био ажуриран. У међувремену је откривен безбедносни пропуст, а надоградња која решава проблем није инсталирана.

Имајте на уму

Пре него што се опет нађете у ситуацији да се питате како да инсталирате неки пакет, програм или било какав други део софтвера, имајте на уму да ће управник пакетима урадити следеће ствари уместо вас, као и ствари које би требало ви да урадите уместо управника пакетима.

- Проверити аутентичност локација одакле је пакет преузет;
- Контролном сумом проверити да ли је пакет исправно преузет;
- Проверити потпис пакета како би се искључила могућност да је неко пресрео преузимање и тако подметнуо злонамеран код;
- Разрешити потребне зависности (поновити претходне ставке за све зависности);
- Испратити процедуру за ручно инсталирање;
- Редовно проверавати обавештења о откривеним безбедносним пропустима и доступним надоградњама.

Једноставније је покренути софтверски центар или написати *sudo apt-get install*, зар не?

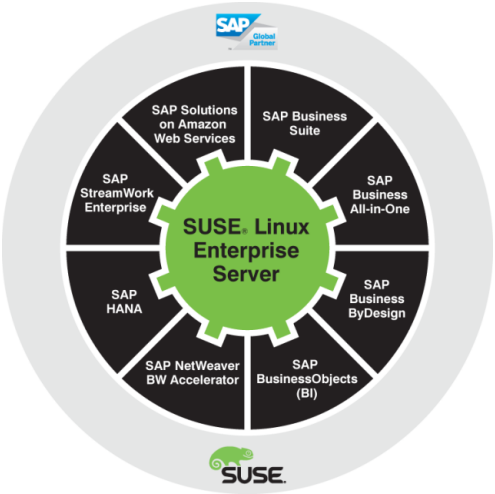
Аутор: Александар Весић

У Франкфурту је 20.01.2015. године одржан један од *SUSE Linux Expert Day* који се од Септембра 2014. године одражавају широм света. За место одржавања изабран је *25Hours Hotel Frankfurt by Levi's* као јако интересантна локација у самом центру Франкфурта. Сам догађај је трајао око пет сати и састојао се од низа занимљивих предавања.

Одмах након уводног говора који је одржао Михаел Јорес (*Michael Jores*), регионални директор *SUSE Central Europe*, директор Сусе Линукс Ентерпрајза (*SUSE Linux Enterprise*) Олаф Кирх (*Olaf Kirch*) представио је тренутно стање и планове везане за даљи развој Сусеових производа дајући нам увид у тренутне трендове развоја.

Након изласка СЛЕС-а 12 (*SUSE Linux Enterprise Server 12*) у октобру 2014. године, ове године (јуни/јули) излази и Сервис пек (*Service Pack*) за СЛЕС 11. Поред уобичајених ажурирања кернела долази и подршка за платформе *IBM z13, POWER8 BE* и *Intel Haswell EX*. Према тренутним плановима, то је заправо последњи Сервис пек, а СП5 тренутно није у плану. Напоменуто је да је САП (*Systems, Applications & Products* - Немачка корпорација) дао СЛЕС-у 12

Слободни професионалац



сертификат за коришћење ЕРП (*Enterprise resource planning*) софтверских компоненти. Званична потврда и одобрење од стране *SAP (Systems, Applications & Products*) се очекује у току овог квартала, тако да купци наведени производ могу да инсталирају и да им притом Сусе и САП пружају пуну подршку.

SUSE OpenStack Cloud

У првој половини ове године ће се појавити и Сусе Облак (*SUSE Cloud*) у петој верзији. Поред врло доступних виртуелних машина биће подржан и Докер-контејнер (*Docker-Container*). Требало би, између осталог, да буде могуће њима управљати путем будућих Сусе Менаџер (*SUSE Manager*) верзија. За трећи квартал 2015. године је предвиђено ажурирање Сусе Облака који би требало да буде базиран на Опен-стеку (*OpenStack*), верзија под именом „Кило”. Код ове верзије би контролни чворови (*Control Nodes*), које користи Опен-стек, могли да функционишу и на СЛЕС-у 12. За 2016. годину је планиран Сусе Облак 7 базиран на Опен-стеку 7.

SUSE Enterprise Storage

Што се тиче Сусе сервера за смештање података (*SUSE Storage Server*), тим производом фирма из Нирнберга жели да добије свој део колача у бизнису везаном за софтвер базиран на складиштењу података. Производ је намењен

првенствено клијентима којима су потребни приватни или хибридни Облаци (*hybrid Cloud*), и/или су у потрази за алтернативама у односу на конвенционална САН (*Storage Area Network*) решења. Основна компонента је Цеф (*Ceph*) који је познат по способностима везаним за повећање капацитета (*upscaling*), добро дуплицирање података и могућност исправљања грешака. Цеф препознаје неисправне дискове и пребацује се на резервне дискове. Три основна дела концепта Цеф су:

- Смештање објеката (*Object Storage*) – приступ путем *C* / *C* ++ - Јаве, Пајтона, ПХП-а, или *RESTful*, имплементира *Striping* и *Snapshot* функције
- Смештање блокова (*Block Storage*) – Смештање објеката као *thin-provisioned Block Storage* (нпр. за виртуелне машине)
- *File System* – *POSIX* компатибилан са директним приступом на *Object Storage*, интегрисан у Линукс кернел од 2010. (2.6.34), опционално је доступан *FUSE-Client*.

Током прошле године је одабраним корисницима био омогућен приступ бета верзији, а прва званична верзија производа појављује се у првом тромесечју ове године, која се темељи на верзији Цефа под називом Фајерфлај (*Firefly*). За трећи квартал 2015. године планирана је верзија 2.0, која као основу треба да користи Цеф „Хамер" (*Hammer*). Док као сервер овде само СЛЕС 12 долази у обзир, по питању клијента је СЛЕС 11 подржан.

За 2016. годину је планиран Сусе Менаџер 3 код кога поред подршке за СЛЕС 12 СП1 такође долазе и промене функција везане за високу доступност (*High Availability*) и надгледање (*Monitoring*). Остаје нам да видимо колико ће Сусе да преради Мониторинг функцију од Спејсвока (*Spacewalk*) након што је Ред Хет (*Red Hat*) објавио да ће се даље бавити његовим даљим развојем. За 2017. годину је планиран Сусе Менаџер 4.

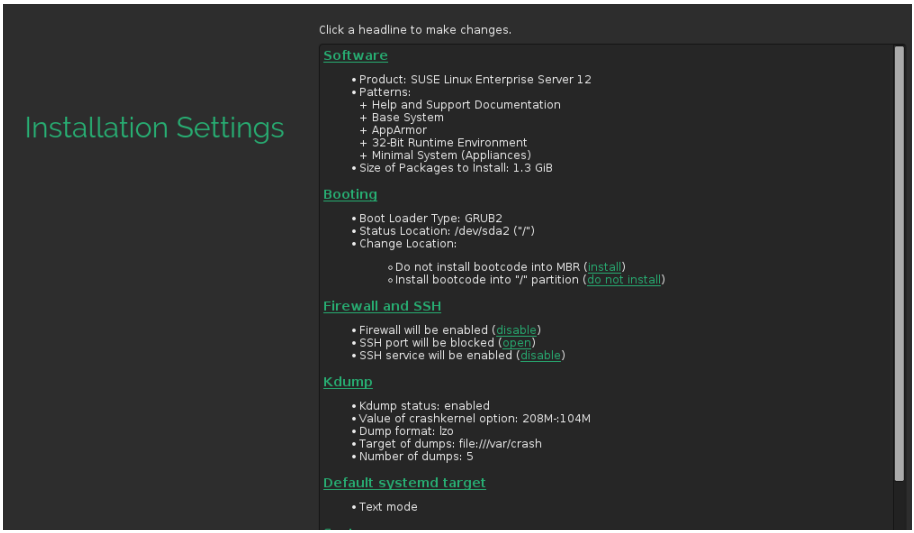
SysVinit vs Systemd

Објављен је после пет година чекања ново главно издање Сусе Линукс Ентерпрајза. У односу на претходну верзију извршене су велике промене, од којих је *Systemd* највећа и најконтроверзнија. Током прошле године је ово ажурирање, које је код многих дистрибуција заменило *SysVinit*, било одговорно за доста дискусија у заједници корисника Линукса. Тако је било и приликом овог семинара, где дискусија „*Systemd vs SysVinit*" није изостала. Без намере да

коментаришемо и да изнесемо мишљење, цитираћемо Олафа Кирха:

> „Сваких десет до петнаест година иста прича. Када је *SysVinit* заменио *RC* скрипте, свако је био ужаснут, иако није имао значајне недостатке. Данас никоме не недостају. Иста прича се понавља када је у питању *Systemd*.“

Systemd је радикална, али сасвим савремена промена која има многе предности. Као и свака друга техничка иновација, *Systemd* носи са собом потребу да се стекну нове вештине. Дистрибуције за фирме, као што су Сусе Линукс Ентерпрајз или Ред Хет Ентерпрајз Линукс пружају клијентима као опцију да користе познате алате (нпр. *service*, *chkconfig*, старе конфигурацијске датотеке) како би се олакшао прелаз. Не би било на одмет да се заједница корисника Линукса опходи ка тој теми са мало више отворености и толеранције.



* * *

У Сусе Линукс Ентерпрајзу су направљене две радикалне промене: *systemd* и укидање подршке за Интелову *i686* 32-битну архитектуру. Коришћење чистих 32-битних система је последњих неколико година у паду у односу на 64-битне алтернативе. СЛЕС 12 такође неће бити доступан за системе базиране на Интел Итанијуму (*Intel Itanium - ia64*) и тиме Сусе сам иде у корак са другим великим дистрибуцијама као што је Ред Хет, на пример. Између осталог, *Xen*, *KVM* (*Kernel-*

based Virtual Machine) и *LXC* (Линукс контејнери) дају нам три могућности за виртуализацију. За тзв. контејнер апликације је доступан Докер.

Btrfs (ButterFS) је нови стандарни фајл-систем код СЛЕС-а 12. Он ће у потпуности бити покривен од стране Сусе подршке (под условом да се користе стандардне опције за конфигурирање фајл-система путем *YaST*-а). Он такође нуди додатне могућности, попут Снепшатса (*Snapshots*). Тако Зипер (*Zypper*), на пример, прави Снепшат (снимак) пре ажурирања система - у случају да након ажурирања систем не може да се подигне, могуће је при старту у ГРУБ-у изабрати претходни Снепшат за подизање система. Предвиђено је и Сусе препоручује да се *BtrFS* користи за оперативни систем а *XFS* за корисничке податке (нпр. *MySQL* база података). У СЛЕС-у 12 *ext4* сада има и подршку за писање (*write*) за разлику од СЛЕС-а 11 који подржава само читање (*read*), јер *ext4* није било могуће довољно тестирати да би био одобрен за продуктивне системе. Додатно је наглашено да је *ext4* пуно лошији код асинхроних улазних-излазних (*I/O*) позива у поређењу са *XFS*-ом.

YasT

Централни алат за конфигурирање *YasT* је такође подмлађен, осим естетских промена он сада користи Руби (*Ruby*) уместо ранијег *YCP*-а. Сусе нам је обећао и лакше одржавање софтвера. Уз уграђен Викед (*Wicked*), *YasT* је добио нови *Network Backend* који се не фокусира само на *Client*-системе и треба да буде користан у хибридним Облацима. Препоручује се да се мрежна конфигурација нових система имплеметира директно путем Викеда – ранији начини конфигурације су још увек подржани. Поједини програмски пакети су сада доступни у модулима. Они ће бити подржавани само неколико година, уместо десет до тринаест година (што је до сада био обичај - прим.ур.). Тренутно доступни модули су:

- *Web and Scripting* : ПХП, Пајтон, *Ruby on Rails* (3 године подршке)
- *Legacy - Sendmail*, старе верзије Јаве, итд. (3 године подршке)
- *Public Cloud - Public Cloud* - програмски пакети (непрекидна интеграција)
- *Toolchain - GNU Compiler Collection* (по годишњем издању годину дана подршке)
- *Advanced System Management – Tools/Frameworks* за администрацију (непрекидна интеграција)

Сусе Машинерија

Програм Сусе Машинерија (*SUSE Machinery*) је техничка претпремијера сервиса за миграцију постојећих система. Машинерија анализира конфигурацију система, консолидује исту и мигрира понуђене сервисе. Основна намера је омогућити миграције са СЛЕС-а 11 на СЛЕС 12, као и за хибридне Облаке. Опоравак од катастрофа (*Disaster Recovery*) је побољшан могућностима које нуди Машинерија, нажалост примену овог програма још увек не подржава Сусеова подршка.

Измена језгра у лету

Мото овогодишњег *SUSE Linux Expert Day* била је „пут према непрекидности рада" (*Towards Zero Downtime*) којим је Сусе рекламирао функцију измене језгра у лету (*Kernel Live Patching*) код СЛЕС-а 12. Као и код Ред Хета ова се функција продаје као додатни производ у покушају да се што више купаца веже за напредну подршку (*Premium Support*). Закрпе за језгро су доступне као *RPM* датотеке које инсталирају модуле и обнављају *Initial ramdisks*. Позиве функција језгра преусмеравају ка новим модулима ф-трејс (*ftrace*) и К-графт (*kGraft*) - компонента коју развија Сусе. Програми, који су у функцији, не морају поново да се покрену. Тренутно су само *x86_64* платформе подржане, а у зависности од реакција купаца следиће подршка за остале архитектуре. К-графтом (*kGraft*) Сусе жели да буде конкурентан у односу на Ред Хет и њихов К-печ (*kPatch*) као и према Ораклу (*Oracle*) који је већ представио К-сплајс (*kSplice*), који такође користе модуле (отвореног кода) језгра за имплементирање измене у току рада (*Live Patching*). Сусе, за разлику од решења које користи Ред Хет, врши детаљне провере повезаности модула. Ред Хет је у новембру прошле године покренуо дискусију о томе да се дође до уједињења сродних решења, а главну реч треба да дâ развојна заједница.

Сусе Облак

У поређењу са конкуренцијом Сусе се поноси сертификованим подешавањима за хардвер и софтвер и подршком других Хипервизорса (*Hypervisors*) које не подржава Опен-стек. Сусе посебно добро подржава инсталације *vSphere*-а на *VMware*-у. На Сусеовој веб страници можете преузети шездесетодневну тест-верзију која у року од тридесет минута изврши потпуно подешавање вашег приватно Облака. У односу на ручну инсталацију Сусе Облак штеди много рада

око конфигурисања, што је јако интересантно купцима који још немају искуства са Опен-стеком (*OpenStack*). Као и Ред Хет, Сусе је такође платинасти члан који јако пуно помаже пројекту Опен-стек на разне начине а поготово финансијски.

Шифровани чет (1. део) - *Subrosa*

Аутор: Криптопанк

Данас за међусобно брзо дописивање користимо много сервиса и апликација, али да ли бринемо о својој приватности користећи их?

Иако смо кроз претходне бројеве научили како да заштитимо своју електронску пошту, то можда не покрива цео спектар наших комуникација и свакојаких размена информација и података преко интернета. Па ако је тако, хајде да видимо како можемо да се дописујемо са другима, брзо, лако и сигурно.

Проблем је следећи: „Хоћемо да се инстант дописујемо, односно да четујемо са једном или више особа у реалном времену на сигуран начин. То значи да желимо сигурни групни чет. Онда бисмо желели да имамо могућност размене података-фајлова или аудио-видео комуникацију. Већином за ово користимо централизоване сервисе попут Фејсбука, и/или оне које нису отвореног кода попут Скајпа (енг. *Skype*) или Вајбера (*Viber*), и не пружају скоро никакву приватност корисницима од администратора самог сервиса, па можда и државних безбедносних компанија."

Међутим, почетак није једноставан јер смо погрешили у старту. Не би требало да користимо апликације и сервисе који нису направљени са идејом о приватности самих корисника нити томе служе. Ако нам је приватност важна, онда треба да потражимо алтернативе. Па, да ли оне уопште постоје? Наравно да постоје, а понекад су чак и боље.

Шифровани чет

У овом (првом) делу серијала поменућемо уједно и најновију међу њима. Суброса је заправо централизована апликација/платформа отвореног кода (http://goo.gl/fAyHev) која може да ради у два мода: клијент и сервер.

Да бисте је користили, потребно је да се региструјете, тј. да направите корисничко име (енг. *username*) и лозинку (енг. *password*), опцију за мејл можете да прескочите или да једноставно упишете лажни мејл да бисте заварали траг. Ни Суброса не зна наше тајне кључеве нити чува лозинке, стога ако заборавите лозинку, можете се поздравити са тим налогом.

Начин функционисања ове чет платформе је следећи:

1. При регистрацији уносите корисничко име и шифру. Шифра не напушта ваш претраживач.
2. Унета шифра се потом користи за генерисање јавног/тајног пара асиметричних кључева (*RSA 2048 bits*).
3. Јавни кључ се шаље Субросиним серверима, а тајни кључ се шифрован чува на рачунару.

```
-----BEGIN RSA PRIVATE KEY-----
MIIBPAIBAAJBAIBDPhLynDZ7SnfLazZwsJT90OKcUvKxDghNHpO2IviH8iP4pEz0
olLqlK8TpmBAOb4A8VGvtwt/bC8jNbjK87kCAwEAAQJBAAhZ2HHNYARgFKImGW3q
+mRI/k+y0+e/RWWe982wPyilQTvyrrQToVVGkX5wfPQUeAiYE86v1rw2bOVs4tFb
J00CIQDc8m0cIrwg71o4sGK9UGSRgcH2S1a5Zbcm2UnRQXkCKwIhAJSchY2jkjzc
KU+uLa1VIJXAxN2zg+wQ3UM+lYXt0AOrAiEAs1dkEU5wfhtKlHE39IroluMIWa7V
ei+B9tqw/FW0r3kCIQCAZL5qAbD9jcTPR4/FGi/90b8EP+0xDdzsFOz/+ddm4wIg
P1Xbhz/V6KLgLRXvmtUXtbZJ6lg62NZwYA7+xe12nOU=
-----END RSA PRIVATE KEY-----
```

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsaCbY3Ly52qiW7wFYNss
5Vj7eDCbH/+uu/bIIj57dTpUXbsAwQRjM5R+6cSEduhMU/Kk1Et6/lPRC0K7q/Xf
doctrTsyYoXIuug79vXbdcYBiPaHgKE+q20lyfBZ+w7hVkkKPJxsRv+ctZYjX1O3
lWlajjBmz8oMTlb9byBmyKOzSkOEWxzFJI9DoIAt0uInXIK68hlVhMGkBzlMSowl
furKzV4Bj7QMWDLa4sWCBEVlfMN2busDuTJMRhxgPNrcbCgrcYmoR3zGDTqq9JZy
GXzTXXWhuvKXBoSLCB28ki9+URfDexjyj2DU3N/9kc0MLMIAkzL1dtUCtWhWaMIA
dwIDAQAB
-----END PUBLIC KEY-----
```

Пошто су кључеви направљени, комуникација може да почне. Све што је потребно је да пронађете особу са којом желите да дискутујете. Оно што се

потом дешава је да иницијатор комуникације добије од Субросиног сервера јавни кључ особе са којом жели да комуницира. Затим:

1. претраживач му генерише симетрични кључ (*AES-GCM-256*);
2. тај кључ се потом шифрује јавним кључем вашег будућег саговорника и затим се тако шифрован шаље њему;
3. саговорник дешифрује поруку, и надаље се тај кључ, који сада поседују обе стране, користи за шифровање и дешифровање порука, и уопште било ког типа података између две стране - тзв. енд-ту-енд (***end-to-end***) шифрована комуникација.

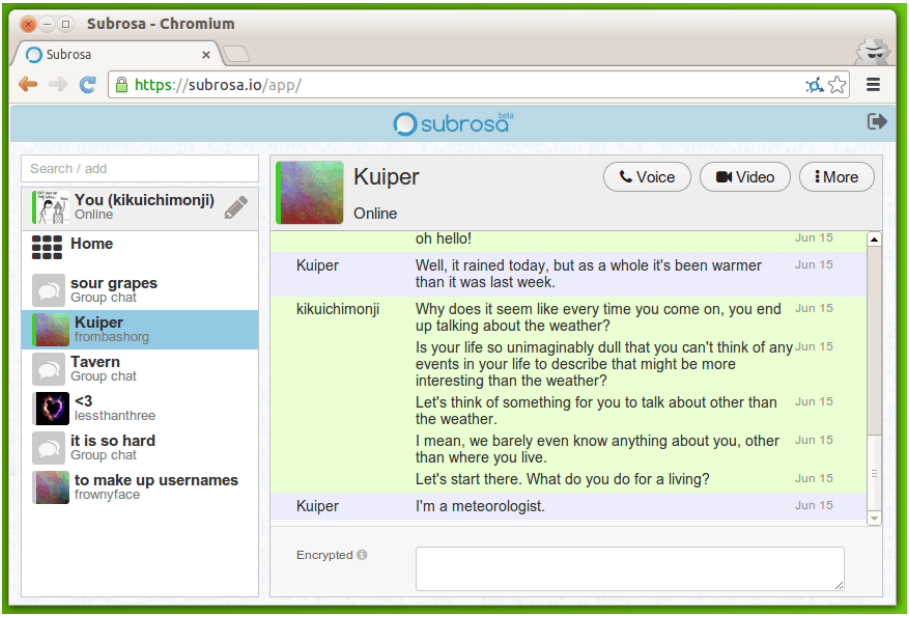
Суброса пружа могућност видео комуникације, аудио позива и дописивања (наравно, све је **енд-ту-енд** шифровано). Шифровање **енд-ту-енд** значи да када особе А и Б комуницирају, порука која је са једног краја (корисник А) шифрована, може бити дешифрована само на свом другом крају (корисник Б).

Суброса такође користи *HSTS* и *PFS*, што ,најпростије речено, значи да се за сваку комуникацију користи други пар кључева. Ово је само мера превенције против неких напада на шифровану комуникацију путем *SSL*-а (https/ /:) и односи се само на комуникацију са *Суброса* сервером, који, као што смо рекли, могу подићи и корисници на својим машинама.

Изворни кôд је проверен независно од стране немачке безбедносне компаније *Curve53*.

Суброса такође подржава и тзв. *warrant cannary*, што је заправо метод којим провајдер интернет-ских комуникационих услуга обавештава кориснике да није примио тајни налог или судски налог за достављање информација о својим корисницима властима у некој земљи (http://goo.gl/z5nSDG).

Дакле, да сумирамо: Суброса је отвореног кода, не захтева адресу мејла нити било какве личне податке; можете подићи сервер на свом рачунару; не чува



логове (барем тако тврде - прим.аут.); покреће се директно из претраживача и не захтева преузимање било каквог софтвера; подржава **HSTS** (енг. *HTTP Strict Transport Security*), **PFS** (енг. *Perfect Forward Secrecy*) и подржава видео, аудио и текстуалну шифровану комуникацију.

Један од потенцијалних проблема, сматрају неки, је коришћење Јаваскрипта и извршавање кода са удаљеног рачунара, пошто ништа није потребно да се инсталира на рачунар. То значи да је платформа потенцијално рањива као већина других на *XSS*, разне врсте инџекшна и *MITM* од стране сервера. Да би избегли ове нападе, (параноични) корисници такође могу да хостују и сервер који ће користити у својој комуникацији.

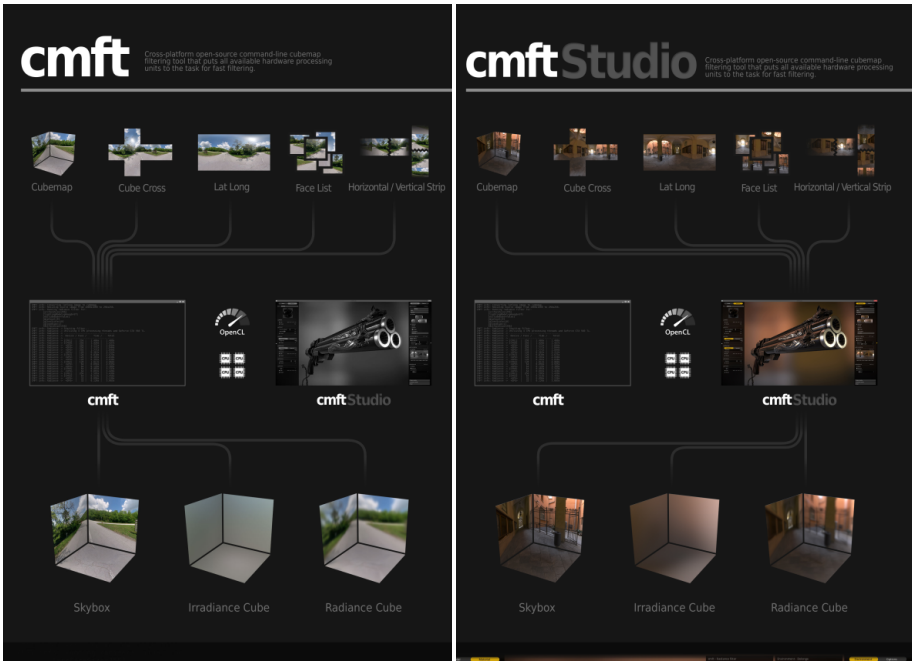
Могуће је да се анонимност и приватност још више унапреде коришћењем проксија или повезивањем на неку виртуелну приватну мрежу или на Tor.

Сам свој мајстор

Cmft и cmftStudio

Аутор: Дарио Манеску

Cmft је алат за филтрирање тзв. коцкастих-текстура (енг. *cubemap texture*) за потребе рачунарских игара и тродимензионалних апликација, а *cmftStudio* је алат који поред филтрирања нуди и интерактивну визуализацију и преглед добијених резултата.



Коцкасте-текстуре су врсте текстура, које имају доста велику примену у рачунарској графици у реалном времену, или, прецизније речено, у рачунарским играма.

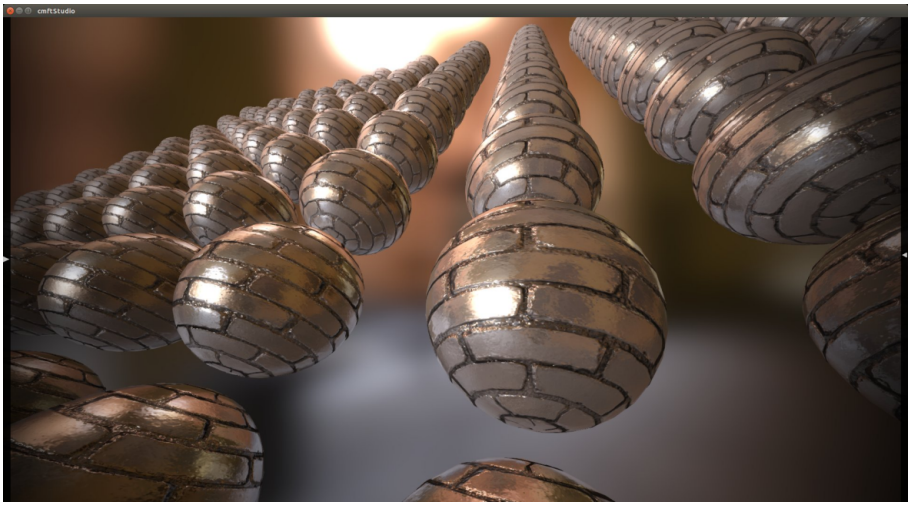
Шест страна у облику коцке чине једну коцкасту-текстуру, а њен узорак се добија путем тродимензионалних вектора (**x**,**y**,**z**) који означава правац од центра коцке. То је једна суштинска разлика у односу на дводимензионалне текстуре, где се узорак добија путем дводимензионалних координата (**x**,**y**).

Као што је лако закључити, коцкасте-текстуре обухватају цео простор око своје централне тачке и, као такве, погодне су за имплементацију разних техника у рачунарској графици.

Једна од конкретних примена коцкастих-текстура је имплементација тзв. *IBL* технике (енг. *Image-Based Lighting*) или осветљења заснованог на слици. Техника обухвата складиштење података о упадним светлосним зрацима са свих страна за жељену тачку у тродимензионалном простору и коришћење тих информација за израчунавање, тј. апроксимацију амбијенталне светлости у окружењу те тачке. Подаци о упадној светлости се најпре морају обрадити и упаковати у жељеном облику у коцкасте-текстуре. Тај облик треба бити погодан за обраду у реалном времену, и управо за то служе алати *cmft* и *cmftStudio*.

Сам свој мајстор

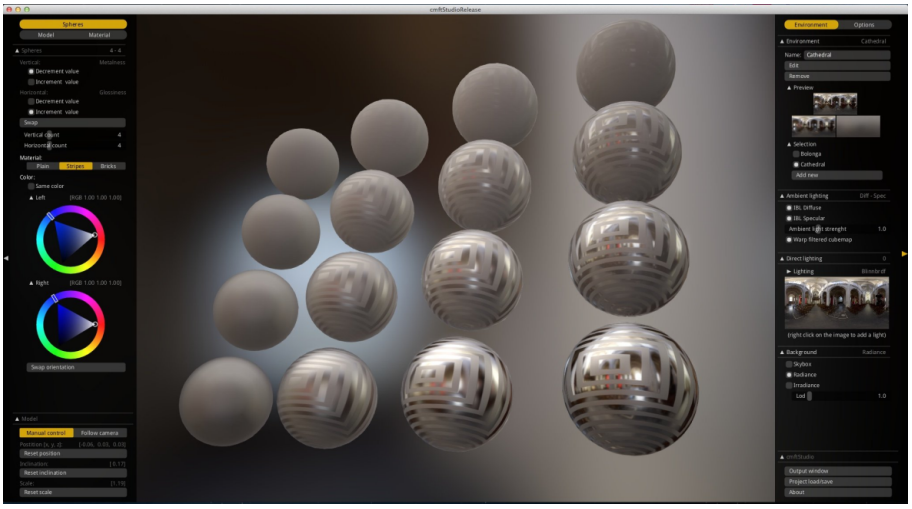
Ова техника је у широкој употреби у, може се рећи, скоро свакој данашњој модерној 3Д игри, јер су резултати који се добијају доста примамљиви, а брзина данашњег доступног хардвера без проблема омогућава примену ове технике у пракси.



Пре настанка *cmft*-а и *cmftStudio*-а, за ову намену људи су углавном користили познати АМД-ов алат Кјубмапџен (енг. *CubeMapGen*), који је био бесплатно доступан на интернету. Септембра 2011. године, АМД је прекинуо даљи развој овог алата и објавио је јавно његов изворни код. Алат је био доста добар - нудио је доста могућности за подешавање параметара за филтрирање. Међутим, имао је једну великуману - филтрирање коцкастих-текстура је често дугачак процес, а Кјубмапџен је вршио обраду искључиво на једном језгру процесора. Појавила се, такође, на интернету и модификација Кјубмапџена која је могла да врши обраду користећи три процесорска језгра. Међутим, то је и даље било доста споро. Филтрирање једне коцкасте-текстуре је могло трајати, у зависности од изабраних параметара, и до шест или седам минута, па чак и преко петнаест минута. Ми смо у то време користили наведени алат и експериментисали са разним подешавањима, и тај процес је трајао доста дуго - на тренутке је чак био и фрустрирајући. Да бисмо сазнали какви ће резултати бити са појединим подешавањима, били смо приморани да чекамо доста дуго, а притом нам је рачунар био оптерећен и једва смо могли њиме да радимо ишта друго.

Ту је настала идеја да покушамо да убрзамо цео тај процес. Нисмо се определили за даљу модификацију Кјубмапцена пошто је његов изворни кôд био уско везан за Дајрект-икс (енг. *DirectX*), тј. Виндоуз платформу, и то нам се није нимало допало. Одлука је била да почнем да правим алат из почетка, са идејом да:

- буде доступан за све актуелне платформе: Виндоуз, Линукс, Мек ОС Икс (енг. *Mac OS X*);
- користи сва доступна процесорска језгра и у исто време и графички процесор путем Опен-си-ел (енг. *OpenCL*) технологије, како би се постигла што већа брзина обраде података;
- има могућност позива из командне линије како би се са лакоћом могао позвати из неког скрипт језика и на тај начин креирао задатак за серијско обрађивање бројних коцкастих-текстура одједном;
- буде јавно доступан свима на интернету, заједно са изворним кодом, како би што више људи било у могућности да га користи за своје потребе.



Након неколико месеци, управо смо то и реализовали. *Cmft* алат се појавио на интернету и био је приметно бржи од Кјубмапцена. Оно што је некада трајало седам минута са Кјубмапценом, *cmft* би то завршио за око двадесетак секунди. Поред тога, имао је и могућност учитавања као улазних података много различитих типова слика окружења. Напокон, обрада коцкастих-текстура није

Сам свој мајстор

више била толико напоран процес. Међутим, пошто је у питању алат који се користи из командне линије, потребно је било документовати и објаснити његово коришћење. То је било нешто што ми, у то време, нисмо имали воље и жеље да радимо, па смо дошли на бољу идеју: да направимо додатни интерактивни алат за визуализацију и приказ резултата филтрираних коцкастих-текстура, тако да је могуће лакоћом користити алат без потребе претходног читања упутства и смерница. Тако је започео развој *cmftStudio*-а.

Развој *cmftStudio*-а је трајао приметно дуже због разлога што је сада требало много ствари урадити и уклопити на једном месту како би се постигло пријатно корисничко искуство. Исти циљеви су превладали: доступност за све платформе и јавно доступан изворни код целе апликације. За развој је коришћена *bgfx* библиотека за исцртавање, о којој је писано раније - у 33. броју овог часописа. Она је такође отвореног кода, нуди приступ функцијама графичког процесора за исцртавање са различитих платформи без скоро икакве додатне модификације и, као таква, представљала је одличан и очигледан избор за потребе развоја *cmftStudio*-а.

Сада већ доступан на интернету, *cmftStudio*, поред филтрирања коцкастих-текстура, нуди и још мноштво других функција, као например: учитавање 3Д модела и његових текстура, креирање и подешавање различитих материјала, преглед исцртавања материјала са разним особинама глаткоће и конституције под утицајем директне и амбијенталне светлости, снимање и учитавање сесије, итд. Због свега овога, чак и 3Д моделари налазе употребу *cmftStudia* за учитавање и преглед својих 3Д модела.

У првих недељу дана након објаве, Гитхаб страница *cmftStudio* пројекта је остварила скоро две хиљаде јединствених прегледа, што показује доста велико иницијално интересовање за овакав алат. Како ће даљи развој *cmftStudia* тећи, у великој мери зависи од утисака самих корисника. За сада алат нуди све есенцијалне опције за функцију коју обавља и може се без проблема користити у продукцији компјутерских игара или 3Д апликација које имају потребу за оваквим алатом.

Корисни линкови:
[1] https://github.com/dariomanesku/cmft
[2] https://github.com/dariomanesku/cmftStudio

BeagleBone Black Rev C

Водич од првог дана (5. део) – Биглбон Блек као Тор егзит

Аутор: Ненад Марјановић

Да бисмо боље разумели конфигурацију Тор сервера, треба посебно обратити пажњу на конфигурациони фајл /etc/tor/torrc. Вредности у подешавањима варирају од ресурса којима располажемо. Уколико желимо да се бавимо администрацијом излазног (енг. *Exit*) нода, за то је потребна брза интернет конекција и, наравно, више расположиве радне меморије (енг. *RAM - Random Access Memory*).

Биглбон Блек рев. Ц (ББББ - енг. *BeagleBone Black Rev C*) располаже са 512MB *RAM*-а, што нам ипак оставља одређене могућности при параметрирању нашег првог излазног нода. Све што треба да урадимо је да ограничимо излазну политику на одређене портове. У прошлом броју ЛиБРЕ! часописа показали смо конфигурисање трансфер сервера.

*ExitPolicy reject *.**

Ово у преводу значи да наш сервер служи искључиво за трансфер ка излазном серверу. У редакцији смо почели да тестирамо одређена подешавања и, након вишемесечног тестирања, дошли смо до закључка да је могуће подржати сервисе као што су *SSH*, *IRC*, *IRC(s)*, *FTP*, *SMTP(s)*, *IMAP(s)*, *POP3(s)*, *XMPP*, *GIT* и *OpenPHP HKP* без проблема на већ постојећем трансфер ноду. Да бисмо додали ова правила, потребно је изменити **torrc** фајл, односно пре уноса **ExitPolicy reject *.*** треба уписати следећа правила.

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

Преведено - прихвати комуникацију за наведене сервисе, остале одбиј.

Као што можете приметити, нигде нисмо подржали портове 443 и 80, али њих и не препоручујемо на овако малим уређајима. Чак и ако сте у ситуацији да поседујете адекватну брзину интернет конекције, подржите комуникацију на порту 443, али не и порту 80, ипак, ово су опције којима морамо посветити дужи период тестирања. И, уколико не желите да ваш сервер буде коришћен за малициозни мејл маркетинг (енг. *SPAM*), не отварајте порт 25.

Више доступних излазних правила можете наћи на овој локацији: https://trac.torproject.org/projects/tor/wiki/doc/ReducedExitPolicy . Како располажемо са више могучности при подешавању сервера, када смо у прилици можемо тестирати нове портове и лимит ББББ Рев Ц уређаја, са друге стране ипак не треба претеривати са бројем излазних портова због могућих законских перипетија услед саобраћаја који пролази кроз излазни сервер.

Тор заставе

По правилима Тор заједнице, сваки сервер добија заставице (енг. *Flags*) на основу функција које обавља.

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

- *Running*

су само неке од заставица, али најбитније су *Fast* (брз), *Valid* (валидан) и *Running* (у функцији). *Exit* (излаз) заставица је резервисана за сервере са довољним ресурсима. Међутим, ако ваш сервер не добије ову заставицу, а ви сте омогућили порт 443, то не значи да нисте учесник излазног саобраћаја, већ само да наш ББББ није довољно дуго у мрежи или нисте дали довољно ресурса излазном серверу.

Након неколико сати активности статус нашег сервера можемо проверити на следећим адресама:

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

У поље претраге унесите надимак сервера и добићете информације о Тор серверу над којим вршите администрацију.

Сигурност Тор сервера

Бити администратор сервера односи се на бригу о сигурности уређаја који је доступан јавности. Подешавања се односе пре свега на *SSH* конекцију и заштитни зид (енг. *Firewall*). Златна правила су следећа:

1. не користити стандардни порт за конекцију на сервер (увек подесити вредност изнад 20000);
2. додајте новог корисника на сервер и искључите логовање за рут (енг. *root*) корисника;
3. креирајте *SSH* кључ за конекцију на сервер, зато што коришћење лозинки није препоручљиво због могућих бруталних напада на рут налог корисника.

* * *

У наредном броју настављамо са писањем о подешавању излазног Тор сервера са акцентом на заштиту сервера и корисника те прављење копије серверских конфигурационих фајлова, као и о напреднијој политици заштите *SSH* сервера употребом технике геолокализације администратора сервера.

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