

**VENTSPILS AUGSTSKOLA**

INFORMĀCIJU TEHNOLOĢIJU FAKULTĀTE

**Patstāvīgais darbs**

studiju kursā

„Tīklu operētājsistēmas”

**FreeBSD 6.2 + SSH + Apache + PHP +  
+ phpMyAdmin + MySQL + FTP**

**Autors:**

Adrians Heidens

Ventspils, 2007

1. IEVADS .....	4
2. FREEBSD INSTALĀCIJA .....	6
2.1. FreeBSD iegūšana .....	6
2.2. Instalācijas sākums .....	6
2.3. Fdisk Partition Editor - taisam daļas (slices).....	8
2.4. Boot Manager izvēlne .....	10
2.5. Partīcijas ar Disklabel.....	11
2.6. Izvēlēties, ko instalēt .....	12
2.7. Instalācijas beigu daļa .....	13
2.8. Pēcinstalācijas konfigurācija.....	14
3. APACHE WEB SERVERIS .....	17
3.1. Instalācijas process.....	17
3.2. Apache darbināšana .....	19
4. PHP5 MODULIS .....	23
4.1. Instalācija un konfigurācija .....	23
4.2. PHP tests.....	24
5. MYSQL DATU BĀZU SISTĒMA .....	26
5.1. MySQL instalācija .....	26
5.2. MySQL darbināšana un tests.....	27
5.3. Lietotāju pievienošana.....	27
5.4. MySQL - PHP tests .....	28
6. PHPMYADMIN .....	29
6.1. phpMyAdmin instalācija.....	29
6.2. phpMyAdmin konfigurācija .....	29
7. FTP PIEEJA .....	31
7.1. FTP iedarbināšana .....	31

7.2. Lietotāju pievienošana, FTP un SSH konfigurēšana.....	32
<b>8. LIETOTĀJU SAGATAVOŠANA DARBAM.....</b>	<b>34</b>
8.1. MySQL sagatavošana.....	34
8.2. Apache sagatavošana.....	35

# 1. Ievads

*Izmantota informācija no FreeBSD mājas lapas*

<http://www.freebsd.org/>

FreeBSD ir bāzēts uz 4.4BSD-Lite, ko izstrādāja **Computer Systems Research Group (CSRG)** Kalifornijas universitātē Bērklījā. FreeBSD nes sev līdzīgu BSD sistēmu attīstības tradīcijas, kur projekta dalībnieki tūkstošiem stundu ilgi strādā pie tā, lai sistēma būtu perfekta reālām dzīves situācijām.

FreeBSD operētājsistēmu var izmantot ļoti dažādiem mērķiem sākot no lietotņu izstrādi, rūpnīcas darba automatizācijas kontroli līdz pat satelīta navigācijas kontrolei.

Pateicoties tam, ka operētājsistēmas kods ir brīvi pieejams, to var pielāgot jebkuram darbam.

Daži veidi, kā izmantot FreeBSD:

- **Interneta servisi:**
  - FTP serveri
  - WWW serveri, standarta un SSL
  - Ugunsmūri un NAT vārtejas
  - E-pasta serveri
  - USENET grupas
- **Mācību nolūks:** FreeBSD patiešām ir ļoti labs, lai apgūtu operētājsistēmas. Pieejami labi materiāli.
- **Tīkla lietas:** Ja vajadzīgs maršrutētājs vai DNS serveris, vai ugunsmūris, tad bez problēmām var pārvērst savu veco datoru, kas mētājas stūrī par ļoti attīstītu maršrutētāju.
- **Desktop:** Varat izmantot FreeBSD, kā desktop. FreeBSD var uzinstalēt X Window sistēmu un Gnome vai KDE un lietot, kā desktop

operētājsistēmu.

- **Programmēšana:** FreeBSD nāk komplektā ar nepieciešamiem rīkiem programmēšanā, kā, piemēram, GNU C/C++ kompilators un atklūdotājs.



## 2. FreeBSD instalācija

*Materiāli ņemti no FreeBSD rokasgrāmatas*

[http://www.freebsd.org/doc/en\\_US.ISO8859-1/books/handbook/](http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/)

### 2.1. FreeBSD iegūšana

Lai uzinstalētu FreeBSD, sākumā mums vajag dabūt kaut kādu instalācijas avotu.

Mans avots nāk no FreeBSD FTP servera.

- <ftp://ftp.freebsd.org/pub/FreeBSD/ISO-IMAGES-i386/6.2>

Tur atradīsiet četrus ISO diskus. Es izmantoju šo:

- 6.2-RELEASE-i386-disc1.iso

Tajā ir viss, kas vajadzīgs instalācijai. Es zinu, ka daudzi izmanto *bootonly* disku, jo tādā gadījumā esot mazāks risks uzdukties kādai I/O kļūdai. Ar *bootonly* disku instalācija notiek no FTP serveriem.

Par pārējiem diskkiem... Tātad, 2 jau ir zināmi. CD2 ir programmas un iespējams kādas dokumentācijas. Docs disks, kā jau pēc nosaukuma var spriest, satur dokumentācijas.

Tālāk ierakstīju ISO failu reālā diskā, lai varētu sākt normālu standarta instalācijas procesu.

### 2.2. Instalācijas sākums

Lai sāktu instalāciju, mums ir nepieciešams pateikt datoram, lai startē no CDROM, nevis HDD. To var izdarīt iekš **BIOS**. Ja nemāk, tad var pameklēt

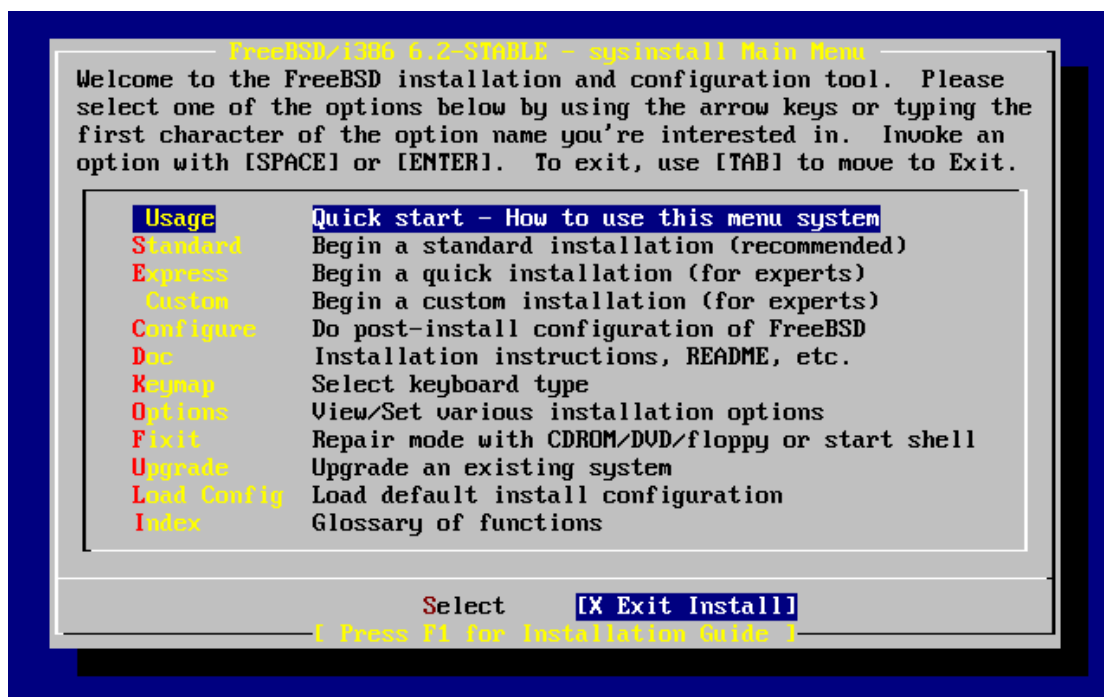
iekš [www.google.lv](http://www.google.lv)

Turpinām ar to, ka ieliekam disku diskdzinī un startējam no tā, startējas FreeBSD instalēšanas sistēma. Mums tiek piedāvātas dažādas *boot* iespējas, izvēlamies noklusēto (default), spiežot Enter.

Turpinās ielādes process, šeit varam redzēt, kā sistēma atpazīst dzelžus. Lai apskatītu aizgājušās rindas, spiežam Scroll Lock un ar bultiņām, vai Page up un Page Down braukājam pa ierakstiem.

Turpinot, mums tiek piedāvāta izvēle starp reģioniem. Izvēlamies Latviju. Vēl tiek piedāvāta iespēja izvēlēties *keyboard layout*. Izvēlējos noklusēto USA ISO.

Tālāk parādās logs.



Nekavējamies šeit un izvēlamies standarta instalāciju (rekomendēto izvēli).

FreeBSD disku numerācija notiek šādā veidā:

tips+nummurs

- ad ATAPI (IDE) disk
- da SCSI direct access disk
- acd ATAPI (IDE) CDROM
- cd SCSI CDROM
- fd Floppy disk

Numerācija ir atkarīga no tā, kurā vietā atrodas ierīce. Piemēram, ja IDE cietais disks atrodas pie IDE1, kā primārais, tad tas saucas ad0. Ja sekundārais, tad ad1.

Ja IDE cietais disks atrodas pie IDE2, kā primārais, tad tas saucas ad2, neatkarīgi no tā, vai pie pirmā porta ir kāds disks.

Ja parādās brīdinājums apmēram tāda veida - „**A geometry of 2552926/1/63 for ad0 is incorrect**”, tad neņem to par pilnu. Man šāds brīdinājums bija, mēģināju uzlikt citu ģeometriju, kas man rādās iekš BIOS, bet tad arī tas pats brīdinājums. Vārdu sakot, spiediet OK. Lasīju, ka arī cietiem ir tāda problēma bijusi, bet nekādas sekas tam nav novērotas.

### 2.3. Fdisk Partition Editor - taisam daļas (slices)

Tagad rīkojamies ar **Fdisk Partition Editor**, kurā izvēlamies, kādas partīcijas, FreeBSD tās sauc par *slices*, veidojam. Sauksim tās par daļām. FreeBSD ar vārdu partīcija (partition) saprot nedaudz kaut ko citu.



```
Disk name:      ad0                                FDISK Partition Editor
DISK Geometry: 16383 cyls/16 heads/63 sectors = 16514064 sectors (8063MB)

Offset          Size(ST)          End          Name  PType          Desc  Subtype  Flags
-----
0              63              62          -     6      unused        0
63          16514001      16514063    ad0s1  3      freebsd       165     CA

The following commands are supported (in upper or lower case):

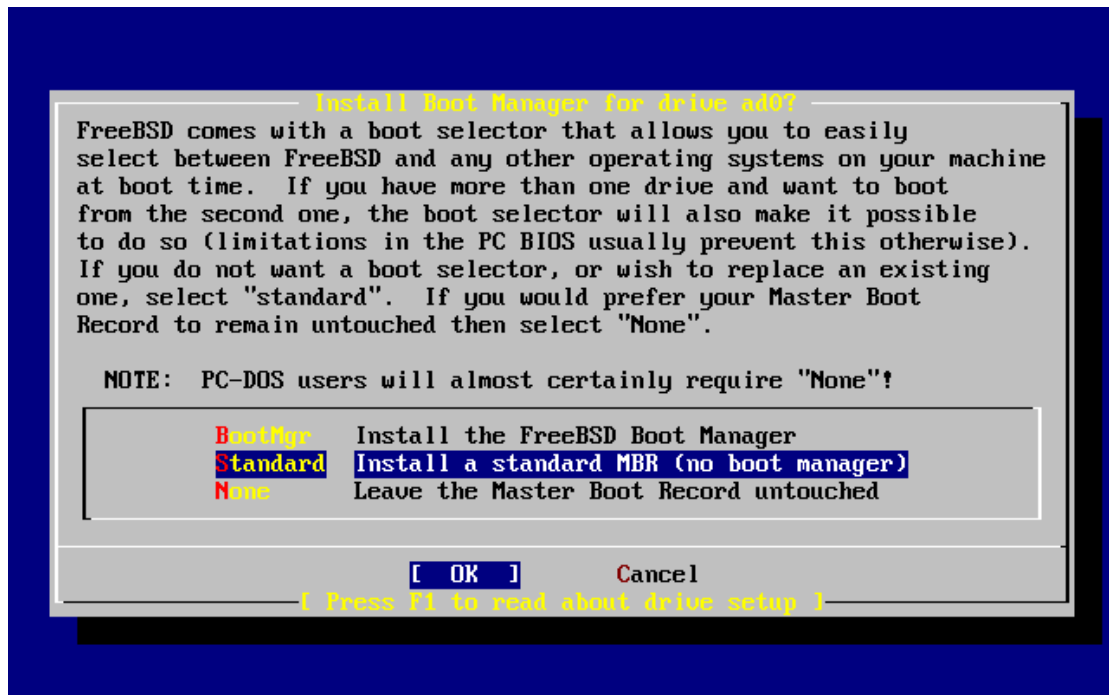
A = Use Entire Disk      G = set Drive Geometry  C = Create Slice      F = `DD' mode
D = Delete Slice        Z = Toggle Size Units   S = Set Bootable     I = Wizard m.
T = Change Type         U = Undo All Changes    Q = Finish

Use F1 or ? to get more help, arrow keys to select.
```

Izvēlos vienkāršāko variantu - izmantošu visu disku. Spiežu taustiņu „a”. Un pēc tam „s” - šķiet, ka šī opcija nav obligāta šajā gadījumā. Bet tā vajadzētu darīt - lai varētu startēt no šīs daļas.

## 2.4. Boot Manager izvēlne

Tālāk piedāvā instalēt **Boot Manager**.



Boot menedžeris ir vajadzīgs divos gadījumos:

- Jums ir vairāki diski un ieinstalējāt ne uz primāro.
- Instalējiet FreeBSD līdzās citām operētājsistēmām

Manā gadījumā neviens no šiem variantiem nav aktuāls, tāpēc lieku standarta variantu - *no boot manager*.

## 2.5. Partīcijas ar Disklabel

Tagad taisām partīcijas ar Disklabel.

```
FreeBSD Disklabel Editor
Disk: ad0 Partition name: ad0s1 Free: 0 blocks (0MB)
Part      Mount      Size Newfs  Part      Mount      Size Newfs
-----
ad0s1a    /           512MB UFS2   Y
ad0s1b    swap        2534MB SWAP
ad0s1d    /var        2048MB UFS2+S Y
ad0s1e    /tmp        1024MB UFS2+S Y
ad0s1f    /usr        143GB  UFS2+S Y

The following commands are valid here (upper or lower case):
C = Create      D = Delete     M = Mount pt.   W = Write
N = Newfs Opts  Q = Finish     S = Toggle SoftUpdates Z = Custom Newfs
T = Toggle Newfs U = Undo       A = Auto Defaults R = Delete+Merge

Use F1 or ? to get more help, arrow keys to select.
```

Šeit varam spiest taustiņu „a”, lai dators mēģinātu uzminēt, kā vajadzētu sadalīt partīcijas.

Partīcijas nummurē no a līdz h.

b un c partīcijām ir īpaša nozīme. B partīcija ir swap, bet c ietver sevī visas partīcijas - tas vajadzīgs procesiem, kuri darbojas ar visām partīcijām.

Piemēram, ad0s1a - pirmā (a) partīcija pirmai daļai (s1 - slice 1) no pirmā diska (ad0)

Principā var dzīvot ar partīcijām, ko dators automātiski izrēķina, bet es uztaisīju savas, jo swap partīciju it kā vajag 2 vai 3 reizes lielāku par RAM apjomu. Pārējās visas uztaisīju tāpat, kā automātiskajā.

Pēc automātiskā varianta bija:

ad0s1a	/	512MB
ad0s1b	swap	998MB

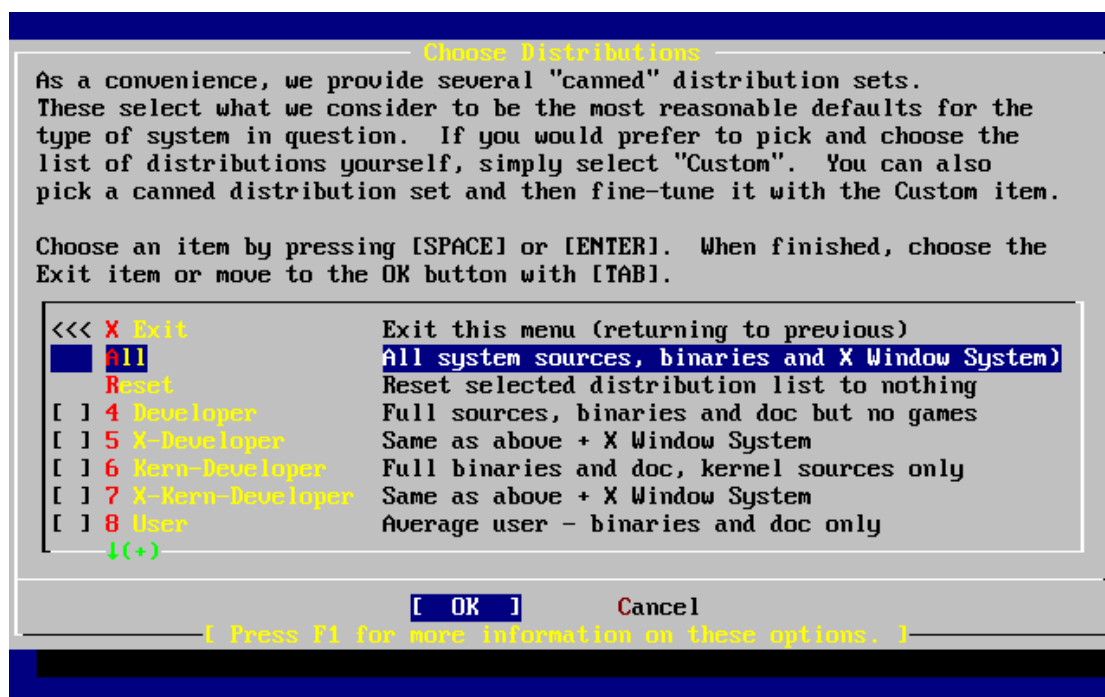
ad0s1d	/var	1523MB
ad0s1e	/tmp	512MB
ad0s1f	/usr	74982MB

Uzliku:

ad0s1a	/	512MB
ad0s1b	swap	1024MB
ad0s1d	/var	2048MB
ad0s1e	/tmp	512MB
ad0s1f	/usr	74432MB

Visas darbības ir vienkārši veicamas ar programmas **Disklabel** dotajām iespējām. Komandu iespējas ir sniegtas programmas loga apakšā.

## 2.6. Izvēlēties, ko instalēt



Liekam All. Principā varam iztikt ar diezgan maz programmām, bet mums diska ietilpība ir pietiekama. Tāda sistēma, protams, ir visdaudzpusīgākā. Šī izvēle nenozīmē, ka mums tagad būs uzinstalēts X Window System. Tas

nozīmē, ka mums būs faili, no kuriem instalēt to.

Kad esam izvēlējušies, spiežam OK.

Tālāk mums piedāvā uzinstalēt **Portu kolekciju**. Instalējām arī šo, jo vieta uz diska ir, un tādā gadījumā mums būs viegli atrast un uzinstalēt vajadzīgās aplikācijas.

Ir 2 viegli veidi, kā uzinstalēt aplikācijas uz FreeBSD

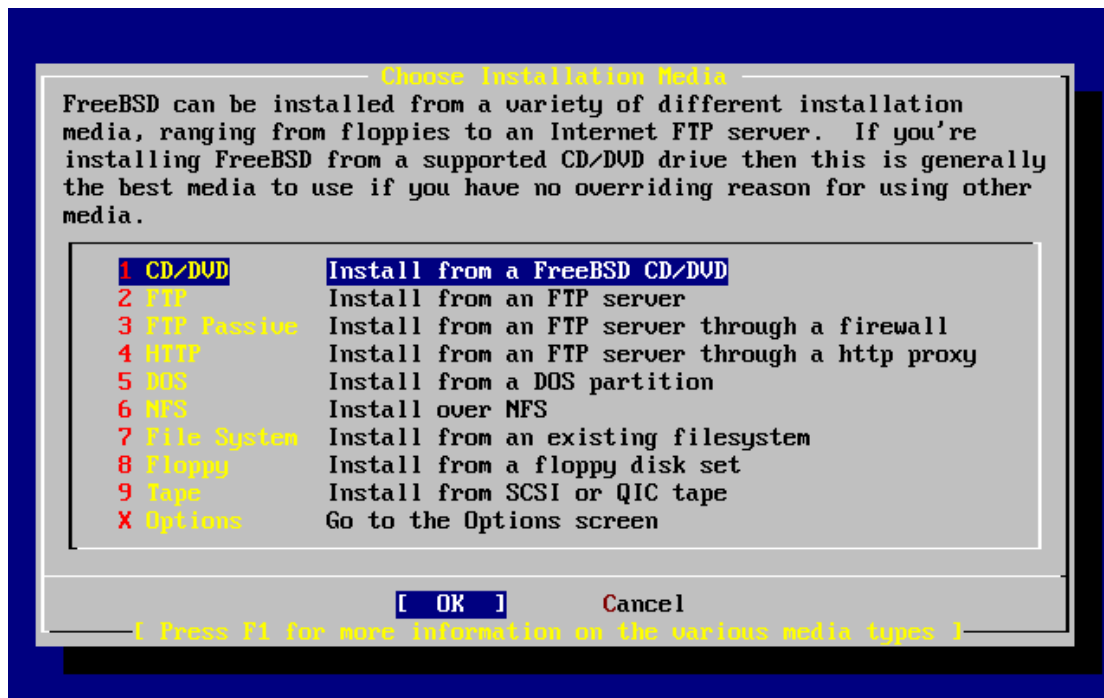
- No Portiem
- No Pakotnēm

Porti ir veids, kā instalēt programmas uz FreeBSD. Tās ir mapes ar failiem, ar kuru palīdzību automātiski tiek novilkta faili, kas tiek kompilēti un tādā veidā uzinstalē programmu.

Pakotnes jau ir sagatavoti binārie faili, no kuriem instalē programmu.

## 2.7. Instalācijas beigu daļa

Šeit mums piedāvā izvēlēties, no kurienes instalēsīm FreeBSD. Vienkārši izvēlamies 1. variantu - **CD/DVD**.



Apstiprinām logu, kur saka, ka ir pēdējā iespēja atkāpties.

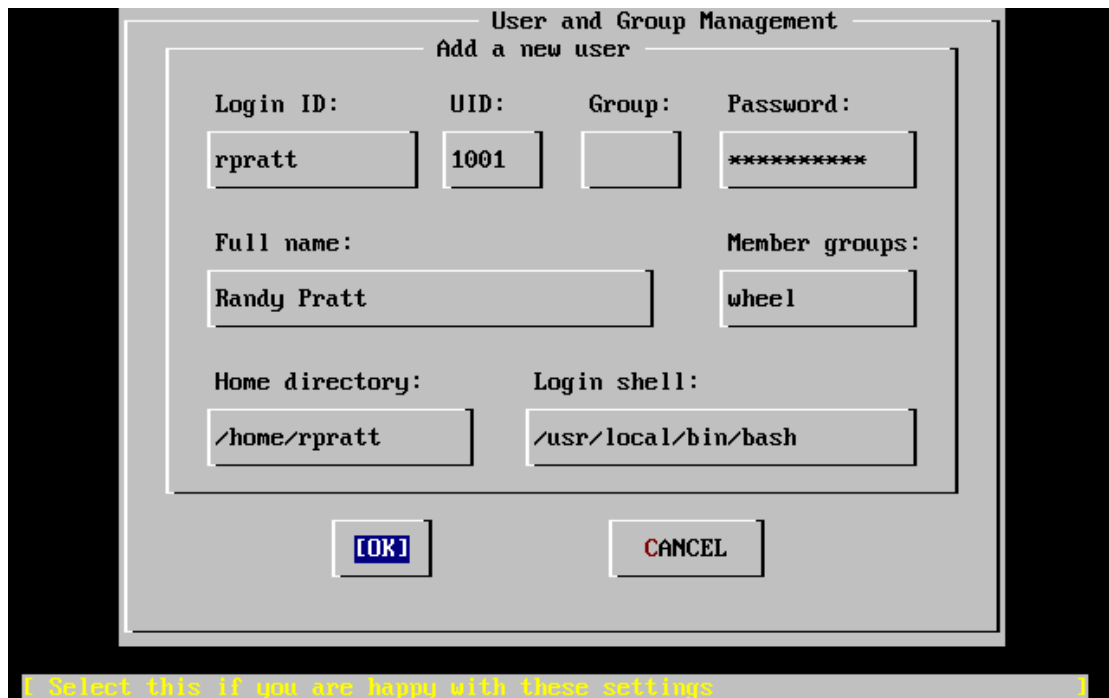
Tālāk, manā gadījumā pēc apmēram 6 minūtēm, pienāca apsveikums, ka FreeBSD ir uzinstalēts.

## 2.8. Pēcinstalācijas konfigurācija

Šeit mums tiek prasīti jautājumi, uz kuriem jāatbild ar YES vai NO.

Atbildu uz visiem ar NO, izņemot šādus jautājumus:

- Enable SSH login
- Set Timezone
- Enable Linux binary compatibility
- Jautājums par peles esamību
- Create user accounts - uztaisu savējo kontu...



levadīju...:

- Login ID
- Password
- Member groups

Login ID - vārds, ar kuru pieteiksieties sistēmā. Manā gadījumā - adrians.

Iekš **Member groups** ir tāda svarīga grupa, kā **wheel**. Ja lietotājs ir piederīgs šai grupai, tad viņam ir iespēja aizvietot savu identitāti ar root ar komandas su palīdzību.

Pievienoju savu lietotāju wheel grupai.

Turpinot, mums jāievada parole **root** lietotājam.

Logā „Visit general configuration menu”, spiežu NO. Jo visu, ko vajadzēs konfigurēt, izdarīsim to vēlāk.

Tālāk mēs atgriežamies sākuma izvēlnē, bet šoreiz izvēlāmies **Exit install**.

Tālāk dators pārstartēsies, un varam nomainīt iekš BIOS boot secību, izņemt

disku un beidzot startēt uzinstalēto FreeBSD.

Ja jums beigās parādīsies tāda rinda... Tad ziniet, ka jums izdevās.

```
login:
```

### Piezīme - Kā izbeigt FreeBSD darbību (shutdown)

Komanda *shutdown -h now*. Tā jāizpilda ar root tiesībām. Te atkal ir divi veidi:

#### 1. Ielogoties ar root kontu

```
login: root
password:
#
```

#### 2. Veikt substitūciju - su

```
login: adrians
password:
$ su
password:
#
```

# - nozīmē, ka jums ir root tiesības. Tas nozīmē, ka tagad varat izpildīt komandu

```
# shutdown -h now
```



## 3. Apache Web serveris

### 3.1. Instalācijas process

Kā jau iepriekš minēju, FreeBSD var uzinstalēt programmas 2 veidos.

Viens no tiem ir ar pakotnēm - tie ir sakompresēti, sagatavoti faili, ar kuru palīdzību tiek ieinstalēta programma.

Te nāk palīgā šādas komandas:

- `pkg_add pakotnes_nosaukums` - ieinstalē programmu
- `pkg_add -r pakotnes_nosaukums` - automātiski novelk pakotni no interneta un uzinstalē to
- `pkg_delete pakotnes_nosaukums` - izinstalē programmu
- `pkg_info` - parāda sarakstu ar ieinstalētām programmām
- `pkg_version` - parāda sarakstu ar ieinstalētām programmām un vēl informāciju par to, vai versija ir pati jaunākā vai nē.

Piemēram:

```
# pkg_add -r lsof
```

Otrs veids, kā instalēt aplikācijas ir no portiem. Tā arī darīsim ar Apache. Lai uzinstalētu šādā veidā, mums ir jāaiziet uz attiecīgo direktoriju. Un tur jāizplida komandas `make install clean`.

Principā tā ir kombinācija no komandām

1. `make` - novelk attiecīgos failus un nokompilē, sagatavo instalācijai
2. `make install` - ieinstalē aplikāciju
3. `make clean` - izdzēš visus failus, kas tika izveidoti procesa gaitā

```
# cd /usr/ports/www/apache22  
# make install clean
```

Te mums rodas vēl viens jautājums - kā zināt, kur atrodas vajadzīgais ports?

Te ir vairanti:

```
# whereis apache22  
  
# cd /usr/ports  
# make search name=apache
```

Otrs variants mums atgriezīs ļoti daudz ierakstus. Tur grūti pēc tam atrast.

Es darīju tā, lai zinātu, kādas versijas ir pieejamas portos. Uzzināju, ka kāda versija atrodas /usr/ports/www es aizgāju uz turieni un sameklēju visus interesējošos ierakstus šādi:

```
# cd /usr/ports/www  
# ls -l | grep apache
```

Pirms instalējam kaut ko, mums jānodrošina interneta klātbūtne. To var vienkārši izdarīt ar dažām komandām.

Pārbaudīt, vai internets ir pieejams var ar **ping** komandu. Piemēram, šādi:

```
# ping net.apollo.lv
```

1) uzzinām, kā saucas mūsu tīkla ierīces

```
# ifconfig
```

Uzzinu, ka manējā tīkla karte saucas **r10**, Linux operētājsistēmā, kā, piemēram, Ubuntu, tā saucas **eth0**.

Manā gadījumā es „sēžu” uz Lattelekom internetu, kur tie izmantota DHCP tehnoloģija. Tātad, man tagad jādabū IP adrese no servera izmantojot **dhclient**.

2) dabūjam IP adresi atbilstošai ierīcei

```
# dhclient r10
```

Lai mums katru reizi nebūtu tas jādara, pievienojam failam

- /etc/rc.conf

rindiņu

- ifconfig\_r10="DHCP"

Piemēram, izmantojot Easy Editor

```
# ee /etc/rc.conf
```

Turpinām ar Apache22 instalāciju. Un tas notiek ļoti vienkārši ar šādām komandām:

```
# cd /usr/ports/www/apache22  
# make install clean
```

Neejiet uzreiz dzer tēju, jo instalācijas gaitā ir jāapstiprina viens logs!

### 3.2. Apache darbināšana

Mēģinām piedarbināt Apache ar komandu

```
# /usr/local/sbin/apachectl start
```

Ups, kaut kur ir kļūda, jo parādās paziņojums...

```
httpd: apr_sockaddr_info_get() failed for  
httpd: could not reliably determine the server's fully qualified  
domain name, using 127.0.0.1 for ServerName  
[Thu May 03 15:30:17 2007] [warn] (2)No such file or directory:  
Failed to enable the 'httpready' Accept Filter
```

Skatāties log failu... Kaut gan vienu no kļūdām jau uztreiz saprotam, tas ir, nav uzlikts ServerName.

Tā vai tā, skatāties log failā, jo vēl kaut kas nav kārtībā.

```
# cat /var/log/httpd-error.log

[Thu May 03 15:30:17 2007] [warn] Init: session cache is not
configured [hint: SSLSessionCache]
[Thu May 03 15:30:17 2007] [alert] (EAI 8)hostname nor servname
provided, or not known: mod_unique_id: unable to find IPv4 address
of ""
Configuration Failed
```

Redzam, ka ir divas kļūdas. Vienu no tām izraisa SSL modulis. Mums nevajadzēs SSL, tāpēc to izslēgsim. SSL ir atsevišķs stāsts. Taisīsim standarta Web serveri. Apache22 SSL ir ieslēgts pēc noklusējuma.

Otru kļūdu arī izraisa viens no moduļiem, kas paredzēts klāsteru sistēmām, mums ir viens *standalone* serveris, tāvad tas arī mums nav vajadzīgs.

Atveram konfigurācijas failu un izlabojam kļūdas...

```
# ee /usr/local/etc/apache22/httpd.conf
```

Atrodam vietas:

```
...
LoadModule unique_id_module libexec/apache22/mod_unique_id.so
...
LoadModule ssl_module libexec/apache22/mod_ssl.so
...
#ServerName www.example.com:80
...
```

Izlabojam uz šādu situāciju:

```
...
#LoadModule unique_id_module libexec/apache22/mod_unique_id.so
...
#LoadModule ssl_module libexec/apache22/mod_ssl.so
...
ServerName localhost:80
...
```

Aizkomentējām ciet nevajadzīgos moduļus, un iedevām serverim vārdu. Vārdam jābūt IP adresei vai arī no */etc/hosts* faila kādam nosaukumam. Localhost atbilst IP adrese 127.0.0.1.

Startējam tagad vēlreiz Apache Web serveri!

Pieejamās komandas ir:

- **start** - iedarbināt serveri
- **stop** - apstādināt
- **restart** - pārstartēt
- **graceful** - pārstartēt serveri neizverot konekcijas uz tā.
- ir arī citas komandas

```
# /usr/local/sbin/apachectl start
```

Tagad parādās tikai brīdinājums.

```
[Thu May 03 15:52:17 2007] [warn] (2)No such file or directory:  
Failed to enable the 'httpready' Accept Filter
```

Šis brīdinājums nenozīmē totālu kļūdu. Tas notiek tāpēc, ka FreeBSD sistēmā starta laikā pēc noklusējuma neielādējas *accf\_http* modulis. To var izdarīt ar komandu `kldload accf_http`. Var arī pateikt sistēmai, lai starta laikā ielādē šo moduli, bet mēs to tagad nedarīsim.

Apskatamies vai serveris strādā.

```
# ps -auxww | grep httpd
```

Ja parādās procesi, tad viss notiek. Pārbaudot ar pārlūku, vajadzētu ieraudzīt uzrakstam **It works!**

Pēc adreses:

- <http://81.198.248.53/>

Manā gadījumā. Protams, izmantojam pareizo IP adresi! IP adresi, kā jau

mēs noskaidrojām, var uzzināt ar komandu **ifconfig**.

Man ir tāda situācija, ka es varu pārbaudīt uzreiz, kā darbojas serveris, jo man ir otrs dators.

Izdarīsim tā, lai serveris startējās reizē ar sistēmu. Tas jādara `/etc/rc.conf` failā. Pieliekot attiecīgu rindiņu:

- `apache22_enable="YES"`

## 4. PHP5 modulis

### 4.1. Instalācija un konfigurācija

PHP arī instalēsim no portiem. To jau mēs mākam, tātad darīsim šitā:

```
# cd /usr/ports/lang/php5
# make install clean
```

Tad mums parādīsies logs, kurā varēsim atņeksēt, lai uztaisa Apache moduli.

- Build Apache module

Šķiet, to var arī izdarīt padodot argumentus pie kompilācijas. Piemēram, kad instalē attālināti caur SSH login, tad tādi logi neparādās.

Kad tas ir izdarīts, mums jāpārlicinās, ka Apache konfigurācijas failā ir dažas iezīmes.

```
# ee /usr/local/etc/apache22/httpd.conf
```

Lūk, šādas:

```
...
LoadModule php5_module libexec/apache22/libphp5.so
...
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
...
<IfModule dir_module>
    DirectoryIndex index.html index.htm index.php
</IfModule>
...
```

Tātad, kas te ir izdarīts. Pievienots php5 modulis - šim te būtu jau jābūt pēc php instalācijas.

Tālāk mēs pateicām, lai interpretē .php , kā PHP failus un .phps, kā PHP avotu failus - nezinu, kāpēc tādu vajag, bet tā mēdz darīt. Tā arī ir rakstīs

[www.php.net](http://www.php.net) lapā.

Vēl mēs pasakām, ka pēc noklusējuma servējam failus index.html, index.htm un index.php, ja norādīta tikai direktorija. Tāds ieraksts jau ir pēc noklusējuma konfigurācijas failā, bet mēs to papildinājām.

Tālāk mums vajag izveidot **php.ini** failu. To darām vienkārši.

```
# cd /usr/local/etc
# cp php.ini-recommended php.ini
# /usr/local/sbin/apachectl restart
```

Ko mēs izdarījām - nokopējām vienu failu un pārstartējām serveri.

## 4.2. PHP tests

Tagad testējam PHP.

Uzrakstam nelielu testa programmiņu - ejam uz Apache root direktoriju un startējam Easy Editor...

```
# cd /usr/local/www/apache22/data
# ee test.php
```

Uzrakstām šādu programmiņu:

```
<?php
phpinfo();
?>
```

Saglabājam un tagad pārbaudām pēc adreses:

- <http://81.198.248.53/test.php>

Jūsu gadījumā adrese būs citādāka.

Ja kaut kas nenostādā, tad jāskatās Apache log fails. Man vienreiz, nez kāpēc gadījās segmentation fault restartējot serveri. Tad stopējam to



pavisam un startējam no sākuma.

```
# cat /var/log/httpd-error.log
```

## 5. MySQL datu bāzu sistēma

### 5.1. MySQL instalācija

```
# cd /usr/ports/databases/mysql51-server  
# make install clean
```

Ar šīm komandām mēs būsīm ieinstalējuši MySQL serveri, kā arī klientu. Iespējams, ka arī vēl kādu programmu, uz kuras balstās viss šis pasākums.

Tagad instalēsīm php moduli, kas ieinstalēs un sakonfigurēs programmas, kuru rezultātā varēsīm MySQL datu bāzēm piekļūt no PHP skriptiem.

```
# cd /usr/ports/databases/php5-mysql  
# make install clean
```

Vēl vienu lietu izdarām - ierakstam `/etc/rc.conf` to, ka gribam, lai MySQL serveris startējas reizē ar sistēmu.

```
# ee /etc/rc.conf
```

Un pievienojam failam ierakstu:

- `mysql_enable="YES"`

Tagad mums jābūt iespējai pieslēgties datu bāzei no PHP skriptiem, kā arī pieslēgties ar MySQL klientu.

Vēl mēs varam pārkopēt konfigurācijas failu, lai būtu vieglāk administrēt serveri.

```
# cd /usr/local/share/mysql  
# cp my-medium.cnf /etc/my.cnf
```

## 5.2. MySQL darbināšana un tests

```
# mysql
welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 5.1.11-beta

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Ja jums parādās šāds paziņojums, tad varat sevi apsveikt!

Pamēģinām apskatīties, kādas datu bāzes mums ir pieejamas!

```
mysql> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| cluster           |
| mysql             |
| test              |
+-----+
4 rows in set (0.01 sec)

mysql>
```

## 5.3. Lietotāju pievienošana

Pievienosim lietotāju datu bāzei, kā arī uzliksim paroli root lietotājam. Jo ar root lietotāju neviens normāli nestrādā un tam vienmēr ir jābūt nodrošinātam ar paroli.

```
mysql> set password for 'root'@'localhost' = password('123456');
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Tagad pievienojam normālu lietotāju.

```
mysql> create user 'adrians'@'localhost' identified by '789012';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

## 5.4. MySQL - PHP tests

Uzrakstīsim nelienu programmiņu, kas pieslēdzas MySQL serverim un izvēlas datu bāzi. Šādā veidā mēs pārlicināsimies, vai no PHP skripta var strādāt ar MySQL.

```
ee /usr/local/www/apache22/data/my.php
```

Un uzrakstam tur šādas rindas:

```
<?php
// Connecting, selecting database
$link = mysql_connect('localhost', 'adrians', 'parole1')
    or die('Could not connect: ' . mysql_error());
echo 'Connected successfully';
mysql_select_db('test') or die('Could not select database');
echo 'Database selected';
?>
```

Un saglabājam.

Tālāk ar Web pārlūka palīdzību notestējam šo failu.

- <http://81.198.248.53/my.php>

Ja pārlūkā ir redzamas rindiņas:

```
Connected successfully
```

```
Database selected
```

Tad viss darbojas nevainojami.

## 6. phpMyAdmin

### 6.1. phpMyAdmin instalācija

```
# cd /usr/ports/databases/phpmyadmin
# make install clean
```

Sākumā mums tiek piedāvāta izvēle instalēt dažādas programmas komponentes. Tā kā īsti nezinu, ko katra no tām dara, tad izvēlējos visas. Pie tam, pēc noklusējuma tur visas izņemot vienu jau ir atzīmētas. Un tā viena neatzīmētā izklausījās sakarīgi.

### 6.2. phpMyAdmin konfigurācija

Tālāk uzstādām konfigurācijas failu šai programmai. Un pielāgojam to savām vajadzībām.

```
# /usr/local/www/phpMyAdmin
# cp libraries/config.default.php config.inc.php
# ee config.inc.php
```

Un nomainām attiecīgos ierakstus.

```
...
$config['blowfish_secret'] = 'rakstietsavusekrettekstu';
...
$config['servers'][$i]['auth_type'] = 'cookie';
...
```

Ko mēs izdarījām - nomainījām autentifikācijas metodi. Cookie metodei ir vajadzīgs kaut kāds teksts, pēc kura ģenerēt *cookies*, tāpēc ierakstam arī tekstu.

Tagad pievienosim Apache konfigurācijas failā rindiņas, kas nodrošinās cilvēcīgu piekļuvi phpMyAdmin.

```
# ee /usr/local/etc/apache22/httpd.conf
```

Un pievienojam šādas rindiņas, kurās pasakam, ka zem adreses *http://81.198.248.53/phpmyadmin* atradīsies attiecīga direktorijs, kurā ir phpMyAdmin rīks.

```
Alias /phpmyadmin "/usr/local/www/phpMyAdmin"  
  
<Directory "/usr/local/www/phpMyAdmin">  
    Options None  
    AllowOverride Limit  
  
    order allow,deny  
    Allow from all  
</Directory>
```

Tagad pārstartējam Apache un ietestējam jauno rīku adresē:

- <http://81.198.248.53/phpmyadmin>

Pārstartēt jau mēs mākam, bet tomēr atgādināšu. To visdrošāk var izdarīt šādi:

```
# /usr/local/sbin/apachectl stop  
# /usr/local/sbin/apachectl start
```

# 7. FTP pieeja

## 7.1. FTP iedarbināšana

FreeBSD sistēmā FTP servera darbība ir tieši atkarīga no **inetd** dēmona. Inetd ir dēmons, kurš atbild par daudziem ar tīklu saistītiem procesiem. Principā FTP varētu darboties, kā atsevišķs dēmons, bet sistēmas optimizēšanas nolūkā tas tiek organizēts citādāk. Ideja ir tāda, ka inetd atbild par samērā reti izmantojamiem servisiem.

Šeit ir jāizdara divas lietas. Pirmkārt, lai darbinātu FTP ir jāatkommentē inetd konfigurācijas failā attiecīga rinda.

```
# ee /etc/inetd.conf
```

Šādai rindai jābūt:

```
ftp stream tcp nowait root /usr/libexec/ftpd ftpd -l
```

Vēl pievienojam rc.conf failam attiecīgu rindu.

```
# ee /etc/rc.conf
```

Pievienojam šādu rindiņu:

- inetd\_enable="YES"

Tad es pārstartēju sistēmu, lai iedarbinātu inetd. Bieži šitāds startēšanas variants ir ērtāks. Man ātrumā neizdevās atrast, kā startēt inetd.

Tagad jums vajadzētu varēt pievienoties sistēmai ar FTP.

## 7.2. Lietotāju pievienošana, FTP un SSH konfigurēšana

Sākumā izveidosim kādu grupu, lai pēc tam nebūtu problēmas ar administrēšanas lietām.

```
# pw groupadd www-users
```

Tālāk pievienojam lietotāju ar ļoti viegli saprotamo `adduser`.

```
# adduser
Username: www-adrians
Full name: Adrians Heidens
Uid (Leave empty for default):
Login group [www-adrians]: www-users
Login group is www-users. Invite www-adrians into other groups? []:
Login class [default]:
Shell (sh csh tcsh nologin) [sh]:
Home directory [/home/www-adrians]:
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username   : www-adrians
Password   : *****
Full Name  : Adrians Heidens
Uid        : 1002
Class      :
Groups     : www-users
Home       : /home/www-adrians
Shell      : /bin/sh
Locked     : no
OK? (yes/no): y
adduser: INFO: Successfully added (www-adrians) to the user
database.
Add another user? (yes/no): y
```

Kad visi vajadzīgie lietotāji ir pievienoti, tagad varam pateikt FTP serverim, lai nomaina attiecīgiem lietotājiem root direktoriju, kad viņi piesakās



sistēmā ar FTP. Tas notiek ar **ftproot** konfigurācijas failu.

```
# cd /etc
# cp /usr/ports/ftp/bsdftpd-ssl/files/ftproot ftproot
```

Pašā failā neko nemainām, pagaidām mums pietiek ar noklusētajiem parametriem. Tas mums nodrošina to, ka lietotāji, kuri ir wheel grupā tiek klāt sistēmas root direktorijai, bet cietiem lietotājiem nomaina FTP root direktoriju ar viņu mājas direktoriju, tādējādi nodrošinot to, ka viņi nevarēs vazāties apkārt pa sistēmu.

Tagad liedzam SSH pieeju sistēmai dažiem lietotājiem.

```
# ee /etc/ssh/sshd_config
```

Un pievienojam failam ierakstu:

```
...
DenyGroups www-users
...
```

Tagad pārstartējam SSH dēmonu

```
# /etc/rc.d/sshd restart
Stopping sshd.
Starting sshd.
#
```

Tagad lietotāji, kas ir grupā www-users, īstenībā visi lietotāji, izņemot tie, kuri ir wheel grupā, nevarēs attālināti piekļūt sistēmai ar SSH login.

## 8. Lietotāju sagatavošana darbam

### 8.1. MySQL sagatavošana

Lai izveidotie lietotāji varētu strādāt ar sistēmu, kā tīmekļa administratori, mums ir jāveic vēl daži sagatavošanas darbi. Tie ir:

Jāizveido katram lietotājam MySQL konts un datu bāze.

```
# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 23 to server version: 5.1.11-beta-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create user 'www-adrians'@'localhost' identified by '123456';
Query OK, 0 rows affected (0.01 sec)

mysql> create user 'www-andris'@'localhost' identified by '123456';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Tālāk izveidosim katram lietotājam vienu datu bāzi un iedosim viņam visas tiesības uz to - varbūt ne labākā izvēle, bet ērtākā gan. Šeit neiedziļināsimies MySQL administrācijas procesā.

```
mysql> create database www_adrians;
Query OK, 1 row affected (0.00 sec)

mysql> create database www_andris;
Query OK, 1 row affected (0.00 sec)

mysql> grant all on www_adrians.* to 'www-adrians'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> grant all on www_andris.* to 'www-andris'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Vienīgā problēma, kas vēl var gadīties ir latviešu burtu pieejamība MySQL datu bāzē. Bet tas ir MySQL jautājums. Mēs to šeit neapskatīsim. Principā phpMyAdmin arī ir pietiekami spēcīgs, lai katrs lietotājs varētu pats noregulēt šo jautājumu vēl iespējams piepalīdzot ar PHP.

## 8.2. Apache sagatavošana

Izveidosim katram lietotājam viņa mājas direktorijā mapi public\_html, kurā viņam būs tiesības publicēt failus.

```
# mkdir /home/www-adrians/public_html
# chown www-adrians /home/www-adrians/public_html
```

Tā darām ar katru no lietotājiem.

Tagad Apache konfigurācijas failā pievienojam dažas rindas.

```
# ee /usr/local/etc/apache22/httpd.conf
```

Un pievienojam šo:

```
Alias /adrians /usr/home/www-adrians/public_html
Alias /andris /usr/home/www-andris/public_html

<Directory /usr/home/*/public_html>
    Options None
    AllowOverride None

    Order allow,deny
    Allow from all
</Directory>
```

Šīs rindas pasaka, ka adresē *http://xxx.xxx.xxx.xxx/adrians* atradīsies faili, kas ir mapē */usr/home/www-adrians/public\_html*. Analogi ar *www-andris*.

Tālāk mēs norādam katra lietotāja *public\_html* mapes dažādas opcijas.

Darbs pabeigts, ja visu izdarījāt, kā vajadzēja, tad jums tagad ir darba spējīgs daudzlietotāju Web serveris ar PHP un MySQL iespējām, kā arī FTP piekļuvi un phpMyAdmin administratoru.