

Apache Web сървър. Инсталиране и конфигуриране на Apache. Инсталиране и конфигуриране на PHP за работа с Apache и MSSQL. Тест на Apache+PHP+MSSQL

Lab No. 4

Цел: Запознаване с основните директиви при конфигуриране на Apache сървър. Създаване на умения за конфигуриране на PHP за работа с Apache сървър и БД сървър MSSQL. Експериментиране на работоспособността на apache, apache+php, apache+php+mssql

Задачи за изпълнение:

- I. ИНСТАЛИРАЙТЕ ИЗПОЛЗВАЙКИ УКАЗАНИЯТА В ПРИЛОЖЕНИЕТО APACHE WEB SERVER. В ПРОЦЕСА НА ИНСТАЛАЦИЯ ИЗПЪЛНЕТЕ СЛЕДНИТЕ ЗАДАЧИ:
- НАСТРОЙКИ ПРИ ИНСТАЛАЦИЯ:
 - МРЕЖОВ ДОМЕЙН TUGAV.BG
 - ИМЕ НА СЪРВЪРА LOCALHOST ИЛИ ИМЕ_NA_PC
 - E-MAIL НА АДМИНИСТРАТОРА НА APACHE: ВАЛИДЕН E-MAIL АДРЕС.
 - СТАРТИРАНЕ НА APACHE КАТО SERVICE НА WINDOWS;
 - APACHE ДА СЕ СЪДЪРЖА В ДИРЕКТОРИЯ D:\APACHE;
 - ИЗПОЛЗВАЙКИ СТАРТ МЕНЮТО НА WINDOWS ИЗБЕРЕТЕ APACHE HTTP SERVER -> CONTROL APACHE SERVER -> MONITOR APACHE SERVERS И ПРЕГЛЕДАЙТЕ СЪСТОЯНИЕТО НА СЪРВЪРА. ПРИ ПОЯВА НА СЪОБЩЕНИЕ ЗА ВЕЧЕ СТАРТИРАН МОНИТОР, С ПОМОЩТА НА TRAY ИКОНАТА НА APACHE (ОКОЛО ИКОНАТА ЗА ДАТА) КЛИКНЕТЕ ДВУКРАТНО ВЪРХУ НЕЯ, СЛЕД КОЕТО ОТНОВО ПРЕГЛЕДАЙТЕ СЪСТОЯНИЕТО НА СЪРВЪРА.
 - СПРЕТЕ СЕРВИЗА НА APACHE;
 - СТАРТИРАЙТЕ СЕРВИЗА НА APACHE
 - МОДИФИЦИРАЙТЕ СЕРВИЗА ДА НЕ СЕ СТАРТИРА АВТОМАТИЧНО ПРИ СТАРТ И LOGON НА WINDOWS (MANUAL)
 - С ИЗПОЛЗВАНЕ НА MSDOS PROMPT В ДИРЕКТОРИЯ /WIN НА APACHE И ФАЙЛА APACHE.EXE:
 - СПРЕТЕ СЪРВЪРА;
 - СТАРТИРАЙТЕ СЪРВЪРА;
 - ПРЕГЛЕДАЙТЕ ВЕРСИЯТА НА СЪРВЪРА;
 - ИЗПОЛЗВАЙКИ БРАУЗЪРА INTERNET EXPLORER ТЕСТВАЙТЕ РАБОТОСПОСОБНОСТТА НА СЪРВЪРА:
 - ЗАДАЙТЕ В ПОЛЕТО ЗА URL АДРЕСА: [HTTP://LOCALHOST:80](http://localhost:80) И АНАЛИЗИРАЙТЕ СИТУАЦИЯТА;
 - ИЗПОЛЗВАЙКИ КОМАНДА В MSDOS ЗАПИШЕТЕ:

TELNET localhost 80 – при което следва да се подаде подходяща заявка за връзка чрез HTTP протокол, запишете:

GET / HTTP/1.0

Отговорът трябва да е подобен на:

HTTP/1.1 200 OK

Date: Sun, 21 Nov 2004 17:03:49 GMT

Server: Apache/2.0.52 (Win32)

Connection: close

Content-Type: text/html; charset=Cyrillic -1251

Content-Language: ru

- II. С ПОМОЩТА НА ПРИЛОЖЕНИЕТО КОНФИГУРИРАЙТЕ APACHE СЪРВЪР СЪС СЛЕДНИТЕ ВЪЗМОЖНОСТИ:
- ГЛАВНАТА ДИРЕКТОРИЯ ЗА ДОКУМЕНТИ И САЙТОВЕ НА APACHE ДА Е D:\WWW (АКО Е НЕОБХОДИМО СЪЗДАЙТЕ ПРЕДИ ТОВА ТАЗИ ДИРЕКТОРИЯ В ДИСКА С WINDOWS EXPLORER):
 - ЗАДАЙТЕ ОПЦИИ ЗА ПРОМЕНЕНАТА DOCUMENTROOT ДИРЕКТОРИЯ, КАТО ПОТРЕБИТЕЛИТЕ:
 - ДА НЕ МОГАТ ДА ПРЕГЛЕЖДАТ СПИСЪКА С ФАЙЛОВЕ СЪДЪРЖАЩ СЕ В ДИРЕКТОРИЯТА;
 - ДА МОГАТ ДА ИЗПЪЛНЯВАТ CGI СКРИПТОВЕ (НАПИСАНИ НА PHP, PERL И Т.Н.);
 - ПРИ НЕЗАДАДЕН ФАЙЛ ЗА ПРЕГЛЕД ОТ URL В БРАУЗЪРА АВТОМАТИЧНО ДА СЕ ПРОВЕРЯВАТ ПОСЛЕДОВАТЕЛНО ФАЙЛОВЕТЕ: INDEX.PHP INDEX.HTML INDEX.HTM
 - В СЪЗДАДЕНАТА ДИРЕКТОРИЯ ДА ИМАТ ДОСТЪП ВСИЧКИ ПОТРЕБИТЕЛИ С ИЗКЛЮЧЕНИЕ НА ПОТРЕБИТЕЛ С IP АДРЕС 192.168.222.44;
 - СЪЗДАЙТЕ ПОДДИРЕКТОРИЯ НА WWW : /AIUT.BG
 - ДА МОГАТ ДА ПРЕГЛЕЖДАТ СПИСЪКА С ФАЙЛОВЕ СЪДЪРЖАЩ СЕ В ДИРЕКТОРИЯТА;
 - ДА МОГАТ ДА ИЗПЪЛНЯВАТ CGI СКРИПТОВЕ (НАПИСАНИ НА PHP, PERL И Т.Н.);
 - ПРИ НЕЗАДАДЕН ФАЙЛ ЗА ПРЕГЛЕД ОТ URL В БРАУЗЪРА АВТОМАТИЧНО САМО ЗА ТАЗИ ДИРЕКТОРИЯ ДА СЕ ПРОВЕРЯВАТ ПОСЛЕДОВАТЕЛНО ФАЙЛОВЕТЕ: INDEX.HTM INDEX.HTML INDEX.PHP
 - В СЪЗДАДЕНАТА ДИРЕКТОРИЯ ДА ИМАТ ДОСТЪП ВСИЧКИ ПОТРЕБИТЕЛИ С ИЗКЛЮЧЕНИЕ НА ПОТРЕБИТЕЛ С IP АДРЕС 192.168.222.1;
 - УТОЧНЕТЕ ИМЕТО НА ФАЙЛА , РЕГЛАМЕНТИРАШ ДОСТЪП ЗА ДИРЕКТОРИИ (ПРОВЕРЯВА СЕ И ИЗЧИТА ЗА ВСЯКА ДИРЕКТОРИЯ) ДА

БЪДЕ HTACCESS.CONF. ЗАДАЙТЕ НЕОБХОДИМИТЕ ПРИВИЛЕГИИ ЗА ТОЗИ ФАЙЛ ЧРЕЗ FILES ДИРЕКТИВАТА (ЗА ПРЕДПОЧИТАНЕ Е ДА НЕ СЕ ВИЖДА ОТ ПОТРЕБИТЕЛЯ)

5. ЗАДАЙТЕ ОПЦИИ ЗА ФАЙЛОВОТЕ В /WWW/AIUT.BG ДИРЕКТОРИЯ, КАТО ПОТРЕБИТЕЛИТЕ:
 - А. ДА НЕ МОГАТ ДА РАЗГЛЕЖДАТ СПИСЪК НА ФАЙЛОВОТЕ В ДИРЕКТОРИЯТА, КАТО САМО РС НА КОЙТО Е ИНСТАЛИРАН СЪРВЪРА ДА ИМА ТЕЗИ ПРАВА - С ПОМОЩТА НА HTTPD.CONF;
 - В. ДА НЕ МОГАТ ДА ОТВАРЯТ ФАЙЛОВЕ С РАЗШИРЕНИЯ *.HTM, С ИЗКЛЮЧЕНИЕ НА РС НА КОЙТО Е ИНСТАЛИРАН АРАСНЕ. ЗА ЦЕЛТА ИЗПОЛЗВАЙТЕ ФАЙЛ HTACCESS.CONF ;
 - С. СЪЗДАЙТЕ ФАЙЛ INDEX.HTML И INDEX.HTM С СЪДЪРЖАНИЕ: `<HR>ТОВА Е ТЕСТ INDEX.HTML<HR>` В INDEX.HTM И `<HR>ТОВА Е ТЕСТ INDEX.HTML<HR>` В INDEX.HTML. СЪЗДАЙТЕ КОПИЯ В ДИРЕКТОРИЯ /WWW И /WWW/AIUT.BG.
6. ЗАДАЙТЕ TIMEOUT ПРИ НЕВЪЗМОЖНОСТ ЗА ДОСТЪП ДО ФАЙЛ ВЪПРЕКИ ЧЕ СА ОТПУСНАТИ ПРИВИЛЕГИИ.
7. С ЦЕЛ ПОНИЖАВАНЕ НА НАТОВАРНОСТТА НА СЪРВЪРА ЗАДАЙТЕ В ДАДЕН МОМЕНТ ДОСТЪП ДО СЪРВЪРА ДА ИМАТ 10 ПОТРЕБИТЕЛЯ (10 КОНЕКЦИИ)
8. ЕКСПЕРИМЕНТИРАЙТЕ ПО-ГОРЕ ЗАДАДЕНИТЕ ОПЦИИ ЗА РАБОТОСПОСОБНОСТ ПОСРЕДСТВОМ БРАУЗЪРА И ЗАДАВАНЕ НА URL С ФАЙЛОВОТЕ СЪЗДАДЕНИ ДО ТУК В ОТДЕЛНИТЕ ДИРЕКТОРИИ.
9. СЪЗДАЙТЕ ПСЕВДОНИМ NEWS НА ДИРЕКТОРИЯ D:\STUDENTI И КОПИРАЙТЕ INDEX.HTML И INDEX.HTM. ЕКСПЕРИМЕНТИРАЙТЕ С URL [HTTP://LOCALHOST/NEWS/INDEX.HTM](http://localhost/news/index.htm) . (ALIAS)
10. ПРЕНАСОЧЕТЕ САЙТА В ДИРЕКТОРИЯ /WWW/KST.BG КЪМ URL HTTP://AIUT.TUGAV.BG
11. АСОЦИРАЙТЕ РАЗШИРЕНИЯТА .PHP С MIME ТИП - ДА СЕ РАЗПОЗНАВАТ ОТ АРАСНЕ (ADDDTYPE)
 - А. MIME ИДЕНТИФИЦИРА ФАЙЛОВИЯ ТИП ЧРЕЗ СХЕМА ЗА ИМЕНУВАНЕ ОТ ВИДА ТИП/ПОДТИП. ПЪРВИЯТ КОМПОНЕНТ ОПРЕДЕЛЯ ОСНОВНИЯ ТИП НА ФАЙЛА, А ВТОРИЯ КОНКРЕТНИЯ ТИП В РАМКИТЕ НА ОСНОВНАТА КАТЕГОРИЯ,
12. СВЪРЖЕТЕ ЗАДАДЕНИЯ ПО-ГОРЕ MIME ТИП С РЕСУРС, КОНФИГУРИРАН КАТО CGI СКРИПТ (В СЛУЧАЯ ИНТЕРПРЕТАТОРА НА PHP). ИЗПОЛЗВАЙТЕ ДИРЕКТИВА ACTION
13. ЗАДАЙТЕ ДИРЕКТОРИЯ В КОЯТО ЩЕ СЕ СЪДЪРЖАТ ИЗПЪЛНИМИ ПРОГРАМИ , КАТО ИНТЕРПРЕТАТОРИ И ДР. ДАЙТЕ ВЪЗМОЖНОСТ PHP ИНТЕРПРЕТАТОРА , ИНСТАЛИРАН В ДИРЕКТОРИЯ D:\PHP ДА СЕ СТАРТИРА! (SCRIPTALIAS);
14. АСОЦИРАЙТЕ РАЗШИРЕНИЯТА .BAT .EXE КАТО CGI ИЗПЪЛНИМИ (ADDHANDLER) В ДИРЕКТОРИЯ D:/WWW РАЗРЕШЕНА ЗА CGI СКРИПТОВЕ СЪЗДАЙТЕ ФАЙЛА test.bat СЪС СЪДЪРЖАНИЕ:

@echo off

echo Content-type: text/html

echo.

echo.

dir

експериментирайте работоспособността чрез: <http://localhost/www/test.bat>

III. С ПОМОЩТА НА ПРИЛОЖЕНИЕТО ИНСТАЛИРАЙТЕ PHP В ДИРЕКТОРИЯ D:\PHP. СПАЗВАЙТЕ СТРИКТНО ПОСЛЕДОВАТЕЛНОСТТА ЗА ИНСТАЛИРАНЕ И КОНФИГУРИРАНЕ НА PHP , КАТО CGI ПРИЛОЖЕНИЕ НА АРАСНЕ:

15. НАПРАВЕТЕ ТЕСТ НА ИНСТАЛАЦИЯТА , ИЗПОЛЗВАЙКИ КОМАНДА В MSDOS

TELNET localhost 80 – при което следва да се подаде подходяща заявка за връзка чрез HTTP протокол, запишете:

GET / HTTP/1.0

Отговорът трябва да е подобен на:

HTTP/1.1 200 OK

Date: Sun, 21 Nov 2004 17:03:49 GMT

Server: Apache/2.0.52 (Win32)

X-Powered-By: PHP/4.2.1

Connection: close

Content-Type: text/html; charset=Cyrillic -1251

Content-Language: ru

16. НАПРАВЕТЕ ТЕСТ НА ИНСТАЛАЦИЯТА. СЪЗДАЙТЕ ФАЙЛ С ИМЕ INFO.PHP В /WWW/AIUT.BG СЪС СЪДЪРЖАНИЕ:

<?

phpinfo();

?>

Напишете в браузъра URL : <http://localhost/aiut.bg/info.php> и анализирайте.

Ако всичко е протекло както трябва прегледайте данните които генерира функцията на php – phpinfo(), в противен случай проверете дали сте спазили всички изисквания на инсталацията!

17. НАПРАВЕТЕ ТЕСТ НА ИНСТАЛАЦИЯТА. СЪЗДАЙТЕ ФАЙЛ С ИМЕ SQL.PHP В /WWW/AIUT.BG СЪС СЪДЪРЖАНИЕТО ПОСОЧЕНО В САМИЯ КРАЙ НА ПРИЛОЖЕНИЕТО.

- а. Ако е необходимо променете данните в:

```
$dblocation = "127.0.0.1";
```

```
$dbname = "MY_1";
```

```
$dbuser = "sa";
```

```
$dbpasswd = "";
```

- б. Напишете в браузъра URL : <http://localhost/aiut.bg/info.php> и анализирайте

ИНСТАЛИРАНЕ НА АРАСНЕ WEB СЪРВЪР ЗА WINDOWS

- Стартиране на оригиналния файл от сайта <http://apache.online.bg/httpd/binaries/win32/> В появилия се диалогов прозорец изберете Next и след това Yes за приемане на лицензните условия.
- След натискане на бутона Next до появяване на екран за попълване на информация за сървъра , като

мрежов домейн, име на сървъра и e-mail на администратора. За предпочитане е избор на опция 'Run as a service for All Users';

- При въпрос за избор на тип инсталация е за предпочитане избор на Typical, след което Next;
- На въпрос за избор на директория за инсталиране на Apache е възможно да се избере директория примерно E:\Apache или предлаганата C:\Program Files\Apache Group\Apache
- След потвърждаване на дотук избраните опции посредством Next следва копирането на файловете.
- След копиране инсталационната програма изисква избор на Finish, с което завършва процеса на инсталация.

Стартиране и спиране:

apache -k start : стартира се apache
apache -k stop|shutdown|restart : спира или рестартира apache
apache -l : показва списък на компилираните модули и излиза
apache -S : показва настройките на виртуалният хост и излиза
apache -f файл : задава конфигурационния файл
apache -d : задава начална стойност за ServerRoot
apache -k install|uninstall|config : инсталира, деинсталира, конфигурира сервиза Apache
apache -v : версия
apache -X : стартира режим на трасиране чрез единичен процес

Топология на директорията:

bin : Програмни файлове
cgi-bin : CGI скриптове
conf : Конфигурационни файлове
htdocs : HTML документи
icons : Икони
include : Файлове на Apache за включване на езика C
Lib
libexec : Модули които се зареждат
logs : Log-файлове
modules : Модули
proxy : Йерархия на кеша на прокси (ако кеширането е разрешено)

КОНФИГУРИРАНЕ НА АРАСНЕ WEB СЪРВЪР – ФАЙЛОВЕ HTACCESS.CONF, HTTPD.CONF

HTTPD.conf - основен конфигурационен файл на Apache (/conf)

Файлът httpd.conf е текстов файл с множество опции, които могат да се настроят чрез добавянето и премахването на коментари и чрез изтриване на различни опции. Файлът httpd.conf използва директиви за конфигурирането. Това са инструкции към сървъра, които помагат на Apache да работи по определен начин и да открива необходимите за работата му ресурси. Конфигурационните директиви в този файл са групирани в три основни раздела. Разделите улесняват намирането на конфигурационните опции при редактирането. Те са:

- o раздел на обкръжението;
- o раздел на конфигуриране на основния сървър
- o раздел на виртуални хостове.

Основни директиви използвани във разделите на конфигурационния файл и директиви необходими за работа с PHP през CGI интерфейс.

o РАЗДЕЛ НА КОНФИГУРИРАНЕ НА ОСНОВНИЯ СЪРВЪР

ServerAdmin – задаване на електронен адрес. Той се появява на определени страници, например страниците с описание на грешка и може да се използва за контакт с администратора на сървъра при появили се проблеми.

ServerAdmin aiut@tugab.bg

ServerName – Директивата задава името, което идентифицира сървъра и което е част от URL адреса попълван от клиента По подразбиране е localhost

ServerName aiut_1

Port – Директивата задава името на порта, на който Apache ще следи за заявки. Стандартния порт за HTTP е 80. Може да се избере порт от 0 до 65535

Port 80

ServerRoot – указва директорията на Apache, където е извършена инсталацията. Ако е необходимо да се прехвърли целия сървър в друга директория, то е необходимо параметъра да се актуализира.

ServerRoot "e: /apache"

DocumentRoot – задава главната директорията на Apache за документи. Тази директория е мястото, където се съхраняват всички файлове (директории) със съдържание свързани с Web сайтове.

DocumentRoot "e: /Apache/htdocs"

<Directory ime na direktoria></Directory>. Тук може да се задават различни опции за дадената директория. Задължително име поне две такива директиви – една за DocumentRoot и една за директорията зададена чрез ScriptAlias. В тази директива може да се задават различни поддирективи, които имат влияние само за дадената директория.

- **Options** – задава различни опции на директорията:
 - o Indexes означава, че ако няма някои от документите по подразбиране (зададени в DirectoryIndex), то ще се показва съдържанието на директорията.
 - o FollowSymLinks е опция указваща на сървъра да следва символичните линкове.
 - o ExecCGI, то в дадената директория ще може да се изпълняват CGI скриптове.
 - o Includes - разрешава SSI (Server Side Includes).
 - o All – Установява всички опции;
- **DirectoryIndex** – указва реда в който ще се търсят документите по подразбиране в дадената директория.
- **AllowOverride** – ако стойността е None, то всички директиви зададени във файла htaccess.conf се игнорират. Ако стойността е All, това позволява да се предефинират опциите зададени в директивата <Directory> от файла htaccess.conf. Името на този файл може да се променя чрез директивата AccessFileName.

Забележка: AllowOverride [options ...]

Опцията AllowOverride определя набора директиви, допустими в файла htaccess.conf. Като параметри могат да се използват:

- AuthConfig - разрешава авторизация по име на потребител и парола;
- FileInfo - разрешава директиви, отговарящи за типа документи;
- Limit - разрешава командите allow и deny, които ограничават достъпа до дайл в зависимост от адреса на клиентския компютър;
- None – указва че се игнорират всички родителски файлове htaccess.conf
- All – разрешава всички директиви на родителския htaccess.conf
- **Allow from all | host** - Директивата се използва за да зададе хостовете които ще имат достъп до ресурсите на сървъра.
- **Deny from all | host** - Директивата се използва за да забрани хостовете които не трябва да имат достъп до ресурсите на сървъра.
- **Order** – указва в какъв ред ще се извършва проверката за това дали даден хост и дадено потребителско име има достъп до сървъра.

Пример за използването на директивата:

```
<Directory /www.aiut.bg>
Options Indexes ExecCGI
DirectoryIndex index.php index.html index.htm
AllowOverride All
Order allow,deny
Allow from all
Deny from 192.168.222.2
</Directory>
```

AddType Понякога се налага да се асоциира дадено разширение на файла към определен MIME тип. Например, за да знае сървъра какво да прави с PHP файловете (освен зареждането на модулите) е необходимо файловете с разширение .php да се асоциират към MIME типа *application/x-httpd-php*.

AddType application/x-httpd-php .php

DirectoryIndex <списък на индекс файлове>

Файл индекс - това е този файл, който ще бъде предаден при обръщение към директорията (без задаване на точно определен файл). Ако са указани няколко файла, сървъра ще търси подходящия файл от ляво на дясно. Първоначално е зададен - index.html, но може да се прибавят и други.

DirectoryIndex index.html index.php login.php

AccessFileName <име на файла регламентиращ достъп>

Директивата се използва за задаване на име на файла, който ще бъде проверяван във всяка директория, преди да бъде разрешен достъп до ресурсите на тази директория

AccessFileName htaccess.conf

Освен това за да нямат достъп потребителите до този файл `htaccess.conf` с помощта на директивата **Files** (съдържа описаните по-горе две поддирективи `Order`, `Deny`, `allow` на `Directory` само че регламентира файлове) се задава:

```
<Files htaccess.conf>
    Order allow,deny
    Deny from all
</Files>
```

Alias `psevdoime realnoime\`

Позволява Apache да съхранява документи и директории, различни от директорията зададена в `documentRoot`. Псевдо-името е псевдопът до файла, докато реалното име представя действителното разположение на файла.

`Alias /nov /all/exper`

Възможен е URL: <http://localhost/nov/info.php> --файла ще бъде изтеглен от `/all/exper`

За да се включи поддръжка например на изпълними PERL, PHP, SAPI скриптове и програми се използва директивата

ScriptAlias - задава директорията на диска, на която ще се съхраняват изпълними програми Аналогична на `Alias`, но тази се използва за разглеждането на файловете в зададената директория като скриптове, които могат след това да се изпълняват:

```
ScriptAlias /cgi-bin/ "e:/Apache/Apache2/cgi-bin/"
```

```
ScriptAlias /php/ "E:/php/" --изискване за изпълнение на php скриптове
```

```
ScriptAlias /sapi/ "E:/php/sapi/"
```

```
ScriptAlias /dll/ "E:/php/"
```

AddHandler свързва файлово разширение към определен изпълнител (handler). определя, че всички файлове с посоченото разширение ще се обработват като изпълними.

```
AddHandler cgi-script .bat .exe .cgi
```

Директивата **ErrorDocument** позволява да се замени стандартното съобщение за грешка със свое съобщение. Например, в случай на най-разпространената грешка - 404 (файлт не е намерен) - може да се предложи на потребителя страница със съответните предложения за продължаване със други страници.

```
ErrorDocument 404 /missing.html --В главната директория на всеки виртуален сървър създаваме файл missing.html.
```

Action се използва за задаване на действие. Това действие активира ресурс в директория, конфигуриран като CGI скрипт. От друга страна действието може да е обработчик или MIME тип на информацията, когато клиентската заявка изисква дадено действие.

```
Action application/x-httpd-php /php/php.exe
```

Redirect - се използва при премахването на документи от сайта. При такива ситуации всички URL адреси, сочещи към тези документи могат да се пренасочат към алтернативен сайт:

```
Redirect /www/aiut.bg http://aiut.tugab.bg
```

ErrorLog - се използва за посочване на пътя към файл за логове в които се пазят грешките на сървъра:

```
ErrorLog /logs/error_log
```

LogLevel - задава степента на подробност на грешките. По подразбиране стойността в `warn`, което означава, че към файла за логове се изпращат грешки от тип `Warning` или по-сериозни. Възможните стойности са `debug`, `info`, `notice`, `warn`, `error`, `crit`, `alert`, `emerg`. Най-високо ниво на съобщаване на грешки е `debug`, а най-ниско - `emerg`

```
LogLevel debug
```

○ РАЗДЕЛ НА ОБКРЪЖЕНИЕТО

Timeout - задава времето, което Apache трябва да изчака преди да изпрати съобщение за изтекло време за отговор на заявката

```
Timeout 300 # 300 секунди
```

MaxClients - задава максималния брой на едновременно свързани клиента. По подразбиране стойността е 150, което може да не е достатъчно за натоварени сървъри

Listen - това е важна директива, която може сериозно да повлияе на бързината и сигурността на сървъра. Чрез нея се задава порта и IP адреса на който ще „слуша“ сървъра за заявени връзки. По подразбиране порта е 80 за локалната машина. Можете да задавате или само порт (`Listen 5000`) или комбинация от порт и IP адрес (`Listen 192.168.7.102:80`). Можете да задавате повече от една комбинация от IP адрес и порт, като подреждате `Listen` директивите последователно една след друга. Ако зададените IP адреси не са достъпни за сървъра, той ще откаже да се стартира.

LoadModule - директива, чрез която може да зареждате различни модули, като по този начин разширявате функциите поддържани от сървъра. Например : `LoadModule mime_module modules/mod_mime.so` Тук `modules` е директорията в която са инсталирани модулите на сървъра.

```
LoadModule php4_module "e:/php/sapi/php4apache2.dll"
```

`mod_perl` - интеграция Apache/Perl обединява мощния език за програмиране Perl и HTTP сървъра Apache. С постоянно стартирания вграден в

сървъра интерпретатор позволява да не се губи време за пускане на външен интерпретатор на Perl като CGI приложение.

AddModule – Всеки зададен модул в Loadmodule трябва да има съответстващ запис в Addmodule. След това модула е активен и достъпен за използване

```
AddModule mod_php4.c
Addmodule mod_cgi.c
```

○ **РАЗДЕЛ НА ВИРТУАЛНИ ХОСТОВЕ**

Virtual Hosts. Можете да пуснете няколко Web сайта, които се хостват на един единствен Web сървър. Тези сайтове може да бъдат част от основния домейн или да не бъдат. Всеки отделен виртуален хост трябва да се поставя в своя собствена <VirtualHost></VirtualHost> секция. Вътре в нея се дефинират отделните директиви специфични за дадения виртуален хост. Виртуалните хостове могат да бъдат два типа IP базирани (IP-Based) и Именно базирани (Name-Based). Разликата между тях е, че при първия тип е необходимо за всеки виртуален сървър да се конфигурира отделен IP адрес. За разлика от него, при втория тип можете да конфигурирате множество виртуални сървъри на един единствен IP адрес. Важно е да се знае, че втория тип виртуални хостове не могат да работят в версия на HTTP по-ниска от 1.1.

Ето примерна конфигурация на IP-Based виртуален хост:

```
<VirtualHost 192.168.1.2>
ServerName aiut.com
DocumentRoot "e:/Apache/Apache2/htdocs/www/aiut.bg"
</VirtualHost>
<VirtualHost 192.168.1.3>
ServerName kst.com
DocumentRoot "e:/Apache/Apache2/htdocs/www/kst.bg"
</VirtualHost>
```

За да работи това е необходимо да конфигурирате вашия мрежови интерфейс да има повече от едно IP едновременно. (192.168.1.2, 192.168.1.3)

Разбира се, трябва да имате и валидни записи в DNS сървърите които да свързват *aiut.com* и *kst.com* с 192.168.1.2 и 192.168.1.3. Необходимо е да се инсталира DNS сървър за Windows!

Конфигурацията на Name-Based виртуални хостове е подобна. Преди секциите с отделните виртуални хостове, трябва да промените директивата:

```
NameVirtualHost 192.168.1.2
<VirtualHost 192.168.1.2>
ServerName ivan.com
DocumentRoot /home/www/public_html/ivan
</VirtualHost>
<VirtualHost 192.168.1.2>
ServerName gosho.com
DocumentRoot /home/www/public_html/gosho
</VirtualHost>
```

В самите секции на различните виртуални хостове, може да слагате различни директиви, които имат по-висок приоритет от тези в секцията описваща основната конфигурация.

○ **HTACCESS.conf** - файл за конфигуриране направата за достъп за всяка директория

В **HTACCESS.conf** се съдържат директиви, описващи правата на достъп до директории и файлове на Web-сървъра. Необходимо е да се посочи в коя директория ще бъдат документите. Инсталационно тя е apache/htdocs. Нека създадем директории, които ще са кореновите директории за съответните виртуални хостове:

```
/www/aiut.bg/
/www/stud_aiut.net/
```

Всяка конфигурация, която сте избрали да включите във файлът **htaccess.conf** може да бъде зададена и в секцията Directory, на главния конфигурационен файл httpd.conf. С използването на файловете htaccess.conf се премахва необходимостта от многобройни рестартирания, които се налагат при промени в главния конфигурационен файл. Могат да се правят локални промени на конфигурацията, които се отнасят до поведението на директории, но не засягат целия сървър. Използват се директивите: **Options, Deny и Allow**

Options [option [+option] [-option] ...], възможните значения на параметрите:

ExecCGI - разрешава изпълнение на CGI-фалове в дадената директория и под директории;
FollowSymLinks – указва на сървъра да проследи символните връзки по отношение на главната директория за документи;
Includes - разрешават SSI (Server Side Includes);
Indexes – разрешава показването на съдържанието на директорията ако няма име на файл посочен в DirectoryIndex;
MultiViews – възможност множество изгледи в процеса на преговори за съдържанието;
All – Установява всички опции без MultiViews.

Allow from all | host

Директивата се използва за да зададе хостовете които ще имат достъп до ресурсите на сървъра

Deny from all | host

Директивата се използва за да забрани хостовете които не трябва да имат достъп до ресурсите на сървъра

За да работи системата за достъп е необходимо в основния конфигурационен файл да се зададе като параметър името на файла предназначен да указва достъпа за съответна директория в директивата

AccessFileName htaccess.conf

Също така с директивата Files да се укажат права забраняващи достъпа им до клиента:

```
<Files htaccess.conf>
    Order allow,deny
    Deny from all
</Files>
<Directory />
    Options Indexes Includes FollowSymLinks MultiViews
    AllowOverride All
</Directory>
```

ИНСТАЛИРАНЕ И КОНФИГУРИРАНЕ НА PHP

Възможни са два начина за инсталиране на PHP: като CGI приложение или като SAPI модул на Apache.

последователност:

1. Инсталиране на php от инсталационния файл : php-x.x.x-Win32.zip (<http://bg.php.net/downloads.php>)
2. Инсталацията се състои в дезархивиране на файла в зададена директория
3. Преименуване на папката на php :

Инсталиране на PHP като Apache модул

4. Копиране на всички *.dll файлове от php/extension във c:\windowst\system32
5. Копиране на всички *.dll файлове от php/dlls във c:\windowst\system32
6. Копиране на php4ts.dll и php4ts.lib от c:\php във c:\winnt\system32
7. Копиране на файла c:\php\php.ini-dist във c:\windows и преименуване в php.ini
8. Пре- конфигуриране на php.ini файла в директория c:\windows . Параметъра register_globals=off се инициализира с register_globals=on, даващ възможност за предаване на параметри с формите.
9. Пре- конфигуриране на php.ini файла в директория c:\windows . Параметъра Sgi.force_redirect=1 се инициализира с Sgi.force_redirect=0, което ще пренебрегне съобщението за безопасност при стартиране на скрипт.
10. Пре- конфигуриране на php.ini файла в директория c:\windows . Параметъра Extension_dir= e:\php\extensions (разширения, библиотеки), и doc_root = e:\apache\htdocs, safe_mode=On, display_errors=On (показване на съобщения за грешки), post_max_size=4M (максимална големина на файла при метод POST)
11. Разрешаване на разширението за достъп до MSSQL – extension=php_mssql.dll
12. Модифициране на http.conf :
AddType application/x-httpd-php .php
LoadModule php4_module c:/php/sapi/php4apache.dll
AddModule mod_php4.c
13. Рестартиране на Apache web Server
14. Тестване на PHP / Apache . създаване на файл info.php в директорията посочена в documentroot със съдържание:

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

Написва се в брауъра URL : http://localhost/info.php

Инсталиране на PHP като CGI приложение

4. Пре- конфигуриране на php.ini (преименуван от php.ini-dist) файла в директория d:\php . Параметъра register_globals=off се инициализира с register_globals=on, даващ възможност за предаване на параметри с формите.
5. Пре- конфигуриране на php.ini. Параметъра Sgi.force_redirect=1 се инициализира с Sgi.force_redirect=0, което ще пренебрегне съобщението за безопасност при стартиране на скрипт.
6. Пре- конфигуриране на php.ini. Параметърите Extension_dir= d:\php\extensions (разширения, библиотеки), и doc_root = d:\apache\htdocs, safe_mode=On, display_errors=On (показване на съобщения за грешки), post_max_size=4M (максимална големина на файла при метод POST)
7. Разрешаване на разширението за достъп до MSSQL – extension=php_mssql.dll
8. Модифициране на httpd.conf :

```
<Directory "e:/php">
Options ExecCGI
</Directory>
```

AddType application/x-httpd-php .php .php3 .php4 .phtml

ScriptAlias /php/ "e:/php/"

Action application/x-httpd-php "/php/php.exe"

9. Рестартиране на Apache web Server

10. Тестване на PHP / Apache . създаване на файл info.php в директорията посочена в documentroot със съдържание:

```
<?phpinfo();
```

```
?>
```

Написва се в браузъра URL : http://localhost/info.php

Тест на Apache + PHP + MSSQL

```
<?php
$dblocation = "127.0.0.1";
$dbname = "MY_1";
$dbuser = "sa";
$dbpasswd = "";

$dbcnx = @mssql_connect($dblocation, $dbuser, $dbpasswd);
if (!$dbcnx) {
    echo "<p>MSSQL сървъра не е достъпен</p>";
    exit();
}
if (!@mssql_select_db($dbname,$dbcnx) ) {
    echo "<p>БД не е достъпна</p>";
    exit();
}
$ver = mssql_query("SELECT getdate()");
if(!$ver) {
    echo "<p>Грешка в заявката!</p>";
    exit();
}
echo mssql_result($ver,0, 0);
?>
```