Загальна характеристика мережі Internet

По своєї суті ***Internet*** представляє собою одну із найбільш популярних глобальних комп'ютерних мереж. При цьому під комп'ютерною мережею розуміють зв'язані між собою комп'ютери (вузли), які можуть обмінюватися інформацією. ***Вузлом Internet*** може бути як окремий комп'ютер, так і локальна мережа. Взаємодія вузлів не залежить від типів комп'ютерів, їх архітектури і операційних систем, а також фізичної реалізації зв'язку між ними. При цьому проблеми окремого вузла не можуть глобально вплинути на роботу мережі Internet. Обмін інформацією між комп'ютерами відбувається по лініям зв'язку, які називають ***середовищем передачі даних.*** У більшості комп'ютерних мереж використовуються кабельні канали зв'язку, хоча існують і безпровідні мережі. Комп'ютери передають інформацію один одному за допомогою електричного струму, світла, інфрачервоних хвиль, радіохвиль. Інформація від одного комп'ютера до іншого передається окремими порціями (блоками), які називаються ***пакетами***. Використання пакетів пов'язано з тим, що у мережі одночасно може відбуватися багато процесів передачі даних між комп'ютерами. А пакети як раз і дозволяють “розділити” у часі мережу між передаючими вузлами. Якщо потрібна інформація передавалась би вся і одразу, то це призвело б до монопольного захвату мережі.

Глобальна мережа об'єднує комп'ютери, які ще називаються ***хостами***. Хости об'єднуються під мережами. Задачею підмережі є передача даних від хоста до хосту. В глобальній мережі підмережа складається з двох розділених компонентів: ліній зв'язку та перемикаючих елементів. Лінії зв'язку (чи канали) переносять дані від однієї машини до іншої. Перемикаючі елементи є спеціалізованими комп'ютерами, які використовують для з'єднання двох і більше ліній зв'язку. Коли дані з'являються на вхідній лінії, перемикаючий елемент повинен вибрати вихідну лінію для подальшого оптимального

маршруту цих даних. Ці перемикаючі елементи називають ***маршрутизаторами***. Маршрутизація є базовою функцією підтримки мережі операційною системою. Маршрутизацію своїх пакетів до місця призначення виконує будь-яка система в мережі, маршрутизацію ж чужих пакетів (форвардинг) виконують тільки маршрутизатори (також відомі як ***роутери*** або ***шлюзи***). Задачі маршрутизатора при форвардингу пакету:

– прийняти пакет;

– знайти машину, на яку прямує пакет чи наступний маршрутизатор по маршруту до машини (в таблиці маршрутів);

– передати пакет або повернути повідомлення про неможливість його доставки з причин:

– призначення недосяжно (у пакета закінчився “час життя” перш ніж він досяг місця призначення);

– хост недосяжний (комп'ютер або наступний маршрутизатор вимкнений або не існує);

– мережа недосяжна (маршрутизатор не має маршруту в мережу призначення).

– якщо пакет не може бути доставлений внаслідок перевантаження маршрутизатора (або мережі) – відкинути пакет без повідомлень.

Мережа Internet базується на багаторівневій ***архітектурі*** "***клієнт–сервер***". ***Клієнт*** мережі Internet (***робоча станція***) – окремий (персональний) комп’ютер або віддалений термінал мережі, з яких користувачі отримують доступ до ресурсів мережі Internet. ***Сервер*** –

програмний компонент обчислювальної системи, що виконує сервісні функції по запиту клієнта, надаючи йому доступ до певних ресурсів.

Для взаємодії з клієнтом (або клієнтами, якщо підтримується одночасна робота з декількома клієнтами) сервер виділяє необхідні ресурси міжпроцесної взаємодії (розділяєма пам'ять, пайп, сокет) і чекає запитів на відкриття з'єднання. Залежно від типу такого ресурсу сервер може обслуговувати процеси в межах однієї комп'ютерної системи або процеси на інших машинах через канали передачі даних або мережні з'єднання. Існують сервери, які при відсутності запитів на обслуговування можуть простоювати в очікуванні. Тоді як інші при цьому виконують деяку роботу (наприклад, роботу по збору інформації) – для таких серверів робота з клієнтами може бути другорядною задачею.

У слова “сервер” також є друге значення – це апаратне забезпечення, спеціалізоване для того, щоб використовувати як апаратну базу для серверів послуг. ***Апаратними серверами*** (апаратним забезпеченням) називаються вузькоспеціалізовані рішення з вбудованим програмним забезпеченням на відміну від комп'ютерів, де програмне забезпечення необхідно встановлювати. Апаратні сервери (чи роутери), як правило, більш прості і надійні в експлуатації, споживають менше електроенергії і в багатьох випадках більш дешеві. Але разом з тим вони менш гнучкі і часто обмежені в ресурсах. *Сервери* послуг можна запускати на робочої станції, розділяючи ресурси комп'ютера з програмами, що запускаються користувачем. Такий режим роботи називається невиділеним. При виділеному режимі роботи комп'ютер виконує тільки сервісні функції. Сервери можна класифікувати по типу послуг, які вони надають:

Універсальні сервери – особливий вид серверної програми, що не надає ніяких послуг самостійно. Універсальні сервери надають серверам послуг спрощений інтерфейс до ресурсів міжпроцесної взаємодії та/або уніфікований доступ клієнтів до різних послуг. Універсальні сервери часто використовуються для написання інформаційних серверів, яким не потрібна якась специфічна робота з мережею. Більшість внутрішніх і мережних специфічних серверів Windows працює через універсальні сервери;

Сервери-маршрутизатори;

Сервери (мережні служби) – забезпечують стартову ініціалізацію серверів і робочих станцій, трансляцію імен в адреси і навпаки;

Проксі-сервери – забезпечують зв'язок з мережею, яка недосяжна роутингом;

Сервери (інформаційні служби) – надають інформацію про хост (time, daytime, motd), про користувачів (finger, ident);

Файл-сервери – сервери для забезпечення доступу до файлів на диску сервера;

Сервери доступу до даних (сервер баз даних) – обслуговують базу даних і віддають дані по запитах;

Сервер електронної пошти – дозволяє клієнту передавати і одержувати повідомлення;

Сервер новин – для організації конференцій;

Сервер чатів – для обміну повідомленнями в реальному часі;

Сервери віддаленого доступу – забезпечують користувача консольним доступом до віддаленої системи;

Сервери розділення ресурсів – для розділення доступу до апаратних пристроїв (наприклад, сервер друку розділяє доступ до принтера);

Ігрові сервери – для одночасної гри декількох користувачів в єдиній ігровій ситуації.

Успішна робота в мережі Internet в значній мірі залежить від вибору **провайдера**. Під цим поняттям розуміють юридичну або фізичну особу, яка надає користувачам доступ до мережі Internet. Провайдер, іншими словами постачальник, має володіти достатніми технічними можливостями для швидкої передачі даних з Internet в локальну мережу, або/та на домашній комп'ютер. Провайдер повинен мати спеціальну ліцензію на право поставки послуг у мережі Internet. Швидкість передачі даних по каналу, що сполучає провайдера із зовнішнім світом, повинна бути не нижче 512 кілобітів в секунду. Кількість користувачів на одну телефонну лінію провайдера не повинна перевищувати 10 чоловік при мінімально допустимій швидкості модемів провайдера 33600 біт в секунду. Серед послуг, що надаються Internet-провайдером, існують такі:

- доступ до Internet через комутовані та виділені канали;
- виділення дискового простору для зберігання і забезпечення роботи сайтів (хостинг);
- підтримка роботи поштових ящиків або віртуального поштового серверу;
- розміщення устаткування клієнта на платформі провайдера;
- оренда виділених і віртуальних серверів;
- резервування даних.

Обмін інформацією між вузлами Internet відбувається за допомогою спеціальних ***протоколів***, які регламентують спосіб обміну інформацією. Базовим протоколом передачі даних в мережі Internet є протокол TCP/IP (Transport Control Protocol/Internet Protocol). Відзначимо, що ***TCP/IP*** – це множина комунікаційних протоколів, що визначають, як комп'ютери різних

типів з різноманітними операційними системами можуть спілкуватись між собою. Досить часто мережі, в яких протокол TCP/IP є базовим, називають мережами TCP/IP.

Щоб уникнути плутанини при передачі інформації, ведеться контроль за унікальністю імені вузла – його адреси в мережі Internet. В Internet усі комп'ютери ідентифікуються шляхом присвоєння їм цифрової Internet-адреси, яка називається IP-адресою. IP-адреса – це послідовність із 4-х чисел, розділених крапками. Кожне з цих чисел представляє собою восьмирозрядне двійкове значення, яке називається октетом. Кожна частина може приймати значення від 0 до 255. Перша частина IP-адреси означає конкретну частину Internet і називається мережним ідентифікатором (NetworkID) чи маскою підмережі. Друга частина означає конкретний комп'ютер в цій мережі і називається ідентифікатором комп'ютера (HostID). Використання такої структури адреси дозволяє в різних мережах мати однакові адреси комп'ютерів. Така адресація більш зручна, ніж призначення адреси окремому комп'ютеру за допомогою простого переліку.

Для людей простіше запам'ятовувати буквені адреси, ніж послідовність цифр IP-адреси. ***DNS*** (Domain Name System) – це система, що дозволяє перетворювати символьні імена доменів в IP-адреси (і навпаки) в мережах TCP/IP. DNS важлива для роботи Internet, бо для з'єднання з вузлом необхідна інформація про його IP-адресу.

Домен – певна зона в системі доменних імен (DNS) Internet, виділена якій-небудь країні, організації або для інших цілей.

Пароль – секретна комбінація цифр, знаків, слів для захисту інформації від несанкціонованого доступу.

Логін – процес аутентифікації користувача в комп'ютерній системі шляхом вказівки облікового запису і пароля. Також під логіном може матися на увазі безпосередньо обліковий запис користувача.

Після реалізації підключення до мережі Internet користувач може використовувати різноманітні сервіси.

На сьогоднішній день найпопулярнішим сервісом Internety є ***WWW***, ***Web*** або ***Web*** (Word Wide Web – Всесвітня павутина). Web – абстрактний інформаційний простір, в якому користувачу доступні незліченні архіви документів, зв'язаних перехресними посиланнями один на одного. Такі документи називають ***гіпертекстами*** або ***гіпермедіа***. Інформація може бути представлена у вигляді текстів, графіки, звуку або відео зображення. Популярність Web пояснюється зручністю роботи з різноманітними видами інформації. Будь-який користувач може опубліковуватися в Web.

WWW викликала справжню революцію в розвитку інформаційних технологій. Винахідниками WWW вважаються Тім Бернерс-Лі і Роберт Кайо. Тім Бернерс-Лі є автором технологій HTTP, URI/URL і HTML. Як і вся мережа Internet, мережа WWW базується на ***багаторівневій архітектурі*** "***клієнт–сервер***", показаній на рис. 1.1.

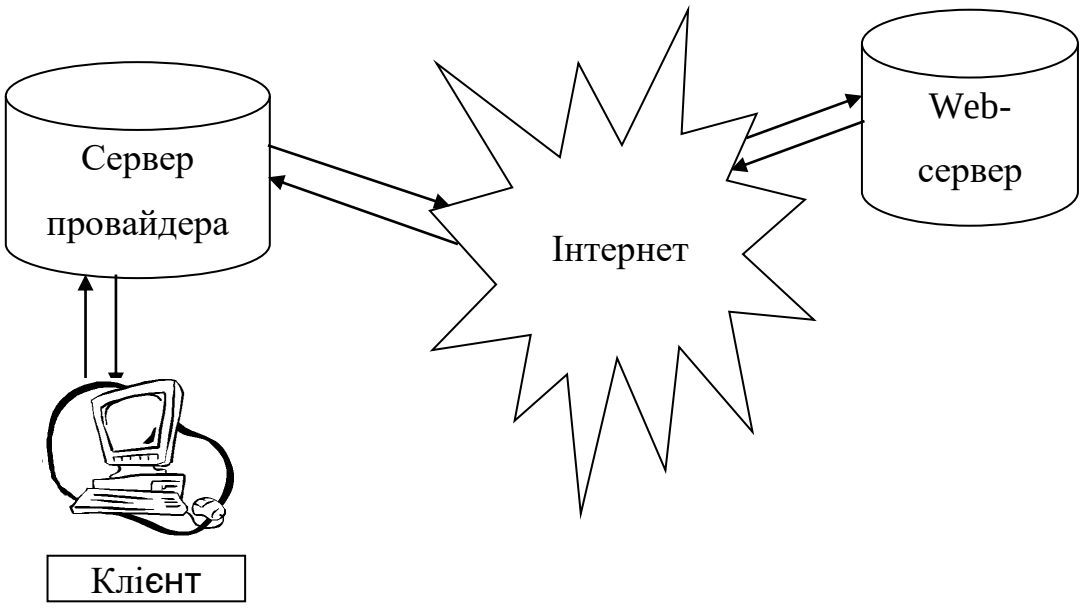

Рис. 1.1 Багаторівнева архітектура мережі WWW

Всесвітню павутину утворюють мільйони Web-серверів мережі Internet, розташованих по всьому світу. Web-сервер є програмою, яка запускається на підключеному до мережі комп'ютері і використовує протокол HTTP для передачі даних. Для того, щоб отримати доступ до певного інформаційного ресурсу мережі WWW (Web-сайту) клієнт (браузер) встановлює TCP-з'єднання з Web-сервером, що обслуговує даний сайт. З'єднання встановлюється за допомогою сервера провайдера та мережі Internet. Після цього браузер відсилає Web-серверу запит на доступ до ресурсу.

Наступним етапом є опрацювання Web-сервером отриманого запиту. В найпростішому випадку опрацювання полягає в пошуку Web-сервером необхідного файлу. В більш складних випадках необхідний інформаційний ресурс повинен бути сформований на основі даних запиту. Під час опрацювання таких запитів Web-сервер досить часто використовує доступне йому прикладне програмне забезпечення та бази даних. Розробка такого

програмного забезпечення дістала назву "програмування на стороні серверу". В процесі розробки найчастіше використовуються технології: CGI, Active Server Pages, JavaServer Pages, Java Servlets, Server-Side JavaScript, SSi та Php.

Після опрацювання запиту відповідь по мережі Internet відсилається клієнту. Відповідь надходить у вигляді ***HTML-документу***. Для перегляду інформації, одержаної від Web-серверу, на клієнтському комп'ютері застосовується спеціальна програма ***Web-браузер***. Основна функція Web-браузера – відображення гіпертексту. Всесвітня павутина нерозривно пов'язана з поняттями гіпертексту і гіперпосилання. Більша частина інформації у WWW представляє собою саме гіпертекст. Для полегшення створення, зберігання і відображення гіпертексту у Всесвітній павутині традиційно використовується мова HTML (мова розмітки гіпертексту). Робота по розмітці гіпертексту називається верстkою, майстри по розмітці називають Web-майстрами. Після HTML-розмітки отриманий гіпертекст поміщається у файл. Такий HTML-файл є найпоширенішим ресурсом Всесвітньої павутини. Після того, як HTML-файл стає доступний Web-серверу, його починають називати Web-сторінкою. Набір Web-сторінок утворює Web-сайт. Потужні Web-сайти на яких роміщена велика кількість різноманітних сервісів іноді називають Web-порталами. Сучасним різновидом Web-сторінки є віджет. Віджет – це Web-сторінка, позбавлена звичного оформлення у вигляді елементів інтерфейсу браузера, яка дозволяє виконувати Web-додатки в браузері. Для створення віджета використовуються стандартні Web-технології, такі як HTML, CSS і JavaScript. Вже існує чимало віджетів для читання рядків новин або здійснення пошуку даних в мережі.

В гіпертекст Web-сторінок додаються гіперпосилання. Гіперпосилання допомагають користувачам WWW легко переміщатися між ресурсами (файлами) незалежно від того, знаходяться ресурси на локальному комп'ютері

або на віддаленому сервері. При знайомстві з документом можна або слідувати за гіперпосиланнями, або ігнорувати їх. Якщо покажчик миші встановити на зв'язок, то він прийме вигляд руки з витягнутим вказівним пальцем. Коли вибираємо гіперзв'язок, то на екрані з'являється новий документ. Знайомим всім аналогом гіперзв'язку є словники. Гіперзв'язки можуть бути виконані і через графічні зображення. Процес слідування за гіперзв'язками від однієї сторінки до іншої називається навігацією або серфінгом.

Для знаходження документу, якій відповідає деякому зв'язку, необхідно знати адресу документу. Кожен ресурс в мережі WWW має свою унікальну адресу (***URL- адреса)***. Доменне ім'я (або IP-адреса) входить до складу URL для позначення комп'ютера (точніше – одного з його мережних інтерфейсів). Цей мережний інтерфейс виконує код потрібного Web-серверу.

Сервер - програмний компонент обчислювальної системи, що виконує сервісні функції за запитом клієнта, надаючи йому доступ до певних ресурсів. Для взаємодії з клієнтом (або клієнтами, якщо підтримується одночасна робота з декількома клієнтами) сервер виділяє необхідні ресурси міжпроцесної взаємодії (розділювальна пам'ять, пайп, сокет) і чекає запитів на відкриття з'єднання. Залежно від типу такого ресурсу сервер може обслуговувати процеси в межах однієї комп'ютерної системи або процеси на інших машинах через канали передачі даних або мережні з'єднання. Існують сервери, які за відсутності запитів на обслуговування можуть простоювати в очікуванні. Тоді як інші при цьому виконують деяку роботу (наприклад, роботу зі збору інформації) - для таких серверів робота з клієнтами може бути другорядною задачею. Досить часто під поняттям сервер також розуміють апаратне забезпечення, на якому функціонує вказаний програмний компонент обчислювальної системи.

Обмін інформацією між вузлами Інтернет відбувається за допомогою спеціальних протоколів, які регламентують спосіб та правила обміну. Базовим протоколом передачі даних в мережі Інтернет є стек протоколів TCP/IP (Transport Control Protocol/ІнтернетProtocol). Відзначимо, що TCP/IP - це множина комунікаційних протоколів, що визначають, як комп'ютери різних типів з різноманітними операційними системами можуть спілкуватись між собою. Відповідно TCP/IP в мережі Інтернет усі комп'ютери ідентифікуються шляхом присвоєння їм цифрової адреси, яка називається IP-адресою. У випадку використання найбільш поширеного протоколу IP v.4, IP-адреса - це послідовність із чотирьох чисел, розділених крапками. Кожне з цих чисел представляє собою восьмирозрядне двійкове значення, яке називається октетом. Кожна частина може приймати значення від 0 до 255. Перша частина IP-адреси означає конкретну частину Інтернет і називається мережним ідентифікатором (NetworkID) чи маскою підмережі. Друга частина означає конкретний комп'ютер в цій мережі і називається ідентифікатором комп'ютера (HostID). Використання такої структури адреси дозволяє в різних мережах мати однакові адреси комп'ютерів. З ідентифікацією комп’ютерів у мережі Інтернет пов’язано використання двох протоколів – DHCP та DNS.

DHCP (Dynamic Host Configuration Protocol) - це мережний протокол, що дозволяє комп'ютерам автоматично одержувати IP-адресу та інші параметри, необхідні для роботи в мережі. Для цього комп'ютер звертається до спеціального серверу, який називається сервером DHCP. Мережний адміністратор може задати діапазон адрес, розподілених серед комп'ютерів. Це дозволяє уникнути ручної настройки комп'ютерів мережі і зменшує кількість помилок.

Для людей простіше запам'ятовувати буквові адреси, ніж послідовність цифр IP-адреси, тому більшість IP-адрес співвідносяться з деяким символьним

іменем, яке дістало назву доменного імені. DNS (Domain Name System) це система, що дозволяє перетворювати символьні імена доменів в IP-адреси (і навпаки) в мережах TCP/IP. DNS важлива для роботи Інтернет, бо для з'єднання з вузлом необхідна інформація про його IP-адресу. Домен певна зона в системі доменних імен (DNS) Інтернет, виділена якій-небудь країні, організації або для інших цілей. Доменне ім'я містить, як мінімум, дві частини (або мітки), розділені крапкою. Права мітка є доменом верхнього рівня (наприклад, для адреси ru.wikipedia.org домен верхнього рівня - org). Кожна наступна мітка справа наліво є піддоменом (наприклад, wikipedia.org - піддомен домену org, а ru.wikipedia.org - піддомен домену wikipedia.org). Теоретично такий розподіл може досягати глибини 127 рівнів, а кожна мітка може містити до 63 символів, поки загальна довжина разом з крапками не досягне 254 символів. Система DNS містить ієрархію серверів DNS. Кожний домен або піддомен підтримується як мінімум одним авторизованим сервером DNS, на якому розташована інформація про домен. Ієрархія серверів DNS співпадає з ієрархією доменів.

Ім'я хоста і IP-адреса є не тотожними поняттями. Хост з одним IP може мати безліч доменних імен, що дозволяє підтримувати на одному комп'ютері безліч Веб-сайтів (це називається віртуальний хостинг). Але і одному імені може бути поставлено у відповідність декілька хостів. За допомогою такого прийому можливо один і той самий ресурс розмістити на декількох комп'ютерах.

Запит на визначення імені зберігається в кеші DNS, який обмежений час пам'ятає відповіді на запити, що проходили через нього раніше. В цьому випадку під терміном «кеш» розуміють інформацію, що тимчасово знаходиться в пам'яті комп'ютера. Організації або провайдери можуть організовувати кеш DNS. Як правило, разом з відповіддю на запит приходить

інформація про те, скільки часу слід берегти цей запис в кеші. DNS використовується в першу чергу для перетворення символьних імен в IP-адреси, але він також може виконувати зворотний процес.

Одна із найбільш популярних мереж, що входить до складу Інтернет, є WWW або Web (Word Wide Web - всесвітня павутина). Популярність цієї мережі пояснюється зручністю роботи з різноманітними видами інформації, від простого тексту до мультимедійних роликів. У свою чергу зручність роботи забезпечується використанням посилань. Посилання (гіпертекстове посилання) – адреса іншого мережевого інформаційного ресурсу у форматі URL (Universal Resource Location), який тематично, логічно або будь-яким іншим способом пов’язаний з документом, у якому це посилання визначене.

Окремий інформаційний ресурс мережі WWW, що має власну адресу, називають Web-сторінкою або Web-сайтом. Для передачі даних в мережі WWW використовується протокол прикладного рівня HTTP (Hyper Text Transfer Protocol).

Програмне забезпечення, що надає користувачам інтерфейс для доступу до інформації Web-сторінок та їх перегляду, називають броузер або браузер. На сьогодні найбільш популярними та поширеними є браузери Microsoft Internet Explorer, Firefox та Opera.

Як і вся мережа Інтернет, мережа WWW базується на багаторівневій архітектурі "клієнт-сервер". Наведемо деякі пояснення стосовно архітектури та функціонування мережі WWW.

Клієнт мережі Інтернет (робоча станція) – окремий (персональний) комп’ютер або віддалений термінал мережі, з яких користувачі отримують доступ до ресурсів мережі Інтернет.

Клієнт мережі WWW – браузер, що використовується клієнтом мережі Інтернет для доступу до ресурсів мережі WWW. Веб-сервер – сервер, який

обслуговує запити користувачів (клієнтів) згідно з протоколом HTTP, забезпечує актуалізацію, збереження інформації WEB-сторінки, зв'язок з іншими серверами. Для отримання доступу до певного інформаційного ресурсу мережі WWW (Веб-сайту) клієнт (браузер) встановлює TCP-з'єднання з Веб-сервером, що обслуговує даний сайт. зЗ'єднання встановлюється за допомогою сервера провайдера та мережі Інтернет. Відзначимо, що кожен ресурс в мережі WWW має свою унікальну адресу (URL-адресу). Механізм іменування та ідентифікації розміщення Web-сторінок реалізується за допомогою URL-адресації. Для цього кожній Web-сторінці призначено уніфікований вказівник інформаційного ресурсу (URL, Uniform Resource Locator). URL-адреса складається із трьох частин:

- протоколу, який також називають схемою;
- DNS-імені комп'ютера, на якому знаходиться Web-сторінка;
- локального імені, що ідентифікує Web-сторінку в межах комп'ютера. Зазвичай в ролі локального імені використовують ім'я файлу, що відповідає Web-сторінці.

Окремі частини URL-адреси розділяються за допомогою спеціальних символів. Розглянемо приклад адреси головної сторінки Web-сайту пошукової системи META: http://www.meta.ua/index.htm. Ця URL-адреса складається із трьох частин: протоколу (http), DNS-імені Web-серверу (www.meta.ua) та імені файлу (index.htm). Крім наведеної абсолютної адресації при визначені гіпертекстових посилань досить часто використовують відносну URL-адресацію. В цьому випадку URL-адреса вказує на ресурс (файл) на тому ж комп'ютері, на якому знаходиться Web-сторінка з гіпертекстовим посиланням. Відносна URL-адреса може містити компоненти відносних шляхів та ідентифікатори фрагментів. Наприклад, вираз "../" означає, що ресурс знаходиться в ієрархії на один рівень вище від файлу, в якому визначене

посилання. Відзначимо, що під час реалізації запитів DNS-ім’я комп’ютера (задане людиною-користувачем в адресному рядку браузера або Web-дизайнером при визначенні гіперпосилання) замінюється відповідною IP-адресою.

Після встановлення з’єднання браузер відповідно протоколу HTTP надсилає Веб-серверу запит на обробку. Запит клієнта поділяється на три розділи. В першому рядку запиту міститься URL-метод та версія HTTP протоколу. Параметр URL-метод може приймати наступні значення:

- GET – запит та отримання документу. При цьому дані надсилаються на сервер також в заголовку запиту.
- HEAD – запит на отримання тільки заголовка документу.
- POST – запит та отримання документу. При цьому дані надсилаються на сервер в тілі запиту у вигляді параметрів.
- PUT – запит на розміщення документу на сервері. Запит з цим методом має тіло, в якому передається сам документ.
- LINK – пов’язує інформацію заголовка з документом на сервері.
- UNLINK – відміняє зв’язок інформації заголовка з документом на сервері.
- DELETE – знищує дані на сервері.
- OPTIONS – запитує інформацію про комунікаційні параметри серверу.
- TRACE – використовується для службових цілей.

Наступні рядки запиту клієнта містять інформацію заголовку. В цій інформації відображаються відомості про клієнта.

Наступним етапом є опрацювання Веб-сервером отриманого запиту. В найпростішому випадку опрацювання полягає в пошуку Веб-сервером необхідного файлу. В більш складних випадках необхідний інформаційний

ресурс повинен бути сформований на основі даних запиту. Під час опрацювання таких запитів

Web-сервер досить часто використовує доступне йому прикладне програмне забезпечення та бази даних. Розробка такого програмного забезпечення дістала назву "програмування на стороні серверу". В процесі розробки найчастіше використовуються технології: CGI, Active Server Pages, JavaServer Pages, Java Servlets, Server-Side JavaScript, SSI, C# та Php. Використання цих технологій потребує професійних знань в галузі програмування. Це є основною з причин того, що "програмування на стороні серверу" не входить до компетенції більшості Web-дизайнерів.

Після опрацювання запиту відповідь по мережі Інтернет відсилається клієнту. Відповідь сервера досить схожа на запит клієнта. Вона також складається з трьох частин. У першому рядку міститься номер версії протоколу HTTP, код помилки та короткий опис коду помилки. У випадку успішного завершення код помилки дорівнює 200, а її опис - "ОК". Друга частина відповіді – це інформація заголовка. Вона може приймати наступні значення:

- Connection – аналогічна відповідному параметру в запиті клієнта.
- Content-Type – тип змісту, який може приймати наступні значення:
    - text/html – текст у форматі HTML (Веб-сторінка);
    - text/plain – простий текст;
    - image/jpeg – рисунок у форматі JPEG;
    - image/gif – рисунок у формате GIF;
    - application/octet-stream - потік "октетів" (тобто просто байт) для запису на диск.

Зазначимо, що перераховано тільки найбільш використовувані типи змісту.

- Content-Length – довжина змісту відповіді в байтах.

- Last-Modified – дата останньої модифікації документа.

Третя частина відповіді – це HTML-документ, за умови його наявності у відповіді.

Після інтерпретації браузером HTML-документ відображається на екрані робочої станції. Зазначимо, що HTML-документ – це файл текстової або нетекстової природи (звук, відео, зображення), створений за допомогою мови гіпертекстової розмітки HTML (Hyper Text Mark-up Language). Застосування універсальної мови HTML зумовлене необхідністю представлення інформації для глобального використання комп'ютерами різноманітних типів, що працюють під управлінням різноманітних операційних систем. Існують стандарти мови HTML, які підтримуються практично всіма провідними компаніями-розробниками програмного та апаратного забезпечення. Ці стандарти передбачають, що HTML-документ буде однаково та коректно представлений всім користувачам мережі WWW. Мова HTML надає розробникам засоби для:

- Публікації HTML-документів з заголовками, текстом, таблицями, списками, рисунками і т.і.
- Навігації по HTML-документам за допомогою гіперпосилань.
- Розробки форм для обміну даними з відділеними службами, що можуть бути використані при пошуку інформації, в електронній комерції і т ін.
- включення електронних таблиць, відео- та аудіокліпів та інших додатків безпосередньо в HTML-документи.

Текстовий HTML-документ представляє собою звичайний текстовий файл з розширенням HTML або HTM. Класичним і до цих пір найбільш поширеним способом передачі інформації від клієнта на Web-сервер є

використання форм. Як правило, результатом такої передачі є відображення у клієнта нового HTML-документа, сформованого Web-сервером на основі переданої інформації.

Коротка характеристика Веб-серверу Apache

Для більшості людей, знайомих з технологіями Internet, поняття Веб-сервер однозначно асоціюється з Apache. І це неспроста – більше 60% Веб-вузлів використовують його як програмну платформу. Ефективність, надійність, можливість використання найсучасніших технологій, реалізація для всіх основних апаратних платформ і операційних систем і на останок відкритий початковий код і ліцензія, що дозволяє вільне розповсюдження, – ось основні чинники, які дозволяють Apache витримувати жорстку конкуренцію з боку комерційних виробників.

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-потоком 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 кросплатформеним програмним забезпеченням, що може функціонувати на операційних системах GNU/Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BEOS. Основними перевагами 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 походить від 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 мовою 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;
- посилені 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 у 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 настройок 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 настройках 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. Парметр 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 прослуховуваня 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 зазвичай 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 у разі виникнення 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 перевизначення слід записати:

AllowOverrides All

17. Директиви виду:

<Directory "ім’я теки">

……

</Directory>

та

<Location "ім’я теки">

……

</ Location >

визначають блок параметрів, що відносяться до цієї 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 та ім'я сервера. Так, рядок <Location http://www.shoop.org/index.html> буде неправильним 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 до 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 (зажадати).

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 ролі 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 доступі 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 парольних даних користувачів та груп користувачів.

– Обмеження доступу до певних тек та файлів, яке базується на методі HTTP-запиту.

Існують модулі, що реалізовують авторизацію через СУБД або PAM. У деяких модулях реалізована можливість запуску кожного процесу Apache із використанням різних 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 даними 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 мережевих 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 у 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 услід іде слово from і список комп'ютерів, яким сервер повинен заборонити або, навпаки, дозволити 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 йти 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 /dlgest_passwd "shoop-users@www.shoop.coni" jem

Adding passowd 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, перелічених 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 form all
allow from nsa.gov
require valid-user
</Limit>

У цьому 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.