

Октобар 2013.



ЛИБРЕ!

Часопис о слободном софтверу

број

18



pySioGame
Educational Activity Pack for Kids



9. октобар

Објављен је
FirefoxOS 1.1.



24. октобар

Android слави 10
година постојања.



Creative Commons Ауторство-Некомерцијално-Делити под истим условима.



FLOSS у Србији

Овог месеца смо били вредни: издали смо први ЛиБРЕ! специјал (приручник *Scribus 1.4*) и успели смо да припремимо нови редовни број који је сада пред вама. Надамо се да ћете нам зато опростити због малог кашњења овог редовног издања.

ЛиБРЕ! пројекат се још увек добро држи, упркос осетном паду активности у свим FLOSS заједницама у Србији. Економска криза је коначно почела да узима данак, поткопавајући сваки друштвено корисни рад. Пројекти који се могу подвести под категорију друштвено корисног рада су увек ниско на скали приоритета, јер не доносе профит нити се од тог рада може живети.

Међутим, FLOSS активисти се још увек боре против ветрењача у Србији, па можда чак и у региону. Не вреди објашњавати предности слободног софтвера у неуређеним државама у којима је пиратерија ако не прећутно одобравана, онда барем често игнорисана. Овакав однос према софтверу ће се ускоро

многима „обити о главу”.

Држава Србија је почела да незграпним корацима уређује и ову област. Потписала је уговор са *Microsoft*-ом којим се легализује софтвер у свим државним органима. Бојимо се да ово није корак напред у правцу борбе против пиратерије него више увођење монопола једне платформе. Реч уредника није место за детаљну анализу овог уговора. Можемо само да најавимо опширнији текст о његовим последицама за следећи број.

Овом најавом желимо да подстакнемо све људе у FLOSS заједницама у Србији да се прегрупишу и организовано делују у јавности како би се бар ублажиле већ најављене последице овог уговора.

Први корак је свакако обелодањивање садржаја документа и анализа његових последица. За ту сврху ЛиБРЕ! свим српским FLOSS заједницама нуди простор да у наредним бројевима објаве своје



виђење овог проблема. Што боље упознамо јавност са проблемом, из више углова, лакше ћемо га решити.

Вреди се борити, у противном, OSS пројекти у Србији неће имати будућност, а српска софтверска индустрија биће сведена на развој апликација за само једну платформу.

На крају желимо све вас да подсетимо да је ЛиБРЕ! пројекат отворен за све оне који желе да му се прикључе било у својству аутора, графичара или дизајнера. Нарочито нам требају аутори за рубрике „Интернет, мреже и комуникације“, „Сервери“ и „Мобилни кутак“. Своје текстове нам можете слати на нашу већ познату адресу [libre\[et\]lugons\[dot\]org](mailto:libre[et]lugons[dot]org).

До читања.

ЛиБРЕ! тим

Моћ слободног
софтвера



Број: 18

Периодика излагања: месечник

**Главни и одговорни уредник:
Никола Харди**

**Извршни уредник:
Александар Станисављевић**

**Лектура:
Ромео Млинар
Јелена Мунћан
Александра Ристовић
Александар Божиновић
Александар Станисављевић**

Редакција:
Горан Мекић Сандрина Димитријевић
Џони Промис Александар Тодоровић
Жељко Шарић Милован Кривокапић
Данило Ђокић Далибор Богдановић
Дарко Стантић Александар Брковић
Златан Васовић Гаврило Продановић
Иван Булатовић Михајло Богдановић
Стефан Ножинић Владимир Цицовић
Жељко Попивоџа Александар Весић
Бојан Богдановић

Сарадници:
Никола Ненадић Момчило Медић
Ненад Мијатовић Велимир Бакса
Тамара Ђорђевић

**Графичка обрада:
Дејан Маглов**

**Дизајн:
Младен Шћекић Зоран Лојпур**

**Контакт:
IRC: #floss-magazin на irc.freenode.net
E-пошта: libre@lugons.org**

<http://libre.lugons.org>



ЛИБРЕ! вести

стр. 6



Пул слободе

стр. 8

FreeOpen.net beta 4

стр. 8



Представљамо

стр. 12

Ubuntu 13.10
Saucy Salamander

стр. 12



ArchBang Linux 2013.09.01

стр. 16



Како да?

стр. 20

Мала школа:
Sigil 0.7.2 (6. део)
Додавање прилога

стр. 20



Ослобађање

стр. 24

PC гејминг индустрија
мења свој правац према
отворености

стр. 24



Слободан софтвер за
образовање деце (2. део)
pySioGame

стр. 28



Слободни професионалац стр. 33

Пуштање *Python web*
пројеката у рад



Пропратни проблеми
и нека решења стр. 33



Сам свој мајстор стр. 40

Увод у *LaTeX* (2. део) стр. 40



Мобилни кутак стр. 44

Firefox OS 1.1 стр. 44



ЛИБРЕ! пријатељи





Нови Mercedes-Benz користи Ubuntu и Xubuntu

7. октобар



Xubuntu.

Прототип новог Mercedes-Benz аутомобила S класе, са интелигентним аутономним управљањем, користи Ubuntu и

Користан линк: <http://j.mp/1fdh8LO>

FirefoxOS 1.1

9. октобар



Објављен је FirefoxOS 1.1. Међу изменама је и MMS подршка, као и подршка за више од 15 језика.

Користан линк: <http://j.mp/15q4DHF>

Објављен је Linphone web 1.0

10. октобар



VoIP позиви из web прегледача уз помоћ open source решења? Linphone web.

Користан линк: <http://j.mp/1cEZ0tq>

Објављен је AbiWord 3.0

13. октобар



AbiWord 3.0 текст процесор са собом доноси GTK3 подршку као и многобројне исправљене бубе.

Користан линк: <http://j.mp/1alk7ur>

Крај развоја за Linux 3.0

13. октобар



Након 100 тачка издања, Linux 3.0 се више неће развијати. Корисницима се препоручује прелазак на Linux 3.4 или 3.10.

Користан линк: <http://j.mp/1hRsQZm>

Cinnamon се осамосталио

15. октобар



Cinnamon 2.0 је постао самостално окружење радне површи као последица брзог и непредвидивог развоја GNOME 3.0 окружења.

Користан линк: <http://bit.ly/19MC88b>

GCC 4.8.2

16. октобар



Објављен је GNU C компјлер са више од 70 отклоњених буба.

Користан линк: <http://j.mp/1avvHDT>

Ubuntu 13.10

17. октобар



Објављени су Ubuntu, Kubuntu, Xubuntu и Lubuntu 13.10 кôдног имена Saucy Salamander.

Користан линк: <http://bit.ly/1bZTW3f>



Calligra 2.7.4

17. октобар



Објављен је *Calligra 2.7.4* канцеларијски скуп алата.

Користан линк: <http://j.mp/1f10KmU>

PHP 5.5.5

17. октобар



Објављен је *PHP 5.5.5* са 20 закрпљених буба и подршком за *GNU Hurd*.

Користан линк: <http://j.mp/H2NFG2>

Wireshark прелази на Qt

19. октобар



Wireshark, популарни анализатор мрежног саобраћаја, прелази на *Qt* пакет алата.

Користан линк: <http://j.mp/1aaATvD>

Twitter u open source

21. октобар



Како би превентивно деловао у циљу спречавања прекида свог сервиса у раду, *Twitter* прелази на *open source*.

Користан линк: <http://j.mp/17EVSqu>

LG Fireweb

23. октобар



LG је издао *Fireweb*, свој први *Firefox OS smart-phone* који је представљен у Бразилу.

Користан линк: <http://j.mp/18vMgOr>

Android слави рођендан

24. октобар



Android слави 10 година постојања. Срећан ти рођендан, *Android-y!*

Користан линк: <http://j.mp/1ikwTxK>

ЛИБРЕ! пријатељи

LUTHERUS

Et in Arcadia ego!



Think about this

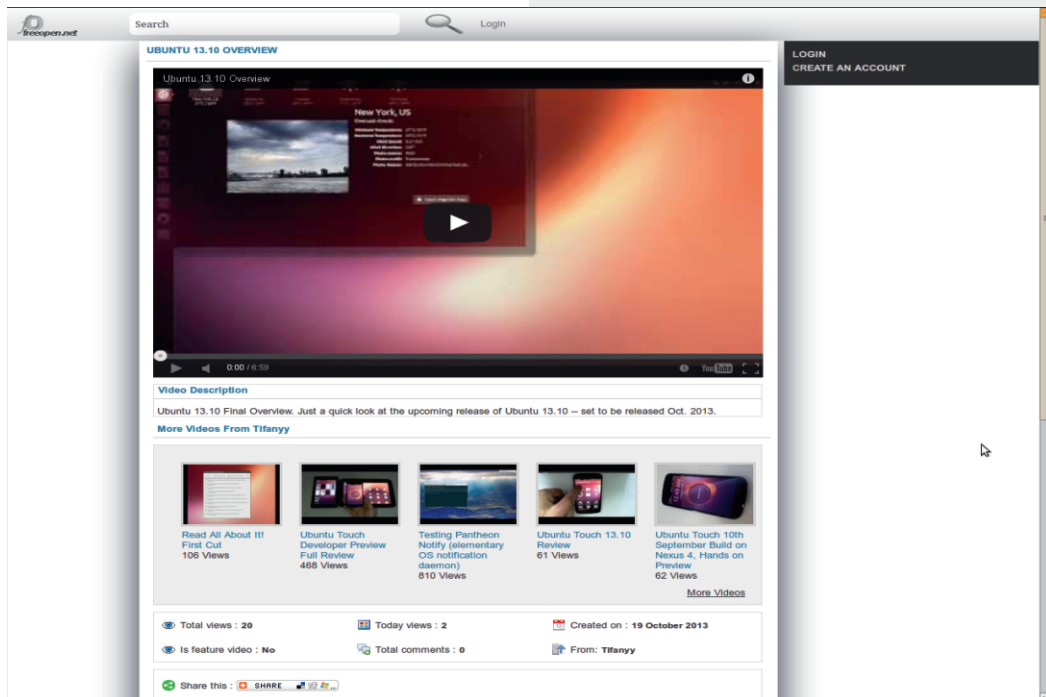


ICT časopis

ictcasopis.ict.edu.rs



FreeOpen.net beta 4



Аутор: freeopen.net

Након четири месеца развоја пред нама се налази *freeOpen.net beta4*. Заиста је напредак и више него очигледан. За оне који још увек не знају, *freeOpen.net* је нова друштвена мрежа (активан портал), намењена првенствено *GNU/Linux* корисницима и девелоперима, али добродошли су сви који на било који начин подржавају слободан и отворен софтвер.

Осмишљена је тако да пружи потпуну слободу приликом презентације ваше или вама омиљене дистрибуције. Како се не бисмо понављали, (чему која опција служи) из једног од претходних бројева (#13) изнећемо само промене које *beta4* доноси. Наспрам *beta3* верзије, *beta4* доноси заиста много тога новог, а акценат је стављен на побољшање корисничког искуства.

Ово је списак основних промена које



freeOpen.net beta4 доноси:

- Јединствени профил (нема више подела на *User* и *ActiveUser* профиле).
- Лакше повезивање између корисника.
- Могућност праћења другог корисника, тј. његовог профила (*Follow*).
- Два функционална *ActiveWall*-а (један за профиле које пратимо, други за профиле пријатеља).
- *Home ActiveWall* - нове објаве корисника којих пратимо.
- *Friends ActiveWall* - приказује шта су наши пријатељи коментарисали.
- Препорука садржаја који су занимљиви вашим пријатељима.
- Лакши унос садржаја (текст, видео...).
- Омогућен унос садржаја (текст, видео...) директно на *ActiveWall*.
- Преглед коментара садржаја на *ActiveWall* страницама.
- Коментарисање садржаја без напуштања *ActiveWall* страница.
- Преглед садржаја који су унели корисници, без напуштања њиховог профила.
- *MyPanel* са новим и функционалним распоредом.
- *Favorite User* листа.
- Унапређен изглед страница профила.
- Омогућено коментарисање сопственог садржаја.
- Омогућено додавање свог профила у листу пријатеља.
- Омогућено да *Friends ActiveWall* приказује који садржај сте коментарисали.

- Сви могу додавати садржај (дистрибуције, апликације, видео, блог и најаве).
- Поједностављено кретање кроз садржај.
- Промењена насловна страна (за нерегистроване).
- Додато уочљивије *Follow* дугме.
- Клик на лого у *header*-у аутоматски враћа на *Home* страницу.

Укратко да појаснимо: најбитнија промена је то што се ипак одустало од две врсте корисничких профила. Тренутно постоји само једна врста корисничких профила, у којој је сажето скоро све из некадашња оба профила, али зато постоје два *ActiveWall*-а (*Home* и *Friends*). Овако је почетницима олакшано сналажење, али и сами корисници сносе већу одговорност, јер је тренутно свима дозвољено постављање текстова, видеа, линкова за *download* дистрибуција, слика...

Такође, једна од новости је праћење другог корисника. Више није потребно слати захтев за одобрење другом кориснику, како бисте могли да пратите на свом зиду (*Home ActiveWall*) његове објаве, довољно је само да кликнете на дугме *Follow*, и аутоматски све његове објаве су видљиве и на вашем зиду (нешто попут *Twitter*-а).

Наравно, уколико желите да пратите активно и његове коментаре (или коментаре неког другог корисника), морате да му пошаљете захтев за пријатељство. У зависности од његовог подешавања, може аутоматски да прихвати ваше пријатељство, или пак морате да сачекате његово одобрење.

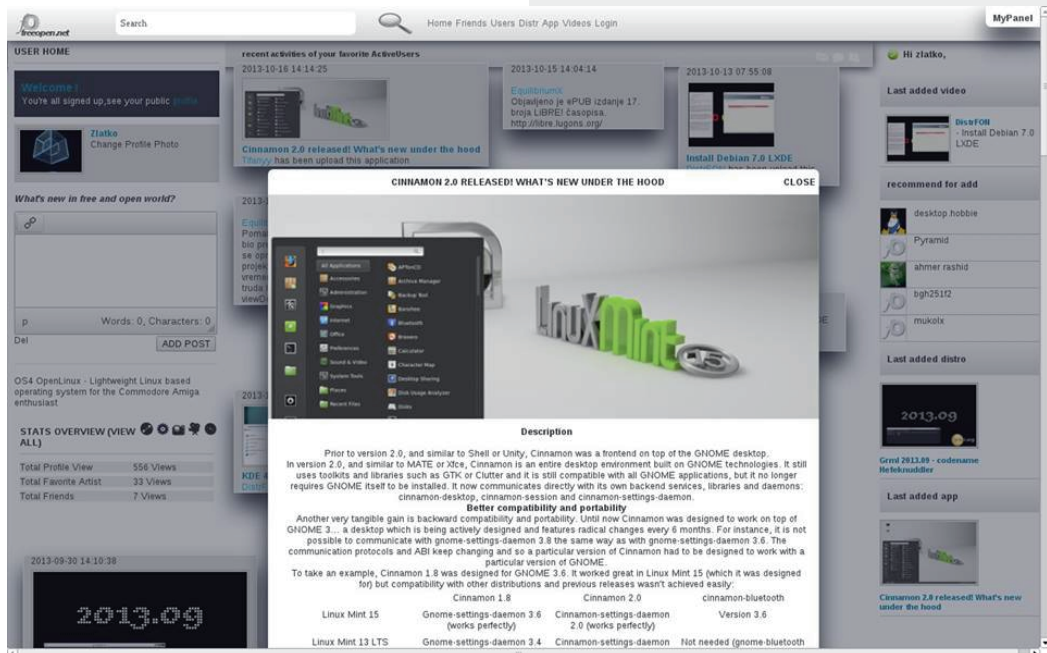


Овом опцијом, сем што пратите његове коментаре, такође, он вам препоручује друге објаве корисника. *Friends ActiveWall* је врло квалитетно и практично решење и уколико имате много активних пријатеља, бићете и више него обавештени о дешавању у слободном свету.

Ово наравно није све, већ само основа за оно што следи, а то је далеко комфорнији преглед и коментарисање објаве. Више не морате да напуштате *ActiveWall*, како бисте прочитали или прокоментарисали неку објаву. Директно на *ActiveWall*-у кликом на поље вам се отвара искочни прозор са комплетним текстом и коментарима везаним за тај текст. Ту се наравно налази поље за унос вашег коментара.

Новост је и додавање садржаја, текстова (објава), видеа и најава директно на *ActiveWall*. Кликком на одређену иконицу која се налази с леве стране, отвара се форма за наменски унос. Ово је практично решење и омогућава брзо и лако додавање садржаја.

Најбољи начин да се упознате са новинама које *freeOpen.net beta4* доноси јесте да сами истражите. Слободно прегледајте садржај, додајте најаве, опишите укратко дистрибуције или програме које користите, изнесите своја искуства или проблеме с којима сте се сусретали приликом инсталације и употребе или пак изнесите корисна решења. Не морате бити професионални блогер, већ једноставно бележите своја запажања,





Search Login

NEW VERSIONS OF KUBUNTU 13.10

Distribution Description

Kubuntu 13.10
New 3.11 kernel
KDE 4.11.2

Upgrading from Previous Versions

To upgrade to Kubuntu 13.10 from previous releases, click on the notification which will appear in your panel. See [13.04 to 13.10 Upgrade Instructions](#) for a complete walkthrough.

KUBUNTU 13.10 HIGHLIGHTS

KDE Plasma and Applications 4.11

Download a Disk Image

Download a [bootable image](#) and put it onto a DVD or USB. [Burning ISO Guide](#).

New versions of KDE's Software Compilation 4.11s featured in Kubuntu 13.10, and adds faster Nepomuk indexing, Kontact improvements such as a new theme editor for e-mails, and prepares the ground for future developments using Wayland and Qt 5.

Muon Discover

Distribution **Released** **Thursday, 17 October 2013**

[Download](#)

Total views : 103
Today views : 5
Created on : 17 October 2013
Total softwares : 0
Total comments : 0
From: [DistrFON](#)

Share this : [SHARE](#)

Report abuse : [Click here](#)
Bookmark : [Click here](#)

More Distributions From DistrFON

- KANOTIX 2013**
563 Views
- Elive 2.1.32 development released**
484 Views
- Release Notes for Gimi 2013.02 - codename Grumpy Grinch**
526 Views

препоручујте странице и наравно умрежавајте се. Што већи круг пријатеља будете имали, тиме ћете стећи и поделити више знања. Активно доприносите развоју мреже, јер ви сте креатори садржаја као и главни критичари истог, а такође можете да утичете и на додавање неких нових проширења на *freeOpen.net*-у.





UBUNTU 13.10

Saucy Salamander





Аутор: Ubuntu Србија

Као што смо навикли, у свом шестомесечном ритму *Ubuntu 13.10* је стигао да попуни дуги низ *Ubuntu* издања најпопуларније *Linux* дистрибуције. Како је у питању најраширенија дистрибуција, нимало не чуди што је у слободном свету ово важан догађај.

Шта нам ново доноси *Salamander*?

Аутор овог представљања ретко ради *clean install* система. Много чешће са издања на издање прелази *upgrade*-ом и то још на развојној грани. Овог пута се улењео па је *Salamander* „слетео“ на његов *desktop* кроз чисту инсталацију. Баш због тога се изненадио изгледом инсталера. Видан напредак је постигнут на том пољу. Могућност да

сопствени *Ubuntu One* сервис повежете са поступком инсталације је добра и корисна ствар. Многи корисници имају навику да поједине пакете, *deb*-ове и свашта нешто чувају на *Ubuntu One* па је згодно имати их при руци кад затребају.

Дакле *Ubuntu One*, као део инсталације, јесте добродошла промена.

Unity се у верзији 7 задржао као подразумевано окружење за *Ubuntu*. *Unity 7* постаје све „зрелији“ пројекат. Осим што је (очекивано) осетно бржи него *Unity* на 12.04 који аутор користи на лаптопу, визуелно је „испегланији“ и значајно богатији. То богатство поготово долази до изражаја када бацимо поглед на количину сочива (*scopes*), чији задатак је да претрагу специфичних података ограничи на поједине области. Сама идеја сочива није лоша. Прескаче се претрага у веб претраживачу, али таква претрага даје





поплаву резултата без финог филтрирања, што онда постаје бесмислено. Због ограниченог простора и дизајна *Dash*-а сама корисност сочива је упитна. Тај задатак тек стоји пред *Ubuntu* развојним тимом, како и на који начин ту врсту претраге учинити паметнијом. У међувремену, кориснику остаје да по сопственом нахођењу поступком укључивања и искључивања појединих сочива, добије жељени резултат. Са друге стране, *Unity* тим обећава да ће временом сам *Unity* учити од корисника, и да ће кориснику нудити боље резултате, што на свеже инсталираном *Ubuntu*-у не можемо проверити.

Са друге стране можемо поздравити, што је напokon *Gnome3* коректно

визуелног изгледа, овај *Ubuntu* је најлепши до сада, барем по мишљењу аутора овог текста. Како је лепота субјективна категорија, а аутор није усамљен у овој тврдњи, можемо слободно рећи да *Ubuntu* визуелно изгледа одлично у кожи *Salamander*-а.

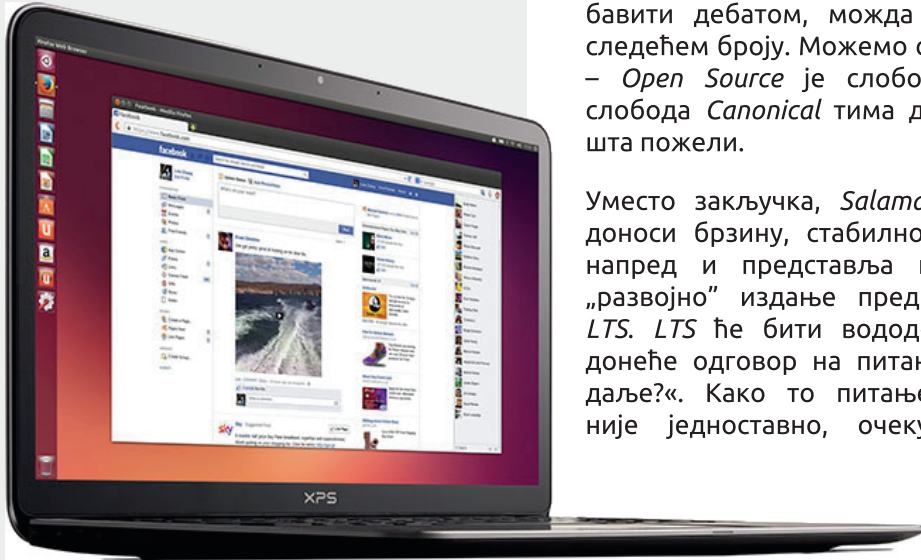
Обрадовали смо се такође, кад само „налетели“ на *Geary*, *lightweight email* програм, у софтверском центру. То сведочи да вредан рад *Elementary* тима није незапажен, што је добро. *Thunderbird* штуца са развојем, па топло препоручујемо миграцију на овај сјајни, лагани *mail* програм.

Mir није подразумевани графички сервер, ту је стари *X*. Са друге стране *Mir* се инсталира лако и једноставно, и



имплементиран у *Ubuntu*, па *Nautilus* изгледа како треба без додатних напора корисника. Кад смо већ код

ради на ауторовој *Nvidia* графичкој картици лепо и глатко. Ово не значи да ће све *Nvidia* картице подржати *Mir*, а нисмо проверили какво је стање са



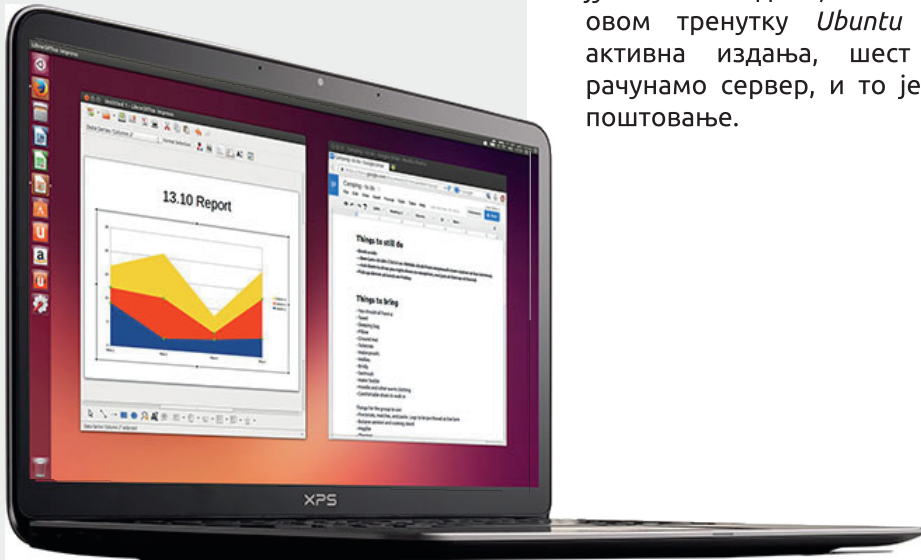
бавити дебатом, можда у неком следећем броју. Можемо само рећи – *Open Source* је слобода, па и слобода *Canonical* тима да развија шта пожели.

Уместо закључка, *Salamander* нам доноси брзину, стабилност, корак напред и представља последње „развијно” издање пред следећи *LTS*. *LTS* ће бити вододелница и донеће одговор на питање: »Куда даље?«. Како то питање уопште није једноставно, очекујемо да

AMD АТI графичким картицама.

Дебата *Mir* против *Wayland* заслужује посебан текст, о томе ће се још много говорити и куцати, па се нећемо овде

следећи *LTS* буде достојан одговор. *Salamander* је ту да том одговору да солидан темељ и тај посао је урадио веома добро. Званична подршка је до јула 2014. године, што значи да у овом тренутку *Ubuntu* има три активна издања, шест уколико рачунамо сервер, и то је за свако поштовање.





ArchBang Linux 2013.09.01

Аутор: Александар Брковић

ArchBang Linux заснован је на принципу минимализма и једноставности. Захваљујући Openbox-у који је подразумевани window manager, постиже високе перформансе чак и на хардверу старије производње. Компатибилан је са i686 и x86-64 рачунарима и претежно је састављен од слободног софтвера. Rolling release принцип даје могућност да се ажурирањем система увек изврши надоградња на најновије издање.

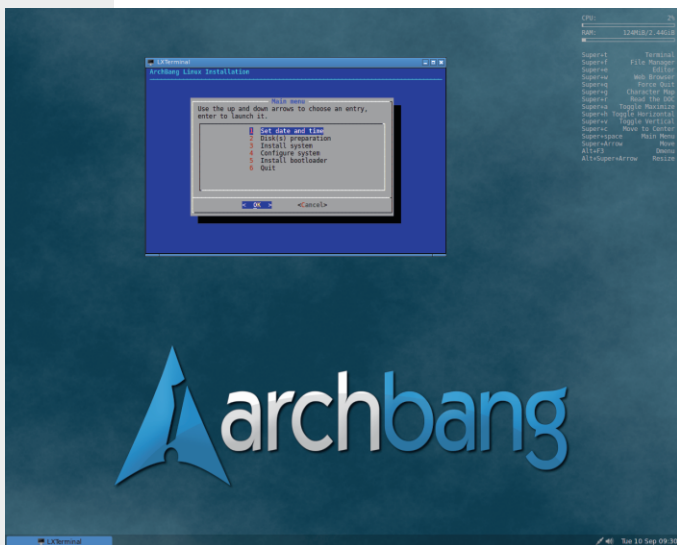
Настанак наведене дистрибуције инспирисан је постојањем CrunchBang Linux-а. Willensky Aristide је заслужан за настанак ArchBang дистрибуције. Циљ је био да се направи Linux дистрибуција чија ће основа бити Openbox window manager, а која ће функционисати по rolling release принципу. У ту сврху изабран је Arch linux, и уз помоћ и подстицај сарадника, настао је ArchBang Linux. Од 16. априла 2012, нови руководилац пројекта је Stan McLaren.



Конфигурација

ArchBang није дистрибуција намењена почетницима, већ искуснијим корисницима: сва се подешавања система врше текстуално унутар командне линије. *ArchBang* нема графичке алате, помоћу којих би се вршила конфигурација и одржавање система. Припада дистрибуцијама са централизованом конфигурацијом, и сличан је *Gentoo* дистрибуцији. Централна конфигурациона датотека налази се у `/etc/rc.conf`, и садржи основне податке и информације о систему, као што су временска зона, локализација, модули који се учитавају приликом покретања система као и сервиси који се покрећу заједно са оперативним системом. Централно место свих сервиса, односно њихових старт скрипти је `/etc/rc.d`.

процес одабира временске зоне и подешавања времена, постављање лозинке за *root* налог, креирање корисничког налога, и инсталацију датотека на хард-диск. Подразумевано *ArchBang* инсталира *GRUB* покретачки програм. Процес инсталације је веома брз, и траје око петнаест минута.



Инсталација

ArchBang долази са модификованим *Arch Linux* графичким инсталером. Скрипта за инсталацију пружа једноставан водич, лак за праћење, који корак по корак приказује инсталациони процес. Корисницима који се први пут срећу са овим системом поједностављен процес инсталације нуди велику олакшицу у односу на *Arch* из којег вуче корене. *Cfdisk* текстуални програм ће извршити поделу партиција на хард диску, форматирање *swap* и *root* партиције, инсталер даље води корисника кроз

Изглед

Након завршетка инсталационог процеса, корисника ће радна површина десктоп са небоплавом позадином, у десном углу *conky* програмом са основним параметрима, и панелом на дну екрана. Нема *pop-up* прозора, нема обавештења о доступним ажурирањима. Оперативни систем не комуницира са корисником све док се не издају одговарајуће команде. Графички управљач датотека не препознаје по *default*-у ознаке типа датотека. Двоклик на текстуалну или аудио датотеку неће проузроковати



подразумевану радњу, већ ће корисник прво морати да успостави везу између типова датотека и апликација којима се одређене датотеке покрећу.

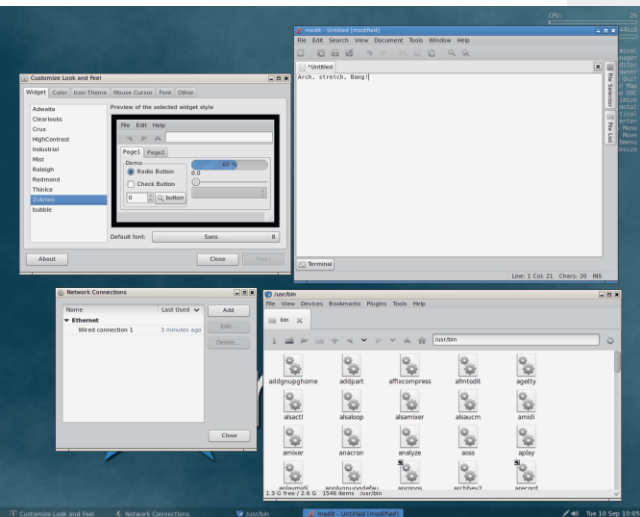
могућностима и својим потребама самостално изгради систем до границе употребљивости.

Управљање пакетима

Инсталација, ажурирање и управљање програмским пакетима, врши се уз помоћ алатке *Rastan*, који је једини управљач пакетима на овој дистрибуцији. Формат пакета је датотека чија је екстензија **pkg.tar.gz**, односно пакет је компресована датотека који садржи бинарне датотеке, информације о пакету, и начину инсталирања.

Пакети се налазе у пет главних бинарних репозиторијума:

- *Current* – садржи основне програмске пакете за потпуну инсталацију система.
- *Extra* – у њему се налазе програмски пакети који нису неопходни, али су добра алтернатива.
- *Unstable* – садржи незваничне пакете који су још увек у развоју.
- *Testing* – ризница нестабилних пакета који још увек имају багове и могу да изазову проблеме.
- *Community* – садржи пакете који су направљени од стране корисника *Arch*-а, често су то пакети из *AUR*-а.
- *AUR (Arch User Repository)* - поред бинарних пакета у наведеним складиштима, постоје и пакети са упутствима како направити и инсталирати пакете из изворног кода. Овај начин користи *PKGBUILD* текстуалне датотеке, у којима је



Апликације

Када је реч о апликацијама, *ArchBang Linux* доноси мали број унапред инсталираних апликација. *Firefox* веб прегледач, калкулатор, прегледач слика и *SpaceFM* управљач датотека. Стандардно је инсталирано и неколико алата за подешавање и прилагођавање изгледа *Openbox window* управљача. *Network manager* је такође прединсталиран да би омогућио конекцију која је потребна за даљу надоградњу система. У позадини налазимо кернел у верзији 3.10. Суштински, пред корисника је постављен задатак да у складу са својим

записано, кодирано упутство за комплетну израду paketa pomoću *makepkg* komande. Korisnici Arch-a sami prave *PKGBUILD* datoteke te ih postavljaju na internet u *AUR* spremište, da bi i ostali korisnici imali koristi od njih. Provereni i priznati paketi iz *AUR*-a se premештају у *Community* repozitоријум и постају службени paketi.

Da bi *Pacman* ispravno функционисао, потребно је након инсталације да изменимо листу repozitоријума, те да омогућимо оне најближе нама, као и да извршимо синхронизацију са серверима. Добра идеја је омогућити више repozitоријума у случају да један није синхронизован са другима. Након тога је могуће извршити инсталацију и ажурирања софтвера.

Софтвер је увек најновији и ажурира се често. То резултира потенцијалном нестабилношћу и може да доведе до пада система, будући да пакети нису имали много тестирања. Због тога се препоручује опрез приликом преузимања надоградњи, те се саветује читање *online* документације о примени тренутних исправки, да би се избегли евентуални проблеми.

Закључак

Од корисника *ArchBang* система се очекује да подешавања система раде ручно. Потребно је чешће дружење са

командном линијом и текстуалним уређивачима. Због тога се не препоручује свим корисницима. Они који имају солидно знање, ентузијастички воле да истражују срж оперативног система, могу да се опробају у коришћењу *ArchBang* дистрибуције. Систем ће изгледати онако како га корисници склопе. Заузврат ће добити веома брз систем са тренутним одзивом апликација, који у исто време троши веома малу количину системских ресурса. Изазов за упорне, храбре, и кориснике који не одустају одмах када се појави први проблем.

Корисни линкови:

- [1] <http://distrowatch.com/weekly.php?issue=20130923>
- [2] <http://en.wikipedia.org/wiki/ArchBang>
- [3] [http://en.wikipedia.org/wiki/Pacman \(package manager\)](http://en.wikipedia.org/wiki/Pacman_(package_manager))





Мала школа: *Sigil 0.7.2* (6. део)

Додавање прилога

Аутор: Дејан Маглов

За ову епизоду претпоставићемо да сте већ успели:

1. да креирате један *HTML* документ у *Sigil*-у,
2. да у њега поставите ваш текст,
3. да текст форматирате и
4. да стилове преместите у посебан *CSS* документ и повежете га са вашим *HTML* документом.

Још само да напоменемо да ће свака накнадна интервенција на формату текста употребом *Sigil*-ових алата за форматирање бити записана као нови стил у облику:

```
<style>
...
</style>
```

а не директно у повезани *CSS* документ.

Ако желите да тај стил буде примењиван на више места у документу, ручно морате да га пребаците у ваш *CSS* документ.

Примена нестандартног фонта

У прошлом броју (#17) оставили смо

недоречено како смо у нашем документу додали и применили нестандардни *Minion Pro* фонт.

Дефинисање фонта је аргумент *HTML* тага као на пример:

```
<p style="font-family: Minion Pro">Helo world</p>
```

Овај **p** таг (таг параграфа) је дефинисан да буде исписан *Minion Pro* фонтом. Да ли ће и бити исписан на тај начин зависи од тога да ли на уређају који чита овај *HTML* документ постоји инсталиран *Minion Pro* фонт. Ако не постоји, биће исписан подразумеваним фонтом за тај уређај.

Web дизајнери проблем са фонтовима решавају тако да никад не дефинишу у *font-family* само један фонт него читаву фамилију сличних фонтова, као на пример:

```
font-family: SwatchCT,Arial, Helvetica,sans-serif
```

Први фонт је идеалан и баш оно што би дизајнер волео да корисник види. Остали фонтови из те фамилије су алтернатива уколико предходног фонта из списка нема. *Web* дизајнери воде рачуна о стандардним фонтовима за све оперативне системе и тако бирају алтернативе.



Epub, као формат за електронске књиге, има могућност да у својој дефиницији има придружен жељени фонт.

Битно је, за почетак, да дизајнер зна где му је жељена фонт датотека.

Постоји велики број фонтова, али и велики број типова фонт датотека. Данас су најчешће у употреби *OTF* (*OpenTypeFont*), *TTF* (*TrueTypeFont*), и то су два типа која прихвата *Sigil*.

У нашем случају, за правилан приказ примера Калибрације, потребне су нам датотеке Минион Про фонт фамилије: **MinionPro-Bold.otf**, **MinionPro-BoldIt.otf**, **MinionPro-It.otf** и **MinionPro-Regular.otf**.

Десним кликом на *Book Browser > Fonts*, и избором *Add Existing Files* додаћемо ове датотеке *Minion Pro* фамилије фонта. Осим *Minion Pro* фонта у примеру се користи *FreeMono* или *monospace* фонт. Да бисмо обезбедили да корисник сигурно има један од та два фонта додаћемо и **FreeMono.ttf** на исти начин.

Када се овако уграде фонтови у *ePUB* документ, *e-book* читач прво тражи уграђене фонтове. Ако не нађе одговарајући фонт у *ePUB*-у, онда тражи исти на свом оперативном систему. Тек кад нигде не нађе одговарајући фонт, примениће подразумевани фонт система на ком ради.

Илустровање ePUB-а

Текст у *ePUB*-у се може илустровати сликама, видео и аудио материјалом. *Sigil* прихвата следеће формате за илустрације:

1. слике у *PNG*, *JPG*, *GIF* (растерска графика), *SVG* (векторска графика) формату,
2. аудио у *MP3*, *AAC*, *MPG*, *MPEG*, *M4A* формату,
3. видео у *MP4*, *OGG*, *WEBM*, *M4V* формату.

Најбитније на шта треба обратити пажњу приликом илустровања *ePUB*-а јесте припрема материјала. Крајњи циљ је да *ePUB* документ буде што мањи, да би то било могуће све компоненте *ePUB*-а морају бити што мање. На величину текста, фонт фајлова не можемо да утичемо али на величину слика, видео материјала и аудио материјала итекако можемо.

Иста слика у различитим форматима даје различите величине датотека. Наше искуство је да *JPG* формат даје најбољи однос величина – квалитет, зато препоручујемо да све слике припремите у том формату. Постоје изузеци, нпр. када нам је потребна транспарентност слике. *JPG* не подржава транспарентност и тада се употребљава *PNG* који је подржава. *GIF* се највише употребљава у облику анимираног *GIF*-а. За илустрације од којих се захтева велика оштрина најбоље је решење, ако је то могуће, да то буде векторска *SVG* слика.

За *ePUB* резолуција слика треба да



буде на нивоу слика припремљених за интернет, то јест максимална резолуција 72 тачке по инчу (*DPI*). *EPUB* ће често бити читан на савременим телефонима. Ови телефони су у стању да прикажу слику резолуције и до 600 *DPI*. У екстремним случајевима, ако вам је врло битно да се виде детаљи на слици, можете припремити и јако мале слике са високом резолуцијом. Однос величина – резолуција би требало да буде у границама да за крајњи резултат слика не прелази 150 *kB*.

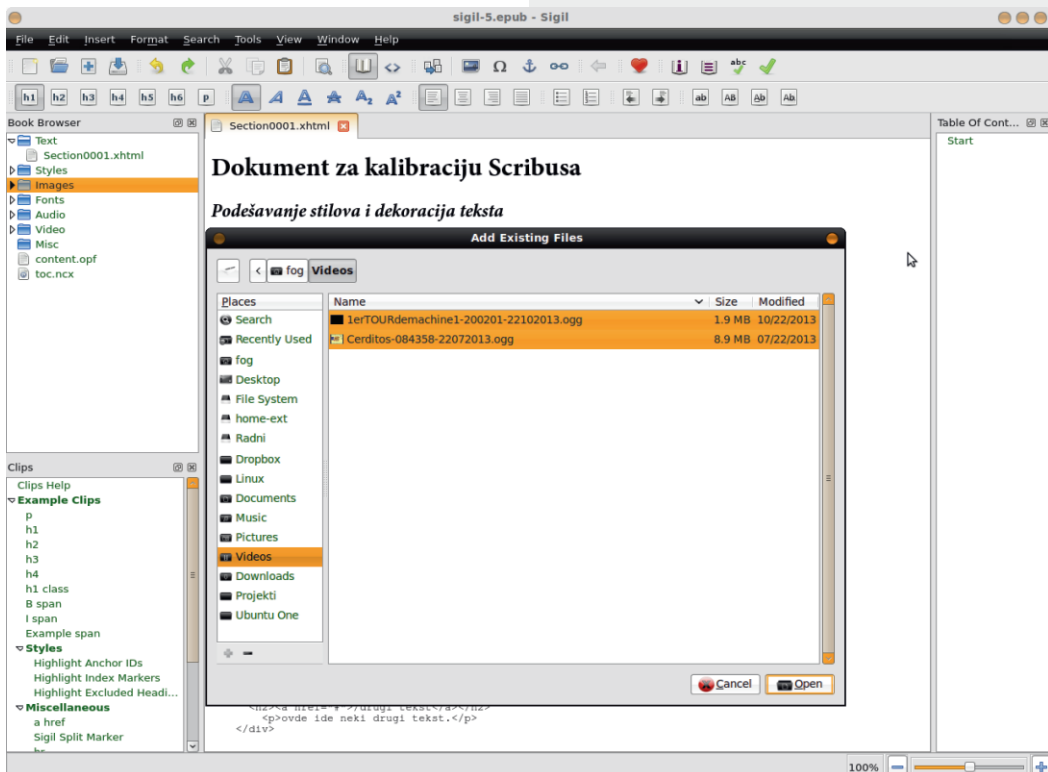
Sigil се исто тако побринуо да подржани аудио и видео формати буду са добром компресијом што обезбеђује

мале датотеке.

Сви припремљени материјали се на сличан начин увозе у *Sigil*:

1. Растерске, *SVG* векторске слике и анимиране *GIF*-ове – десним кликом на *Book Browser* > *Images* и избором *Add Existing Files*,
2. Аудио фајлове – десним кликом на *Book Browser* > *Audio* и избором *Add Existing Files*,
3. Видео фајлове – десним кликом на *Book Browser* > *Video* и избором *Add Existing Files*,

Сваки пут кад изаберете *Add Existing Files* отвара се дијалог за претраживање *file system*-а и могуће је

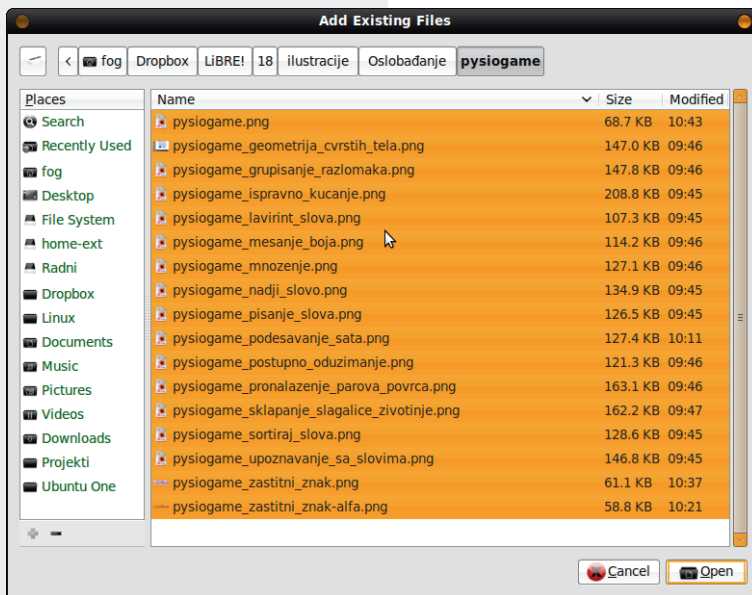




увести више датотека истовремено. Можете истовремено увозити слике, видео, аудио датотеке и фонтове, *Sigil* ће непогрешиво распоредити слике у *Images* директоријум, видео у *Video*, аудио у *Audio*, фонт у *Fonts* а све остало што не препозна у *Misc* фасциклу.

За крај

Овим смо додали све што је потребно нашем *ePUB* документу из спољњих извора и сада само треба лепо да распоредимо. О томе ћемо писати у следећем броју.

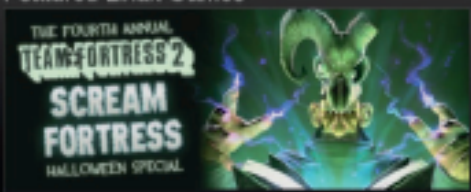


SIGIL

sigil

A WYSIWYG ebook editor.

Featured Linux Games



Scream Fortress Update
Zvill Witches, Zorblax, Magic Spells, Forbidden Bases, Shrunken Heads, Hay Bales. This is as scary as it gets!
Free to Play



New DLC Available
The new DLC, Jewel of the Mile, is now available!
\$19.99

PC гејминг индустрија мења свој правац према отворености



Аутор: Александар Тодоровић

Сви знамо да је *Microsoft* већ дуже година главна платформа за *PC* гејминг. Знамо да нисте задовољни тиме, јер да јесте вероватно не бисте читали наш часопис. Међутим, прошлог месеца су најављене огромне промене за гејминг индустрију у корист отворености. Пре почетка да са се мало подсетимо. Верзија *Steam*-а за *Mac* рачунаре постоји већ три године, а за то време је избачено 525 игара за *Mac* платформу. Средином фебруара је изашао *Steam* за *Linux* дистрибуције и за само осам месеци је број подржаних игара за *Linux* оперативни систем прешао бројку од 200 игара. Међутим, оно што је *Valve* и девелопере навело на размишљање су заправо перформансе, где се на неколико тестова *Linux* платформа показала као доминантна у односу на *Windows* на којем тренутно има убедљиво највише наслова (преко 2100).

На недавно одржаној северноамеричкој *Linux* конференцији у Чикагу, *Gabe Newell*, суоснивач *Valve*-а -

компаније која је развила *Steam*, најавио је како његова фирма не жели више да буде овисна о *Microsoftu* и његовим производима. Крајем септембра је *Valve* објавио како то планира да оствари: избацивањем властите дистрибуције базиране на *Linux*-у. Од тада се много тога променило. Прво је *Nvidia* понудила помоћ девелоперима који стоје иза драјвера са отвореним кодом за њихове графичке картице. Два дана касније је *Valve* најавио да ће

да почне производити хардвер представљајући *Steam-Machines* као сопствене *PC* уређаје који ће да покрећу *SteamOS*. Исти дан су два највећа произвођача графичких картица (*Nvidia* и *AMD*) најавили да ће далеко већу позорност посветити својим драјверима за *GNU/Linux* дистрибуције, а *AMD* је направио први корак пред-

стављањем *MANTLE*-а. *MANTLE* је *API* нижег нивоа који ће моћи боље да искористи могућности графичких картица. Израда у *MANTLE*-у значи мало више посла за девелопере, боље искуство за играче и оно најважније – алтернативу за *DirectX* (узимајући у обзир да ће *MANTLE* бити мултиплатформски и да ће његов код бити јавно доступан). Последњи новитет је нови контролер од стране *Valve*-а.





UNITING BOTH WORLDS: MANTLE



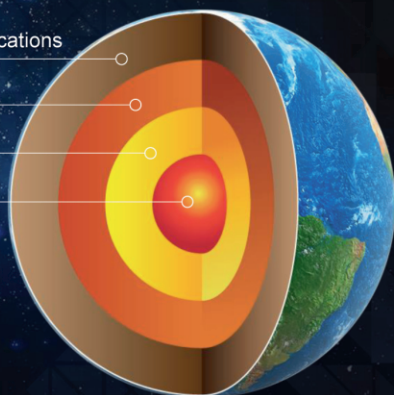
- ▲ New low level programming interface for PCs
- ▲ Designed in collaboration with top game developers
- ▲ Lightweight driver that allows direct access to GPU hardware
- ▲ Compatible with DirectX® HLSL for simplified porting

Graphics Applications

Mantle API

Mantle Driver

GCN



MORE WITH MORE

122 | AMD RADEON™ GPU14 PRODUCT SHOWCASE | SEPTEMBER 25, 2013 | APPROVED FOR PUBLIC DISTRIBUTION



Дакле, корисници *Linux*-а добијају још једну дистрибуцију на избор, *Valve*-ов фокус више није *Microsoft*-ова платформа, а девелопери су добили и

разлог и средства да много више времена проведу фокусирајући се на *Linux* платформе. Невероватно је колико се промена за *PC* гејминг најавило у тако кратком времену. *SteamMachines* се може посматрати као први покушај да се *PC* гејминг приближи конзолама. Међутим, оно што га издваја од осталих (заједно са свим новитетима које је *Valve* објавио) јесте потпуна могућност прилагођавања. У најави је речено како имате у потпуности могућности као што су инсталирање другог система, додавање програма, брисање постојећег, замењивање неког дела



хардвера... Дакле, било шта што вам падне на памет. Све се ставља под контролом корисника и зато су ови новитети постали огромна вест за љубитеље отвореног кода и хакере широм света.

Разлога за прелажење на *Linux* је све више. Када би сада питали некога зашто не пређе на *Linux*, велики постотак корисника ће вам рећи да због тога што не може да покрене одређену игру. Узимајући у обзир нову лавину промена коју је *Valve* најавио, та тренутна предност на страни *Microsofta* убрзо ће да нестане. Број разлога за прелазак на *Linux* је све већи, а број разлога да се остане на *Windows-у* се полако смањује. У име

наше редакције, а вероватно и у име сваког љубитеља слободног софтвера, велико хвала *Valve-y!*





Слободан софтвер за образовање деце (2. део)



v.1.30.811

Аутор: Александар Станисављевић

pySioGame је играца или, боље речено, скуп образовних програма намењен деци узраста од 3 до 10 година.

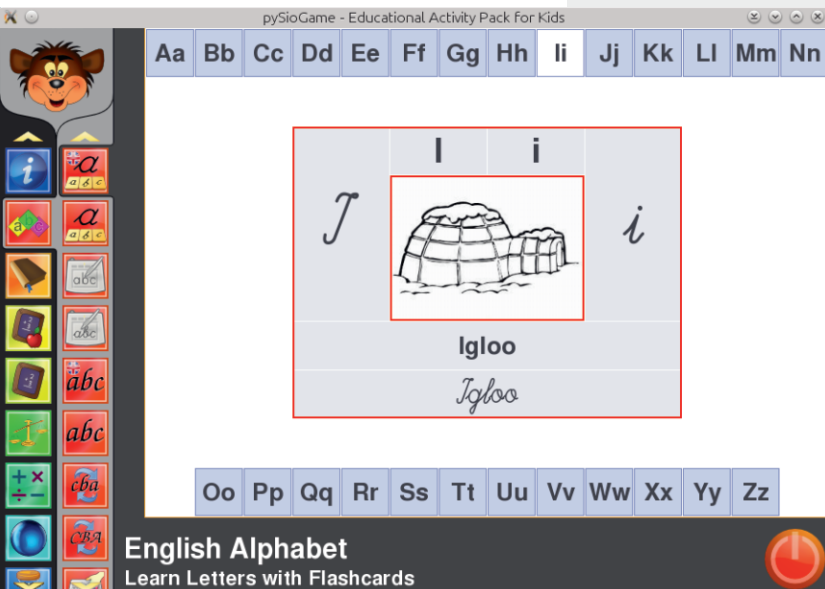
Цела играца је подељена на следеће области: језик, математика, сналажење у времену, уметност, памћење и

забава. Као највећа област посебно се истиче математика.

Језик

Ова област садржи две категорије: учење слова и бројева (*Discover Letters*) и учење нових речи (*Learn New Words*).

Код учења слова и бројева, деца се најпре упознају са енглеским алфабетом путем слика које су именоване.



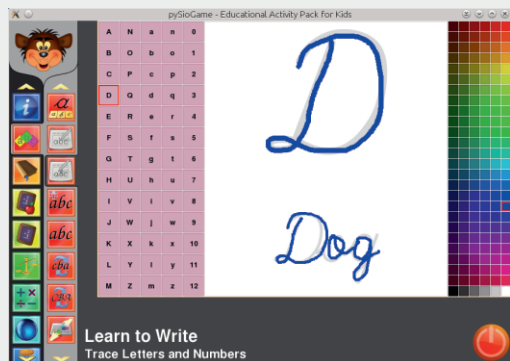
Слика 2:

Упознавање са словима путем именованих слика

Кликом на одређено слово енглеског алфавета, отвара се слика која почиње тим

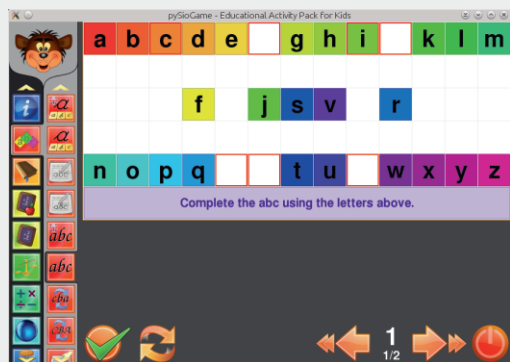
словом, при чему је почетно слово написано као велико и мало, а уједно и као штампано и писано.

Ту је затим и учење писања научених слова. Прво је дато почетно слово, а затим и реч која тим словом почиње, а које је потребно написати.



Слика 3: Учење писања слова и речи

Након тога долази на ред учење енглеског алфабета. Најпре је потребно пронаћи слово које недостаје, а затим и сортирати дата слова енглеског алфабета.



Слика 4: Учење енглеског алфабета проналажењем слова која недостају



Слика 5: Учење енглеског алфабета сортирањем датих слова

Ту је такође и учење слова и бројева путем исправног куцања (о чему је већ било речи у 16. броју ЛИБРЕ! часописа).



Слика 6: Учење слова и бројева путем исправног куцања

Код учења нових речи, деца се најпре упознају са писањем задатих речи уз помоћ скупа понуђених слова, а затим кроз игру, у лавиринту састављају задату реч тако што управљају бубом која треба да „поједе” дата слова а да сама не буде „поједена” од стране других буба.



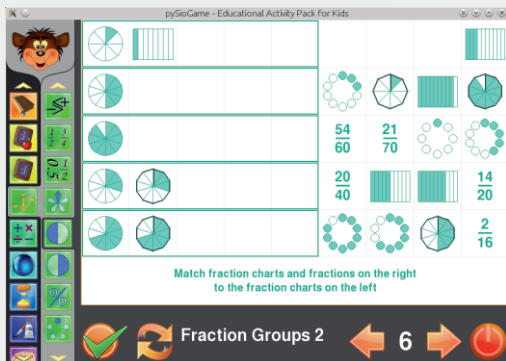
Слика 7: Учење слова кроз лавиринт игрицу

Математика

Математика, као највећа област у овој игрици, подељена је на више категорија: математика 1 (бројеви и основне рачунске операције), математика 2 (напредне рачунске операције), математика 3 (сортирање и упоређивање), рад са великим бројевима, геометрија и препознавање облика. Нећемо се детаљније задржавати на овој области пошто је велика, већ ћемо навести само пар могућности које нуди.

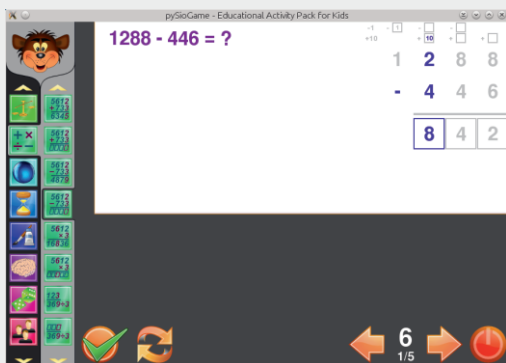


Слика 8: Математика 2 – Учење множења

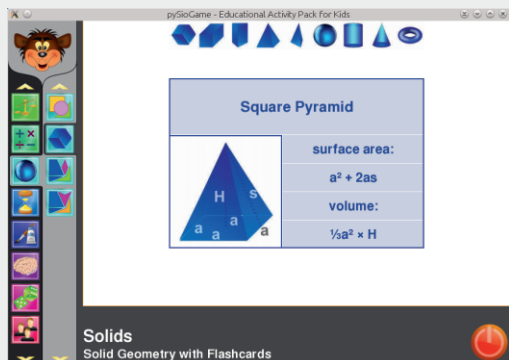


Слика 9: Математика 3 – Груписање разломака

Важно је напоменути да код груписања разломака морате вршити груписање оним редоследом који је дат на десној страни што, по нашем мишљењу, додатно отежава задатак, имајући у виду да се овиме отежава коришћење метода елиминације приликом решавања задатка.



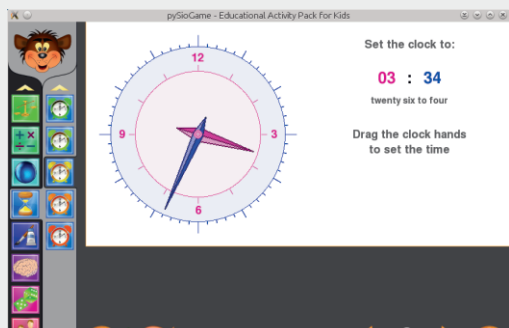
Слика 10: Рад са великим бројевима – Учење поступног одузимања



Слика 11: Геометрија и препознавање облика – Геометрија чврстих тела

Време

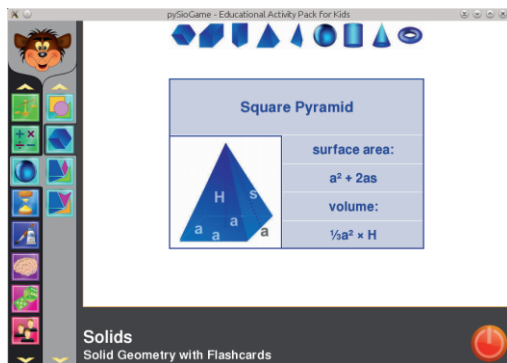
У овој области деца се најпре упознавају са начином рада аналогног часовника тако што, померајући произвољно његове казаљке, могу видети које време часовник показује. Након тога уче да читају које време дати часовник показује и на крају, како сами да подесе часовник, при чему се задатак усложњава до те мере да су им напослетку, у облику бројева, приказани само квартални часови, а остали часови и минути су приказани само у облику цртица.



Слика 12: Време – Подешавање часовника

Уметност

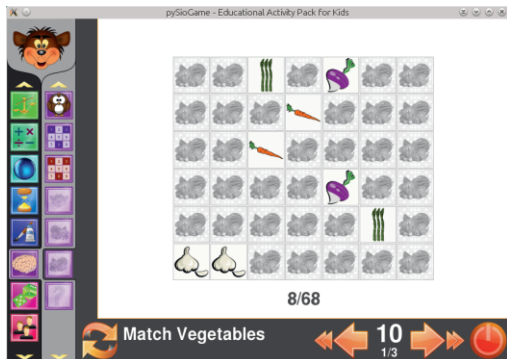
У овој области је најпре доступан обичан програм за цртање (*Paint*). Но, поред њега, занимљиве су остале категорије у којима је акценат стављен на боје (препознавање боја, мешање боја и сл.).



Слика 13: Уметност – Мешање боја

Памћење

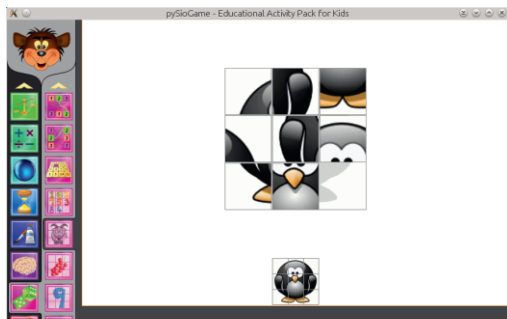
Ова област се заснива на памћењу појава на екрану. Најпре је потребно запамтити смерове у којима се креће птица, затим бројеве који су на кратко истакнути на екрану и на крају, треба пронаћи парове откривањем скривених поља, при чему се иза тих поља могу налазити животиње, воће, поврће и бројеви.



Слика 14: Памћење – Проналажење парова поврћа

Забава

Циљ ове области је да децу мало више забави, али опет стављајући акценат на образовање. Посебно су раздвојене оне игрице које су за једног од оних које су за више играча, при чему је највећа пажња посвећена првој групи игрица. Тако деца кроз игру могу да науче ког је облика нека животиња, да спајају одређене елементе истих боја, да ређају бројеве у растућем редоследу, или да склапају слагалицу одређене животиње или одређеног боја.



Слика 15: Забава – Склапање слагалице пингвина

Игрицама које су намењене вишекорисничком игрању није посвећена већа пажња, па се у овој категорији налазе само две „икс-окс“ игрице.

Према нашем мишљењу, ово је одличан пример како се деца могу образовати коришћењем рачунара. Идеално би било када би овај софтверски пакет домаћи програмери (у сарадњи са изворним програмером) прилагодили нашем говорном подручју, али и без тога, овај програм је веома употребљив. Зато позивамо све домаће програмере да се у овом погледу укључе у даљи развој овог програм, а децу позивамо да кроз забаву науче нешто ново.

Наставиће се...

Корисни линкови:

- [1] <http://sourceforge.net/projects/pysiogame/>
- [2] <https://www.facebook.com/pysiogame>
- [3] <http://www.youtube.com/watch?v=aqr1LqR0RwI>





Пуштање *Python web* пројеката у рад (пропратни проблеми и нека решења)

Аутор: Никола Харди



```
def add5(x):  
    return x+5  
  
def dotwrite(ast):  
    nodename = getNodeName()  
    label=symbol.sym_name.get(int(ast[0]),ast[0])  
    print '%s [label="%s' % (nodename, label),  
    if isinstance(ast[1], str):  
        if ast[1].strip():  
            print '=' % ast[1]  
        else:  
            print ''  
    else:  
        print ''  
    children = []  
    for n, child in enumerate(ast[1:]):  
        children.append(dotwrite(child))  
    print '%s -> (' % nodename,  
    for name in children:  
        print '%s' % name,
```



PHP је доминантан, али постоје и друге могућности

*P*HP је још увек доминантан језик у сфери *web* програмирања, али одувек је присутна и алтернатива. Некада су „озбиљне“ *web* апликације писане у *Javi*, хакерима је био занимљив *Perl*, а љубитеље *MS* технологије су привлачили *ASP*, *ADO* и други. У последње време су *JavaScript*, *Ruby* и *Python* врло популарни, а и у нашем часопису смо у више наврата писали о *web* програмирању у *Python*-у, па и сада желимо да поделимо нека наша искуства.

Шта је проблем и шта је циљ

Многе колеге и пријатеље, који већ дуго користе *PHP*, покушавамо полако да наговоримо да пређу на *Python* и упорно им показујемо нове вратоломије које изводимо, али они још увек имају један снажан аргумент, а он је: „Где ћу ја то да *hostujem*?”.

Хајде да се укратко присетимо шта је потребно за покретање једне *PHP* и једне *Python web* апликације и каква све решења имамо у понуди.



Класичан „*shared*“ *web hosting* план обухвата простор за складиштење датотека на серверу са *FTP* приступом (*File Transfer Protocol*), одређени број база података и неку врсту контролне табле, као што је то рецимо *Cpanel*. Наравно, уз све то нас чека и *web* сервер подешен за наш сајт и чију конфигурацију углавном у оваквој ситуацији не можемо да прилагођавамо. Када се одлучимо за овакво решење, све нас лепо чека подешено и ми само треба да поставимо свој кôд, подесимо базу података уколико је то неопходно – и можемо да користимо наш нови сајт.

Са друге стране, постоје и „*virtual private server*“ (*VPS*) решења која подразумевају да на располагању имамо цео рачунар. *Hosting* провајдер нам обезбеди виртуелну машину са жељеном дистрибуцијом, односно оперативним системом и достави нам параметре за приступање тој машини путем *SSH* протокола (*Secure Socket Shell*). Као *root* можемо да радимо шта

хоћемо, а за једну *python* апликацију то значи следеће: инсталирати потребне пакете, обезбедити сервис који ће покретати апликацију (рецимо *uwsgi* или *gunicorn*), *proxy* сервер за прихватање спољашњих захтева и прослеђивање тих захтева претходно поменутом сервису (практично је то у питању *proxy* и *Nginx* је сјајан избор) и базу података која нам одговара.

Морамо још једном да нагласимо да је на клијенту обавеза да обезбеди ове ресурсе и подеси их за рад јер је добио рачунар на којем је инсталиран само оперативни систем. За *PHP* нас је све чекало подешено и било је потребно само да поставимо кôд нашег сајта.

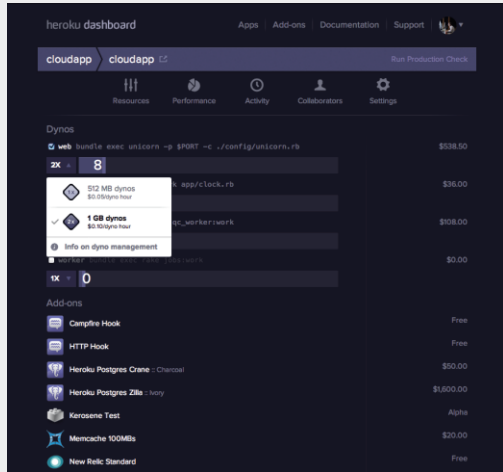
Постоји и трећа страна коју још увек нисмо сретали у понудама *hosting* компанија из региона. У питању су такозвана *PaaS* решења (*Platform as a Service*), а реч је о томе да компанија клијентима нуди разне платформе већ подешене за различита окружења која су врло свежа, занимљива и можда

егзотична. Осим избора платформе, имамо могућност и за скалирање, да изаберемо колико нам је сервера потребно, који ће шта да ради, да направимо *cluster* према нашим потребама. Овом приликом споменућемо *Heroku* сервис и компанију која има сјајне услове за бесплатне апликације и која је избор многих *web* програмера који се баве овим новим, шареним

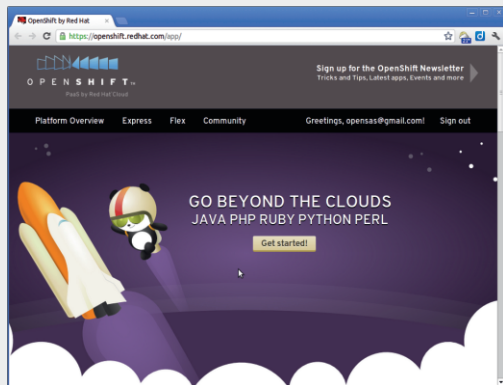


технологијама. Сервис као што је *Heroku* можемо и сами да одржавамо јер су присутни пројекти као што је *OpenShift*. *Heroku* и *OpenShift* су једна сасвим посебна прича.

инфраструктуру. Наравно, највише шта можете да очекујете да ћете затећи у таквом окружењу су редован *web* сервер и евентуално неко *Java* решење (*Tomcat* или *.NET*, *ASP*, *ADO* и дружина). Наравно, то нам не одговара за *Python* и можемо, или да се договоримо са запосленим да на неку од постојећих машина инсталирају и подесе оно шта нам је потребно за рад, или да нам обезбеде једну посебну машину коју ћемо спремити себи онако како нама одговара. Следи објашњење зашто је добро имати посебну машину и неколико корисних алата који ће нам олакшати да то све средимо.



Сви ми на нашим рачунарима имамо гомилу софтвера за које ни не знамо да су присутни. Подешавали смо системе на свакаке начине и мала је вероватноћа да ћемо наићи на исту конфигурацију, а у продукцији и то може да доведе до врло озбиљних проблема када наше ново испрограмирано чедо преселимо на други рачунар и многе ствари више не раде. Чак и ако успемо брзо да похватимо шта је недостајало и правило проблеме, следећи пут када нешто изменимо, не можемо бити сигурни да ли је та промена успешно обављена само на нашем систему или ће проћи безболно и у продукцији. Решење је да сваки програмер који ради на одређеном пројекту има на располагању и окружење какво га очекује у продукцији и пре пуштања новог кода или сервиса да може да га тестира у условима који верно пресликавају стварно стање.



Помоћни алати

Многе компаније у својем *IT* арсеналу често имају и сервере о којима се брину, имају запослене који су одговорни када нешто закаже (тзв. *sysadmin*) и донекле разрађену



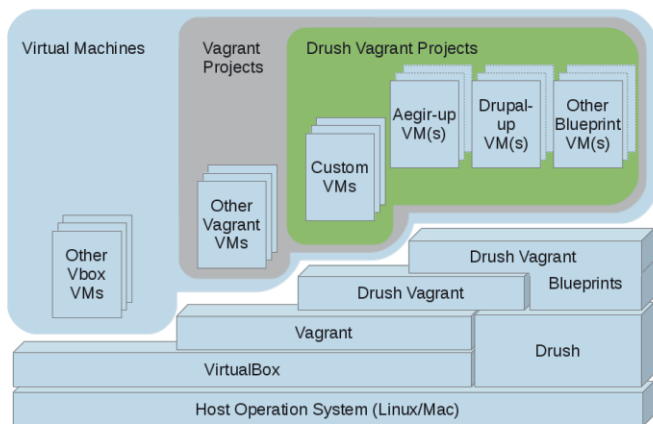
Vagrant и Puppet

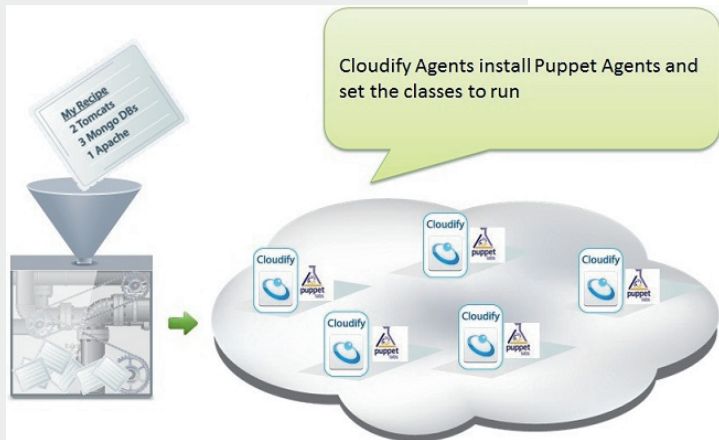
Sysadmini већ имају неку своју рутину у подешавању разноразних сервиса и то се своди на „инсталираш ово, и онда измениш ону датотеку, рестартујеш онај даемон, а све ти то пише на нашем *wiki-ju*“. То тамо пише, али је упутство застарело, компликовано је или програмер не жели да се бави тиме. Срећом на располагању су нам виртуелизација и софтвер који тај посао још више поједностављује. Алатима као што су *Vagrant* и *Puppet* можемо да креирамо нову виртуелну машину на нашем рачунару са жељеним инсталираним оперативним системом, а *Puppet* може да нам обезбеди инсталирање и подешавање осталих сервиса. Окружење из продукције можемо да добијемо у три команде: *vagrant init*, *vagrant up* и *vagrant ssh*. Врло корисна стварчица, јер ако кренемо уназад, и прво тако припремимо наше развојно окружење, онда немамо проблема са креирањем окружења за продукцију, јер ће *Puppet* исто тако једноставно инсталирати и

подесити све сервисе који смо изабрали. Причу о *Vagrantu* и *Puppetu* ћемо остати дужни за неки други број, а до тада проверите сами о чему се ради.

Други врло битан проблем који *Puppet* донекле решава су различите верзије софтвера за који развијамо наш код и који ће нас дочекати у продукцији. *Python* то решава на врло елегантан начин помоћу виртуелних окружења (*virtualenv - virtual environment*) и алата за управљање *Python* модулима. Предност *virtualenv+pip* комбинације у односу на инсталирање *Python* пакета из званичних ризница наше дистрибуције је у томе што за сваки пројекат можемо да направимо посебно радно окружење са својим сетом пакета и са одређеним верзијама тих пакета. Тако да, ако одлучимо да развијамо пројекат у односу на *Django 1.4.x* серију, а наша дистрибуција нуди 1.3.x или 1.5.x, *virtualenv* може да реши овај проблем. Штавише, ако имамо два пројекта на

истој машини који захтевају различите верзије пакета, то је сасвим нормална ситуација која се решава са два радна окружења. Уз *python* и *pip* је обавезно споменути и *virtualenvwrapper*, помоћну скрипту за *bash* која олакшава управљање и активирање виртуелних окружења, и наравно њихово креирање.





рестартуј сервисе на серверу и провери да ли је све завршено како треба;

2. *fab test*: покрени тестове и провери да ли су нове измене у коду можда пореметиле нешто;
3. *fab init*: повежи се на удаљени сервер, преузми конфигурациону датотеку за *puppet*

Библиотека *Fabric*

Сада већ имамо завидан сет алата у нашој радионици: *Vagrant* и *Puppet* за инсталирање и подешавање машине и потребних сервиса на њој, *pip* и виртуелне да обезбеде жељене верзије и да окружење независно од остатка оперативног система који смо наменили за продукцију. Уз ова два пара врло корисних алата, веома је користан и *fabric*. *Fabric* је *python* библиотека за аутоматизацију задатака разних врста. Предности које пружа *fabric* су скриптовање у *pythonu*, једноставна интеграција са *virtualenv* и многим другим алатима и подршка за управљање удаљеним машинама путем *SSH* протокола помоћу *paramiko* библиотеке. Циљ је да у једној датотеци (најчешће *fabfile.py*) дефинишемо функције и задамо упутства за аутоматизацију одређених задатака. Ево неколико идеја:

1. *fab deploy*: покрени тестове, ако се тестови успешно изврше додај нов код у репозиторијум, повежи се на сервер и освежи код на њему,

(манифест), постави потребне сервисе, подеси радно окружење, инсталирај зависности, преузми моју апликацију и иницијализуј базу података, рестартуј сервисе и провери да ли је све у реду.

Три сложена примера, који су наизглед компликовани да се искодирају, али врло брзо ћете се уходати и након тога ћете без размишљања бити спремни да клијенту у року од 15 минута решите проблем и надоградите му систем са потпуном сигурношћу да ће све проћи без проблема јер ако ради код вас, мора да ради и код њега.

Тестирање као саставни део програмирања

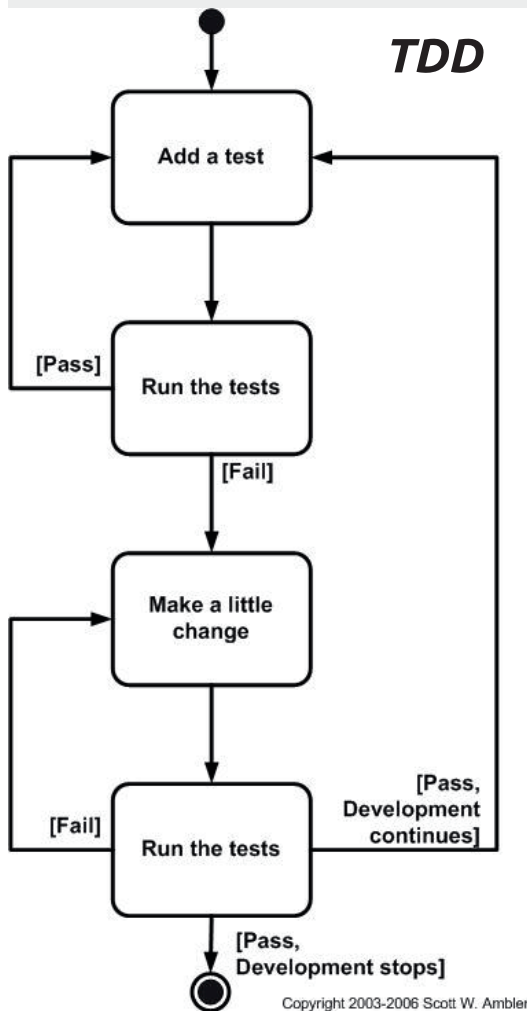
У претходном одељку смо више пута споменули тестирање. Један важан део развоја софтвера заузимају и разне врсте тестирања. Морамо бити уверени пре свега да је код синтаксно исправан, да може да буде покренут. Након тога, треба тестирати и



семантику целог система и видети да ли се код понаша очекивано. Тестирају се поузданост, робусност, перформансе и још много ствари. Почетком 2000-их година се појавио термин „agile software development“ који обухвата разне технике за развој софтвера на такав начин да се нове могућности брзо додају, грешке што примећују и исправљају и да се скрати време потребно за путовање кода од развојног окружења до пуштања у рад.

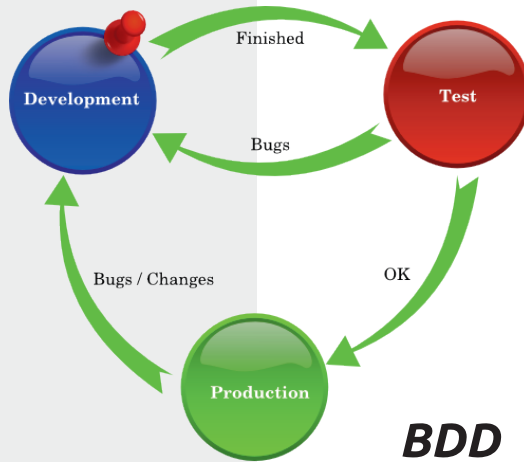
Саставни део оваквог начина развоја софтвера је писање тестова и појавили су се термини као што су Test Driven Development (TDD, развој вођен тестирањем) и Behaviour Driven Development (BDD, развој вођен понашањем софтвера). Два врло слична приступа, TDD је нешто старији него BDD, али оба приступа деле врло битне принципе. Један је да прво напишемо тест за могућност коју треба да имплементирамо, а потом додајемо или мењамо код све док та могућност не проради. Други принцип се директно надовезује на претходни и подразумева да измене у коду не смеју да покваре ни један други тест, односно нисмо завршили посао све док имамо неуспешних тестова.

У чему је предност и зашто би неко писао неколико пута више линија кода за тестове него за сам код који представља функционалност? Показало се да током животног века пројекта, 90% кода буде врло брзо написано, а потом следи пакао исправљања грешака. Управо ови тестови нам дају слободу да се слободно крећемо по коду и безбрижно га мењамо, јер ћемо



у сваком тренутку знати шта смо када покварили и моћи ћемо да погледамо чак и део кода који је у том тренутку измењен и ко га је изменио. Код TDD-а, тестови су у надлежности програмера, јер је он дужан да осмисли и имплементира тест примере. BDD је отишао корак даље и омогућио послодавцима да напишу тест примере језиком који разумеју, а

потом програмери на врло једноставан начин креирају парсирање и извршавање тих примера. Добит за програмере је непроцењива. Пре свега, послодавци су натерани да добро размисле о својим идејама и жељама које су некада толико фантастичне, да ни они сами не могу да их запишу речима, а очекују од програмера да их преточе у кôд. Друго, ако тестови који представљају све њихове тест примере успешно пролазе, а они се жале да нешто још не ради, увек можете од њих да затражите да вам покажу у којем се то примеру налази, а углавном ће они бити криви, јер ту могућност нису задали у спецификацији. Пројекти вредни пажње су *Cucumber*, *Nose Test*, *Lettuce*, *Robot Framework*, *Selenium* и мноштво других.



BDD

буде што сличнији, у чему нам помажу *pip* и *virtualenv*. Споменули смо и пример како на једноставан начин можемо наше исправке брзо да пласирамо клијенту и о све популарнијем начину развоја пројеката вођеним писањем тестова. Ово је само врх леденог брега, само појмови, алати и вештине сакупљене на једном месту.

Било би врло добро када бисмо могли да поразговарамо, разменимо искуства и можда мало скренемо пажњу *hosting* компанијама на *PaaS* услуге као што пружа *Heroku*, јер је то у интересу свима нама.

Закључак

Представили смо неколико практичних проблема у развоју софтвера уопште и неколико решења која су углавном везана за *Python* свет, али наравно да еквиваленти постоје и за друге „програмерске културе“. Писали смо о проблему постављања инфраструктуре који решавају *vagrant* и *pipet*, о томе да је важно да окружење за развој и за продукцију

Увод у

L^AT_EX

Аутор: Данило Ђокић

У овом броју часописа позабавићемо се са основним облицима форматирања, као што су поравнања, подебљан и искошен текст и слично. Пре него што почнемо са тим требало би да прво објаснимо шта је **околина** и шта је то **пакет**.

ОкоLINE (енгл. *Environment*) су посебан део синтаксе *LaTeX*-а, јер нам омогућавају да на одређени део кода применимо неки посебни вид обраде. ОкоLINE се користе увођењем наредби `\begin` и `\end` на следећи начин:

```
\begin{назив окоLINE}[опције  
окоLINE]  
\end{назив окоLINE}
```

Мада је ово стандардно, неретко се дешава да су опције окоLINE наведене у витичастим заградама (`{}`). Једну од окоLINE смо већ упознали и она се појавила у претходном броју, наравно то је окоLINE *document* која ће се појавити у сваком *LaTeX* коду и она указује компајлеру да се ту налази код који треба обрадити да би се он приказао у коначном документу. ОкоLINE можемо правити сами или увозити из разних пакета.

Пакети (енгл. *Package*) су већ припремљени низ окоLINE и наредби за неки специфични тип обраде докумената, те тако имамо пакете за додавање других писама (грчког алфабета, ћирилице и сл.), увођење слика у документ, исцртавање висококвалитетне векторске графике у самом коду и још пуно тога.

И пакете можемо сами да направимо, а можемо их и увести наредбом:

```
\usepackage[**потпакети**]{**назив  
пакета**}
```

Упознајмо се прво са неколико најосновнијих окоLINE и наредби које ћемо користити приликом форматирања текста, а за које нам нису потребни никакви нови пакети.

Основи форматирања

Поравнања (енгл. *justification*) извршавамо коришћењем окоLINE *flushleft*, *center* и *flushright*. **Подебљани текст** (енгл. *bold*) можемо произвести наредбом *textbf* а **искошени** (енгл. *italic*) коришћењем наредбе *textit*. Један пример који користи све ове опције би гласио:



```

\documentclass[a4paper]{article}

\begin{document}

\begin{flushleft}
\textit{Iskoseni tekst poravnan
uz levu ivicu}
\end{flushleft}

\begin{center}
\textbf{Podebljani centrirani
tekst}
\end{center}

\begin{flushright}
Poravnanje uz desnu ivicu
\end{flushright}

\end{document}

```

Рецимо да је тај ко̀д сачуван под називом *main.tex*. Тада би смо компајлирање истог обавили наредбом

```
pdflatex main.tex
```

(ова наредба се користи када имамо *texlive* компајлер који смо вам препоручили да инсталирате у предходном броју). Након тога он би изгледао као:

Слова српске латинице и српски наводници

Због далеко веће једноставности писања српске латинице у *LaTeX*-у, њу уводимо прву (у каснијим бројевима објаснићемо и увођење ћирилице). Слова која се у њој појављују са „квачицом” користе наредбе да би се приказивале (класа наредби које производе слова специфична за неко друго писмо осим енглеског алфабета називају се **специјални карактери**). Наредбе `\v{ }` и `\' { }` ће доцртати одговарајућу квачицу на слово које је њихов аргумент. Па ће тако табела **специјалних карактера** српског језика изгледати овако:

č = `\v{ c}`

ć = `\v{ c}`

ć = `\' c`

č = `\' c`

ž = `\v{ z}`

ž = `\v{ z}`

š = `\v{ s}`

s = `\v{ s}`

Iskoseni tekst poravnan uz levu ivicu

Podebljani centrirani tekst

Poravnanje uz desnu ivicu



Као што можемо приметити, ове наредбе се могу писати и на мало стандарднији начин тиме штедећи време и дужину кода. Ово потиче од синтаксе самих наредби. Како наредбе могу бити слова енглеског алфавета (у неким случајевима и јединствени карактери), наредба `\v` препознаје први знак иза белине као њен аргумент (белина може бити произвољан број размака, табулатора или нових линија). А наредба `\'` препознаје непосредно први карактер иза ње као њен аргумент. Са словом `\D` је ситуација за нијансу другачија. Да би се користило ово слово без неких посебних потешкоћа, треба увести пакет *fontenc* посебно његов део `T1` (`\usepackage[T1]{fontenc}`), тада се оно може добити наредбама `\DJ` и `\dj`.

Српске наводнике је најлакше увести дефинисањем нове наредбе без аргумента. Нећемо се сада бавити дефинисањем нових наредби, за сада ћемо само рећи да је потребно у преамбули додати следећи код:

```
\def \dvn{\, \kern-0.09em,}
\def \gnv{\` }
```

И сада ћемо доње и горње наводнике добијати као специјалне карактере `\dvn` и `\gnv` редом. Као неки пример за ово наведимо обраду одсечка чланка о *BalCCCon-y* из прошлог броја у *LaTeX-y*:

```
\documentclass[a4paper]{article}

\usepackage[T1]{fontenc}

\def \dvn{\, \kern-0.09em,}
\def \gnv{\` }

\begin{document}
Po\v cetkom septembra, ta\v cnije
6. i 7. septembra, u Novom Sadu u
Master centru, odr\v zan je prvi
balkanski ra\v cunarski kongres
koji je privukao pa\v znju kako
lokalnih, tako i hakerskih
zajednica \v sirom svijeta. Ovaj
doga\dj{}aj je jasan indikator
nivoa kvaliteta \textit{IT}-a
balkanskih zemalja. Prvi kongres
je specifi\v cno nazvan \dvn
First contact\gnv - na na\v sem
jeziku \dvn Prvi kontakt\gnv.
\end{document}
```

Што би након компајлирања изгледало као:

Početkom septembra, tačnije 6. i 7. septembra, u Novom Sadu u Master centru, održan je prvi balkanski računarski kongres koji je privukao pažnju kako lokalnih, tako i hakerskih zajednica širom svijeta. Ovaj događaj je jasan indikator nivoa kvaliteta *IT*-a balkanskih zemalja. Prvi kongres je specifično nazvan „First contact“ - na našem jeziku „Prvi kontakt“.



Приметимо неколико важних ствари, пре свега нови ред се не обрађује директно. Он је еквивалентан било којој белини и биће препознат само као раздвајање речи, наредби и сл. Да бисмо увели прави нови ред, треба користити наредбу `\` (двоструки *backslash*). Једини начин да нови ред има посебног утицаја су два узастопна знака за нови ред (што значи једна празна линија између редова) која ће означити нови ред и увлачење текста. Ово увлачење је дефинисано и за прву реч у тексту. Ако некад желимо да избегнемо увлачење текста испред речи која ће по стандардној синтакси бити увучена стављамо наредбу `\noindent`, а ако се пак по синтакси не би увлачио текст а ми то ипак желимо, испред речи коју треба увући стављамо наредбу `\indent`.

Закључак

У овом броју објаснили смо основе форматирања текста на које ћете наићи у сваком текст-процесору. У следећем броју објаснићемо неке особености везане за сам *LaTeX*, као што је јединствени начин формирања наслова, бројаче, и за *LaTeX* чувено и јако популарно укрштено референцирање (тзв. *cross-референцирање*).

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

Distrowatch

1	Mint	3714>
2	Ubuntu	2457<
3	Debian	2140>
4	Mageia	1843>
5	PCLinuxOS	1464<
6	openSUSE	1348>
7	Fedora	1339<
8	Manjaro	1157<
9	Ubuntu GNOME	1043<
10	CentOS	997>
11	Arch	975<
12	Lubuntu	860<
13	FreeBSD	850<
14	Puppy	840>
15	elementary	810>
16	Ultimate	789>
17	CrunchBang	785>
18	Simplicity	668>
19	Kubuntu	657<
20	Xubuntu	656<
21	Zorin	654<
22	SparkyLinux	635<
23	Slackware	630>
24	OS4	626<
25	wattOS	603>

Пад <

Пораст >

Исти рејтинг =

(Коришћени подаци са *Distrowatch*-а)



Firefox OS 1.1

Аутор: Златан Васовић

Нова верзија *Firefox OS*-а је објављена само неколико недеља након што су се први *Firefox OS* паметни телефони појавили на тржишту. Свежа верзија доноси нове могућности, побољшања перформанси и нове преводе. Кодно име нове верзије је *Leo* и он долази са *Gecko*-ом верзије 18+.

Могућности

Могућности су знатно проширене у новој верзији *Firefox OS*-а.

Оно што ће сигурно бити драго млађим корисницима је *MMS* подршка, поред класичног *SMS*-а.

Прилагодљива претрага је премештена на почетни екран, тако да не морате да мењате почетни екран како бисте стигли до претраге. Од помоћи може бити *SUMO (Mozilla Support)* чланак – <https://support.mozilla.org/en-US/kb/what-happened-to-adaptive-search>.

Gmail и *Hotmail* (заправо *Outlook*)



контакти се могу увести директно, као на осталим мобилним оперативним системима. Сада у контакте можете додати било кога ко вас је позвао,



послао вам *SMS*, *MMS* или *e-mail*; не морате више да копирате и лепите (*copy paste*) бројеве телефона или *e-mail* адресе. Додата је и подршка за сугестије приликом куцања бројева или имена контаката, наравно уз аутоматско допуњавање.

Још једна битна промена је могућност преузимања слика, видео и аудио материјала директно из *Firefox* прегледача.

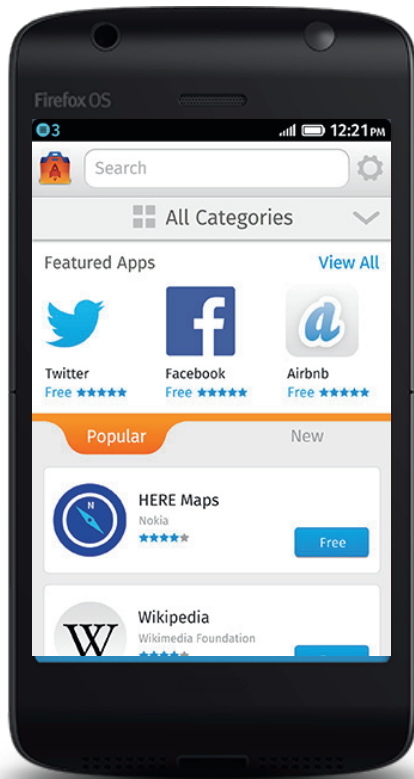
Ваши недовршени *e-mail*-ови се аутоматски чувају, тако да их можете завршити касније, када не будете заузети. Сада можете сачувати видео и аудио прилоге из *e-mail*-ова и слати слике. Планира се омогућавање и слања видео и аудио материјала.

Музику је могуће једноставно претражити, једноставним превлачењем са врха апликације ка дну. Од помоћи можете бити *SUMO* чланак – <https://support.mozilla.org/en-US/kb/play-music-on-firefox-os>.

За оне који греше при куцању, ту је аутоматска исправка основних граматичких и правописних грешака. Нажалост, та опција још није прилагођена српском језику, што се баш и не може очекивати у скорије време.

Прављење догађаја је поједностављено и може се извести притиском на неки датум у календару. Ту су и подсетници за догађаје.

Програмерима ће бити интересантан *Push Notifications API* који знатно побољшава временско одређивање



обавештења и смањује потрошњу батерије.

Перформансе

Систем је додатно оптимизован тако да је учитавање апликација тренутно. Скоровање је такође убрзано коришћењем нових *web* технологија.

Ова побољшања су резултовала да упркос слабијем хардверу у односу на нове вишејезгарне *Android* и *iOS* телефоне, *Firefox OS* уређаји раде брзо.



Интернационализација

Firefox OS је сада преведен на 14 језика, што и није тако мало с обзиром на број *Firefox OS* волонтера. Оно што је заиста импресивно је да је *Firefox OS* преведен и на српски, па је спреман за наше тржиште.

Развојне компоненте

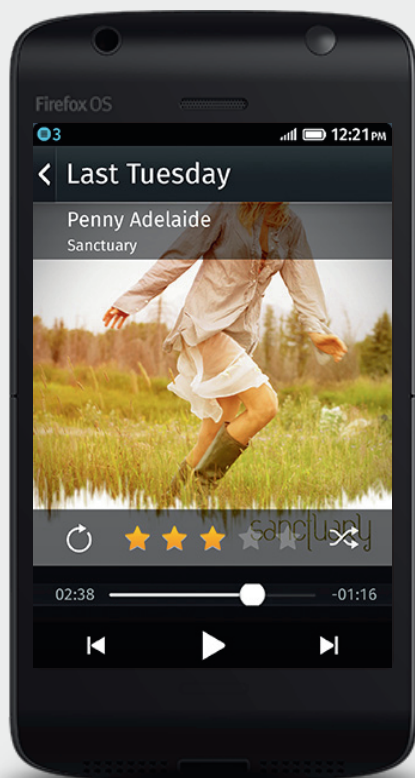
Основу *Firefox OS*-а чине компоненте отвореног кода. Три основне компоненте су *Gonk* (језгро), *Gecko*

(функционалност) и *Gaia* (изглед).

Gonk [1] је језгро система базирано на свеprisутном *Linux* керналу. Он садржи неке драјвере за *Android* уређаје, првенствено из *hardware abstraction layer*-а (*HAL*). У будућности се планира шира хардверска подршка.

Gecko [2] је покретач апликација. Он имплементира отворене стандарде за *HTML*, *CSS* и *JS*. Он је главна компонента *Firefox* прегледача и *Firefox OS*-а.

Gaia [3] је најједноставније речено





кориснички интерфејс *Firefox OS*-а.

Изворни код *Firefox OS*-а можете наћи на <https://github.com/mozilla-b2g/>.

Корисни линкови:

- [1] https://developer.mozilla.org/en-US/docs/Mozilla/Firefox_OS/Platform/Gonk
- [2] <https://developer.mozilla.org/en-US/docs/Mozilla/Gecko>
- [3] <https://wiki.mozilla.org/Gaia>

Развојни уређаји

Постоје и посебни уређаји за програмере. Они увек имају свежу верзију *Firefox OS*-а и могу да користе развојне верзије апликација. Најпознатији је *Geekspone*.

Развојни телефони:

- <http://www.geekspone.com/>
- https://developer.mozilla.org/en-US/docs/Mozilla/Firefox_OS/Developer_phone_guide/ZTE_OPEN



Закључак

Firefox OS 1.1 делује као интересантан систем који има битне компоненте *Android* и *iOS* дизајна и има добру функционалност, али троши много мање ресурса од својих конкурената.

Уколико имате проблем са *Firefox OS*-ом, погледајте развојно упутство (https://developer.mozilla.org/en-US/docs/Mozilla/Firefox_OS) и чланке на *Mozilla Support*-у (<https://support.mozilla.org/en-US/products/firefox-os>).



CRYPTO PARTY No.2

CryptoParty

CryptoParty je svetska serija AdHoc događaja usmerenih na širenje informacija i znanja o privatnosti, sigurnosti i alatima za kriptovanje.

ŠTA?

Drugi crypto party biće u fokusu mail servisa. Objasnićemo zašto je bitno koristiti mail kao secure servis, kako podesiti mail da možete da primete i šaljete kriptovane poruke.

GDE? KADA?

Univerzitet u Novom Sadu
Fakultet tehničkih nauka – FTN
Trg Dositeja Obradovića 2
učionica broj 208, drugi sprat
Subota, 09. novembra 2013. godine u 15.00h

