

Atskaite par kursu *Tīkla operētājsistēmas III* (DatZ3191)

Jānis Erdmanis

2013. gada 6. aprīlis

Linux Mint skolai un mājām

Darba uzdevumi

- Uztādīt *Linux Mint* dotajam datoram
 - Veikt manuālu diska dalīšanu
 - Uztādīt latviešu valodu saskarsnei un klaviatūrai
 - Konfigurēt *GRUB2*, lai noklusējums būtu jau esošais *Windows 7*
- Veikt *OS* atjaunināšanu gan komandrindā gan ar piedāvātajiem *GUI* rīkiem.
- Izskatīt piedāvāto *Linux* programmu alternatīvu sarakstu, lai varētu sekmīgi migrēt no *Windows*
- Piedāvāt savas *OSS* alternatīvas
- Uztādīt *VirtualBox*

Darba piederumi

- *Linux Mint 13 XFCE* DVD ierakstīta instalācija
- Interneta pieslēgums ar aptuveni novērtējamo ātrumu *5Mbps*
- Dators:
 - *4GB* operatīvā atmiņa, kas ir pietiekami, lai uz *VirtualBox* instalētu citu *OS*
 - divkodolu procesors
 - *500 GB* cietais disks, kas nav piepildīts
 - perifērija - monitors,...,datorpele

Darba gaita un rezultāti

Veicot datora cietā diska sadalījuma izpēti, tika pieņemts lēmums visu *Linux Mint* instalēt jau pieejamās tukšās partīcijas *20GB* vietā. Tika noteikts šāds diska sadalījums:

- **/home** 10 GB, kur šī kursa ietvaros galvenokārt glabātos *VirtualBox* iedalītā diska vieta *Ubuntu 12.04 server* operētājsistēmai.
- **/boot** 150 MB, kur atrodas *GRUB2* un *Linux* kerneli, kas var būt noderīgi, kad tiek instalēta paralēli cita OS¹.
- **/** 9 GB, kur glabāsies OS un pieinstalētās programmas
- **SWAP** 1 GB, kura pie dotā operatīvās atmiņas apjoma ir pilnīgi pietiekama, ja vien nav nepieciešams izmantot *hibernate* režīmu.

Pēc tam, kad šis svarīgais uzdevums tika veikts, tika piedāvātas izvēlnes, lai iestatītu gan valodu, gan uzstādītu pareizu laiku. Tās laikā varēja pārlūkot internetu, kas nav iespējams Windows instalācijas laikā.

Instalācijai sekmīgi noslēdzoties tika veikta datora pārstartēšana. Kā būtisks jaunums parastam lietotājam varētu likties jauns melns logs *GRUB2*, no kura automātiski ielādējās *Mint*², kas arī noteica, ka pēc iespējas ātrāk nepieciešams izmainīt iestatījumus, lai automātiski tiktu ielādēts *Win7*³.

Internetā tika atrasts, ka jāredīgē `/etc/default/grub` fails, kas `update-grub` komandas automātiski ģenerē `/boot/grub/grub.cfg`. Redzams, ka šāds failu izkārtojums būtu noderīgs, lai katrs jaunais *Unix* varētu arī nepārrakstīt *GRUB2* vai arī, ja nolemts OS likvidēt, tad var atstāt tikai `\boot` partīciju un izdzēst liekās rindas no `/boot/grub/grub.cfg`.

Turpinot iepazīšanos ar *Mint* tika veikta tā atjaunināšana ar `sudo apt-get update; sudo apt-get upgrade -y` komandu terminālī. Pēc tam tika pārbaudīts vai ir pazuduši piedāvātie atjauninājumi no grafiskā risinājuma, kas manuprāt rada lēnuma sajūtu strādājot ar jebkura tipa *Unix*.

Lai iepazītos ar piedāvātajām *OSS* alternatīvām, tika izmantots gan komandrindas gan grafiskais instalētājs, kur otrais īpaši noderēja, kad vajadzēja sameklēt, kādas programmas nosaukumu pēc tās funkciju apraksta. Tās izmantošana ļāva atrast labāk izstrādātu *Paint* alternatīvu par *TuxPaint* - *MyPaint*.

Pārsvārā liela daļa *OSS* programmu dotajā tabulā vēl joprojām tiek aktīvi attīstītas, kas nozīmē, ka *Linux* literatūra tik ātri nenoveco. Kā labāku alternatīvu par *libre-office-calc* uzskatu *gnumerics*, jo tajā bez papildu spraudņiem pieejama regresijas analīze, kuru (programmu) aktīvi izmantoju pirmo gadu izstrādājot fizikas praktikuma laboratorijas darbus. Manu vislielāko uzmanību piesaistīja programma *Geany*.

Tas vien, ka to var izmantot ļoti plašam programmēšanas valodas klāstam⁴, lai bez jebkādas modifikācijas ar to varētu kompilēt un izpildīt komandas, padara to par ļoti labu alternatīvu *IDE* vide *Codeblocks*. Ja vien būtu iespējams nomainīt šīs programmas izskatu, lai tā būtu ar gaišiem burtiem uz tumša fona, tad tā kļūtu par manu noklusējumu *gedit* vietā, rakstot *python* modulus vai šo atskaiti *pandoc markdown* formātā.

Kā ļoti labu alternatīvu programmēšanas valodu, ko varētu mācīt skolēniem, uzskatu *python*, jo tā:

- Liek lietotājam lietot tabulāciju, kas būtiski palielina koda pārskatāmību

¹<http://www.linuxquestions.org/questions/linux-newbie-8/separate-boot-partition-why-933988>

²Turpmāk izmantotais saīsinājums pilnajam - *Linux Mint 13 XFCE*

³*Microsoft Windows 7*

⁴Tika pārbaudīts ar *free pascal*, *C*, *C++*, *python* programmēšanas valodām. Ja kompilators uz datora nav uzstādīts tad tiek izmests kļūdas paziņojums ar to, kāds kompilators tiek meklēts, kas ir ērti, lai to pieinstalētu `apt-get` komandu, piemēram, `sudo apt-get install fpc`, lai darbotos ar *free pascal* rakstītām programmām.

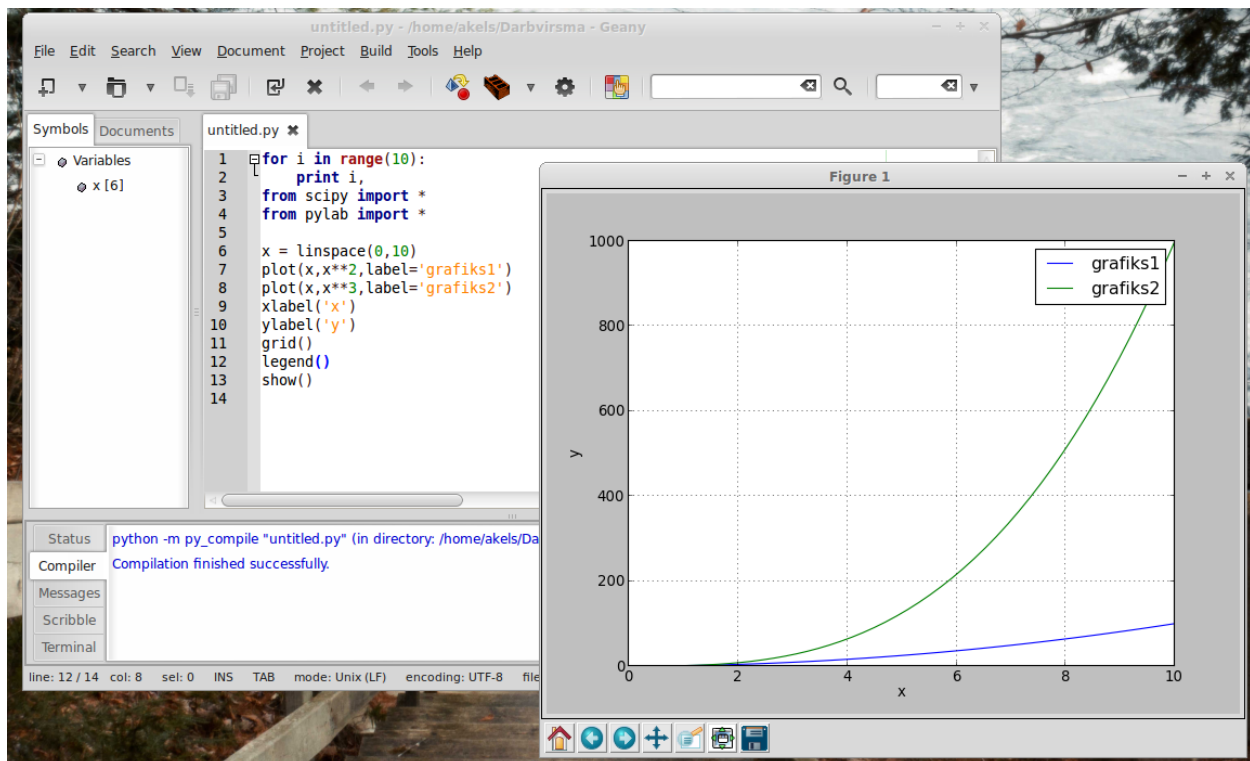


Figure 1: Geany ar python programmēšanas valodas pielietojumu. Redzams *pylab* bibliotēkas pielietojums, lai zīmētu grafikus, kas skolā būtu noderīgi mācot matematikā par elementāru funkciju grafiku transformācijām.

- Ļauj veikt cikliskas operācijas ar masīviem tos neindeksējot, kas 10 rindiņas C valodā var tikt uzrakstāmas 1 python rindā. Turklāt šī python rinda paliek tik nemainīga dažādos pielietojumos, ka skolēns spētu ātri vien atrast tās pielietojumu.
- Jāatceras mazāk komandu, lai panāktu, ka vienkārša programma strādātu
- Valodai ir izaugsmes iespējas, jo tai ir ļoti liels klāsts ar dokumentētām bibliotēkām un augošs lietotāju īpatsvars.

Programmēšanas valodai ir bibliotēka arī *.csv* failu ielasišanai un apstrādei, ko es izmantoju *gnumerics* vietā, lai reizi divās nedēļās apstrādātu ievāktos laboratorijas datus. Uzskatu, ka tā ir arī alternatīva *MS Excell* arī skolā, jo datus var apstrādāt interaktīvi *ipython qtconsole* aplikācijā.

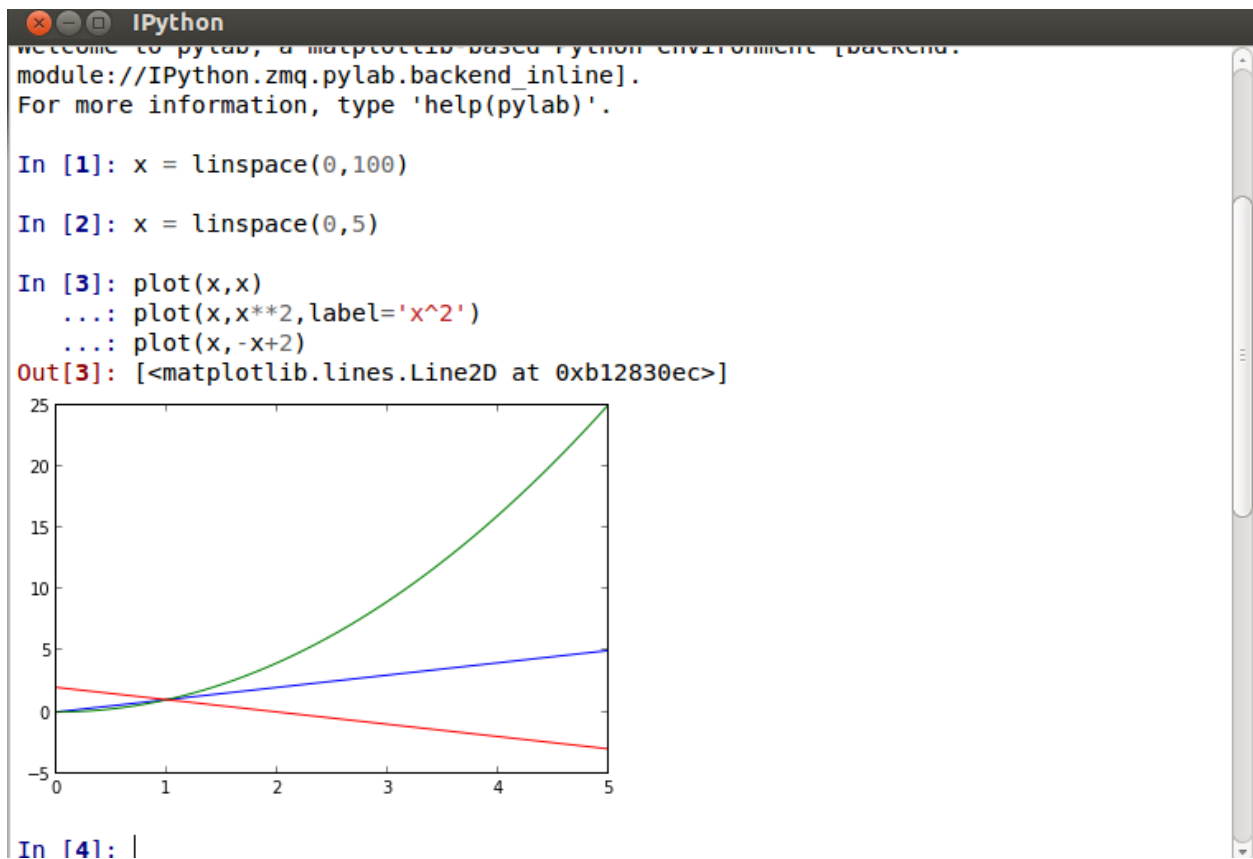


Figure 2: Modernā programmēšana interaktīvi caur *ipython qtconsole*. Šo aplikāciju izmantoju, kad apstrādāju arī savus laborus

Par nākotnes rediģēšanas alternatīvu redzu *pandoc markdown*, jo tā ļauj rakstīt informāciju jebkurā teksta redaktorā nedomājot par formatēšanas rezultātu, kā to es izmantoju rakstot šo atskaiti.

Izskatot visu alternatīvu sarakstu turpināju ar *virtualbox* uzstādīšanu, kura ļaus turpmāk to izmantot, lai instalētu *Ubuntu serveri*. Tas bija pieejams oficiālajā *paku datubāzē*, tādēļ tas problēmas nesagādāja.

Secinājumi

Esmu iepazinies ar *Mint* un piedāvāto *OSS* sarakstu, kurā atradu *Geany* citā perspektīvā. Piedāvāju savas alternatīvas ikdienas uzdevumiem ar datiem un piedāvāju arī savu viedokli par *python* programmēšanas valodu skolā, un veidu, kā to izmantot.

Daudzfunkcionāla skolas Linux servera izveide

Darba uzdevumi

- Veikt *Ubuntu Server 12.04* uzstādīšanu uz *VirtualBox*
- Uzstādīt un nokonfigurēt *SSH*, lai turpmāk varētu serveri pārvaldīt caur to
- Uzstādīt Moodle jaunāko versiju
- Izveidot koplietojamo mapi, kur atsevišķi būtu atdalīta skolēnu un skolotāju mape. Katram lietotājam būtu mape atbilstošajā kategorijā, kur lietotājs spētu piekļūt tikai vienai mapei savā kategorijā, bet katram skolotājam būtu piekļuve jebkurai skolēna mapei.

Darba gaita

Darbs tika sākts ar *VirtualBox* sagatavošanu, kurā ietilpa resursu piešķiršana, galvenokārt cietā diska vietas iedalīšana, ko varēja turpināt ar *Ubuntu* instalāciju norādot tikko nolādētā *.iso* faila vietu.

Instalēšana notika caur komandrindas saskarsni, kas bija labi noorganizēta, lai šo procesu padarītu pēc iespējas mazāk sāpīgāku. Tās laikā tika noteikts šāds partīciju sadalījums:

- `\www` 2 GB, kur glabāsies moodle
- `\home` 2 GB, viens liels ar `\www`, jo šeit atradīsies koplietojamā mape
- `\` 3 GB, sistēmas mape

Tika atzīmēts, ka serverim nepieciešamas *metapakas* - *lamp*, *samba*, *ssh*.

Pabeidzot instalāciju tika pārstartēts servera OS un tika sākts ar *ssh* konfigurēšanu. Zināmus sarežģītumus radīja *VirtualBox* ip adreses piešķiršanas mehānisms, kas pēc noklusējuma neļauj pieslēgties ar *ssh* norādot tīkla adresi, tādēļ tika rediģēts `/etc/network/interfaces` fails. Pēc tam tika rediģēts `/etc/ssh/sshd_config` fails, ar kuru tika panākts, ka virtuālo serveri var pārvaldīt no jebkuras vietas.

Darbu turpināju ar moodle instalāciju, kur datu bāzu izveidošanai tika izmantots rīks *phpmyadmin*, kura saskarsne bija pietiekami vienkārša, lai veiktu nepieciešamo uzdevumu. Tālāk bija nepieciešams sagatavot mapes moodles sistēmai un tās failiem, kas tika darīts caur *ssh*. Izmantojot *gftp* tika serverī ielādēti moodle instalācijas faili, kas ļāva sākt instalāciju no pārlūkprogrammas. Instrukcijas bija labi saprotamas, kā rezultātā tika nokonfigurēts moodle serveris. Lai pārbaudītu vai tas strādā, kā nākas tika izvēlēta jauna saskarsnes tēma un izveidots pirmais kurss.

Lai panāktu pilnīgu skolas servera darbību būtu nepieciešams, lai skolotāji, skolēni tajā varētu augšuplādēt failus un tos pēc vajadzības dzēst, kārtot. Šī funkcija tiek panākta ar *SAMBA*, kuras koplietojumiem var piekļūt arī ar Windows. Tās iestatīšanai var tiešā veidā rediģēt `/etc/samba/smb.conf` vai arī izmantot aplikāciju *webmin*, kas arī tika darīts.

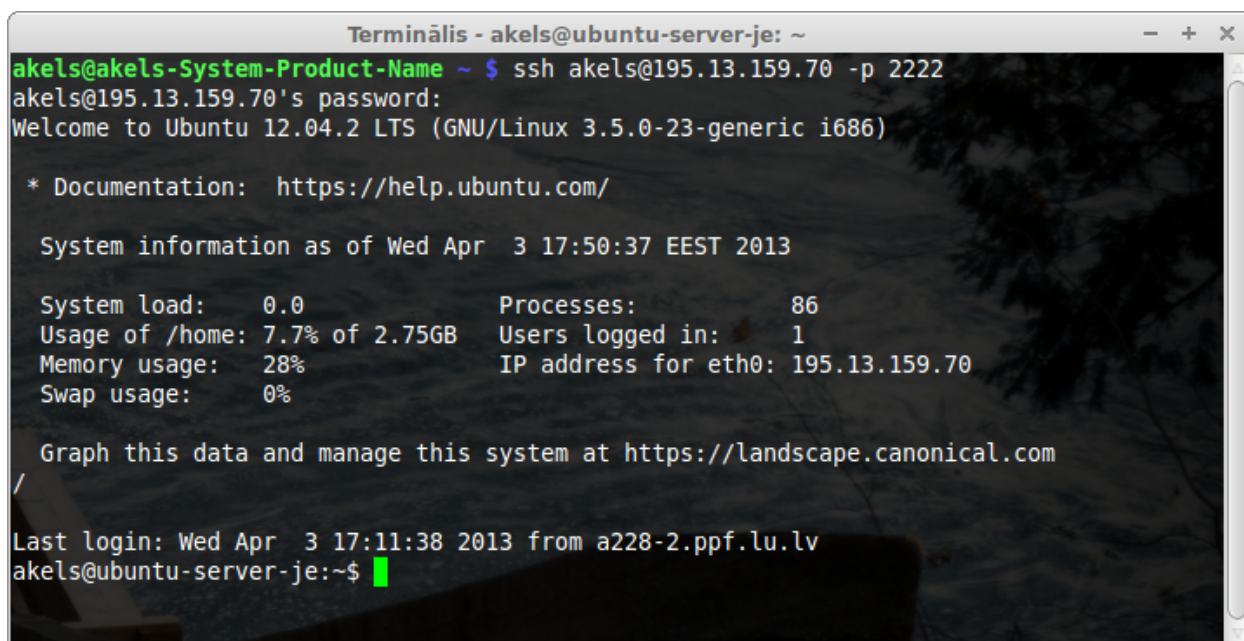


Figure 3: Servera darbības demonstrējums, kur tam esmu pieslēdzies ar ssh

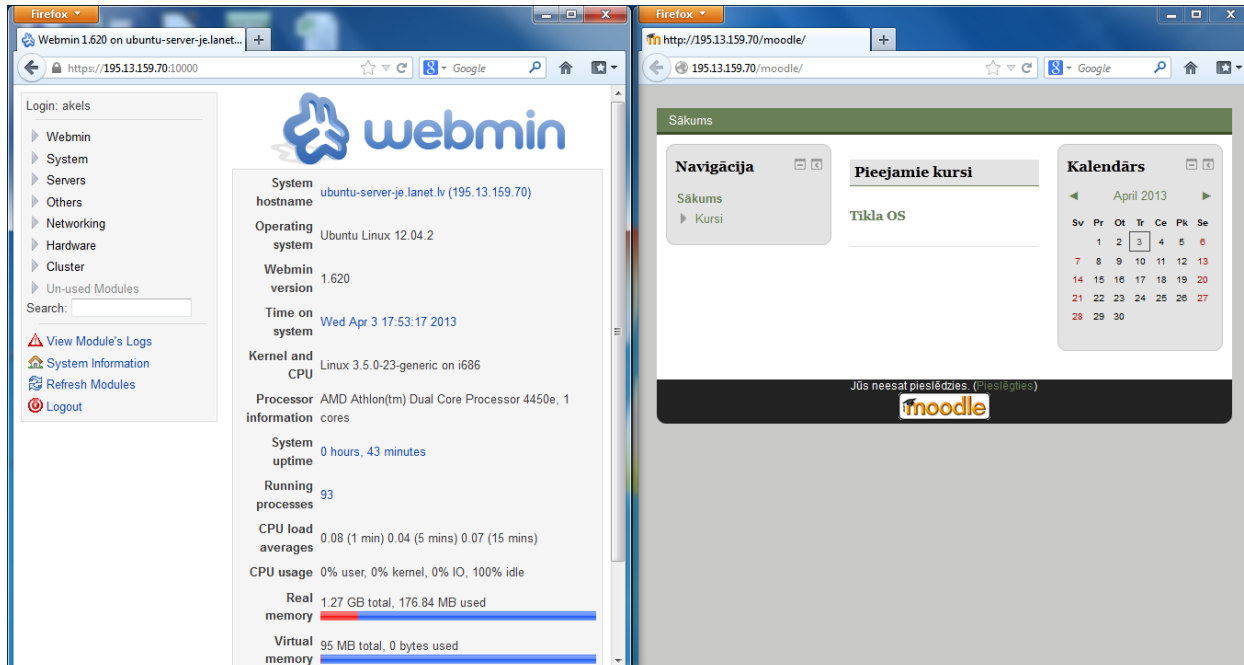


Figure 4: Webmin un Moodle no cita klases datora

Grūtības sagādāja *webmin* saskarsnes sarežģītība šim vienkāršajam uzdevumam tādēļ salīdzinot to ar tiešu *smb.conf* rediģēšanu var secināt, ka *webmin* lietošana var radīt problēmas, lai spētu dokumentēt, kā ir iestatīts serveris.

Secinājumi

Visvienkāršāk serveri ir iestatīt izmantojot komandrindu un rediģējot */etc* mapes failus, jo tādas dokumentācijas ir lasīt vienkāršāk, tādēļ darbs ritētu raitāk, ja *GUI* rīki netiktu pieminēti. Turklāt pieslēgšanās ar *gftp*, lai ielādētu tajā atarhivētu arhīvu bija pārāk sarežģīta un lēna salīdzinot *scp* komandu, kura nāk komplektā ar *ssh*. Turklāt izmantojot tikai komandrindu var visas darbības apkopot vienā *.sh* failā, kuru izpilda uzreiz pēc servera OS uzstādīšanas tādējādi tas iegūtu *POST-SCRIPT* apzīmējumu uzstādot nākamo serveri.