

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Уманський державний педагогічний університет імені Павла Тичини

СУЧАСНІ МЕРЕЖНІ ТЕХНОЛОГІЇ

Навчальний посібник

Укладачі Стеценко В.П., Тітова Л.О.

Умань
Візаві
2022

УДК 004.7(075.8)

С91

Рецензенти:

Дякон В. М., кандидат фізико-математичних наук, доцент, директор Уманської філії Європейського університету;

Жмурко О. І., кандидат фізико-математичних наук, доцент, доцент кафедри інформатики і інформаційно-комунікаційних технологій Уманського державного педагогічного університету імені Павла Тичини.

Рекомендовано до друку вченою радою факультету фізики, математики та інформатики Уманського державного педагогічного університету імені Павла Тичини (протокол №8 від 20 січня 2022 р.)

Сучасні мережні технології : навч. посіб. / МОН України, С91 Уманський держ. пед. ун-т імені Павла Тичини ; уклад.: Стеценко В. П., Тітова Л. О. – Умань : Візаві, 2022. – 153 с.

У навчальному посібнику особлива увага приділена принципам організації та функціонування комп'ютерних мереж. Висвітлені питання використання мережних сервісів, класифікації мереж, характеристик середовища передавання даних, еталонної моделі взаємодії відкритих систем, мережних топологій, методів передавання даних та доступу до середовища передавання. Описані основні архітектури сучасних комп'ютерних мереж, розглянуто окремі питання адресації та маршрутизації в мережі Інтернет, дано порівняльні характеристики найбільш поширених мережних операційних систем.

Призначений для студентів, що навчаються за ОПП «Середня освіта (Інформатика)» на освітньому ступені «бакалавр» усіх форм навчання, а також широкого кола користувачів комп'ютерних мереж. Навчальний посібник укладено відповідно до навчальної програми дисципліни «Сучасні мережні технології».

УДК 004.7(075.8)

© Стеценко В. П., Тітова Л. О., уклад., 2022

Зміст

Тема 1. Основи комп'ютерних мереж. Апаратні та програмні засоби комп'ютерних мереж	5
1.1. Історія виникнення та розвитку комп'ютерних мереж	5
1.2. Основні поняття комп'ютерних мереж	6
1.3. Класифікація комп'ютерних мереж	7
1.4. Апаратні та програмні засоби комп'ютерних мереж.	9
Питання для самоконтролю	15
Тема 2. Локальні мережі та особливості їх побудови. Тестування коректності роботи мережевих служб	16
2.1. Локальні мережі	16
2.2. Топології локальних комп'ютерних мереж	17
2.3. Базові технології побудови локальних мереж	20
Технологія Ethernet	20
Технологія Token Ring.....	22
Технологія FDDI	23
Технологія Arcnet.....	24
Технологія LocalTalk	24
2.4. Мережевий стандарт OSI	25
2.5. Адресація в комп'ютерних мережах. Мережеві протоколи.	26
Питання для самоконтролю	30
Тема 3. Захист інформації в комп'ютерних мережах	31
3.1. Основи захисту інформації у комп'ютерних мережах	31
3.2. Антивірусні рішення	32
3.3. Мережеві екрани	34
3.4. Сканери вразливостей	37
3.5. Системи виявлення і попередження вторгнень	37
3.6. Рішення попередження витоку інформації	39
3.7. Шифрування інформації	40
Питання для самоконтролю	43
Тема 4. Проху-сервер. Організація бездротового зв'язку. Стан і перспективи використання комп'ютерних мереж	44
4.1. Проху-сервер	44
Види проху-серверів.....	45
4.2. Організація бездротового зв'язку	47
4.3. Стан і перспективи використання комп'ютерних мереж	53
Питання для самоконтролю	57
Тема 5. Принципи побудови глобальних комп'ютерних мереж. Мережа Інтернет. Основні сервіси. Провайдери	58
5.1. Принципи побудови глобальних комп'ютерних мереж	58
5.2. Структура і основні принципи побудови мережі Інтернет	61
5.3. Способи під'єднання та адресація в мережі Інтернет. Провайдери ..	63
5.4. Основні сервіси мережі Інтернет	66
Служба World Wide Web	66
Служба FTP мережі Інтернет	68

Система обміну миттєвими повідомленнями	69
<i>Питання для самоконтролю</i>	71
Тема 6. Браузери, їх призначення та функціональні можливості. Пошук даних в мережі Інтернет. Налаштування роботи вебсервера.	72
6.1. Призначення та функції браузерів	72
Google Chrome	72
Mozilla Firefox.....	73
Opera	75
6.2. Пошукові системи і каталоги	76
Метапошукові системи.....	83
Інтелектуальні пошукові системи	86
Спеціалізовані пошуковики	88
6.5. Налаштування роботи вебсервера	91
Програмний комплекс «Денвер»	91
Програмний комплекс «Open Server».....	95
<i>Питання для самоконтролю</i>	99
Тема 7. Основи мови HTML. Технології створення вебсторінок	100
7.1. Основні поняття мови HTML. Способи створення вебсторінок	100
7.2. Теги мови HTML	101
<i>Питання для самоконтролю</i>	120
Тема 8. Основи програмування мовою PHP	121
Арифметичні оператори	124
Оператори рядків	124
Оператори присвоювання.....	124
Бітові оператори	125
Логічні оператори	125
Оператори порівняння	126
Умовні оператори.....	126
Цикли.....	129
Математичні функції довільної точності.....	132
Функції Дати/Часу	134
Функції для роботи з каталогами	139
Поштові функції	140
<i>Питання для самоконтролю</i>	141
Тема 9. Робота з електронною поштою, групами новин та розсиланнями 142	
9.1. Електронна пошта (E-mail)	142
9.2. Телеконференції та групи новин (Usenet)	144
9.3. Списки розсилок	146
<i>Питання для самоконтролю</i>	147
Тема 10. Хмарні технології	148
10.1. Основні характеристики хмарних технологій	148
10.2. Хмарні технології для навчальних закладів	149
<i>Питання для самоконтролю</i>	151
Список використаних джерел	152

Тема 1. Основи комп'ютерних мереж. Апаратні та програмні засоби комп'ютерних мереж

1.1. Історія виникнення та розвитку комп'ютерних мереж

Перші електронно-обчислювальні машини з'явилися в 50-роках минулого століття. Вони були дуже великими, складними в експлуатації та мали дорогі апаратні компоненти. Їх використовувала обмежена кількість користувачів. У той час одна і та ж група людей брала участь і в проектуванні, і в експлуатації, і в програмуванні обчислювальної машини. Це була швидше науково-дослідна робота в галузі обчислювальної техніки, а не використання комп'ютерів як інструменту вирішення практичних завдань. Такі комп'ютери не були призначені для інтерактивної роботи. Їх використовували у режимі пакетної обробки. Користувачі підготовляли перфокарти, що містили дані і команди програм, та передавали їх в обчислювальний центр. Оператори вводили ці карти в комп'ютер, а роздруковані результати користувачі одержували зазвичай лише на наступний день.

Питання про передачу даних між кількома окремими ЕОМ виникало з початку їх появи. Тому з розвитком програмного та апаратного забезпечення вдосконалювались і мережеві технології.

Історія виникнення комп'ютерних мереж розпочалась на початку 60-х років. У міру здешевлення процесорів з'явилися нові способи організації обчислювального процесу, що дозволили врахувати інтереси користувачів. Почали розвивати інтерактивні багатотермінальні системи, які для доступу до обчислювального середовища використовували принцип поділу часу. У таких системах комп'ютер віддавався в розпорядження одразу декільком користувачам. Кожен користувач одержував власний термінал, за його допомогою він міг вести діалог з комп'ютером. Термінали, вийшовши за межі обчислювального центру, розосередилися по всьому підприємству. І хоча обчислювальна потужність залишалася цілком централізованою, деякі функції – такі як ввід і вивід даних – стали розподіленими. Така структура називалась централізованою обчислювальною системою.

Застосовуючи інтелектуальні термінали, користувачі могли запускати програми, здійснювати доступ до ресурсів і копіювати файли. Головний комп'ютер мав ідентифікувати користувачів і координувати взаємодію користувача з програмою. Таким чином,

централізовані обчислювальні системи, що працювали у режимі поділу часу, стали першим кроком на шляху створення локальних обчислювальних мереж. Але до появи локальних мереж потрібно було пройти ще великий шлях. Централізовані обчислювальні системи, хоча і мали зовнішні риси розподілених систем, усе ще зберігали сутність централізованої обробки даних [9, с.11].

1.2. Основні поняття комп'ютерних мереж

Сучасний розвиток комп'ютерних мереж починається з моменту появи на робочих місцях персональних комп'ютерів, що давало змогу не робити розподілу часу роботи між користувачами, як це практикувалось з ЕОМ. Проте, персональні комп'ютери, що працювали автономно, не давали безпосереднього доступу до даних усієї організації й не дозволяли спільно використовувати програми й устаткування. Тому виникала потреба у створенні зв'язку між ними [18, с.40].

Комп'ютерна мережа – це комплекс програмних і апаратних засобів, за допомогою яких комп'ютери, накопичувачі даних та офісні електронні пристрої об'єднуються у систему спільного використання ресурсів (пам'яті, принтерів, модемів) і обміну даними.

Основним призначенням комп'ютерних мереж є забезпечення простого, зручного і надійного доступу користувачів до спільних розподілених ресурсів мережі (апаратних, програмних та інформаційних), організація їх колективного використання з надійним захистом від несанкціонованого доступу; забезпечення зручними і надійними засобами обміну даними між користувачами мережі; розподілу обчислень. За допомогою комп'ютерних мереж ці проблеми вирішуються незалежно від територіального розташування користувачів [12, с. 54].

Усі комп'ютери, що підключені до мережі та безпосередньо беруть участь в обміні даними класифікуються як *вузли* або *кінцеві пристрої*. Вузли можуть приймати та надсилати повідомлення по мережі. У сучасних мережах комп'ютерні вузли можуть працювати як клієнт (робоча станція), як сервер, або ж і як клієнт, і як сервер [11, с. 5].

Клієнт (вузол, робоча станція) – це будь-який комп'ютер, що під'єднаний до мережі, через який користувач дістає доступ до її ресурсів. Робоча станція мережі функціонує яку мережевому, так і у

локальному режимі. Вона оснащується власною операційною системою, користувач забезпечується базовим набором інструментів для розв'язування прикладних задач. Робочі станції призначені переважно для напіваавтоматичного розв'язування задач користувачем.

Сервер (від англ. to serve – «обслуговувати») – це потужний комп'ютер або кілька комп'ютерів (один з яких – головний, а решта – резервні), які використовують для об'єднання і розподілу ресурсів комп'ютерної мережі між клієнтами (share – «розподіл»). За допомогою серверів здійснюють зберігання даних, управління базами даних, віддалене опрацювання і друкування даних тощо. Існує кілька типів серверів, орієнтованих на різне застосування:

- *файл-сервер* або *файловий сервер* (для роботи із файлами комп'ютерної мережі);
- *сервер бази даних* (для роботи з базами даних);
- *принт-сервер* (для роботи з принтерами);
- *обчислювальний сервер* (для виконання складних обчислень);
- *сервер прикладних програм* (для роботи з пакетами прикладних програм);
- *сервер зв'язку* (для збільшення пропускних характеристик факсу або модему).

Тип сервера визначається типом ресурсів, які опрацьовують за його допомогою [13, с.58].

1.3. Класифікація комп'ютерних мереж

Різноманіття комп'ютерних мереж можна класифікувати за рядом ознак.

1. За технологією передачі даних мережі поділяються на два типи:

– *віщання* (або з'єднання один – до багатьох). При віщанні повідомлення, відправлене одним комп'ютером, одержують усі комп'ютери мережі;

– *з'єднання «точка-точка»*. З'єднання «точка-точка» розуміє під собою використання індивідуального каналу зв'язку для обміну інформацією між комп'ютерами.

2. За принципом організації обміну даними між абонентами розрізняють мережі, побудовані на комутації каналів, повідомлень, пакетів.

Під *комутацією* розуміють технологію вибору напрямку й організації передачі даних у мережах, що мають кілька альтернативних маршрутів, за якими може проводитися обмін інформацією між двома вузлами.

Інформаційні потоки, що передаються при цьому мережею, називаються *мережевим трафіком* (від англ. traffic – рух).

Мережа з комутацією каналів – тип комунікаційної мережі, у якій кожній парі абонентів надається фізичне з'єднання протягом сеансу їх інформаційної взаємодії. При цьому на час, протягом якого здійснюється сеанс зв'язку між двома абонентами, канал зв'язку стає недоступним для використання іншими абонентами.

Комутація повідомлень розуміє під собою передачу між абонентами інформації у вигляді логічно завершених порцій даних, наприклад телеграм, листів або звітів. При цьому, мережа з комутацією повідомлень працює аналогічно мережі з комутацією каналів, але фізичні канали зв'язку займаються не на період усього сеансу зв'язку, а тільки на період передачі повідомлення.

Мережі з комутацією повідомлень були прообразом для створення мереж з комутацією пакетів і на сьогоднішній день практично не використовуються.

Комутація пакетів – технологія доставки повідомлень, при якій дані, розбиті на окремі блоки малих розмірів, що називаються пакетами, можуть пересилатися з вихідного пункту в пункт призначення різними маршрутами. Пакети у пункті призначення потім збираються у початкові дані.

Мережа з комутацією пакетів – комунікаційна мережа, яка складається зі з'єднаних один з одним магістральними каналами вузлів комутації, у яких дані передаються у вигляді пакетів, із проміжним зберіганням цих пакетів на вузлах комутації. Зважаючи на малі розміри пакетів, що пересилаються, фізичні канали зв'язку виявляються зайнятими протягом мінімальних інтервалів часу, що дозволяє практично завжди забезпечити передачу даних між будь-якими вузлами мережі без тривалих затримок, викликаних необхідністю очікувати звільнення необхідного каналу зв'язку.

3. За територіальною поширеністю мережі можуть бути: локальні, кампусні, міські, глобальні.

Локальна мережа або локальна обчислювальна мережа (Local Area Network – LAN) – це мережа ЕОМ, яка містить у собі комп'ютери, що розташовані у межах одного приміщення, будинку

або невеликої території, і дозволяє обмін даними та спільне використання різних пристроїв (принтерів, сканерів тощо).

Кампусна мережа (від англ. campus – університет, територія університету) – мережа, що охоплює територію університету або студентського містечка.

Міська мережа (Metropolitan Area Network – MAN) поєднує комп'ютери на території міського району або всього міста в цілому.

Глобальна мережа (Wide Area Network – WAN) – сукупність мереж, що поєднують територіально розосереджені комп'ютери, які перебувають у різних містах і країнах.

4. Крім того, мережі можуть поділятися за *топологією*. Топологія мережі може бути шина, зірка, кільце та змішані топології (сітка, дерево та ін.).

5. За швидкістю передачі даних мережі поділяються на:

- низької швидкості (до 10 Мбіт/с);
- середньої швидкості (до 100 Мбіт/с);
- високої швидкості (більше 100 Мбіт/с).

6. За типом середовища передачі даних мережі розділяються на: дротові (коаксіальні, на витій парі, оптоволоконні) і бездротові (радіопередача, супутникові канали).

7. За принципом організації ієрархії комп'ютерів мережі бувають однорангові та з виділеним сервером.

В однорангових мережах усі комп'ютери мають однакові, рівні права (ранги). У мережах з виділеним сервером розрізняють дві архітектури використання сервера:

– файл-сервер – дані та програми на вимогу користувача пересилаються із сервера на комп'ютер клієнту, де можуть бути оброблені;

– клієнт-сервер – виконання програм і обробка даних відбуваються на сервері за запитом користувача, клієнт якого одержує лише результати запиту [5, с.22].

1.4. Апаратні та програмні засоби комп'ютерних мереж.

Апаратне забезпечення функціонування комп'ютерних мереж

До комунікаційної системи належать мережевий адаптер та середовища передавання даних (лінії зв'язку).

Мережевий адаптер (карта, інтерфейсна плата) – технічний пристрій, призначений для з'єднання комп'ютера з каналами зв'язку.

У процесі організації комп'ютерної мережі треба встановити *мережеві плати* (board) на кожний комп'ютер, а потім з'єднати ці плати за допомогою кабелю. Часто як мережеві адаптери використовуються модеми.

Під час підтримки програмних засобів за допомогою мережевих адаптерів під час обміну даними відбуваються такі основні операції:

1. Передавання даних. Дані надходять з оперативної пам'яті в адаптер і навпаки через канал прямого доступу, програмований канал введення-виведення.

2. Буферизація. Під час передавання даних через мережевий адаптер вони спочатку поміщаються в буфер, наявність якого дає змогу через адаптер здійснити доступ до всього пакета, і який повинен мати таку ємність, як і весь пакет даних.

Використання буферів потрібно для узгодження між собою швидкостей (синхронізації) функціонування різних компонентів комп'ютерної мережі.

3. Формування пакета. Через мережевий адаптер файли діляться на частини (під час приймання – об'єднуються), розмір яких залежить від типу мережі (в Ethernet – 1 Кбайт, в Token Ring – 4 Кбайти), додаються до пакетів заголовки і символи кінця, тобто створюється пакет даних, готовий до передавання.

4. Доступ до кабелю. Перед початком передавання даних адаптер чекає надходження сигналу про можливість зайняти лінію або впевнюється, що лінію не зайнято.

5. Перетворення даних. Дані передаються через кабель послідовно, біт за бітом, для чого здійснюється їхнє відповідне подання.

6. Кодування (декодування) даних, тобто формування електричних сигналів, які використовують для передавання даних.

7. Приймання (передавання) імпульсів. Дані як закодовані електричні сигнали надходять через кабель.

Адаптери відрізняються за методами доступу до середовища передавання даних, протоколами передавання і мають такі основні характеристики:

- швидкість передавання даних;
- тип шини комп'ютера;
- розрядність (8, 16, 32, 64 біт);
- ємність буфера для пакета даних;
- сумісність з різними мікропроцесорами;

– топологію – конфігурацію з'єднання елементів у мережі.

У будь-якій комп'ютерній мережі існує середовище обміну, в якому передавання даних здійснюється за допомогою електричних чи електромагнітних сигналів. Середовище, де поширюються електромагнітні хвилі, може бути обмеженим (фізичний провідник сигналу – кабель: телефонний, коаксіальний, вита пара, волоконно-оптичний) або необмеженим (передавання мікрохвильових, інфрачервоних або радіосигналів через відкритий ефір).

Кожен з фізичних провідників має свої обмеження щодо швидкості затухання сигналу, стійкості до впливів сторонніх електромагнітних перешкод тощо. В оцінці якості каналів зв'язку використовують величину – пропускні характеристики, що характеризують максимальну кількість біт, що можуть бути передані від відправника до отримувача за 1 секунду.

Як середовища передавання даних в комп'ютерних мережах можна використовувати:

1. Телефонні кабелі. Їх найчастіше використовують для під'єднання домашнього або офісного комп'ютера до глобальної мережі за допомогою модему. Пропускні характеристики телефонних кабелів становлять кілька десятків Кб/с, тому їх вважають «повільними» каналами зв'язку.

2. Коаксіальний кабель (coaxial) складається з мідного проводу, покритого ізоляцією, що екранує металеву опліткою, і зовнішньою оболонкою [1, с. 30]. За своєю будовою та зовнішнім виглядом нагадують кабелі телевізійних антен, однак відмінні від них за електричними характеристиками.

Розрізняють одно- та багатоканальні кабелі. Їхні переваги – висока стійкість до перешкод, легкість монтажу, середні пропускні характеристики (до 10 Мбіт/с), простота під'єднання нових вузлів.

3. Виті пари (twisted pair) містять 2 або більше (парну кількість) взаємоізольованих звитих між собою на зразок спіралі провідників. Скручування зменшує дію електро-магнітних впливів. Існує кілька різних категорій цих кабелів залежно від розмірів, ізоляції, ступеня скручування, наприклад: UTP (Unshielded twisted pair – неекранована вита пара), STP (Shielded twisted pair – екранована вита пара), SFTP (Shielded Foiled twisted pair – фольгована екранована вита пара). Особливості: легкість монтажу, невелика захищеність від наводок (захищеність підвищується для витих пар, поміщених в екрануючу оболонку) і механічних пошкоджень, використання в невеликих

локальних мережах. Пропускні характеристики витих пар в середньому сягають до 100 Мб/с.

4. Волоконно-оптичні (оптоволоконні) кабелі (fiber optic) виготовляють зі скла або світлопровідних пластикових волокон. Волоконно-оптичні кабелі проводять дані, які попередньо перетворюються у світлові імпульси за допомогою лазера або світлодіодів та які приймає фотодетектор, де вони знову перетворюються на електричні сигнали. Використовують ці кабелі для передавання даних з високою швидкістю (пропускні характеристики – до кількох Гб/с) і мінімальними втратами. Особливості: висока вартість кабелю і обладнання, складний монтаж, складна технологія і створення розгалужень, швидкість – до кількох Гб/с на відстані між станціями-регенераторами сигналу до 50 км, передавання сигналу тільки в одному напрямі.

5. Радіоканал (radio channel). Особливості: відсутність кабелів, погана захищеність, екранування сигналу стінами будівель, труднощі з виділенням вільного діапазону частот.

6. Інфрачервоний канал (infrared channel). Інфрачервоні промені використовують на невеликій відстані. Особливості: мобільність в межах офісу, не потребує спеціального діапазону частот, вплив погодних умов.

7. Супутниковий канал (satellite channel). Особливості: дорогий спосіб передавання даних, але максимально можлива площа охоплення [13, с. 58].

До складу комп'ютерної мережі також можуть бути включені:

– *Повторювач (repeater)*. Основною функцією повторювача є повторення сигналів, що надходять до його порту. Повторювач відновлює і підсилює електричні характеристики сигналів та їх синхронність, і за рахунок цього з'являється можливість збільшувати загальну довжину кабелю між віддаленими вузлами в мережі;

– *Концентратор (concentrator), хаб (hub)*. Концентратором або хабом називають багатопортовий повторювач. Він виконує не лише функцію повторення сигналів, але і виконує функції об'єднання комп'ютерів мережі. Практично у всіх сучасних мережних стандартах концентратор є необхідним елементом мережі, що сполучає окремі комп'ютери у мережу.

– *Міст (bridge)*. Перші пристрої, що дозволяли об'єднувати кілька мереж, були двохпортовими і отримали назву мостів. З розвитком даного типу обладнання, вони стали багатопортовими і

отримали назву комутаторів. Певний час обидва поняття існували одночасно, а пізніше замість терміну «міст» стали застосовувати «комутатор».

– *Комутатор (switch)*. Комутатор за функціональністю є подібним до моста і відрізняється від моста в основному вищою продуктивністю. Часто термін «комутатор» використовується у вузькому сенсі, позначаючи конкретний тип пристрою. Кожен порт комутатора оснащено спеціальним процесором, який обробляє кадри незалежно від процесорів інших портів. Комутатор одночасно підтримує потоки даних між всіма своїми портами. Іншими словами, комутатор передає кадри паралельно.

– *Маршрутизатор (router)*. Маршрутизатор – система ретрансляції, що сполучає дві комунікаційні мережі або їх частини. Маршрутизатори працюють на третьому (мережному) рівні моделі OSI, що спілкується з протоколами вищих рівнів.

– *Шлюзи (gateway)*. Шлюз є системою ретрансляції, що забезпечує взаємодію інформаційних мереж. Шлюз дозволяє об'єднувати мережі, що побудовані на істотно різних програмних і апаратних платформах. Наприклад, шлюз може дозволити користувачам, що працюють в мережі Unix, взаємодіяти з користувачами мережі Windows [6, с. 88].

– *Джерела безперебійного живлення (UPS – Uninterruptible Power Supply)*. UPS використовується для забезпечення завершення роботи і збереження даних у разі аварійного зникнення напруги в електричній мережі.

Програмне забезпечення функціонування комп'ютерних мереж

Драйвер мережевого адаптера – це програма, за допомогою якої здійснюють управління функціонуванням мережевого адаптера.

Мережева операційна система (ОС) – це пакет програм для забезпечення функціонування мережі та управління нею, а також забезпечення клієнтам можливостей користуватись мережевими сервісами. Основним призначенням мережевої ОС є забезпечення спільного використання та розподілу ресурсів мережі; обміну повідомленнями між вузлами мережі; синхронізації процесів у мережі; надійного зберігання даних та інших завдань, пов'язаних із функціонуванням мережі. Важливою функцією мережевої ОС є забезпечення системи захисту даних і програм – конфіденційності зберігання даних, розмежування прав доступу до ресурсів, парольний захист, виявлення спроб несанкціонованого доступу,

відслідковування дій користувача, ведення журналів системних подій тощо.

Через мережеву ОС забезпечується підтримка функціонування різноманітних периферійних пристроїв, мережевих адаптерів, протоколів та можливість їхнього конфігурування.

За допомогою програмного забезпечення клієнтської частини перетворюються запити від прикладної програми на використання мережевих ресурсів у відповідні мережеві протоколи, забезпечуються пересилання через середовища передавання даних та здійснюються зворотні перетворення. Клієнтська частина залежить від ОС, що встановлена на робочій станції, та типів мереж.

Терміни «клієнт» та «сервер» використовують не тільки щодо апаратної складової, а також і для означення програм, у яких використовують так звані технології «клієнт-сервер».

Сервери – це програми, за допомогою яких передають повідомлення до клієнтів. На комп'ютері, на якому запущено програму-клієнт, можна запустити і програму-сервер, що буде надавати повідомлення і послуги клієнтові. Сервери акумулюють дані та способи їх опрацювання, роблячи програму-клієнт придатною для виконання завдань, де без серверів вона непридатна. На одному комп'ютері можна запустити одразу кілька програм-серверів, наприклад, сервер баз даних, вебсервер або сервер електронної пошти. Прикладом програм-клієнтів є веббраузери, програми для читання та відправки електронної пошти тощо.

Інформаційне забезпечення комп'ютерних мереж

Інформаційне забезпечення комп'ютерної мережі – це єдиний інформаційний фонд, який орієнтовано на розв'язування в мережі різноманітних задач. У цьому фонді звичайно міститься масив даних загального використання, доступний для всіх користувачів (абонентів) мережі і масиви індивідуального використання, призначені для окремих абонентів. Сюди входять бази знань, автоматизовані бази даних (локальні і розподілені, суспільного та індивідуального призначення) [13, с. 60].

Питання для самоконтролю

1. Що таке комп'ютерна мережа?
2. Які основні складові комп'ютерної мережі?
3. Яка мережа називається локальною?
4. Що таке мережевий адаптер?
5. Для чого призначені драйвери мережевих адаптерів?
6. Що таке сервер?
7. Який комп'ютер мережі називають клієнтом?
8. Назвіть середовища передавання даних у комп'ютерних мережах.
9. Які існують види комп'ютерних мереж?
10. Мережева операційна система – функції, склад та структура.
11. Принцип архітектури «клієнт-сервер».
12. Чим відрізняється глобальна мережа від локальної?
13. Поясніть різницю між поняттями клієнт і сервер в мережі.
14. Що таке міст?
15. Що таке комутатор?
16. Що таке повторювач?
17. Дайте визначення мережевої операційної системи.
18. Що є основним заданням будь-якої мережевої операційної системи?
19. Дайте означення концентратора.
20. Що належить до програмного забезпечення мережі?
21. Які середовища передавання даних в комп'ютерних мережах можна використовувати?
22. Що таке мережеві ресурси?
23. Які мережі називають одноранговими?
24. Які з мережевих кабелів володіють найбільшою швидкістю і якістю передачі даних?
25. Які типи витої пари Вам відомі?

Тема 2. Локальні мережі та особливості їх побудови. Тестування коректності роботи мережевих служб

2.1. Локальні мережі

Важлива подія, що вплинула на еволюцію комп'ютерних мереж, відбулася на початку 70-х років. У результаті технологічного прориву в області виробництва комп'ютерних компонентів з'явилися великі інтегральні схеми (ВІС). Їх невисока вартість і відмінні функціональні можливості призвели до створення міні-комп'ютерів, що стали реальними конкурентами мейнфреймів. Навіть невеликі підрозділи підприємств одержали можливість мати власні комп'ютери. Міні-комп'ютери виконували задачі керування технологічними процесами та технологічним устаткуванням, складом і іншими задачами рівня відділу підприємства.

Таким чином, з'явилася концепція розподілу комп'ютерних ресурсів по всьому підприємству.

Потреби користувачів обчислювальної техніки росли. Їх уже не задовольняла ізольована робота на власному комп'ютері. Постала потреба обміну комп'ютерними даними з користувачами інших підрозділів в автоматичному режимі. Відповіддю на цю потребу стала поява перших локальних обчислювальних мереж.

Локальні мережі (Local Area Networks, LAN) – це об'єднання комп'ютерів, зосереджених на невеликій території, зазвичай в радіусі не більш 1-2 км, хоча в окремих випадках локальна мережа може мати і більш протяжні розміри, наприклад у кілька десятків кілометрів. У загальному випадку локальна мережа є комунікаційною системою, що належить одній організації.

Спочатку для з'єднання комп'ютерів один з одним використовували нестандартні програмно-апаратні засоби. Різноманітні пристрої сполучення, що використовують свій власний спосіб представлення даних на лініях зв'язку, свої типи кабелів та ін., могли з'єднувати тільки ті конкретні моделі комп'ютерів, для яких вони були розроблені.

У середині 80-х років такий підхід до побудови локальних мереж змінився. Затвердили стандартні технології об'єднання комп'ютерів у мережу – Ethernet, Arcnet, Token Ring, Token Bus, трохи пізніше – FDDI. Потужним стимулом для їх появи слугували персональні комп'ютери, які вже виробляли масово. Вони стали

ідеальними елементами для побудови мереж – з одного боку, були досить потужними для роботи мережевого програмного забезпечення, а з іншого – мали потребу в об'єднанні своєї обчислювальної потужності для розв'язку складних задач, а також одночасного використання периферійних пристроїв і дискових масивів.

Усі стандартні технології локальних мереж базувались на тому ж принципі комутації, що був з успіхом випробуваний і довів свої переваги при передаванні даних у глобальних комп'ютерних мережах – принцип комутації пакетів.

Кінець 90-х виявив явного лідера серед технологій локальних мереж – сімейство Ethernet, в яке ввійшли класична технологія Ethernet (10 Мбіт/с), а також FastEthernet (100 Мбіт/с) і GigabitEthernet (1000 Мбіт/с). Прості алгоритми роботи визначили низьку вартість устаткування Ethernet. Широкий діапазон ієрархії швидкостей дозволив раціонально будувати локальну мережу, застосовуючи ту технологію сімейства, яка найбільшою мірою відповідає розв'язку завдань підприємства і потребам користувачів [9, с. 14].

2.2. Топології локальних комп'ютерних мереж

Фізична топологія

Геометричну форму проекції середовища передавання даних на площину називають фізичною топологією (конфігурацією) мережі. Залежно від способу з'єднання вузлів мережі можуть використовуватись такі фізичні топології:

- шина (Bus);
- кільце (Ring);
- зірка (Star);
- змішані топології.

Шинна топологія відповідає конфігурації мережі, в якій усі вузли підімкнені до спільного лінійного каналу з допомогою відносно коротких з'єднань:



Рис. 2.2.1. Топологія «шина»

Переваги:

- вимагає мінімальної кількості кабелю, оскільки канал підводиться до кожного вузла найкоротшим шляхом;
- нескладне додавання в мережу нового вузла;
- просте управління трафіком між під'єднаними вузлами.

Недоліки:

- пошкодження одного з'єднання між двома вузлами виводить з ладу всю мережу;
- важко локалізувати дефекти середовища передавання.

Кільцева топологія відповідає конфігурації мережі, в якій кожен вузол пов'язаний з двома іншими, а спільний канал утворює кільце:

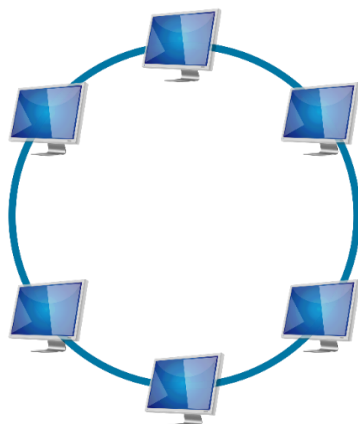


Рис. 2.2.2. Топологія «кільце»

Переваги:

- проста логічна організація;
- відсутність перевантажень середовища передавання.

Недолік, як і в шинній топології, пов'язаний з наявністю одного спільного каналу – пошкодження з'єднання між двома вузлами виводить з ладу всю мережу.

Топологія зірки відповідає конфігурації мережі, в якій усі вузли з'єднані з центральним вузлом (концентратором).

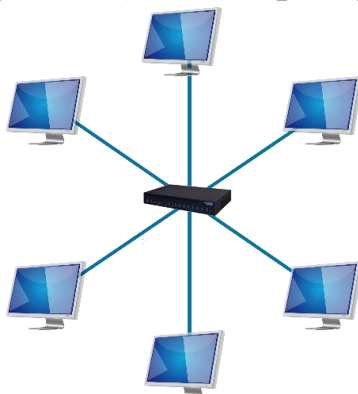


Рис. 2.2.3. Топологія «зірка»

У мережах зіркоподібного типу, звичайно, центральний вузол відповідає за маршрутизацію пакетів та локалізацію несправностей.

Переваги:

- легкість управління;
- надійність роботи та можливість швидкого виявлення дефектів;
- нескладне додавання в мережу нового вузла.

Недоліки:

- вимагає надлишкової кількості кабелю;
- пошкодження концентратора спричиняє вихід з ладу всієї мережі.

Змішані топології

Сіткова топологія відповідає конфігурації мережі, в якій усі вузли з'єднані між собою безпосередніми з'єднаннями.

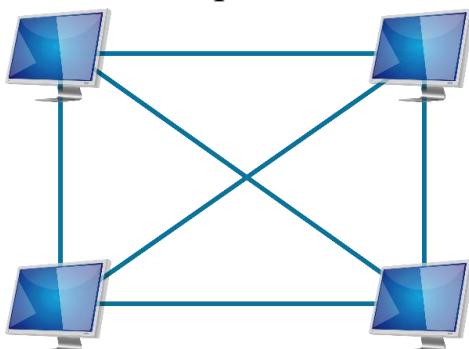


Рис. 2.2.4. Топологія «сітка»

Ідеальна сіткова топологія з N вузлами вимагає наявності у кожного вузла $(N-1)$ -го мережного адаптера.

Переваги:

- висока надійність роботи;
- мінімальний час передавання даних.

Однак висока вартість обладнання, надлишкова кількість кабелю та інші недоліки, звичайно, перешкоджають широкому поширенню такої топології)[4, с. 20].

Топологія «дерево» складається з кількох концентраторів, ієрархічно сполучених між собою зв'язками типу «зірка» [1, с. 51].

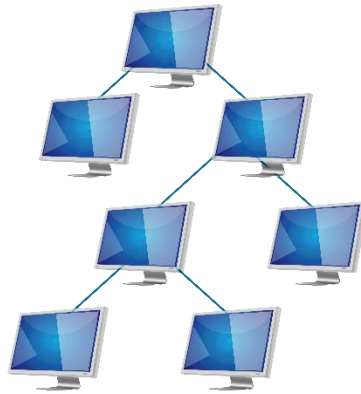


Рис. 2.2.5. Топологія «дерево»

Гібридна (комбінована) топологія використовує різні типи мережних топологій, звичайно, для об'єднання декількох локальних мереж (кампусні, глобальні мережі)[4, с. 21].

Логічна топологія описує можливості з'єднання між парами кінцевих точок мережі, що взаємодіють [1, с. 50].

2.3. Базові технології побудови локальних мереж

Мережева технологія – це узгоджений набір стандартних протоколів та програмно-апаратних засобів, що їх реалізують, достатній для побудови локальної обчислювальної мережі.

Мережева технологія або архітектура визначає топологію і метод доступу до середовища передачі даних, кабельну систему або середовище передачі даних, формат мережеских кадрів, тип кодування сигналів, швидкість передачі в локальній мережі.

Іноді мережеві технології називають базовими технологіями, маючи на увазі те, що на їх основі будується базис будь-якої мережі. До основних базових мережеских технологій належать такі відомі технології локальних мереж як Ethernet, Token Ring та FDDI [5, с. 63].

Технологія Ethernet

Ethernet – це найпоширеніший на сьогоднішній день стандарт локальних мереж. Загальна кількість мереж, що працюють за протоколом Ethernet у цей час, оцінюється в 5 мільйонів, а кількість комп'ютерів із установленими мережними адаптерами Ethernet – в 50 мільйонів.

Коли говорять Ethernet, то під цим звичайно розуміють кожен з варіантів цієї технології. У більш вузькому сенсі Ethernet – це мережний стандарт, заснований на експериментальній мережі

Ethernet Network, що фірма Xerox розробила й реалізувала в 1975 році. Метод доступу був випробуваний ще раніше: у другій половині 60-х років у радіомережі Гавайського університету використовувалися різні варіанти випадкового доступу до загального радіосередовища, що одержали загальну назву Aloha. У 1980 році фірми DEC, Intel і Xerox спільно розробили й опублікували стандарт Ethernet версії II для мережі, побудованої на основі коаксіального кабелю, що став останньою версією фірмового стандарту Ethernet. Тому фірмову версію стандарту Ethernet називають стандартом Ethernet DIX або Ethernet II.

На основі стандарту Ethernet DIX був розроблений стандарт IEEE 802.3, що багато в чому збігається зі своїм попередником, але деякі відмінності є. У той час як у стандарті IEEE 802.3 розрізняються рівні MAC і LLC, в оригінальному Ethernet обидва ці рівні об'єднані в єдиний каналний рівень. У Ethernet DIX визначається протокол тестування конфігурації (Ethernet Configuration Test Protocol), що відсутній в IEEE 802.3. Трохи відрізняється й формат кадру, хоча мінімальні й максимальні розміри кадрів у цих стандартах збігаються. Часто для того, щоб відрізнити Ethernet, визначений стандартом IEEE, і фірмовий Ethernet DIX, перший називають технологією 802.3, а за фірмовим залишають назву Ethernet без додаткових позначень.

Залежно від типу фізичного середовища стандарт IEEE 802.3 має різні модифікації – 10 Base-5, 10 Base-2, 10 Base-T, 10 Base-FL, 10 Base-FB.

У 1995 році був прийнятий стандарт Fast Ethernet, що багато в чому не є самостійним стандартом, про що говорить і той факт, що його опис просто є додатковим розділом до основного стандарту 802,3 – розділом 802.3ч. Аналогічно, прийнятий у 1998 році стандарт Gigabit Ethernet описаний у розділі 802.3z "Основні документи".

Для передачі двійкової інформації з кабелю для всіх варіантів фізичного рівня технології Ethernet, що забезпечують пропускну здатність 10 Мбіт/з, використовується манчестерський код.

Усі види стандартів Ethernet (у тому числі Fast Ethernet і Gigabit Ethernet) використовують той самий метод поділу середовища передачі даних – метод CSMA/CD.

У мережах Ethernet використовується метод доступу до середовища передачі даних, який називається методом колективного доступу (до якого відносяться й радіомережі, що породили цей метод). Усі комп'ютери такої мережі мають безпосередній доступ до

загальної шини, тому вона може бути використана для передачі даних між будь-якими двома вузлами мережі. Одночасно всі комп'ютери мережі мають можливість негайно (з урахуванням затримки поширення сигналу по фізичному середовищу) одержати дані, які кожен з комп'ютерів почав передавати на загальну шину. Кабель, до якого підключені всі станції, працює в режимі колективного доступу (Multiply Access, MA). Простота схеми підключення – це один з факторів, що визначили успіх стандарту Ethernet [6, с. 100].

Технологія Token Ring

Мережі Token Ring, так само як і мережі Ethernet, характеризує поділюване середовище передачі даних, що у цьому випадку складається з відрізків кабелю, що з'єднують всі станції мережі в кільце. Кільце розглядається як загальний поділюваний ресурс, і для доступу до нього потрібен не випадковий алгоритм, як у мережах Ethernet, а детермінований, заснований на передачі станціям права на використання кільця в певному порядку. Це право передається за допомогою кадру спеціального формату, називаного *маркером* або *токеном (token)*.

Технологія Token Ring була розроблена компанією IBM у 1984 році, а потім передана як проект стандарту в комітет IEEE 802, що на її основі прийняв у 1985 році стандарт 802.5. Компанія IBM використовує технологію Token Ring у якості своєї основної мережної технології для побудови локальних мереж на основі комп'ютерів різних класів – мейнфреймів, міні-комп'ютерів і персональних комп'ютерів. У цей час саме компанія IBM є основним законодавцем моди технології Token Ring, роблячи близько 60% мережних адаптерів цієї технології.

Мережі Token Ring працюють із двома бітовими швидкостями – 4 і 16 Мбіт/с. Змішування станцій, що працюють на різних швидкостях, в одному кільці не допускається. Мережі Token Ring, що працюють зі швидкістю 16 Мбіт/с, мають деякі вдосконалення в алгоритмі доступу порівняно зі стандартом 4 Мбіт/с.

Технологія Token Ring є більш складною технологією, ніж Ethernet. Вона має властивості відмовостійкості. У мережі Token Ring визначені процедури контролю роботи мережі, які використовують зворотний зв'язок кільцеподібної структури – посланий кадр завжди вертається в станцію-відправника. У деяких випадках виявлені помилки в роботі мережі усуваються автоматично, наприклад, може

бути відновлений загублений маркер. В інших випадках помилки тільки фіксуються, а їхнє усунення виконується вручну обслуговуючим персоналом.

Для контролю мережі одна зі станцій виконує роль так званого активного монітора. Активний монітор вибирається під час ініціалізації кільця як станції з максимальним значенням MAC-адреси. Якщо активний монітор виходить із ладу, процедура ініціалізації кільця повторюється й вибирається новий активний монітор. Щоб мережа могла виявити відмову активного монітора, останній у працездатному стані кожні 3 секунди генерує спеціальний кадр своєї присутності. Якщо цей кадр не з'являється в мережі більше 7 секунд, то інші станції мережі починають процедуру виборів нового активного монітора [6, с. 134].

Технологія FDDI

Технологія FDDI (Fiber Distributed Data Interface) – оптоволоконний інтерфейс розподілених даних – це перша технологія локальних мереж, у якій середовищем передачі даних є волоконно-оптичний кабель. Роботи зі створення технологій і пристроїв для використання волоконно-оптичних каналів у локальних мережах почалися в 80-ті роки, незабаром після початку промислової експлуатації подібних каналів у територіальних мережах. Проблемна група X3T9.5 інституту ANSI розробила в період з 1986 по 1988 р. початкові версії стандарту FDDI, що забезпечує передачу кадрів зі швидкістю 100 Мбіт/с по подвійному волоконно-оптичному кільцю довжиною до 100 км.

Мережа FDDI будується на основі двох оптоволоконних кілець, які утворюють основний і резервний шляхи передачі даних між вузлами мережі. Наявність двох кілець – це основний спосіб підвищення відмово стійкості в мережі FDDI, і вузли, які хочуть скористатися цим підвищеним потенціалом надійності, повинні бути підключені до обох кілець.

У нормальному режимі роботи мережі дані проходять через усі вузли й усі ділянки кабелю тільки первинного (Primary) кільця, цей режим названий режимом Thru – "наскрізним" або "транзитним". Вторинне кільце (Secondary) у цьому режимі не використовується.

У випадку якого-небудь виду відмови, коли частина первинного кільця не може передавати дані (наприклад, обрив кабелю або відмова вузла), первинне кільце поєднується із вторинним, знову

утворюючи єдине кільце. Цей режим роботи мережі називається **Wrap**, тобто "згортання" кільця. Операція згортання виробляється засобами концентраторів і/або мережних адаптерів **FDDI**. Для спрощення цієї процедури дані по первинному кільцю завжди передаються в одному напрямку (на діаграмах цей напрямок зображується проти годинникової стрілки), а по вторинному – у зворотному (зображується за годинниковою стрілкою). Тому при утворенні загального кільця із двох кільць передавачі станцій, як і раніше, залишаються підключеними до приймачів сусідніх станцій, що дозволяє правильно передавати й приймати інформацію сусідніми станціями.

У стандартах **FDDI** багато уваги приділяється різним процедурам, які дозволяють визначити наявність відмови в мережі, а потім зробити необхідну реконфігурацію. Мережа **FDDI** може повністю відновлювати свою працездатність у випадку одиничних відмов її елементів. При множинних відмовах мережа розпадається на кілька не зв'язаних мереж. Технологія **FDDI** доповнює механізми виявлення відмов технології **Token Ring** механізмами реконфігурації шляхи передачі даних у мережі, заснованими на наявності резервних зв'язків, забезпечуваних другим кільцем [6, с. 143].

Технологія Arcnet

Використовується в топології "зірка". Повідомлення до комп'ютерів передаються за допомогою маркерів. Під час відправлення повідомлень до них приєднуються маркери, і дані посилаються в мережу; під час отримання повідомлень аналізується заголовок маркера, і в разі його відповідності адресі комп'ютера повідомлення відокремлюються та залишаються на комп'ютері.

Технологія LocalTalk

Це протокол фізичного і канального рівнів, що належить **Apple Computer**. Він сконструйований для маленьких робочих груп і вбудований в кожний комп'ютер **Apple Macintosh**. В **LocalTalk** використовується шинна топологія на екранованій витій парі. Метод доступу – конкурентний. На відміну від інших протоколів цікавим є динамічне призначення мережеских адрес. Швидкість передавання даних – 230 Кбіт/с [16, с. 31].

2.4. Мережевий стандарт OSI

Одним з основних завдань при створенні комп'ютерних мереж є забезпечення сумісності обладнання за електричними й механічними характеристиками та забезпечення сумісності програм і даних за системами кодування й форматами даних. Розв'язання цієї задачі відноситься до галузі стандартизації та засноване на так званій моделі OSI (модель взаємозв'язків відкритих систем – Model of Open System Interconnections). Вона створена Міжнародною організацією стандартизації ISO (International Standards Organization) і є еталонним міжнародним стандартом з передавання даних. Дана модель має ієрархічну структуру і складається з семи рівнів:

- фізичний;
- канальний;
- мережевий;
- транспортний;
- сеансовий;
- рівень подання даних;
- прикладний.

Відповідно до моделі OSI архітектуру комп'ютерних мереж потрібно розглядати на різних рівнях. Найвищий рівень – прикладний. На цьому рівні користувач працює з обчислювальною системою. Найнижчий рівень – фізичний. На ньому забезпечується обмін сигналами між пристроями. Обмін даними в системах зв'язку відбувається шляхом їхнього переміщення з верхнього рівня на нижній, потім транспортування і, нарешті, зворотним відтворенням на комп'ютері клієнта.

Розглянемо детальніше кожний з цих рівнів:

Фізичний рівень. На цьому рівні розглядаються проблеми передавання потоку даних через передавальне середовище. Тут визначаються електричні, механічні, функціональні та процедурні параметри фізичного зв'язку.

Канальний рівень (рівень з'єднання). Формуються дані, що утворюються на першому рівні, у так звані кадри чи послідовності кадрів. На цьому рівні відбуваються управління доступом до передавального середовища, синхронізація, виявлення і виправлення помилок.

Мережевий рівень. Встановлюється зв'язок між 2-ма вузлами мережі. З'єднання виникають завдяки функціям маршрутизації, що

потребують наявності мережевої адреси в пакеті, що передається. На цьому рівні забезпечується також відслідковування та виправлення помилок, мультикомплексування, управління потоками даних.

Транспортний рівень. Підтримується направлене передавання даних між двома користувачькими процесами.

Сеансовий рівень. Координується приймання і передавання даних в одному сеансі зв'язку. Для координації необхідний контроль робочих параметрів, управління пакетами даних. На цьому рівні використовуються такі функції: управління пакетами даних та паролем, розрахунку плати за використовуваний ресурс, синхронізації і скасування зв'язку в сеансі при виникненні помилки на нижчих рівнях.

Рівень подання даних. Призначений для інтерпретації даних і підготовки їх для прикладного рівня. Тут описуються методи перетворення даних (з одного формату до іншого, шифрування, кодування, стискування тощо), що передаються для використання на прикладному рівні.

Прикладний рівень. На цьому рівні дані автоматично опрацьовуються за програмами, змістом яких можуть бути інформаційно-довідкові або обчислювальні роботи [16, с. 28].

2.5. Адресація в комп'ютерних мережах. Мережеві протоколи.

Кожному комп'ютеру користувача, який під'єднано до мережі, надаються *реквізити (адреса)*. **Адресація** – спосіб ідентифікації абонентів у мережі.

Для однозначної ідентифікації комп'ютерів у мережі використовується 32-бітова адреса, так звана «*IP-адреса*» (Internet Protocol), що дає змогу однозначно визначити даний комп'ютер і є унікальною в рамках однієї мережі. Запис IP-адреси містить чотири десяткові числа, розділені крапками. Кожне число відповідає 1 байту і може набувати значень від 0 до 255. Прикладом IP-адреси є 192.168.32.3. Такий крапковий десятковий запис набагато легше запам'ятовується і є більш читабельним, ніж єдиний рядок з восьми шістнадцяткових цифр.

Тому значна частина IP-адрес подається у крапковому десятковому записі.

Для визначення розміру мережі у протоколі IP використовують поняття субмережі і маски субмережі. Термін «субмережа» з'явився

внаслідок того, що маска дає змогу виділити підмножину IP-адрес, унаслідок чого у великій мережі можна виділити їхні частини – невеликі мережі (субмережі).

Використання маски субмережі дає змогу на комп'ютерах цієї субмережі розділяти адреси на локальні і нелокальні. Локальним комп'ютерам пакети посилаються безпосередньо. Програмне забезпечення протоколу використовує маску субмережі як бітову маску адреси комп'ютера. У масці субмережі вказується, які біти адреси комп'ютера є бітами адреси мережі, а які – бітами адреси комп'ютера. Будь-яка адреса, біти адреси мережі якої збігаються з бітами адреси мережі комп'ютера, з якого відправляються дані, є локальним щодо цього комп'ютера. В іншому разі адреса не є локальною.

Наприклад, якщо маска субмережі є 255.255.255.0, то перші 24 біти задають адресу мережі, а інші вісім – комп'ютер. Це означає, що 192.168.32 (або 11000000 10101000 00100000) – адреса мережі, а 3 (або 00000011) – номер комп'ютера.

Оскільки 1 байт складається з вісім двійкових розрядів, то за допомогою восьмирозрядного двійкового коду можна подати 256 різних значень від 00000000 до 11111111, тобто теоретично за допомогою 4 байтів можна подати більше 4 млрд унікальних IP-адрес (2⁵⁶⁴). На практиці ж через особливості адресації кількість можливих адрес складає біля 2 млрд [13, с. 65].

Мережеві протоколи

Для забезпечення необхідної сумісності на кожному із семи можливих рівнів архітектури комп'ютерної мережі діють спеціальні стандарти, які називають *протоколами*.

Протокол – це сукупність правил синхронного функціонування об'єктів однойменного рівня під час обміну даними між станцією-відправником і станцією-одержувачем; формати блоків даних, що передаються; контроль помилок, методи кодування даних тощо.

У протоколах описуються характер апаратних взаємозв'язків компонентів мережі (*апаратні протоколи* або *протоколи низького рівня*) і характер використання програм і даних (*програмні протоколи* або *протоколи верхнього рівня*). Фізично підтримку забезпечують через апаратні пристрої (*інтерфейси*) і програмні засоби (*програми підтримки протоколів*). Програми, за якими здійснюють підтримку протоколів, також називають *протоколами*.

Інтерфейс – це опис процедури взаємозв'язків об'єктів суміжних рівнів (наприклад, системи і середовища) для управління фізичним передаванням даних, а також формати даних, що передаються.

Наприклад, якщо два комп'ютери з'єднані між собою прямим з'єднанням, то на нижчому (фізичному) рівні протокол їхнього зв'язку визначають конкретні пристрої фізичного порту (паралельного чи послідовного) і механічні компоненти (роз'єм, кабель тощо). На вищому рівні зв'язки між комп'ютерами визначають через програмні засоби, за допомогою яких здійснюють управління передаванням даних через порти. Для стандартних портів такі програми знаходяться в базовій системі введення-виведення (BIOS). На найвищому рівні протокол взаємозв'язків забезпечується застосуванням операційної системи.

У комп'ютерах локальної мережі, як правило, використовують єдиний комплект протоколів для всіх комп'ютерів. У глобальній мережі можуть з'єднуватися як окремі комп'ютери, так і окремі локальні мережі, зокрема і з різними протоколами.

Розглянемо найпоширеніші протоколи верхнього рівня (програмні протоколи).

1. Apple Talk. Це сімейство мережевих протоколів Apple Computer. У ньому підтримуються архітектури Ethernet, Token Ring та LocalTalk. «Серцем» Apple Talk є AppleTalk Filing Protocol (AFP – файловий протокол AppleTalk). За допомогою AFP можна легко виконувати операції файлової системи, пропонується система безпеки. Apple підтримує різноманітні протоколи і служби рівня прикладних програм. Найширше використовують 3 програми – *AppleShare File Server*, *AppleShare Print Server* та *AppleShare PC*. У першій з них використовують служби AFP для доступу до віддалених файлів. За її допомогою користувачам надається можливість зберігання файлів на мережевому сервері та отримання до них доступу. За допомогою *AppleShare Print Server* надсилаються дані на принтери, що знаходяться в мережі AppleTalk. За допомогою *AppleShare PC* можна отримати доступ до файлів, що перебувають у файловій системі AppleTalk; надається також MS-DOS комп'ютерам можливість друкувати матеріали на принтерах, сумісних з AppleTalk.

2. NetWare. Це група мережевих протоколів і відповідних сервісних служб, створених фірмою Novell. У NetWare описуються специфікації для рівнів 3-7 моделі OSI (мережевий, транспортний,

сеансовий, рівень подання даних, прикладний), використовується з багатьма середовищами з різними типами доступу.

Протоколи NetWare для рівнів 3 і 4 називають Internetwork Packet exchange (IPX – міжмережевий пакетний обмін), Sequenced Packet eXchange (SPX – упорядкований обмін пакетами).

3. TCP/IP. TCP/IP – сукупність двох протоколів, що регулюють передавання даних між комп'ютерами в мережі Інтернет. Своєю назву протокол TCP/IP дістав від двох протоколів. Це *Transmission Control Protocol (TCP)* – протокол транспортного рівня, що визначає, як відбувається передавання даних, та *Internet Protocol (IP)* – адресний протокол мережевого рівня, що визначає, куди передаються дані. Комплекс протоколів TCP/IP включає чотири верхніх рівні моделі OSI. Протоколи базуються на IP-протоколах негарантованого доставляння пакетів без встановлення з'єднання.

Згідно з протоколом TCP дані, що відправляються, «нарізуються» на невеликі пакети, після чого кожний пакет маркується так, щоб у ньому були адреси одержувача та відправника, а також інші дані, необхідні для правильного збирання документа на комп'ютері одержувача.

У протоколі IP передбачається, що в кожного користувача повинна бути своя унікальна адреса (IP-адреса). Вона подається чотирма байтами, наприклад, 104.17.126.10. Структура IP-адреси така, що кожний комп'ютер, через який проходить пакет, може за цими чотирма числами визначити маршрут відправлення пакетів у даний момент з урахуванням умов зв'язку та пропускних характеристик лінії. Пакети можуть передаватися незалежно один від одного різними лініями зв'язку.

Незважаючи на те, що в мережі Інтернет використовуються й інші протоколи, її часто називають TCP/IP-мережею, оскільки ці два протоколи, безумовно, є найважливішими [13, с. 72].

Питання для самоконтролю

1. Що таке локальна мережа?
2. Що таке топологія та архітектура комп'ютерної мережі?
3. Які є топології комп'ютерних мереж?
4. Що таке протокол? Які існують протоколи?
5. Що таке мережеві ресурси?
6. Що таке мережевий диск?
7. Які бувають рівні доступу до ресурсів в мережі?
8. Які облікові записи вбудованих груп користувачів існують у мережевих ОС?
9. Скільки рівнів має еталонна модель OSI?
10. Наведіть схему еталонної моделі OSI.
11. Дайте означення еталонної моделі OSI.
12. Опишіть фізичний рівень моделі OSI.
13. Які функції виконує мережевий рівень моделі OSI.
14. Які недоліки кільцевої топології?
15. Які є переваги у топології «зірка»?
16. Що таке мережева технологія?
17. Що визначає мережева технологія?
18. Що таке Ethernet?
19. Яку топологію може мати мережа Ethernet?
20. Що таке Arcnet?
21. Дайте означення Token Ring.
22. Яку конфігурацію має мережа Token Ring?
23. Опишіть технологію FDDI.
24. Чим схожі і чим відмінні технології Token Ring і FDDI?
25. Що таке маркерний метод доступу?

Тема 3. Захист інформації в комп'ютерних мережах

3.1. Основи захисту інформації у комп'ютерних мережах

Мережеві технології у відриві від комплексного захисту інформації, яка передається мережевими каналами зв'язку, розглядати неможливо. Якщо раніше перенесення небезпечного програмного забезпечення (ПЗ) відбувалося в основному на таких носіях як дискети, оптичні диски тощо, то в умовах стрімкого розвитку Інтернету на перше місце виходить розповсюдження шкідливого програмного забезпечення за допомогою комп'ютерної мережі. У першу чергу, через Інтернет.

У той же час, на даному етапі розвитку інформаційних технологій, атаки організуються не одинаками, а створена ціла індустрія з багатомільйонним оборотом. Створюються «бот-мережі» (комп'ютери, на яких запущене автономне ПЗ – і які керуються віддалено), які в подальшому можуть продаватися. За допомогою заражених комп'ютерів можуть бути організовані DDoS атаки, розсилка спама (небажаної пошти) або ж перебір можливих паролів на віддаленому сервері. Зважаючи на вищесказане, до організації захисту інформації потрібно підходити комплексно, не обмежуючись єдиним інструментом.

Розглядаючи безпеку передачі даних у мережі організації, слід виділити наступні рішення:

- антивіруси;
- мережеві екрани;
- системи виявлення і попередження вторгнень;
- сканери вразливостей;
- рішення, що попереджують витік інформації;
- контентні фільтри;
- інструменти резервного копіювання і відновлення інформації.

У той же час багато рішень пропонують захист в комплексі, тобто, наприклад, мережеві екрани інтегрують в собі ще й захист від вірусів, а системи виявлення і попередження вторгнень є, як правило, міжмережевими екранами. Однак, напевно говорити, що дана система є міжмережовим екраном, до якої було інтегровано функції системи виявлення і попередження вторгнень, або навпаки, не можна без конкретного аналізу розвитку даної системи забезпечення захисту інформації. Розуміючи, що і захист інформації потребує

комплексного підходу, і системи, як правило, інтегрують в собі декілька рішень захисту інформації, вважаємо за доцільне розглянути кожну із вищенаведених систем окремо, як закінчену систему. Це дозволить сформулювати більш чітке уявлення щодо функцій, які виконує та чи інша система у комплексному захисті інформації організації.

3.2. Антивірусні рішення

Антивірусна програма (або просто антивірус) – це програма, яка призначена для знаходження небажаного ПЗ (сюди віднесемо комп'ютерні віруси, троянські програми, програми-шпигуни тощо), для лікування заражених файлів, а також попередження зараження інформаційної системи.

Часто все небажане ПЗ називають вірусами, що є не зовсім правильним. **Комп'ютерний вірус** – це різновид комп'ютерних програм, особливістю яких є здатність до самореплікації (розмноження). Також до небажаного ПЗ відносяться *троянські програми* – небажана програма, яка проникає до системи під виглядом корисної (кодеку, різного роду корисного ПЗ тощо), та *шпигунське ПЗ* – це програмне забезпечення, яке інсталюється в інформаційну систему для повного або часткового контролю над нею без відповідної згоди на це користувача даної системи. Даний вид ПЗ визначають як не санкціоновано встановлений.

З початку створення антивірусних програмних засобів пройшло досить багато часу. На початкових етапах еволюціонування антивірусів існував поділ антивірусних програм на ті, що виявляли віруси під час сканування, ті що знаходилися постійно в оперативній пам'яті комп'ютера, не даючи, таким чином провести зараження інформаційної системи. Також додатково потрібно було встановлювати ПЗ, яке покликане протидіяти шпигунським, троянським програмам тощо. Наприклад, Євгенієм Касперським у 1992 році була зроблена наступна класифікація антивірусних рішень:

– *сканери* (або «поліфаги») – виявляють наявність вірусу по базі сигнатур. Ефективність даних програм вимірюється актуальністю сигнатур, які зберігаються в базі даних;

– *ревізори* – запам'ятовують стан файлової системи, що дозволяє в подальшому робити аналіз наступних змін;

– *монітори* – виявляють потенційно небезпечні операції, видаючи запит на дозвіл або заборону такої операції;

– *вакцини* – змінюють кінцевий файл таким чином, щоб вірус думав, що файл вже заражений (у сучасних умовах з неймовірною кількістю вірусів даний підхід неактуальний).

Насьогодні всі сучасні антивіруси забезпечують постійний захист як від вірусів, так і від іншого небажаного ПЗ. Більше того, широко розповсюджені рішення, які дозволяють встановлювати більше одного антивірусного засобу (як правило, у складі міжмережевого екрану вже є вбудований антивірус, однак можна ще встановити інший, на вибір користувача). Самостійно встановлювати два і більше антивіруси в одну систему категорично забороняється, оскільки антивірусні рішення будуть сприйматися один одним, як комп'ютерний вірус.

Перші антивірусні програми з'явилися одразу після створення перших вірусів. Сьогодні розробкою даного виду ПЗ займаються такі виробники, як Kaspersky, Symantec, McAfee, Microsoft та багато інших. На ринку пропонується найрізноманітніше антивірусне ПЗ, від безкоштовних до дорогих корпоративних рішень.

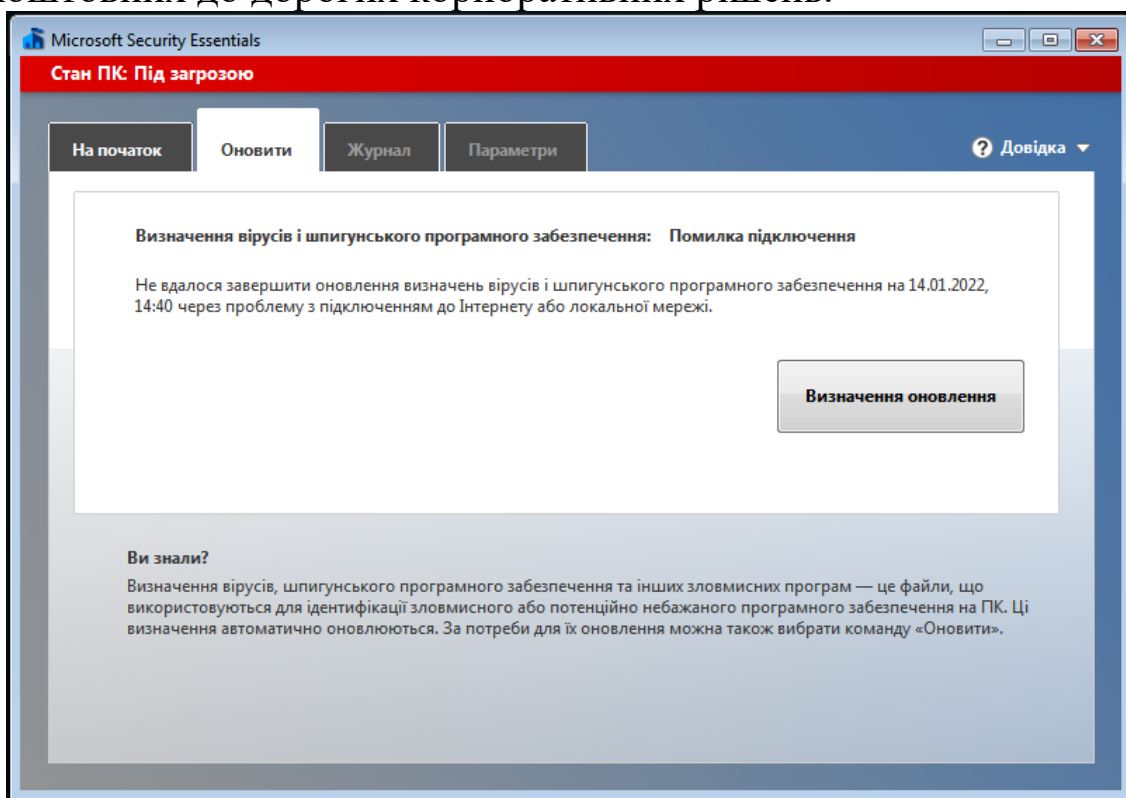


Рис. 3.2.1. Вікно безкоштовного антивірусу від Microsoft

Якщо говорити про класифікацію антивірусних програм, то можна її робити відповідно до визначального фактору. Наприклад,

якщо вибрати ціну, то можна виділити: безкоштовні антивіруси, умовно безкоштовні, платні тощо.

Для виявлення і знешкодження небажаного ПЗ антивірусні програми використовують різні методи:

– відповідність вірусу в описі бази наявних сигнатур. Антивірусна програма шукає відповідність опису вірусу в базі сигнатур, яку має у своєму розпорядженні. Недоліком даного підходу можна назвати те, що таким методом не можна віднайти небажане ПЗ, опис якого не було додано до бази сигнатур;

– віднаходження неадекватної поведінки програм. Відслідковується поведінка програм, які працюють у системі, й у випадку небезпечної дії програми (наприклад, зміни виконуючого файлу) антивірус повідомляє про це користувача. Перевагою такого методу є можливість віднаходити небажане ПЗ, яке ще не було додано до бази даних сигнатур. Недоліком є ймовірність невірної спрацювання при певних режимах роботи користувача (наприклад, встановлення поновлень для ПЗ);

– емуляція поведінки ПЗ. Перед передачею прав на виконання безпосередньо ПЗ антивірус намагається провести емуляцію початку виконання. Якщо програма буде вести себе по іншому, вона буде вважатися шкідливою для системи. Даний метод також має недоліки у вигляді невірних спрацювань;

– «білий список». Дозволяється використовувати лише те ПЗ, яке безпосередньо дозволено в системі. Таким чином, у системі не буде виконуватися навіть ПЗ, яке не несе у собі загрози у вигляді вірусів чи іншого ПЗ. Даний підхід, як правило, використовують при корпоративному управлінні антивірусним ПЗ. Хоча на комп'ютері, особливо на комп'ютері, який взаємодіє з іншими за допомогою мережі, повинно бути встановлене антивірусне ПЗ, слід чітко усвідомлювати, що жоден антивірус не зможе надати 100% захисту від небажаного ПЗ.

3.3. Мережеві екрани

Найбільш важливим засобом захисту мереж є мережеві екрани (або міжмережеві екрани, файрволи чи брандмауери) та проксі-сервера. Основна функція мережевого екрану – екранування мережевого трафіку з метою недопуску несанкціонованого доступу між комп'ютерними мережами. Проксі-сервери використовуються

для обробки запитів користувачів із внутрішньої мережі та транслявання їх до зовнішньої мережі, і навпаки. У таких запитах, як правило, розрізняють мережі, що користуються довірою, та такі, що не користуються (наприклад, мережа Інтернет).

Мережевий екран – це програмне або апаратне рішення, яке здійснює контроль і фільтрацію мережевих пакетів, що проходять через нього на різних рівнях моделі OSI у відповідності до заданих правил.

Мережеві екрани надзвичайно популярні завдяки тим перевагам, які вони у собі несуть:

- вони є засобом реалізації корпоративних політик безпеки;
- мережеві екрани надають можливість обмежувати доступ до певних служб (наприклад, до служби telnet, яка непотрібна звичайним користувачам);
- мережевий екран можна використовувати як засіб аудиту. Збирати інформацію можна не лише про трафік, який пройшов, але і про той, який був заблокований, таким чином попереджуючи можливі майбутні атаки на систему.

Утім, мережеві екрани, будучи лише ланкою у загальній системі захисту інформаційної системи, мають недоліки:

- мережеві екрани не можуть блокувати небажану інформацію, яка потрапляє до інформаційної системи через авторизовані канали. Таким чином, якщо шпигунське ПЗ надійшло до системи через дозволений канал, воно не зможе бути зупинене мережевим екраном;
- ефективність мережевих екранів визначається ефективністю правил, які покладено в основу функціонування мережевого екрану. Тому потрібно підійти з усією відповідальністю до формулювання правил фільтрації трафіку;
- мережеві екрани не зможуть попередити атаки користувачів, які користуються авторизованими каналами інформації;
- мережеві екрани не зможуть протидіяти атакам, якщо є можливість пустити інформаційний потік в обхід екрану.

Розподіл мережевих екранів можна провести відповідно до класифікаційної ознаки, наприклад, відповідно до місця знаходження мережевого екрану в мережі, у залежності від розташування механізму контролю в моделі OSI чи в залежності від можливості слідкування за активними з'єднаннями.

Однією із найбільш поширених класифікацій є класифікація мережевих екранів відповідно до охопту територій, що контролюються:

1. Прикладний шлюз або традиційний мережевий екран. Це програмне або апаратне рішення, яке контролює як вхідні, так і вихідні потоки даних між підключеними мережами. Фактично при встановленні з'єднання користувача із мережею зовні проходить два з'єднання: одне із мережевим екраном, а інше – мережевого екрану з мережею зовні. Даний вид мережевих екранів ще називають проксі-серверами або проксі-шлюзами. У випадку здійснення атаки на систему користувача фактично атака буде здійснюватися на проксі-сервер. Утім наряду із перевагами в захисті інформації, проксі-сервера мають і недоліки, а саме: зменшення швидкодії, оскільки кожне з'єднання представляє собою два з'єднання, та зменшення прозорості, коли деяке ПЗ не може повноцінно функціонувати в мережі;

2. Пакутий фільтр або персональний мережевий екран. Даний тип мережевих екранів встановлюється на кінцеві системи, і тому захищають вони лише систему, на якій встановлені, а не цілі мережі. Мережеві екрани цього типу здійснюють фільтрацію трафіку, базуючись на наступній інформації:

- IP-адреса джерела;
- IP-адреса одержувача;
- протокол, який використовується;
- вихідний порт;
- порт призначення;
- тип повідомлення.

Ефективність мережевих екранів вимірюється правильністю побудови правил, що в них покладені, тому при побудові правил потрібно дотримуватися наступних інструкцій:

– побудова правил повинна здійснюватися від найбільш конкретних до загальних. Оскільки перевірка йде за правилами до першої відповідності, то неправильна побудова правил може створити умови, коли більш загальний набір правил перекриє більш конкретні правила;

– розміщення правил, які використовуються найчастіше, повинно бути у верхній частині списку. Оскільки перевірка йде до першого збігу, то розміщення активних правил у кінці списку може суттєво зменшити швидкість роботи інформаційної системи.

3.4. Сканери вразливостей

Сканери вразливостей – це програмні або апаратні рішення, які призначені для діагностики і моніторингу мережевих комп'ютерних систем.

Вони дозволяють сканувати мережі, інформаційні системи та прикладні програми на предмет виявлення потенційних загроз у системі організації безпеки, оцінювати і знешкоджувати вразливості.

Типи сканерів вразливостей:

- сканери портів;
- сканери, які досліджують топологію комп'ютерної мережі;
- сканери, які досліджують вразливості мережевих сервісів;
- CGI - сканери, які дозволяють виявити вразливі скрипти.

Роботу сканера вразливостей можна розподілити на наступні кроки:

1. Виявлення активних IP-адрес, відкритих портів, активних операційних систем та прикладних програм;
2. Створення звіту з безпеки;
3. Спроба виявлення рівня можливого втручання до операційної системи чи прикладної програми;
4. Можливе використання вразливості для створення збою в системі (лише для шкідливих сканерів).

Говорячи про сканери вразливостей, слід зауважити, що вони можуть слугувати як інструментом виявлення загроз, так і інструментом злому інформаційної системи, тому четвертий крок використовують сканери, які створені для здійснення атаки на інформаційну систему.

До сканерів вразливостей відносяться наступні рішення: Microsoft Baseline Security Analyzer (MBSA), Security Administrator's Integrated Network Tool (SAINT), сканер вразливостей (веб-сервіс) (QualysGuard), сканер для дослідження мережевих вразливостей (X-scan), оцінка вразливостей на рівні прикладних програм (ISS Internet Scanner) та інші.

3.5. Системи виявлення і попередження вторгнень

Система виявлення вторгнень (Intrusion Detection System) – це програмне або апаратне рішення, яке націлене на виявлення фактів неавторизованого доступу в інформаційну систему чи мережу та

несанкціонованого керування ними, як правило, через мережу Інтернет.

Архітектура систем виявлення вторгнень складається з:

- сенсорної підсистеми, яка направлена на збір даних в області інформаційної безпеки;
- підсистеми аналізу, яка виявляє підозрілі дії на основі даних, що були отримані із сенсорів;
- сховища, необхідного для зберігання первинної інформації, а також результатів аналізу;
- консолі керування даною системою.

Основними видами систем виявлення вторгнень є:

– мережева, яка відслідковує вторгнення, перевіряючи мережевий трафік, а також слідкує за декількома хостами. Мережева система виявлення вторгнень отримує доступ до мережевого трафіку, підключаючись до концентратора чи комутатора;

– система, основана на протоколі. Система, яка аналізує комунікаційні протоколи з пов'язаними системами чи користувачами. Для вебсервера дана система виявлення вторгнень, як правило, слідкує за HTTP та HTTPS протоколами. Дана система повинна бути налаштованою так, щоб могла проглядати пакети ще до їх відправки в мережу;

– система, основана на прикладних протоколах. Система, яка слідкує і аналізує дані, які передаються з використанням специфічних протоколів прикладних програм;

– вузлова. Система, яка розташована на хості, відслідковує вторгнення, використовуючи аналіз системних викликів, модифікації файлів та інших джерел;

– гібридна. Система, яка включає більше одного підходу у розробці систем виявлення вторгнень.

Крім того, системи виявлення вторгнень можна розділити на пасивні і активні. У пасивній системі отримана в результаті аналізу інформація записується до спеціального архіву, а також відсилаються повідомлення адміністратору системи.

Якщо система виявлення вторгнень є активною, то вона має назву *система попередження вторгнень* (Intrusion Prevention system – IPS). Даний тип систем не лише веде збір інформації щодо дій в системі, але і здійснює активні дії, направлені на попередження можливих вторгнень до інформаційної системи. Дані дії можуть виконуватися як в автоматичному режимі, так і в ручному.

Системи виявлення і попередження вторгнень досить близькі за своєю сутністю з мережевими екранами. Більше того, деякі мережеві екрани виконують роль систем виявлення і попередження вторгнень. Основна ж відмінність між даними системами виявляється в тому, що мережевий екран відсікає неавторизований трафік, пропускаючи авторизований, та не займається слідкуванням за вторгненнями, які можуть мати місце всередині самої мережі. Система ж виявлення вторгнень навпаки пропускає трафік та аналізує його, подаючи сигнали у випадку підозрілої активності. Щодо наявних систем виявлення вторгнень можна назвати наступні: Snort NIDS, Endian Firewall, Tripwire, Untangle та інші.

3.6. Рішення попередження витоку інформації

Попередження витоку інформації (Data Leak Prevention – DLP) – це технології, які попереджують витік конфіденційної інформації з інформаційної системи на зовні, а також технічні засоби для попередження такого витоку.

Системи даного типу будуються на аналізі потоків даних, які пересікають периметр інформаційної системи. Якщо в інформаційному потоці віднаходиться конфіденційна інформація, то спрацьовує система захисту і потік блокується.

Розпізнавання інформації, витік якої є небажаним, може відбуватися двома основними методами: аналізом формальних ознак (наприклад, грифу документу, спеціально створених міток, порівнянням хеш-функції) та аналізом самого контенту. У першому випадку виключені неправильні спрацювання, однак, потрібен час на проведення індексації документів, введення до них міток тощо. У другому випадку можливі невірні спрацювання, і якість системи буде залежати від правильності налаштованих фільтрів. Як правило, дієва система попередження витоку інформації повинна включати обидва методи перевірки контенту.

Системи попередження витоку інформації включають до свого складу компоненти мережевого рівня і модулі хоста. Компоненти мережевого рівня контролюють трафік, який перетинає кордони інформаційної системи через мережу. Компоненти мережевого рівня повинні бути встановлені на проксі-серверах, серверах електронної пошти, а також окремих серверах. Компоненти рівня хоста встановлюються на комп'ютери співробітників і контролюють шляхи

запису інформації на носії, відслідковують встановлення неавторизованого ПЗ тощо.

Основною задачею систем попередження витоку інформації є недопущення витоку інформації з інформаційної системи. Крім основної задачі дана система повинна ще виконувати ряд другорядних задач:

- архівування повідомлень, які пересилаються. Архіви можуть знадобитися для подальшого розслідування можливих справ;
- попередження передачі зовні не лише конфіденційної інформації, але й іншої, яка є небажаною для компанії (наприклад, значних об'ємів, що може знизити продуктивність каналів чи збільшити оплату за використання Інтернету);
- попередження витоку інформації не лише зовні, а й із зовнішнього середовища до інформаційної системи;
- попередження використання співробітниками службових інформаційних ресурсів в особистих цілях;
- оптимізація загрузки каналів передачі інформації;
- контроль присутності співробітників на робочому місці.

3.7. Шифрування інформації

Шифрування – це спосіб перетворення відкритої інформації в закриту, і навпаки. Шифрування поділяється на процес зашифровування і розшифрування. При шифруванні інформації формується ключ – секретна інформація, яка використовується криптографічним алгоритмом шифрування. Використання криптографії при передачі важливої інформації практикувалося досить давно. Існують докази, що древні римляни використовували шифрування при передачі інформації кур'єрами на полі бою. Шифрування залишається одним з основних механізмів забезпечення захисту інформації завдяки тому, що за допомогою шифрування можна одразу забезпечити вирішення декількох задач в області захисту інформації. До цих задач належать:

- збереження конфіденційності. Інформація, яка була зашифрована залишається недоступною для сторонніх осіб (за умов належного зберігання ключа і використання крипостійкого алгоритму);

– збереження цілісності. Неможливо провести зміну даних несанкціоновано, оскільки для її зміни зашифровану інформацію потрібно спочатку розшифрувати;

– неможливість відмови від авторства. Завдяки наявності цифрового підпису й асиметричних методів шифрування інформації дозволяє однозначно ідентифікувати автора повідомлення.

Шифрування інформації використовується в сучасному світі постійно: при авторизації користувачів на Інтернет порталах, при пересиланні електронної пошти і використанні цифрового підпису. Бази даних користувачів і їх паролі обов’язково повинні зберігатися в зашифрованому вигляді, та навіть звичайний текст документу MS Word можна зашифрувати вбудованими засобами і встановити на нього пароль.

Існують два основні алгоритми шифрування інформації: симетричний і асиметричний, – у кожного є свої переваги і недоліки, а також область, де їх можна застосовувати. При використанні симетричного методу шифрування ключ для шифрування і для розшифровки інформації співпадає, тому він повинен зберігатися в секреті (рис. 3.7.1).

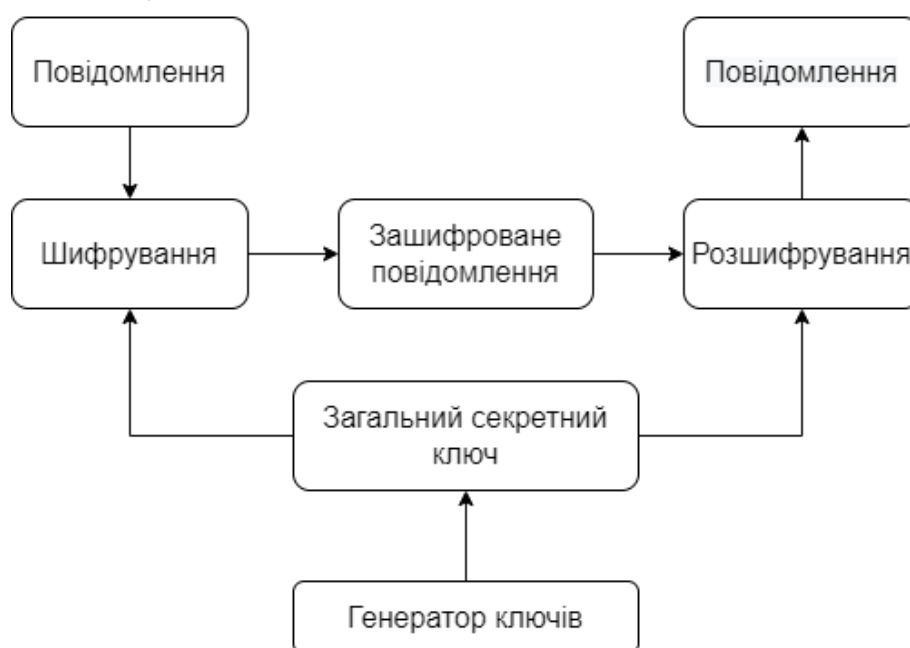


Рис.3.7.1. Використання симетричного методу шифрування

До переваг *симетричного методу* шифрування можна віднести простоту реалізації та швидкість роботи симетричних алгоритмів. Однак, існують недоліки і, у першу чергу, проблема розповсюдження ключів, оскільки для розшифровки повідомлення потрібно мати ключ, аналогічний тому, яким дане повідомлення було зашифроване.

Окрім того, за допомогою симетричних методів шифрування не можливо вирішити задачу щодо авторства повідомлення, оскільки неможливо однозначно виявити автора зашифрованої інформації.

Для вирішення задач, згаданих вище, використовуються *асиметричні методи* шифрування інформації. В асиметричних методах використовується пара ключів: один для шифрування, інший для розшифровки повідомлення. Ключ, за допомогою якого шифрується повідомлення, не є секретним, оскільки за допомогою нього не можна розшифрувати повідомлення, тому він називається відкритим. Разом із відкритим ключем формується секретний, який повинен зберігатися в секреті, оскільки він використовується для розшифровки повідомлень. Одним із відомих методів асиметричного шифрування є RSA, який використовує операції з великими простими числами та їх множенням. Загальний принцип роботи асиметричного методу шифрування наведений на рис. 3.7.2.

Асиметричний метод шифрування вирішує недоліки, які існують при використанні симетричних методів, однак, у асиметричних методів існує один значний недолік, а саме низька швидкодія. Для усунення цього одного недоліку асиметричних методів використовують досить простий і у той же час зручний спосіб: основне повідомлення шифрується за допомогою швидкого симетричного методу, а його ключ за допомогою асиметричного алгоритму, таким чином можна шифрувати значні обсяги інформації і передавати її. У той же час гарантується достовірність авторства [5].

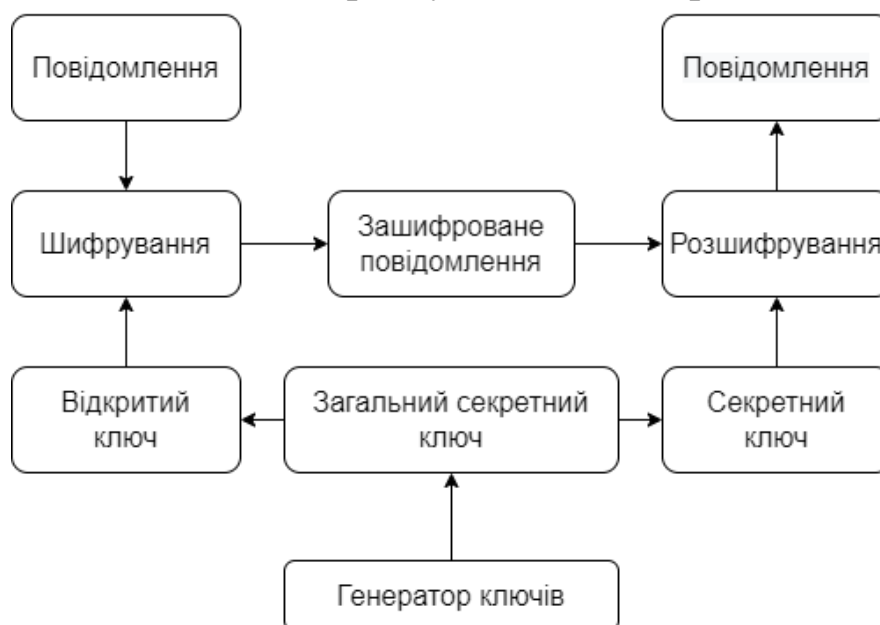


Рис.3.7.2. Використання асиметричного алгоритму шифрування

Питання для самоконтролю

1. Які існують види небажаного ПЗ і чим вони відрізняються?
2. Які основні методи виявлення небажаного ПЗ притаманні антивірусним програмам?
3. Які переваги та недоліки в області захисту інформації притаманні мережевим екранам?
4. Які основні принципи роботи сканерів вразливостей?
5. Призначення систем виявлення і попередження вторгнень.
6. Призначення рішень попередження витоку інформації.
7. Які основні алгоритми шифрування інформації Ви знаєте, у чому їх переваги і недоліки?
8. Як називається спосіб перетворення відкритої інформації в закриту, і навпаки?
9. Який метод шифрування недопустимий при передачі зашифрованого повідомлення електронною поштою?
10. Навіщо потрібен захист інформації?
11. Як може відбуватися втрата інформації?
12. Яка основна функція мережевого екрану?
13. Як класифікують мережеві екрани відповідно до охопту територій, що контролюються?
14. Які типи сканерів вразливостей Вам відомі?
15. Наведіть приклади сканерів вразливостей.
16. Архітектура систем виявлення і попередження вторгнень.
17. Які види систем виявлення і попередження вторгнень Ви знаєте?
18. Охарактеризуйте вузлову систему виявлення вторгнень.
19. Які функції виконують системи попередження вторгнень?
20. Яка основна задачаю систем попередження витоку інформації?
21. Які методи шифрування Ви знаєте?
22. Назвіть основні переваги та недоліки використання асиметричного методу шифрування інформації.

Тема 4. Проху-сервер. Організація бездротового зв'язку. Стан і перспективи використання комп'ютерних мереж

4.1. Проху-сервер

Проху-сервер (від англ. Проху – «представник, уповноважений») – сервер (комп'ютерна система або програма) в комп'ютерних мережах, за допомогою якого клієнти можуть виконувати непрямі (через посередництво проху-сервера) запити до мережеских сервісів. Спочатку клієнт з'єднується з проху-сервером і здійснює запит до ресурсу (наприклад, e-mail), який розташований на іншому сервері. Потім проху-сервер або підключається до вказаного сервера і отримує ресурс у нього, або повертає ресурс з власного кешу (у випадках, якщо проху має свій кеш). У деяких випадках запит клієнта або відповідь сервера може бути змінена проху-сервером з певною метою. Також за допомогою проху-сервера можна захищати клієнтський комп'ютер від деяких мережеских атак і допомагати зберігати анонімність клієнта.

Найчастіше проху-сервери застосовуються для:

– Забезпечення доступу з комп'ютерів локальної мережі в Інтернет.

– Кешування даних: якщо часто відбуваються звернення до одних і тих же зовнішніх ресурсів, то можна зберігати їх копію на проху-сервері та видавати за запитом, знижуючи тим самим навантаження на канал у зовнішню мережу і прискорюючи отримання клієнтом потрібних даних.

– Стиснення даних: за допомогою проху-сервера завантажуються дані з Інтернет і передаються кінцевому користувачеві в стислому вигляді. Такі проху-сервери використовуються в основному з метою економії зовнішнього трафіка.

– Захист локальної мережі від зовнішнього доступу: наприклад, можна налаштувати проху-сервер так, що локальні комп'ютери будуть звертатися до зовнішніх ресурсів тільки через нього, а зовнішні комп'ютери не зможуть звертатися до локальних взагалі.

– Обмеження доступу з локальної мережі до зовнішньої: наприклад, можна заборонити доступ до певних вебсайтів, обмежити використання Інтернет якимось локальним користувачем, встановлювати квоти на трафік або смугу пропускання, фільтрувати рекламу і віруси.

– Анонімізація доступу до різних ресурсів. За допомогою проху-сервера можна приховувати відомості про джерело запиту або користувача. У такому разі цільовий сервер «бачить» лише відомості про проху-сервер, наприклад, IP-адресу, але не можна визначити дійсне джерело запиту. Існують також спотворюючі проху-сервери, за допомогою яких передаються цільовим серверам неправдиві відомості про справжнього користувача.

Проху-сервер, до якого може отримати доступ будь-який користувач мережі Інтернет, називається *відкритим*.

Види проху-серверів

– **Прозорий проксі** – схема зв'язку, при якій трафік, або його частина, перенаправляється на проху-сервер неявно (засобами маршрутизатора). При цьому клієнт може використовувати всі переваги проху-сервера без додаткових налаштувань, але з іншого боку, не має вибору.

– **Зворотний проксі** – проху-сервер, який на відміну від прямого, ретранслює запити клієнтів із зовнішньої мережі на один або кілька серверів, логічно розташованих у внутрішній мережі. Часто використовується для балансування мережного навантаження між кількома вебсерверами і підвищення їх безпеки, відіграючи при цьому роль міжмережевого екрану на прикладному рівні.

Технічна реалізація: Клієнтський комп'ютер має налаштування (конкретної програми або операційної системи), відповідно до якої всі мережеві з'єднання за деяким протоколом здійснювати не на IP-адресу сервера (ресурсу), що визначається з DNS-імені ресурсу, або безпосередньо заданий, а на IP-адресу (і інший порт) проху-сервера.

За необхідності звернення до будь-якого ресурсу за цим протоколом, на клієнтському комп'ютері відкривається мережеве з'єднання з проху-сервером (на потрібному порті) і здійснюється звичайний запит, так якщо б він звертався безпосередньо до ресурсу.

Розпізнавши дані запиту, перевіривши його коректність і дозвіл для клієнтського комп'ютера, за допомогою проху-сервера, не розриваючи з'єднання, відкривається нове мережеве з'єднання безпосередньо з ресурсом і робить той же самий запит. Отримавши дані (або повідомлення про помилку), за допомогою проху-сервера передаються клієнтському комп'ютеру.

Під час становлення мережі Інтернет проху-сервери були найпопулярнішим способом виходу в Інтернет з локальних мереж. Цьому сприяли такі обставини:

– Основний протокол, що використовувався – http, який легко проксіюється;

– Підтримка проксі більшістю браузерів і/або операційних систем;

– Контроль доступу та облік трафіку за користувачами;

– Проху-сервер – це звичайна програма (а не системна), яка може виконуватися з мінімальними правами на будь-якій ОС з підтримкою мережі (стека TCP/IP);

– Відсутність доступу в Інтернет за іншими протоколами часто було більш перевагою, ніж недоліком.

У зв'язку із зростанням ролі інших мережевих протоколів, переходом до тарифікації послуг Інтернет за швидкістю доступу, а також появою дешевих апаратних маршрутизаторів з функцією NAT, використання звичайних проху-серверів для виходу користувачів до Інтернет застосовується вкрай рідко.

Однак великого поширення набули прозорі проху-сервери (протоколу http, іноді деяких інших), в тому числі і ті, що входять до складу багатьох апаратних маршрутизаторів для доступу до Інтернет, з метою збору статистики та контролю доступу до сайтів. Інші порти (протоколи) при цьому проходять через NAT.

Основними вимогами до Проху-сервера є:

– надання доступу до Інтернету користувачам з локальної мережі;

– простота встановлення й налаштування;

– можливість керування правами кожного користувача;

– можливість використання заборон на завантаження файлів більших від певного розміру;

– прискорення доступу (кешування сторінок);

– можливість перегляду запитів;

– налаштування списку заборонених веб-адрес.

Найбільш поширені проксі-сервери

– Zproxу (BSD, багатоплатформений);

– CoolProxy (proprietary, Windows);

– HandyCache (shareware, Windows) безкоштовний для домашнього використання;

– Kerio Control (proprietary, Windows, Linux);

– Microsoft Forefront Threat Management Gateway (proprietary, Windows);

- nginx (вебсервер, що має режим роботи в якості reverse proxy і часто для цього використовується);
- Squid (GPL, багатоплатформений);
- Traffic Inspector (proprietary, Windows);
- UserGate (proprietary, Windows);
- Інтернет Контроль Сервер (shareware, FreeBSD);
- TOR (shareware, багатоплатформений) [7].

4.2. Організація бездротового зв'язку

Відповідно до концепції «бездротового майбутнього», незабаром саме бездротові комунікації «захоплять» значну частину ринку комунікацій, адже бездротовий зв'язок дуже зручний і крім того надзвичайно мобільний.

В бездротовому зв'язку використовується широкий діапазон електромагнітного спектру, від радіохвиль низької частоти в кілька кілогерц до видимого світла, частота якого складає приблизно $8 \cdot 10^{14}$ Гц.

Ідея цифрового бездротового зв'язку не нова. Вже в 1901 році італійський фізик Г. Марконі (Guglielmo Marconi) продемонстрував телеграфний зв'язок між кораблем і узбережжям за допомогою азбуки Морзе, що складається з крапок і тире. Сьогоднішні цифрові радіосистеми мають вищу продуктивність, проте в їх основі лежить та сама ідея.

У першому наближенні бездротові мережі можна поділити на наступні три категорії:

- взаємопов'язані системи;
- бездротові локальні мережі;
- бездротові глобальні мережі.

Під взаємопов'язаними системами розуміється перш за все зв'язування між собою компонентів комп'ютера з використанням радіохвиль малого радіусу дії. У простому випадку взаємозв'язки усередині системи здійснюються за принципом "головний – підлеглий". Системний блок найчастіше використовується як головний пристрій, а всі інші як підлеглі.

Наступним кроком в розвитку цього напрямку стали бездротові локальні обчислювальні мережі. В них кожний комп'ютер обладнано радіомодемом і антеною, за їх допомогою можна обмінюватися даними з іншими комп'ютерами.

Третій тип бездротових мереж використовується в регіональних та глобальних мережах. Прикладом може служити система стільникового зв'язку, що є, насправді, низькопродуктивною цифровою бездротовою мережею.

Зараз розвиваються не лише низько-, але і високопродуктивні глобальні бездротові мережі. Початкова установка така: необхідно організувати доступ до Інтернет з нормальною швидкістю, щоб при цьому не було задіяно телефону. Таку послугу іноді називають *локальною багатовузловою системою розподілу* (стандарт IEEE 802.16).

Майже у всіх бездротових мережах в якомусь місці є шлюз, через який забезпечується зв'язок зі звичайними комп'ютерними мережами, інакше просто неможливо було б організувати, наприклад, доступ до Інтернету. Такі комбінації можуть використовуватися в різних видах і ситуаціях.

Досліджувати і розробляти технології бездротового зв'язку вчені почали по всьому світу ще у 50-ті роки минулого століття. При цьому пробували досліджувати все: і радіо, і інфрачервоний спектр, і навіть з'єднання за допомогою лазера. Саме електромагнітні поля та хвилі найбільш цікаві з точки зору передавання комп'ютерних даних.

Якщо кабельне середовище строго визначає напрям поширення сигналу в просторі, то бездротове середовище є ненапрямленим. Оскільки за ненапрямленого поширення електромагнітні хвилі заповнюють весь простір (в межах певного радіуса, що визначається згасанням потужності сигналу), то цей простір може служити розділювальним середовищем.

Пристрої, спеціально призначені для випромінювання і (або) приймання електромагнітних хвиль, – це *антени*. Передавальні і приймальні антени мають властивість взаємності (оборотності), відповідно до якої одна і та сама антена може використовуватись як для випромінювання, так і приймання електромагнітних хвиль. Якщо в електричний ланцюг включити антену відповідного розміру, то електромагнітні хвилі можна з успіхом приймати за допомогою приймача на деякій відстані. На цьому принципі засновані всі бездротові системи зв'язку.

Характеристики бездротової лінії зв'язку – відстань між вузлами, територія охоплення, швидкість передавання сигналів і т.п. – багато в чому залежать від частоти використовуваного електромагнітного спектру.

Чим вище несуча частота, тим вище можлива швидкість передавання сигналів. Чим вище частота, тим гірше проникає сигнал через перешкоди. Чим вище частота, тим швидше спадає енергія сигналу з віддаленням від джерела

Слід також нагадати, що радіосигнали будь-яких частот можуть перешкоджатися електричним обладнанням.

При роботі на низьких частотах радіохвилі добре проходять крізь перешкоди, проте потужність сигналу в повітрі різко падає у міру віддалення від передавача. Сигнали частот від 2 до 30 МГц відбиваються іоносферою землі, тому вони можуть поширюватися на значні відстані – кілька тисяч кілометрів (за достатньої потужності передавача). Радіохвилі діапазонів HF (High Frequency – висока частота) і VHF (very high frequency – дуже висока частота) поглинаються землею. Проте ті з них, які доходять до іоносфери, що є шаром заряджених частинок, розташованим на висоті від 100 до 500 км, відбиваються нею і посилюються назад до поверхні Землі. За певних атмосферних умов сигнал може відбитися кілька разів. На високих частотах (вище 30 МГц) радіохвилі поширюються майже прямо (тобто є сигналами прямої видимості) і відбиватися від перешкод. Кілька діапазонів від 300 МГц до 3000 ГГц мають нестандартну назву мікрохвильових діапазонів. Мікрохвильові системи являють собою найширший клас систем. Це радіорелейні лінії зв'язку, супутникові канали, бездротові локальні мережі і системи фіксованого бездротового доступу. При частоті сигналу понад 4 ГГц хвилі, що мають довжину лише кілька сантиметрів, починають поглинатися водою, а це означає, що не лише дощ, але й туман можуть стати причиною різкого погіршення якості передавання сигналів за допомогою мікрохвильових систем. Зважаючи на це, деякі оператори зв'язку тримають близько 10 % своїх каналів вільними і тимчасово перемикаються на них у разі виникнення перешкод на якій-небудь частоті.

Вище мікрохвильових діапазонів розташовується інфрачервоний діапазон. Оскільки інфрачервоне випромінювання не може проникати через стіни, то системи інфрачервоних хвиль використовуються для утворення невеликих сегментів локальних мереж в межах одного приміщення.

Останніми роками видиме світло теж почало застосовуватися для передавання сигналів (за допомогою лазерів). Системи видимого світла використовуються як високошвидкісна альтернатива

мікрохвильовим двоточковим каналам для організації доступу на невеликих відстанях [16].

Бездротові технології останньої милі:

- WiFi;
- WiMax;
- RadioEthernet;
- MMDS;
- LMDS
- мобільний GPRS – Internet;
- мобільний CDMA – Internet.

Розглянемо докладніше бездротові технології останньої милі:

1. **WiFi** (Wireless Fidelity – точна бездротова передача даних) – технологія широкопasmового доступу до мережі Інтернет. Швидкість передачі інформації для кінцевого абонента може досягати 54 Мбіт/с. Радіус їх дії не перевищує 50 – 70 метрів. Бездротові точки доступу застосовуються в межах квартири або в громадських місцях великих міст. Маючи ноутбук або кишеньковий персональний комп'ютер з контролером Wi-Fi, відвідувачі кафе чи ресторану (в зоні покриття мережі Wi-Fi) можуть швидко з'єднатися з Інтернетом.

2. **WiMAX** (Worldwide Interoperability for Microwave Access), аналогічно WiFi – технологія широкопasmового доступу до Інтернет. WiMAX, на відміну від традиційних технологій радіодоступу, працює і на відбитому сигналі, поза прямою видимістю базової станції. Експерти вважають, що мобільні мережі WiMAX відкривають набагато цікавіші перспективи для користувачів, ніж фіксований WiMAX, призначений для корпоративних замовників. Інформацію можна передавати на відстані до 50 км зі швидкістю до 70 Мбіт/с. В даний час WiMAX частково задовольняє умовам мереж 4G, заснованих на пакетних протоколах передачі даних. До родини 4G відносять технології, які дозволяють передавати дані в стільникових мережах зі швидкістю вище 100 Мбіт/с. і підвищеною якістю голосового зв'язку. Для передачі голосу в 4G передбачена технологія VoIP.

3. **RadioEthernet** – технологія широкопasmового доступу до Інтернет, забезпечує швидкість передачі даних від 1 до 11 Мбіт / с, яка ділиться між всіма активними користувачами. Для роботи RadioEthernet-каналу необхідна пряма видимість між антенами абонентських точок. Радіус дії до 30 км.

4. **MMDS** (Multichannel Multipoint Distribution System). Ці системи здатні обслуговувати територію в радіусі 50-60 км, при цьому пряма видимість передавача оператора є не обов'язковою. Середня гарантована швидкість передачі даних складає 500 Кбіт/с – 1 Мбіт/с, але можна забезпечити до 56 Мбіт/с на один канал.

5. **LMDS** (Local Multipoint Distribution System) – це стандарт стільникових мереж бездротової передачі інформації для фіксованих абонентів. Система будується за стільниковим принципом, одна базова станція дозволяє охопити район радіусом в декілька кілометрів (до 10 км) і підключити декілька тисяч абонентів. Самі БС об'єднуються один з одним високошвидкісними наземними каналами зв'язку або радіоканалами (RadioEthernet). Швидкість передачі даних до 45 Мбіт/с.

6. **Мобільний GPRS – Internet**. Для користування послугою «Мобільний Інтернет» за допомогою технології GPRS необхідно мати телефон з вбудованим GPRS – модемом і комп'ютером. Технологія GPRS забезпечує швидкість передачі даних до 114 Кбіт/с. При використанні технології GPRS тарифікується не на часі з'єднання з Інтернетом, а сумарний обсяг переданої та отриманої інформації. Ви зможете переглядати HTML-сторінки, перекачувати файли, працювати з електронною поштою і будь-якими іншими ресурсами Інтернет. Технологія GPRS – це удосконалення базової мережі GSM або протокол пакетної комутації для мереж стандарту GSM. EDGE є продовженням розвитку мереж GSM / GPRS. Технологія EDGE (покращений GPRS або EGPRS) забезпечує більш високу швидкість передачі даних в порівнянні з GPRS (швидкість до 200 Кбіт/с).

7. **Мобільний CDMA – Internet**. Мережа стандарту CDMA – це стаціонарний і мобільний зв'язок, а також швидкісний мобільний Інтернет. Для користування послугою «Мобільний Інтернет» за допомогою технології CDMA необхідно мати телефон із вбудованим CDMA – модемом або CDMA модем і комп'ютер. Технологія CDMA забезпечує швидкість передачі даних до 153 Кбіт/с або до 2400 Кбіт/с – за технологією EV-DO Revision 0. У даний час технологія CDMA надає послуги мобільного зв'язку третього покоління. Технології мобільного зв'язку 3G (third generation – третє покоління) – набір послуг, який забезпечує як високошвидкісний мобільний доступ до мережі Інтернет, так і організовує відеотелефонний зв'язок і мобільне телебачення. Мобільний зв'язок третього покоління будується на основі пакетної передачі даних. Мережі третього покоління 3G

працюють в діапазоні близько 2 ГГц, передаючи дані зі швидкістю до 14 Мбіт/с. Мережі третього покоління 3G реалізовані на різних технологіях, заснованих на наступних стандартах: W-CDMA (Wideband Code Division Multiple Access) і його європейському варіанті – UMTS (Universal Mobile Telecommunication System), який є приймачем GSM / GPRS / EDGE; CDMA2000 1X, що є модифікацією стандарту CDMA; китайським варіантом – TD-CDMA / TD-SCDMA.

Слід зазначити, то в даний час для «останніх метрів» доступу в Інтернет застосовуються технології Home PNA (HPNA) і HomePlug. Доступ в Інтернет по виділених лініях Home PNA або HPNA (телефонних лініях) і доступ через побутову електричну мережу напругою 220 вольт (HomePlug, Plug – це штепсель).

Зазвичай доступ до Інтернету по виділеним лініям Home PNA і HomePlug комбінується з такими методами доступу як DSL, WiFi, і іншими, тобто для «останніх метрів» доступу застосовуються технології Home PNA і HomePlug, а в якості «останньої милі» доступу використовуються DSL, WiFi і інші технології.

Швидкість передачі даних HPNA 1.0 складає 1 Мбіт/с, а відстань між найбільш віддаленими вузлами не перевищує 150 метрів. Специфікація HomePNA 2.0 забезпечує доступ зі швидкістю до 10 Мбіт/с і відстань до 350 м.

Технологія Home PNA застосовується в основному для організації домашньої мережі за допомогою мережевих адаптерів. Підключення до глобальної мережі можна здійснити за допомогою роутера через мережі загального доступу. Крім того, технологія HPNA призначена для організації колективного доступу в Інтернет (наприклад, для підключення житлового будинку або під'їзду будинку до Інтернет по існуючій телефонній лінії).

Стандарт HomePlug 1.0 доступ до Інтернет через побутову електричну мережу підтримує швидкість передачі до 14 Мбіт/с. максимальна протяжність між вузлами до 300 м.

Технологія PLC (Power Line Communication) дозволяє передавати дані по високовольтних лініях електропередач, без додаткових ліній зв'язку. Комп'ютер підключається до електричної мережі і виходить в Інтернет через одну і ту ж розетку. Для підключення до домашньої мережі не потрібно ніяких додаткових кабелів. До домашньої мережі можна підключити різне устаткування: комп'ютери, телефони, охоронну сигналізацію, холодильники і т.д. [12].

4.3. Стан і перспективи використання комп'ютерних мереж

Наше повсякденне життя важко уявити без досягнень обчислювальної техніки та телекомунікацій. Розвитку комп'ютерних мереж та телекомунікацій передувало винайдення телеграфу, радіо і телефону. Об'єднувати комп'ютери в мережі почали тільки 60-х роках ХХ ст., та вже зараз, завдяки вдосконаленню апаратури та програмних засобів, освічена людина може експлуатувати комп'ютерну мережу.

Зв'язок на невеликій відстані в комп'ютерній техніці існував ще до появи перших персональних комп'ютерів. До великих комп'ютерів приєднувалися численні термінали. Правда, вони практично не обробляли інформації, і основна мета організації зв'язку полягала у тому, щоб розділити машинний час великого потужного й дорогого комп'ютера між користувачами, що працюють за цими терміналами. Це називалося режимом поділу часу, тому що великий комп'ютер послідовно в часі вирішував завдання багатьох користувачів. У цьому випадку досягалося спільне використання найдорожчих у той час ресурсів – обчислювальних.

Потім були створені мікропроцесори й перші мікрокомп'ютери. З'явилася можливість розмістити комп'ютер на столі в кожного користувача, тому що обчислювальні, інтелектуальні ресурси подешевіли. Але всі інші ресурси залишалися ще досить дорогими. Об'єднавши кілька мікрокомп'ютерів, можна було організувати спільне використання ними комп'ютерної периферії. При цьому все оброблення інформації проводилося на місці, але її результати передавалися на централізовані ресурси.

Тут знову ж спільно використовувалося найдорожче, що є в системі, але вже зовсім по-новому. Такий режим одержав назву режиму зворотного поділу часу.

Потім з'явилися персональні комп'ютери, які відрізнялися від перших мікрокомп'ютерів тим, що мали повний комплект досить розвиненої для повністю автономної роботи периферії: магнітні диски, принтери, засоби інтерфейсу користувача (монітори, клавіатури, миші й т. п.). Периферія подешевіла й стала за ціною цілком порівнянною з комп'ютером.

Створення комп'ютерних мереж було викликано практичною потребою – мати можливість спільного використання даних. За відсутності мереж доводилося роздруковувати документи, щоб інші

користувачі могли працювати з ними, чи копіювати інформацію на дискети або інші накопичувачі даних. При цьому перевагою мережі є спільне використання комп'ютерів, даних, принтерів, модемів та інших пристроїв. Наприклад, мережа дозволяє об'єднати обсяги дисків всіх комп'ютерів, забезпечивши доступ кожного з них до всіх інших дисків як до власного. Без мережі також неможливо обійтися в тому випадку, коли необхідно забезпечити узгоджену роботу кількох комп'ютерів. Мережа дозволяє синхронізувати дії комп'ютерів, розпаралелити і відповідно прискорити процес оброблення даних.

Перші комп'ютерні мережі були невеликими і об'єднували близько десятка комп'ютерів і принтер. Технології обмежували розміри мереж, у тому числі кількість комп'ютерів у мережі та її фізичну довжину.

Наприклад, на початку 80-х років ХХ ст. найпоширеніший тип мереж складався не більше ніж із 30 комп'ютерів, а довжина кабелю мережі не перевищувала 185 м. Такі мережі легко розташовувалися в межах одного поверху чи будинку невеликої організації. Для маленьких фірм подібна конфігурація підходить і сьогодні. Такі мережі називаються локальними.

Проте локальні мережі не могли відповідати потребам великих підприємств, офіси яких зазвичай розташовані в різних місцях. Коли переваги комп'ютерних мереж стали незаперечними і мережеві програмні продукти почали заповнювати ринок, то перед корпораціями для збереження їх конкурентоспроможності постало завдання розширення мереж. Почали з'являтися нові стандарти, на основі локальних мереж стали створюватися великі системи. Більшість організацій отримала змогу зберігати і спільно використовувати в мережевому середовищі величезні обсяги життєво необхідних даних.

Сьогодні ідея з'єднання комп'ютерів за допомогою кабелю (чи безпроводного середовища) не здається нам особливо видатною, але свого часу вона ознаменувала нову епоху в історії комунікацій.

На сучасному етапі відбувається становлення інформаційного суспільства. Сучасні інформаційні технології мають вирішальний вплив на зміни, які відбуваються в соціальній структурі суспільства, економіці, розвитку інститутів демократії тощо.

Інформаційні технології, які покладено в основу інформатизації, на рубежі ХХ-ХХІ ст. стали важливим інструментом науково-технічного і соціально-економічного розвитку суспільства. Вони

продовжують виконувати суттєву роль у прискоренні процесів отримання, використання і розповсюдження нових знань і на їх базі прийнятті оптимальних обґрунтованих управлінських рішень, що, у свою чергу, сприяє підвищенню рівня добробуту країни.

Інформатизація – це організований соціально-економічний і науково-технічний процес створення оптимальних умов із метою задоволення інформаційних потреб на основі формування і застосування інформаційних ресурсів за допомогою інформаційних технологій і розвиненої інфраструктури. Інформатизація здійснюється шляхом вдосконалення і щільної взаємодії процесів і технологій споживання (створення документованої інформації, методів, засобів її оброблення, пошуку, розповсюдження, збирання, накопичення і збереження) із процесами створення і розвитку засобів, що повинні їх забезпечити (програмних, технічних, лінгвістичних, правових, організаційних щодо забезпечення документування інформації і виконання інформаційних процесів).

Тематика інформаційного суспільства, розвитку світового ринку телекомунікацій, мережі Інтернет не сходить зі сторінок провідних ділових вітчизняних і зарубіжних журналів. Колосальні обсяги інформації, присвячені цій тематиці, доступні зараз у мережі Інтернет. Особливо це стосується висвітлення діяльності міжнародних і національних організацій з розроблення і реалізації стратегій інформаційного розвитку. До найбільш цікавих джерел можна віднести Європейську Комісію Ради Європи, Організацію економічної співпраці розвитку, Всесвітній союз зв'язку, адміністративні органи, відповідальні за побудову "інформаційної магістралі" в США, Канаді, Великобританії, Німеччині, Франції, Австралії, Японії та інших країнах. Актуальність входження України в інформаційне суспільство поступово усвідомлюється вітчизняним науковим і політичним співтовариством, про що свідчить велика кількість законодавчих актів у сфері інформації.

Комп'ютерні мережі отримали дуже широке поширення. Сучасні мережі великі і складні, часто потребують кваліфікованого персоналу для їх обслуговування.

Наразі кількість користувачів, які регулярно використовують Інтернет, перевищила чверть земного населення.

У даний час підключитися до локальних мереж можна через супутники зв'язку, кабельне телебачення, радіосигнал, телефон,

стільниковий зв'язок, спеціальні оптоволоконні лінії або електропровід.

На сьогодні більше 80 % об'єднані в комп'ютерні мережі, починаючи від малих локальних мереж до глобальних мереж Інтернет [1, с. 25].

Щоб приблизно уявити, скільки приладів зараз реально претендують на IP-адреси, досить згадати, що в 1996 році в світі діяло 300 млн персональних комп'ютерів, у 2000-му ця цифра вже наблизилася до 600 млн., а до 2010 року цифра досягнула 3 мільярдів. До них варто додати сервери та мережеве обладнання, яке становить фізичну основу мережі Інтернет, ПК, смартфони та інші пристрої з виходом в мережу.

Функціонування комп'ютерних мереж довело їх технічну життєздатність, економічну доцільність і педагогічну перспективність. Включення комп'ютерів у мережі дозволило суттєво підвищити ефективність їх використання у різних галузях промисловості та науки. Навчальним закладам це дало змогу поліпшити наукову та методичну співпрацю між окремими кафедрами, факультетами та навчальними закладами взагалі.

Проблема перенесення та сумісності програмного забезпечення, що є досить складним процесом у галузі масового застосування обчислювальної техніки, набула іншої форми – можливості доступу через комп'ютерну мережу до різних комп'ютерів і програм, а спеціалізація обчислювальних центрів призвела до підвищення продуктивності праці програмістів.

Така проблема також виникає та розв'язується у комп'ютерних методиках навчання. Існує багато видів та форм педагогічних програмних засобів, що ускладнює роботу педагога. Багато вчених висловлюють думки про необхідність стандартизації педагогічного програмного забезпечення. Це стосується їх зовнішнього вигляду, призначення, структури та змісту допомоги, вмонтованої у програмний засіб та ін.

Характерною властивістю створення й експлуатації комп'ютерних мереж є комплексність виникаючих проблем. Коло завдань, що підлягають розв'язанню, не обмежується тільки технічними задачами, а включає також організаційні, економічні, соціальні, педагогічні, юридичні, а інколи й політичні проблеми. Що стосується педагогічних проблем, то це перш за все проблеми використання комп'ютерних мереж у навчанні та дистанційній освіті.

Питання для самоконтролю

1. Що таке проху-сервер?
2. Яке основне призначення проху-серверів?
3. Які види проху-серверів Ви знаєте?
4. Назвіть основні вимоги до проху-серверів.
5. Наведіть приклади проху-серверів.
6. Що сприяло виникненню бездротових мереж?
7. Назвіть основні характеристики бездротових ліній зв'язку.
8. Які бездротові технології Ви знаєте?
9. Опишіть принцип дії проху-сервера.
10. Що являє собою прозорий проксі?
11. Що являє собою локальна багатовузлова система розподілу?
12. Опишіть технології бездротової передачі даних WiFi.
13. Опишіть технології бездротової передачі даних WiMax.
14. Які умови є необхідними для роботи RadioEthernet-каналу?
15. Що являє собою технологія EDGE?
16. Що являє собою процес інформатизації?
17. Які, на Вашу думку, перспективи розвитку комп'ютерних мереж?

Тема 5. Принципи побудови глобальних комп'ютерних мереж. Мережа Інтернет. Основні сервіси. Провайдери

5.1. Принципи побудови глобальних комп'ютерних мереж

Нагальні потреби ринку інформаційних послуг поруч із бурхливим розвитком телекомунікаційних технологій в останні десятиріччя стали вагомими спонукальними причинами стрімкого розвитку технологій широкого обміну інформацією, побудови потужних глобальних комп'ютерних мереж. Створення і розвиток таких мереж є цілком закономірним і природним процесом. Розглядаючи сучасне суспільство як деяку складну динамічну систему, всередині якої постійно відбуваються процеси інформаційного обміну, необхідно особливо виділити як базовий елемент побудови інформаційної інфраструктури такого суспільства технології інформаційного обміну, технології передавання даних на великі відстані, технології побудови систем оперативного доступу до різноманітної інформації.

Глобальна або територіально-розподілена мережа (ГМ) сполучає кілька локальних мереж або ПК, географічно віддалених один від одного (рис. 5.1.1). Для глобальних мереж характерний значний масштаб (як за площею мережі, так і за числом вузлів) та неоднорідність, тобто різний тип архітектури та програмного забезпечення їх вузлів.

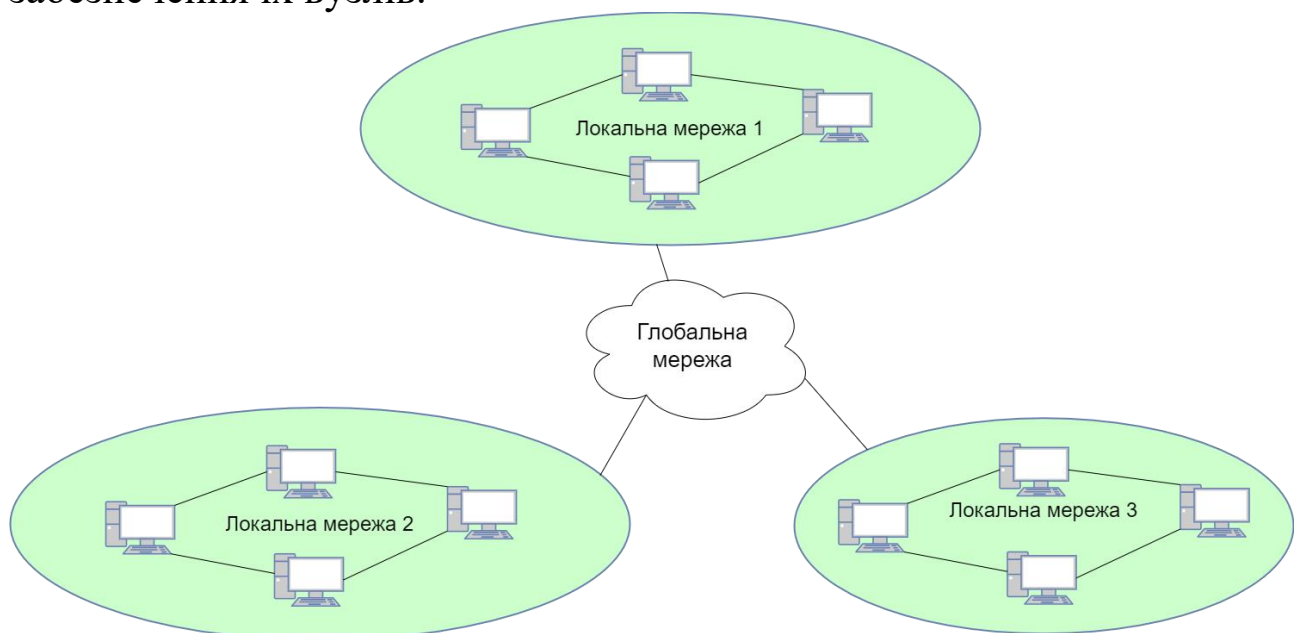


Рис.5.1.1. Глобальна або територіально-розподілена мережа

Розвиток глобальних мереж розпочався наприкінці 60-х рр. ХХ ст. у США. Першою була створена мережа ARPANET для спілкування науковців різних інститутів, що працювали над військовими проблемами. Головним заданням було створення мережі, яка могла б розширюватися, щоб приєднання чи вихід з ладу одного з вузлів не впливало на функціонування інших. Однак протокол NCP, згідно з яким працювала ARPANet, не передбачав взаємодії між мережами різної архітектури, тому в 1973 році було створено новий протокол TCP/IP. Перехід мережі на новий протокол дав їй змогу в 1984 р. розділитися на дві: MILNet – для військового відомства і ARPANET – для невійськового використання. У 1990 р. ARPANet припинила своє існування.

Кожна з глобальних мереж (Internet, FIDONet, ALONA, Bitnet, DECnet та ін.) організовувалися для конкретних цілей, а в подальшому поширювалися за рахунок під'єднання локальних мереж, які використовували їх послуги та ресурси.

Залежно від розмірів охопленої мережею території ГМ поділяють на: міжнародні, які функціонують на території кількох країн чи континентів; регіональні або національні, які охоплюють територію країни, регіону, міста; корпоративні або відомчі, що зв'язують між собою локальні мережі одного підприємства, установи, відомства.

Першою технологією глобальних мереж стала технологія E1...E4/T1...T4, що була розроблена на початку 60-х рр. ХХ ст. Почали використовувати цифрові лінії для передавання телефонних повідомлень. Сигнали від різних джерел накопичувалися в мультиплексорі і передавалися одним кабелем. При прийманні сигналів відбувалося їх перетворення на початкову форму.

У 70-х рр. ХХ ст. почали розвиватися XDSL-технології. Принцип цих технологій заснований на тому, що мовний сигнал не використовує всієї смуги пропускання телефонної лінії. Отже, є можливість розмістити цифровий сигнал. Для цього використовується спеціальне обладнання фільтрування сигналів, що надходять телефонними лініями та DSL-модеми. При використанні DSL-модемів можна паралельно користуватися і телефоном.

Найпоширеніша технологія у світі – технологія ISDN (цифрова мережа з інтеграцією послуг). Вона передбачає перетворення всіх видів даних у цифрові з наступним їх передаванням одним каналом

зв'язку. Це дозволяє швидко і якісно передавати мову, дані, музику, відео.

Технологія FDDI (розподілений оптичний інтерфейс даних) – це специфікація, що описує високошвидкісну мережу зі структурою типу «подвійне кільце» на основі оптоволокна. Вона була розроблена у 1986 р. FDDI забезпечує високошвидкісний зв'язок між мережами різних типів. Однак довжина кільця має обмеження – до 100 км. На основі FDDI можна також будувати локальні мережі.

FDDI базується на технології спільного використання мережі (рис. 5.1.2). Це означає, що одночасно можуть передавати дані кілька комп'ютерів (що може спричинити перевантаження мережі). FDDI використовує систему передавання маркерів у подвійному кільці. Трафік у мережі FDDI складається із двох подібних потоків, що рухаються в протилежних напрямках двома кільцями. Головна перевага топології типу «подвійне кільце» – надмірність. Одне кільце використовується для передавання даних, а друге є резервним. Зазвичай дані передаються лише основним кільцем. Якщо в основному кільці відбувається збій, то мережа автоматично переконфігурується і дані починають передаватися додатковим кільцем у протилежному напрямку.

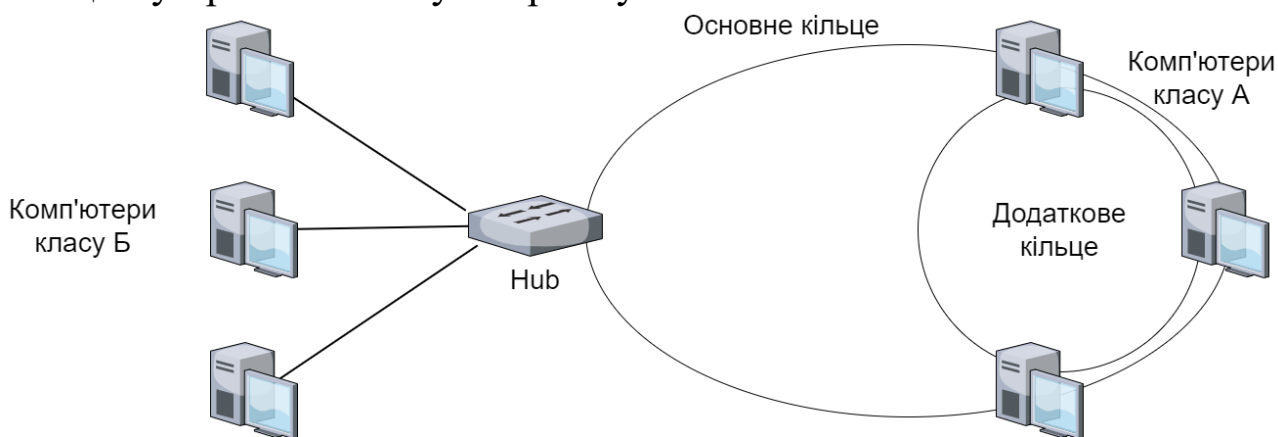


Рис.5.1.2. Схема підключення комп'ютерів до кілець FDDI

Ця технологія має певні обмеження: загальна довжина кабелю об'єднаних кілець має бути не більшою ніж 200 км; довжина одного кільця не повинна перевищувати 100 км; до одного кільця не може бути підключено більше, ніж 500 комп'ютерів; щонайменше через кожні 2 км має бути встановлений репітер.

Комп'ютери можуть підключатися до одного чи до обох кілець FDDI. Комп'ютери, підключені до обох кілець, називаються станціями класу А, а ті, що підключені тільки до одного кільця, –

станціями класу В. Якщо відбувається збій у мережі, то станції класу А беруть участь у переконфігуруванні мережі, а станції класу В – ні.

Ще відомі сучасні технології такі, як X.25, Frame Relay – протокол ретрансляції кадрів, асинхронний режим передавання (АТМ), на основі яких будуються як локальні, так і глобальні мережі, а на основі ТСР/ІР працює мережа Інтернет [1].

5.2. Структура і основні принципи побудови мережі Інтернет

Інтернет – всесвітня інформаційна комп'ютерна мережа, що є об'єднанням безлічі регіональних комп'ютерних мереж і комп'ютерів, які обмінюють один з одним інформацією по каналах громадських телекомунікацій (виділеним телефонним аналоговим і цифровим лініям, оптичним каналам зв'язку і радіоканалами, в тому числі супутникових лініях зв'язку).

Інформація в Інтернет зберігається на серверах. Сервери мають свої адреси і управляються спеціалізованими програмами. Вони дозволяють пересилати пошту і файли, проводити пошук в базах даних і виконувати інші завдання.

Обмін інформацією між серверами мережі виконується по високошвидкісних каналах зв'язку (виділеним телефонним лініям, оптоволоконним і супутниковим каналам зв'язку). Доступ окремих користувачів до інформаційних ресурсів мережі Інтернет зазвичай здійснюється через провайдера або корпоративну мережу.

Основними осередками глобальної мережі є локальні обчислювальні мережі. Якщо деяка локальна мережа безпосередньо підключена до глобальної, то і кожна робоча станція цієї мережі може бути підключена до неї.

Існують також комп'ютери, які безпосередньо підключені до глобальної мережі. Вони називаються хост-комп'ютерами (host-господар). **Хост** – це будь-який комп'ютер, який є постійною частиною мережі Інтернет, тобто з'єднаний по Інтернет-протоколу з іншим хостом, який в свою чергу, з'єднаний з іншим, і так далі.

Для під'єднання ліній зв'язку до комп'ютерів використовуються спеціальні електронні пристрої, які називаються мережевими платами, мережними адаптерами, модемами і т.д.

Практично всі послуги мережі Інтернет побудовані на принципі клієнт-сервер. Вся інформація в Інтернет зберігається на серверах. Обмін інформацією між серверами здійснюється по

високошвидкісних каналах зв'язку або магістралях. Сервери, об'єднані високошвидкісними магістралями, становлять базову частину мережі Інтернет.

Окремі користувачі підключаються до мережі через комп'ютери місцевих постачальників послуг Інтернету, Інтернет-провайдерів (Internet Service Provider – ISP), які мають постійне підключення до Інтернет. Регіональний провайдер, підключається до більш великого провайдеру національного масштабу, що має вузли в різних містах країни. Мережі національних провайдерів об'єднуються в мережі транснаціональних провайдерів або провайдерів першого рівня. Об'єднані мережі провайдерів першого рівня становлять глобальну мережу Інтернет.

Передача інформації в Інтернет забезпечується завдяки тому, що кожен комп'ютер в мережі має унікальну адресу (IP-адреса), а мережеві протоколи забезпечують взаємодію різнотипних комп'ютерів, що працюють під управлінням різних операційних систем.

В основному в Інтернет використовується сімейство мережевих протоколів (стек) TCP/IP. На каналному і фізичному рівні стек TCP / IP підтримує технологію Ethernet, FDDI і інші технології. Основою сімейства протоколів TCP/IP є мережевий рівень, представлений протоколом IP, а також різними протоколами маршрутизації. Цей рівень забезпечує переміщення пакетів в мережі і керує їх маршрутизацією. Розмір пакета, параметри передачі, контроль цілісності здійснюється на транспортному рівні TCP.

Прикладний рівень об'єднує всі служби, які система надає користувачеві. До основних прикладних протоколів відносяться: протокол віддаленого доступу Telnet, протокол передачі файлів FTP, протокол передачі гіпертексту HTTP, протоколи електронної пошти: SMTP, POP, IMAP, MIME [12, с. 45].

Адреси в мережі Інтернет мають як вузли (сервери), так і файли на цих серверах. Ліва частина (наприклад, ftp.cdrom.com) задає адресу вузла (деякого комп'ютера, під'єданого до мережі), а далі (після символу "/") вказується адреса файлу або директорії на вузлі. Адреса вузла будується за доменним принципом. Самий правий – домен верхнього рівня (для України це ua). В URL-адресах як домени верхнього рівня використовуються: com – для комерційних фірм; org – для некомерційних організацій; edu – для університетів та інших

навчальних закладів; net – для різних суто мережевих проектів; gov – для урядових; mil – для військових.

Лівіше (через крапку) іде домен другого рівня. Це може бути назва фірми або ім'я її провайдера, назва міста, просто назва мережевого проекту. Ще лівіше – домен третього рівня, четвертого і т. п.

Найлівіша частина адреси сервера вказує на вебсторінку (www) чи файловий сервер (ftp). Ця частина адреси не є обов'язковою. Адреса файлу чи директорії на вузлі задається так само, як і на ПК, тільки замість символу "\" використовується символ "/". Ім'я магнітного диску задавати не потрібно, а великі і маленькі літери розрізняються. Можливі довгі (до 255 символів) імена, але пробіли та деякі інші символи не припускаються.

Префікс http:// – Hyper Text Transfer Protocol – протокол пересилання гіпертекстів – означає адресацію до WWW. Адреси файлових або FTP-серверів починаються з префікса ftp://. Префікси mail to: та news: задають адреси електронної пошти і груп новин (телеконференцій) [1, с. 98].

5.3. Способи під'єднання та адресація в мережі Інтернет. Провайдери

Для під'єднання комп'ютера до глобальної мережі Інтернет необхідно:

- фізично під'єднати комп'ютер до одного з вузлів мережі;
- мати постійну або тимчасову IP-адресу;
- встановити і налагодити потрібне програмне забезпечення – (програми-клієнти тих сервісів, послугами яких потрібно буде користуватись);
- вибрати провайдера, за допомогою якого буде здійснюватись під'єднання до мережі Інтернет.

Доступ користувачам до мережі Інтернет надають спеціальні організації (юридичні особи), що називаються **провайдерами**.

Провайдер має постійно ввімкнений сервер, сполучений з іншими вузлами каналами з відповідними пропускними характеристиками і засоби для одночасного під'єднання багатьох користувачів (багатоканальний телефон, багатопортова плата тощо). Провайдери надають подібні послуги на договірній основі, найчастіше орієнтуючись на час роботи користувача або обсяг даних,

які пересилаються мережею. Під час укладення договору провайдер повідомляє клієнтові всі атрибути, необхідні для під'єднання та налагодження з'єднання (ідентифікатори – login, паролі, номери телефонів тощо).

Найвідомішими провайдерами в Україні зараз є: Укртелеком, Київстар, Воля, Тріолан та інші. Залежно від масштабу і рівня провайдера, його спеціалізації та інших факторів, набори і кошторис (а також якість) послуг різних компаній можуть відрізнятися [13].

В даний час відомі наступні способи доступу до мережі Інтернет:

1. Dial-Up (коли комп'ютер користувача підключається до серверу провайдера, використовуючи телефон) – комутований доступ по аналоговій телефонній мережі, швидкість передачі даних до 56 Кбіт/с.

2. DSL (Digital Subscriber Line) – сімейство цифрових абонентських ліній, призначених для організації доступу по аналоговій телефонній мережі, використовуючи кабельний модем. Ця технологія (ADSL, VDSL, HDSL, ISDL, SDSL, SHDSL, RADSL під загальною назвою xDSL) забезпечує високошвидкісне з'єднання до 50 Мбіт/с (фактична швидкість до 2 Мбіт/с). Основною перевагою технологій xDSL є можливість значно збільшити швидкість передачі даних по телефонних дротах без модернізації абонентської телефонної лінії. Користувач отримує доступ в мережу Інтернет із збереженням звичайної роботи телефонного зв'язку.

3. ISDN – комутований доступ по цифровій телефонній мережі. Головна особливість використання ISDN – це висока швидкість передачі інформації, в порівнянні з Dial-Up доступом. Швидкість передачі даних складає 64 Кбіт/с при використанні одного і 128 Кбіт/с, при використанні двох каналів зв'язку.

4. Доступ в Інтернет по виділених лініях (аналоговим і цифровим). Доступ по виділеній лінії – це такий спосіб підключення до Інтернету, коли комп'ютер користувача з'єднаний з сервером провайдера за допомогою кабелю (витої пари) і це з'єднання є постійним, тобто некомутованими, і в цьому головна відмінність від звичайного телефонного зв'язку. Швидкість передачі даних до 100 Мбіт/с.

5. Доступ до мережі Інтернет по локальній мережі (Fast Ethernet). Підключення здійснюється за допомогою мережевої карти (10/100 Мбіт/с) зі швидкістю передачі даних до 1 Гбіт/с на

магістральних ділянках і 100 Мбіт/с для кінцевого користувача. Для підключення комп'ютера користувача до Інтернету в квартиру підводиться окремий кабель (вита пара), при цьому телефонна лінія завжди вільна.

6. Супутниковий доступ до мережі Інтернет або супутниковий Інтернет (DirecPC, Europe Online). Супутниковий доступ в Інтернет буває двох видів – асиметричний і симетричний [12, с. 45].

У разі *симетричного* під'єднання клієнт здійснює передавання запиту на отримання і приймання даних через супутник. Але такий спосіб доступу є надзвичайно дорогим, як у частині клієнтського устаткування, так і у вартості обслуговування і застосовується, в основному, у тих випадках, коли його використання є або єдино можливим, або дешевшим, порівняно з використанням провідників або радіоканалів (наприклад, у важкодоступних районах з нерозвиненою інфраструктурою зв'язку).

У разі *асиметричного* під'єднання зі супутника здійснюється лише приймання даних, у той час як передавання запитів відбувається через наземні канали зв'язку, через провайдера мережі Інтернет, наприклад, використовуючи модем [13, с. 114].

7. Доступ до мережі Інтернет з використанням каналів кабельної телевізійної мережі, швидкість прийому даних від 2 до 56 Мб/с. Кабельний Інтернет («soax at a home»). В даний час відомі дві архітектури передачі даних – це симетрична і асиметрична архітектури. Крім того, існує два способи підключення: а) кабельний модем встановлюється окремо в кожній квартирі користувачів; б) кабельний модем встановлюється в будинку, де живе відразу декілька користувачів послуг Інтернету. Для підключення користувачів до загального кабельного модему використовується локальна мережа і встановлюється загальне для всіх устаткування Ethernet.

8. Бездротові технології [12, с. 46].

Завдяки розповсюдженню безпроводних мереж у великих містах з'явилося багато публічних точок доступу (хот-спотів).

Хот-споти – це точки, в радіусі дій яких можна під'єднатись до безпроводної комп'ютерної мережі, враховуючи наявність на комп'ютері або ноутбуці відповідного безпроводного мережевого адаптера. В ідеальному разі безпроводні точки дають змогу під'єднуватись з відстані 300-400 м. У реальній ситуації відстань обмежена потужністю антени і стінами, які екранують сигнал [13, с. 114].

5.4. Основні сервіси мережі Інтернет

На сьогоднішній день у мережі Інтернет існує безліч нарізноманітніших сервісів, що забезпечують роботу з усіма типами ресурсів.

Розглянемо найбільш поширені сервіси:

– World Wide Web (WWW) – єдиний інформаційний простір, що включає в себе різні мережеві ресурси. Наприклад, блоги, вебфоруми, вікі-проекти, Інтернет-аукціони і магазини, соціальні мережі та сайти знайомств та ін.;

– електронна пошта (E-mail), яка забезпечує можливість не лише обмінюватися текстовими повідомленнями між необмеженим числом абонентів, але і пересилати прикріплені файли;

– телеконференції та групи новин (Usenet) – дають можливість колективно обмінюватися різними повідомленнями;

– FTP сервіс – системи файлових архівів, які забезпечують зберігання і поширення різних типів файлів;

– Telnet сервіс – з його допомогою можна управляти віддаленими комп'ютерами в режимі терміналу;

– DNS сервіс – система доменних імен, яка забезпечує можливість використання літерних імен (наприклад, <http://www.myvin.com.ua>), замість числової адреси 195.248.234.141;

– IRC сервіс – сервіс підтримки чатів, миттєвий обмін текстовими повідомленнями в реальному часі.

Це стандартні сервіси мережі Інтернет, отже, всі принципи роботи програмного забезпечення, протоколи взаємодії клієнт-серверного забезпечення сформульовано і прописано в міжнародних стандартах.

У той же час існують і нестандартні сервіси, такі як Інтернет-пейджери: ICQ, Інтернет-телефонія, трансляція відео і радіо, файлообмінні мережі, пошукові системи, які є оригінальною розробкою окремо взятої компанії. Відсутність міжнародних стандартів при розробці таких систем нерідко призводить до технічних конфліктів з іншими сервісами [1, с. 10].

Служба World Wide Web

У 1992 р. було запропоновано ідею створення служби всесвітньої інформаційної мережі (WWW – World Wide Web – всесвітня павутина) як ефективного способу надання інформаційних

ресурсів, що дає змогу користувачам усього світу вільно обмінюватися даними. В основі цього проекту було використання гіпертексту – поняття, вперше введеного Тедом Нельсоном (Ted Nelson) у 60-х роках ХХ століття. У гіпертексті встановлюються зв'язки між деякими фрагментами тексту та іншими документами (які можуть бути текстом, зображенням, аудіо або відеофайлом тощо). Це означає, що у процесі звернення до цього зв'язку (або посилання) за програмою клієнтом відкриється відповідний документ. Унаслідок цього служба мережі Інтернет WWW набула великої популярності. У 1997 р. у WWW з'явилися динамічні сторінки з використанням мультимедійних технологій (так звані «вебсторінки»).

Гіпертекст (hypertext) – це метод подання даних у вигляді тексту, окремі фрагменти якого з'єднані за допомогою системи гіперпосилань.

Гіперпосилання (hyperlink) – спосіб зв'язку між різними компонентами даних у WWW-системі. Гіперпосилання відрізняються від звичайних посилань на інші документи своєю динамічністю. Наприклад, для того щоб скористатися наведеним у книзі звичайним посиланням на інший документ, скоріш за все, потрібно замовляти цей документ у бібліотеці. У разі гіперпосилання достатньо тільки встановити на нього курсор мишки та клацнути її лівою клавішею, і відповідний документ буде відображений на моніторі. Цей документ, у свою чергу, може мати гіперпосилання на багато інших документів тощо.

Отже, структура гіпертекстових документів справді нагадує павутину, вузлами якої є сторінки документа, пов'язані з іншими сторінками цього та інших документів «павутинками» гіперпосилань.

Документ у всесвітній інформаційній павутині називається вебсторінкою. **Вебсторінка** – це текстовий файл, у якому міститься опис мультимедійного документа мовою гіпертекстової розмітки (HTML – Hyper Text Markup Language). Групи тематично об'єднаних вебсторінок утворюють вебсайт (вебвузол).

Текст – це один довгий рядок символів, який читається в одному напрямі. Гіпертекстова технологія полягає в тому, що текст подається з ієрархічною структурою, подібною до мережі. Матеріал тексту поділяється на фрагменти. Кожний фрагмент, який можна переглянути на екрані комп'ютера, має багато зв'язків з іншими фрагментами, що дає змогу детальніше з'ясувати відомості про

об'єкт, який вивчається, і рухатися в одному чи кількох напрямках за обраним зв'язком.

Для передавання даних у WWW використовується протокол HTTP. Одним з призначень системи WWW була розробка стандартного способу вказування посилань на доступні в мережі Інтернет-ресурси, застосовного для будь-якого типу ресурсів. Для розв'язування цієї задачі було введено поняття URL (Universal Resource Locator – універсальний покажчик ресурсу).

Для відображення гіпертекстових документів на екрані комп'ютера і забезпечення швидких переходів між ними за допомогою гіперпосилань призначені спеціальні програми, що називаються *браузерами* (to browse від англ. переглядати).

Протокол HTTP (Hypertext Transfer Protocol – протокол передавання гіпертексту) – один з протоколів, який використовують у мережі Інтернет. Він призначений для передавання вебсторінок і містить правила, за якими вебдокументи передаються з вебсервера до браузера на комп'ютер користувача.

Вебсервер – комп'ютер у мережі, на якому встановлене серверне програмне забезпечення для опрацювання запитів від браузерів [13].

Служба FTP мережі Інтернет

FTP (File Transfer Protocol) – це протокол передавання файлів, один із перших сервісів мережі Інтернет. Служба FTP дає можливість абоненту обмінюватися двійковими і текстовими файлами з будь-яким комп'ютером мережі.

Такі файли найчастіше зберігаються в FTP-вузлах. Для звернення до FTP-вузла в адресному рядку браузера необхідно набрати URL-адресу вузла з використанням протоколу ftp://, за яким іде DNS-ім'я комп'ютера (або його IP-адреса), а також шлях до каталогу, який містить необхідні файли (якщо цей шлях відомо).

Як правило для отримання доступу до ресурсів необхідно зареєструватися на сервері. Однак на деяких FTP-серверах передбачено також анонімний доступ, при якому можна тільки переглядати каталоги і завантажувати файли. Тип доступу до даних визначає власник сервера. Як «гостьові» ім'я і пароль часто використовуються імена anonymous, ftp, а також адреси електронної пошти.

Зазначимо, що використання спеціальних програм для роботи за FTP-протоколом (наприклад, CuteFTP) надає більше можливостей, ніж використання браузерів.

Значну частину FTP-архівів складає безкоштовне (freeware) і умовно-безкоштовне (shareware) програмне забезпечення для різних типів комп'ютерів і операційних систем, технічна документація, архіви періодичних видань тощо.

В українському сегменті мережі Інтернет також існує багато FTP-серверів. На них накопичується вітчизняні дані і розташовується велика кількість файлів, скопійованих із закордонних, насамперед американських, серверів. Таке "віддзеркалювання" даних необхідне для зменшення трафіку в глобальній мережі.

Деякі FTP-сервери створюються в локальних мережах установ для внутрішніх потреб, хоча до окремих розділів може існувати і анонімний доступ ззовні.

FTP-сервіс можна використовувати для комерційного поширення програмних продуктів, баз даних, моделей, рекламних презентацій, великих за обсягом документів (книг) тощо. До появи і швидкого розвитку служби World Wide Web загальнодоступні файлові архіви FTP-серверів були основним засобом накопичення і поширення різних даних серед користувачів глобальної мережі.

Останнім часом, внаслідок інтенсивного розвитку служби WWW мережі Інтернет, відбувається поглинання сервісу FTP сервісом WWW [16].

Система обміну миттєвими повідомленнями

Миттєві повідомлення або система обміну миттєвими повідомленнями (англ. Instant messaging, скорочено IM) – телекомунікаційна служба для обміну текстовими повідомленнями між комп'ютерами або іншими пристроями користувачів через комп'ютерні мережі (як правило через Інтернет).

Зазвичай і від початку, це були невеликі текстові повідомлення. Але з розвитком у систему були додані й інші функції, такі як передавання файлів, зображень, звукових сигналів та повідомлень, відео, а також можливість здійснювати спільні дії, такі як спільне малювання або ігри.

Для користування цим видом комунікації необхідна клієнтська програма. Клієнтську програму системи миттєвих повідомлень часто називають *Інтернет-пейджером* або *месенджером*.

Обмін повідомленнями може бути або між двома або між кількома співрозмовниками (конференція, чат).

Кожна система миттєвого обміну повідомленнями повинна мати такі складові:

- Система ідентифікації (адресації) клієнтів.
- Система обліку стану клієнтів (хто в мережі, а хто ні).
- Система доставки повідомлень (зазвичай передається повідомлення через комп'ютерні мережу, але може наприклад безпосередньо до іншого користувача на тому самому комп'ютері).

Основними службами, що підтримують роботу ІМ є:

- Windows Live Messenger;
- AOL Instant Messenger;
- Yahoo! Messenger;
- Skype;
- Miranda NG.

Питання для самоконтролю

1. Що таке глобальна або територіально-розподілена мережа?
2. Що є характерним для ГМ?
3. Що таке Інтернет?
4. Що таке TCP/IP?
5. Що таке IP-адреса?
6. Які послуги надає Інтернет?
7. Назвіть функціональні можливості електронної пошти.
8. Яким сервісом мережі Інтернет Ви користуєтеся найчастіше?
9. Що таке WWW?
10. Охарактеризуйте принципи адресації в глобальних мережах.
11. Для чого призначена служба управління віддаленим комп'ютером?
12. Для чого призначена служба передавання файлів в мережі Інтернет?
13. За яким протоколом працює служба передавання файлів в мережі Інтернет?
14. Які є системи обміну миттєвими повідомленнями?
15. Які Інтернет-провайдери Ви знаєте?
16. Що таке статичне і динамічне виділення IP-адрес в мережі Інтернет? У чому їхня відмінність?
17. Що таке провайдер?
18. Що являє собою хост?
19. Що таке хот-споти? Яке їх призначення?
20. Які способи доступу до мережі Інтернет Ви знаєте?
21. Охарактеризуйте спосіб доступу до мережі по виділених лініях.
22. Що являє собою Dial-Up доступ до мережі Інтернет?

Тема 6. Браузери, їх призначення та функціональні можливості. Пошук даних в мережі Інтернет. Налаштування роботи вебсервера.

6.1. Призначення та функції браузерів

Браузер – програма, призначена для перегляду вебсторінок і переміщення в документі за посиланнями [13, с. 139]. Вона встановлює зв'язок із вебсервером, завантажує на комп'ютер сторінку, розміщену за вказаною користувачем адресою, та відтворює цю сторінку на екрані.

Вигляд вебсторінки (гіпертекстового документа) у вікні браузера залежить, від використаних у ній *тегів* – команд мови розмітки HTML. Стиль оформлення сайтів визначає також інформація, записана мовою CSS (Cascade Style Sheets – каскадні таблиці стилів) у HTML-документах чи окремих файлах. Сучасні браузери здатні відображати ілюстрації, відтворювати анімацію, відеоролики і звук.

Одна з головних особливостей браузерів полягає у тому, що вони надають можливість переміщатися всім простором Всесвітньої павутини. Переглядаючи з їх допомогою вебсторінки, на яких є гіперпосилання, користувач з легкістю, лише клацаючи відповідні посилання, переходить з однієї такої сторінки на іншу.

Усі сучасні браузери мають приблизно однакові можливості: забезпечують високу швидкість виведення вебсторінок, підтримують роботу як з багатьма вікнами (кожна вебсторінка відкривається в окремому вікні браузера), так і в режимі одного вікна (кожна вебсторінка відкривається на окремій вкладці одного вікна браузера), дозволяють з допомогою панелі інструментів вводити критерії пошуку і вибирати пошукову систему, пропонують вбудовані завантажувачі файлів і здатні забезпечити високий рівень безпеки під час роботи. Далі ми стило опишемо переваги та недоліки найпопулярніших браузерів.

Google Chrome

Google Chrome (www.google.com/chrome) – порівняно молодий браузер, випущений компанією Google (рис.6.1.1). Він позиціонується розробниками як швидкий, надійний та безпечний. І дійсно, стосовно надійності він найкращий. Справа в тому, що у разі збою закривається лише та вебсторінка, яка стала причиною помилки, а всі інші

вебсторінки залишаються відкритими. Інші ж браузери в разі збою закривають свої вікна повністю, з усіма відкритими вкладками, тож їх доводиться завантажувати заново.

Google Chrome також підтримує розширення і теми, доступні за адресою <https://chrome.google.com/webstore/category/extensions>. Щоб встановити розширення, потрібно клацнути посилання Увійти в правому верхньому куті сайту і ввести дані свого облікового запису Google. Після цього слід знайти потрібне розширення, навести на нього вказівник миші та клацнути кнопку ДОДАТИ В CHROME.

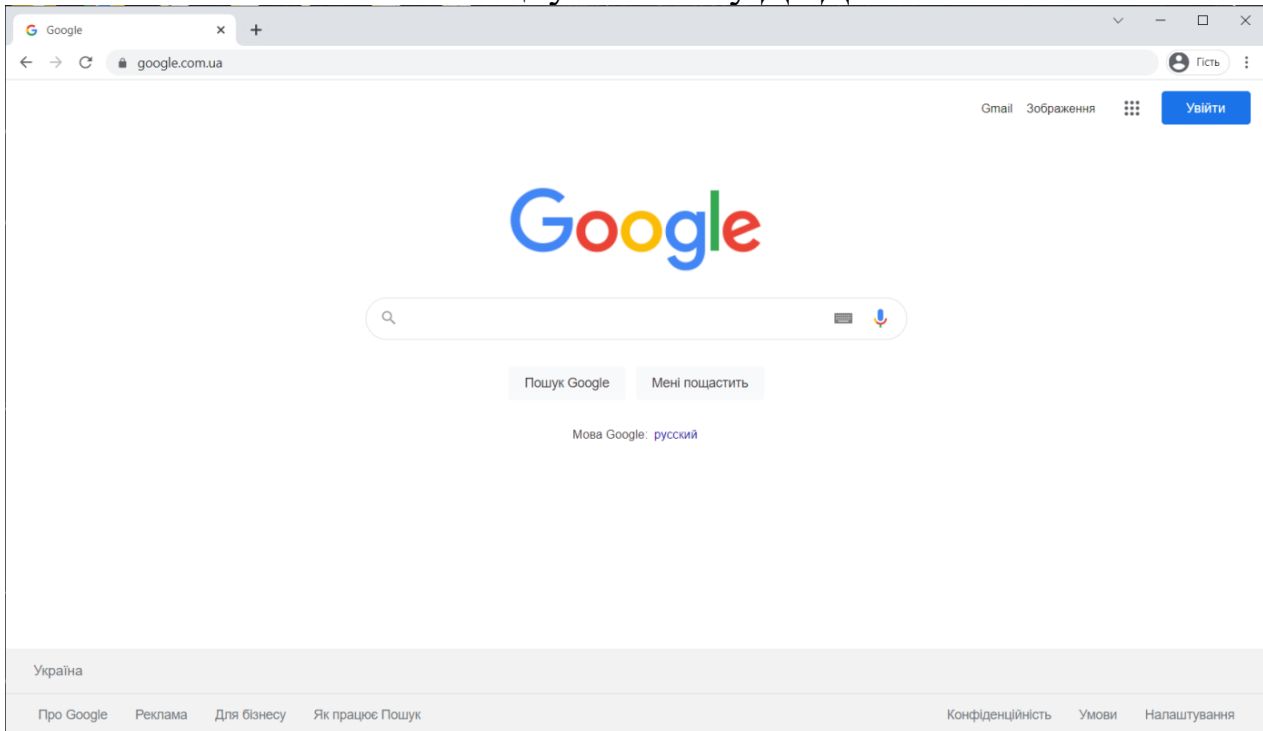


Рис. 6.1.1. Вікно браузера Google Chrome

Mozilla Firefox

Головна перевага Mozilla Firefox (www.mozilla.com/ru/firefox) – браузера, створеного компанією Mozilla (рис.6.1.2), полягає у тому, що кожен користувач може вибрати доповнення (міні-програми), які розширюють вбудовані функціональні можливості браузера, а також дозволяють змінити його вигляд (теми). Велика кількість доповнень міститься на сайті Firefox Add-ons за адресою <https://addons.mozilla.org>.

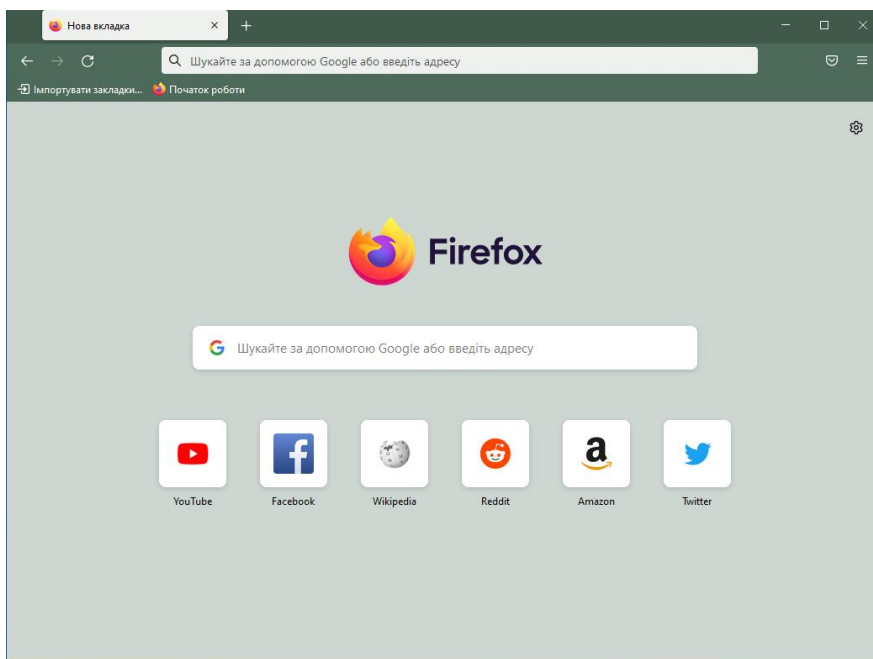


Рис. 6.1.2. Вікно браузера Mozilla Firefox

З допомогою доповнень можна розширити можливості пошуку, додати в браузер ігри, зробити більш зручним керування вкладками, підвищити безпеку браузера, відключити набридливу рекламу. Кожне доповнення має автора, назву і рейтинг (чим більше жовтих зірочок, тим він вищий). Встановити доповнення дуже просто – достатньо клацнути кнопку Add to Firefox, яка з’являється після наведення на нього вказівника миші (рис. 6.1.3). Одним з найпопулярніших вважається доповнення Easy YouTube Video Downloader, яке дозволяє зберігати відеоролики з сайту YouTube на жорсткий диск.

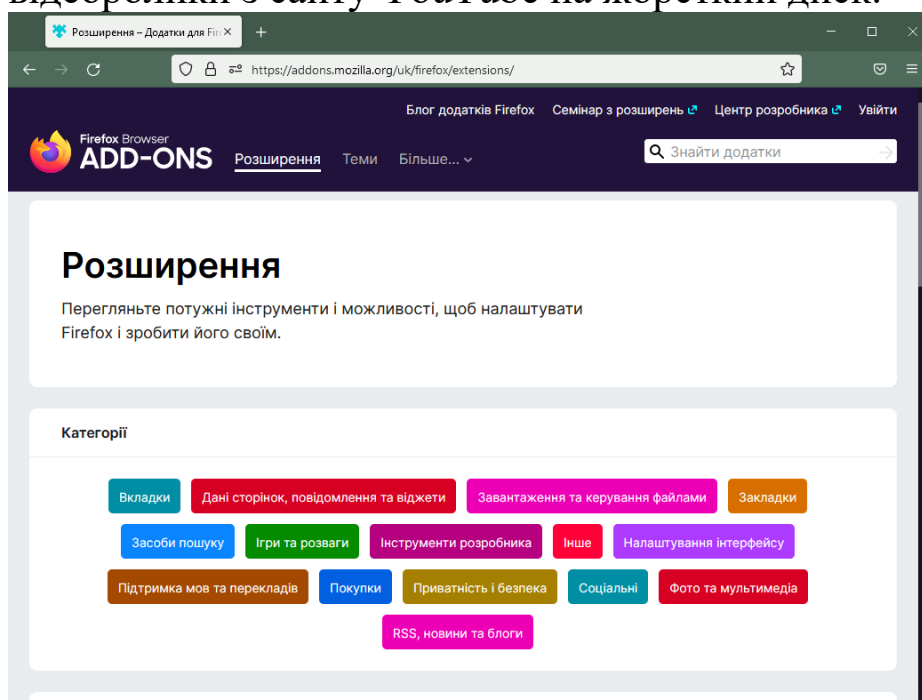


Рис. 6.1.3. Сторінка доповнень

Недоліком Mozilla Firefox є те, що цей браузер не вирізняється високою стабільністю роботи і потребує багато оперативної пам'яті, що інколи призводить до «гальмування» всієї системи.

Opera

Інтегрований пакет програм норвезької компанії Opera Software (<http://www.opera.com/>) є серйозним конкурентом продуктів Microsoft і Mozilla. Браузер Opera (рис. 6.1.4) невимогливий до ресурсів комп'ютера, характеризується високою швидкістю роботи, має зручний інтерфейс, панель інструментів, що налаштовується, а також чудовий засіб для масштабування відображуваних сторінок, що містять не лише текст, а й графіку. Стан браузера Opera може зберігатися після виходу з його вікна та відновлюватися під час його наступного відкриття.

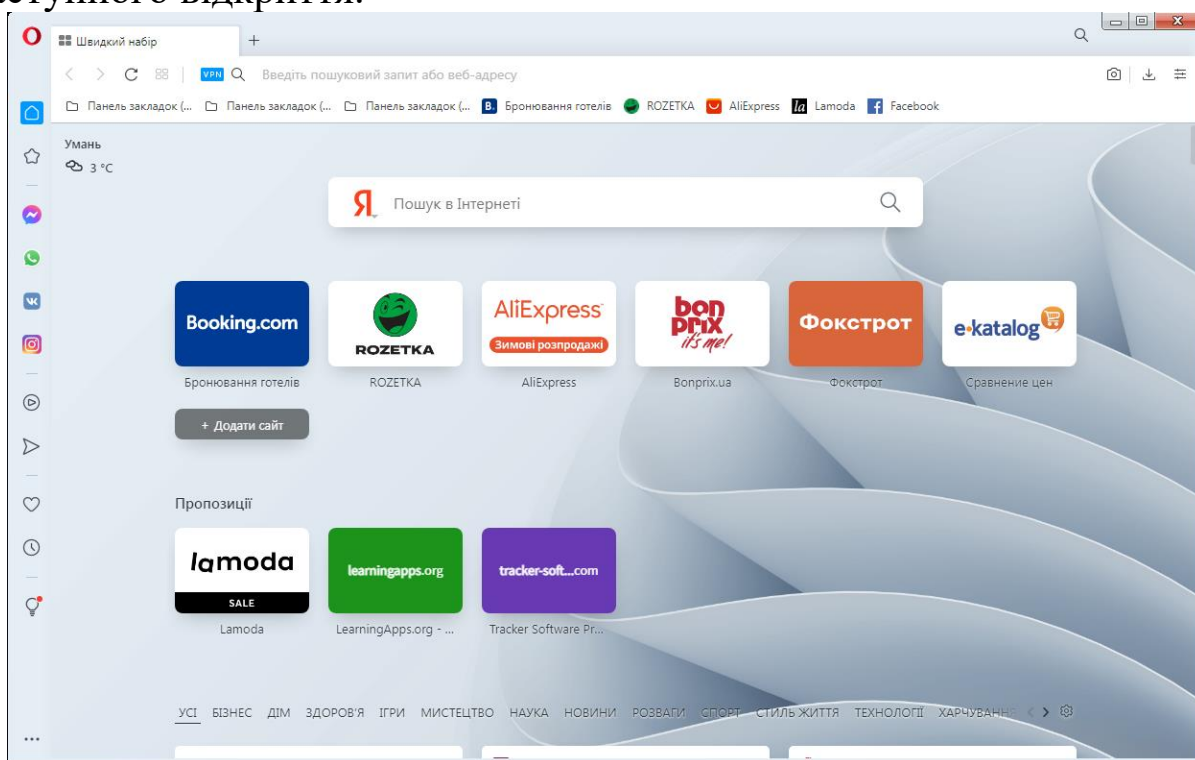


Рис. 6.1.4. Вікно браузера Opera

Саме у браузерах Opera були вперше розширені функції миші. Наприклад, для повернення на попередню сторінку достатньо, утримуючи праву кнопку миші, трохи перемістити курсор ліворуч.

Як і в Mozilla Firefox, в Opera можна встановлювати розширення і теми. Щоб встановити розширення, потрібно знайти його на сайті <https://addons.opera.com/uk/extensions> і клацнути розташовану праворуч кнопку Додати до Opera.

Мабуть, єдиним недоліком Opera, який іноді дає про себе знати і призводить до зависання усієї системи (на щастя, не на всіх комп'ютерах), є раптове збільшення використання ресурсів процесора до 100%.

6.2. Пошукові системи і каталоги

Пошукові системи – це складні технічні комплекси, що містять десятки швидкодіючих комп'ютерів, обслуговування яких ведуть сотні фахівців.

Якщо розумно використати пошукову систему, можна протягом достатньо короткого часу знайти інформацію, на пошук якої без використання Інтернет можна витратити місяці і навіть роки. Але, практика доводить, що ефективно і правильно використовувати пошукові системи вміють не більше 3% користувачів. Зазвичай, пересічний користувач на запит з 1-2 слів отримує інформацію, що охоплює значно більший простір ніж потрібно.

Пошукові системи інакше називають: пошуковими засобами або пошуковими машинами, жаргонною назвою є пошуковики. Еквівалентними іноземними термінами є: англійською – Search Engines, німецькою – Suchmaschinen, французькою – Le systeme de prospection.

Пошукові системи містять три основні компоненти:

– вебсторінка з пошуковим механізмом, яку користувачі використовують як інтерфейс для взаємодії з базою даних;

– база даних, де міститься інформація, що зібрана спеціальними програмами пошукової системи. Власне наявністю баз даних пояснюється висока швидкість виведення результатів пошуку на сторінку пошукової системи;

– пошукові роботи (Robots), павуки (Spiders) або хробаки (Worms) – спеціальні програми, які автоматично періодично «відвідують» сайти, збирають відомості про вміст сторінок, тобто індексують їх і наповнюють бази даних пошукової системи.

Щоб скористатися пошуковою системою потрібно завантажити її вебсторінку і сформулювати запит, за яким відбувається відбір документів з розподіленої бази даних, що зберігається на серверах в Інтернеті. Запит формується за допомогою ключових слів (одного або кількох). Результати пошуку видаються користувачу у вигляді переліку адрес сторінок (гіперпосилань) і короткої анотації до них.

Пошук – це процедура відбору потрібних документів, що зберігаються в Інтернеті. Пошук здійснюється або автоматично за допомогою ключових слів, або шляхом послідовного проходження користувачем по рубриках в каталогах.

Ключове слово – це будь-яка лексична одиниця природної мови, яка найбільшою мірою відображає зміст шуканого документа. При формуванні запиту можуть використовуватися не лише окремі ключові слова, але і словосполучення, що складаються з кількох ключових слів.

Запит – це набір ключових слів, за допомогою яких пошукова система автоматично проводить пошук і відбір необхідних документів. Іншими словами, запит – це інструкція (команда) для пошукової системи на пошук потрібних документів.

Запити бувають двох типів: прості і складні (або розширені, *advanced*). Прості запити складаються з окремих ключових слів або словосполучень. Складні запити, окрім ключових слів, містять логічні чи інші оператори.

Індексація – це процедура автоматичного заповнення бази даних пошукової системи, в якій зберігаються короткий зміст та анотації до сторінок і доменні адреси, за якими розміщені сторінки.

Коли пошукова система отримує конкретний запит на пошук потрібної інформації, то її пошуковий механізм порівнює ключові слова, що введені користувачем, з ключовими словами, що отримані під час індексації і збережені в базі даних системи. При збіганні цих слів користувачу видається назва сторінки, короткий опис, або абзац, де зустрічаються пошукові слова і доменна адреса даної сторінки.

Сучасні пошукові системи постійно вдосконалюють свої алгоритми роботи, і прагнуть проводити індексацію всього документа, а не лише його назви чи перших абзаців тексту. Найбільш досконалі роботи при індексації сканують і головну сторінку і решту сторінок, заходячи за гіперпосиланнями вглиб сайту. Попередня індексація документів мережі Інтернет дозволяє згодом за кілька секунд обслуговування запиту обробити (відсортувати) гігабайти різноманітної інформації. За допомогою механізму попередньої індексації можна отримати якісний результат пошуку у випадках, коли вдається точно сформулювати запит за допомогою невеликого числа ключових слів. В інших випадках користувач або отримує велике число посилань, які є лише "інформаційним шумом", або зовсім не знаходить необхідного документа.

Значно підвищують вибірковість пошуку фільтри, якими забезпечуються пошукові системи.

Фільтри дозволяють:

- обмежити список відібраних документів за допомогою логічних операторів (так званий складний пошук);

- обмежити простір пошуку типом протоколу, за допомогою якого було створено документ (пошук на сайтах або у телеконференціях);

- обмежити відібраний матеріал часовим відрізком, певними датами створення шуканого документа (наприклад, між 1 вересня 2005 р. і 20 вересня 2007 р.);

- відібрати документи, що складені однією мовою (українською, російською тощо);

- обмежити відбір документів територією розміщення серверів (наприклад, тільки з Європи, України чи навіть, лише з Львівської області);

- обмежити пошук певною частиною документа (заголовок, доменна адреса);

- відібрати документи, які містять фразу із заданим порядком розташування ключових слів.

При пошуку інформації пошукова система може робити дві помилки: не відбирати потрібні користувачеві документи і, навпаки, відбирати зайві документи, що містять інформаційний шум. Для уникнення цих помилок варто ознайомитися з поняттями "чутливість" і "вибірковість".

Чутливість – здатність пошукової системи відбирати документи, що відповідають запиту, не пропускаючи потрібних документів. Чутливість характеризується числом пропусків потрібних документів. Чим вищою є чутливість, тим менше вірогідність пропуску потрібного документа.

Вибірковість – здатність пошукової системи відбирати документи, що відповідають запиту і не виводити невідповідні документи. Якість вибіркової характеризується числом відібраних документів, що не відповідають запиту. Чим вищою є вибірковість, тим менше сторонніх документів потрапляє до користувача.

Вибірковість можна змінювати (регулювати) за допомогою логічних операторів (фільтрів).

Для уточнення запиту (фільтрації інформації) призначені логічні оператори OR, AND, NOT. Використання логічного оператора AND

(I) приводить до відбору документів, які обов'язково містять всі перелічені в запиті ключові слова, що сполучені цим оператором.

Оператор NOT (НІ) дозволяє виключити документи, які містять ключове слово, вказане після цього оператора.

За допомогою оператора NEAR (поблизу, поряд) користувач може відбрати документи, в яких ключові слова, що сполучені цим оператором, будуть міститися поблизу один від одного, а не в різних кінцях документа.

Оператор FOLLOWED BY дозволяє відбрати документи, в яких ключові слова слідує один за одним в заданому порядку.

Оператор ADJ відбирає документи, в яких ключові слова є суміжними (слідують одне за іншим).

Сучасні пошукові системи стають інтелектуальними. Використовуючи принципи штучного інтелекту, вони формують список документів (посилань на них), відповідно до ступеня їх релевантності.

Релевантність – відповідність знайденого в процесі пошуку документа до зробленого запиту. При цьому пошукові системи аналізують розташування знайдених ключових слів в документі (заголовку або тексті), число повторень ключових слів, їх взаємне розташування в документі. Найбільш точно відібрані документи розташовуються на початку списку документів, що знайдені в процесі пошуку. Для ранжирування знайдених документів використовують наступні показники.

Розташування ключового слова на сторінці (keyword prominence) – показник, що визначає, як близько від початку документа знаходиться задане ключове слово. Як правило, чим ближче до початку сторінки розташовується ключове слово, тим точніше документ відповідає запиту.

Частота ключового слова (keyword frequency) – показник, що враховує абсолютну частоту використання ключових слів (скільки разів зустрічається дане ключове слово на сторінці). Найбільшу «вагу» при ранжируванні документів мають слова, що розташовані в заголовку сторінки (так званий титул, він міститься між тегами title). Саме цей заголовок відображається в рядку заголовка браузера при перегляді сторінки, і слова, що там вказані мають найбільшу цінність (вагу) для пошукової системи.

Деякі пошукові системи при ранжируванні враховують індекс цитування (посилальна популярність, авторитетність, тематичний

індекс цитування, ІЦ, ТІЦ, Page Rank, Link Popularity) – кількість посилань в Інтернеті, що вказують на конкретний сайт. Чим їх є більше, тим вищим є індекс цитування.

Результати пошуку за допомогою різних пошукових систем будуть різнитися між собою. Це пояснюється тим, що в кожній пошуковій системі застосовуються різні алгоритми індексації, різна періодичність оновлення інформації в базі даних, охоплюється різний простір навколишніх серверів, і відповідно індексується різне число документів.

Спеціалізовані пошукові системи

Метапошукова система дозволяє вести автоматичний пошук за запитом з використанням відразу кількох пошукових машин.

Метапошукові системи:

– <http://www.metacrawler.com>

– <https://www.startpage.com>

Пошукові системи, що призначені для пошуку файлів:

– <https://globalfilesearch.com.webstatdata.com>

– <http://www.filesearch.ru>

Інші спеціалізовані пошукові системи:

– <https://www.findsounds.com/> Музичні файли.

– <https://tineye.com> Малюнки.

– <http://biblio.redaktor.biz/> Книги.

В спеціалізованих системах пошук часто відбувається не за допомогою ключових слів, а за відомими іменами файлів.

Окрім пошукових систем, використовуються інші механізми пошуку і збереження інформації в глобальній мережі. Природним чином утворилися комбіновані (гібридні) пошукові системи, які називають порталами. Вони містять відомості про сторінки різних тематичних напрямів і дозволяють проводити пошук, як за допомогою ключових слів, так і за допомогою каталогів.

Каталоги – це є ієрархічні структури, що містять впорядкований перелік сайтів, які супроводжуються короткими коментарями (анотаціями). Ресурси каталогів можна сортувати за датою долучення, за популярністю, за алфавітом тощо.

Каталоги можуть бути глобальними, національними, загальними і спеціалізованими (галузевими, тематичними тощо).

Каталоги з'явилися самим природним чином: користувачі відбирали для себе посилання на улюблені місця в Інтернеті і

складали впорядковані тематичні списки з деталізацією кожного розділу (рубрики). Саме так виник всесвітньо відомий каталог Yahoo!, що створено студентами Стендфордського університету Девідом Філо і Джері Янгом.

Рубрикація – розміщення документа у відповідний розділ (підрозділ, рубрику), яке, як правило, проводиться власноруч модераторами, розробниками чи власниками сайтів. Процедура рубрикації є достатньо суб'єктивною і здійснюється на підставі індивідуальних уявлень людей про дану наочну область. Автори при розміщенні своїх сторінок деколи керуються не точною відповідністю документа до назви рубрики, а іншими міркуваннями. Наприклад, з метою збільшення відвідуваності своїх сторінок їх поміщають в популярні рубрики. Сайт, що містить різноманітну інформацію інколи поміщають в кілька рубрик.

Модератор – співробітник, що відповідає за вміст каталогу (його окремого розділу, підрозділу тощо), за його достовірність та актуальність.

Окрім ієрархічних каталогів, зустрічаються і інші типи каталогів – неієрархічні, в яких немає супідрядності, наприклад, каталоги, що використовують алфавітно-наочну класифікацію, наприклад, енциклопедії, музичні сайти, телефонні довідники тощо.

Особливості та відмінності пошукових систем та каталогів.

Отже, пошукові системи і тематичні каталоги мають свої особливості і взаємно доповнюють один одного.

Пошукові системи «оглядають» велике число сайтів, що знаходяться на різних серверах. Тут, нажаль, автоматичний відбір шуканих документів супроводжується чималим числом помилок.

База даних каталогу зберігається на одному сервері і містить менший об'єм інформації, ніж в базі даних пошукової системи. Проте, результати пошуку за допомогою каталогів мають більшу релевантність в порівнянні з пошуком за допомогою пошукової системи.

Пошукові машини:

– індексація сайтів відбувається в автоматичному режимі за допомогою спеціальних програм-роботів;

– пошукові системи переглядають кожну сторінку сайту окремо. Може бути проіндексована як одна, так і всі сторінки сайту;

– зазвичай, сучасні пошукові системи самостійно знаходять та індексують нові сайти. Але, іноді за певних умов, пошукові системи

цього не роблять і сайт потрібно самостійно зареєструвати у пошуковій системі. Термін долучення сайту до бази даних (за умови відповідності сайту до вимог, що пред'являються) може складати від 1 до 2 тижнів;

- сайт може бути знайдений як результат обробки різноманітних запитів, що можуть відноситися до абсолютно різних галузей або видів діяльності;

- база даних в пошуковій системі постійно оновлюється. Алгоритм роботи пошукової системи є циклічним;

- при роботі з пошуковими системами кожна окрема сторінка сайту може бути оптимізованою під визначені ключові слова окремо від решти сторінок;

- алгоритм кожної пошукової системи є унікальним і може бути змінений у будь-який час без повідомлення власників тих сайтів, які розміщені в базі даних пошукової системи. Високий рейтинг сайту в результатах пошуку однієї системи, за певними ключовими словами, не гарантує високий рейтинг в результатах інших пошукових систем.

Каталоги:

- каталоги містять опис сайту як єдиного цілого. В каталогах немає опису окремих сторінок сайту. Власник сайту самостійно складає короткий опис сайту, що відображає його суть, призначення тощо. Після реєстрації сайту в каталозі, його зазвичай переглядає модератор, що відповідає за вміст каталогу або його окремого розділу. Змінити опис сайту в каталозі може модератор або власник сайту;

- складання каталогу (класифікація і анотація ресурсів) здебільшого є ручною роботою, і зазвичай в базі каталогу міститься значно менше сайтів, ніж в базі пошукової системи;

- у каталозі, як правило, зазначено доменну адресу головної сторінки сайту;

- при пошуку в каталозі вирішальну роль має короткий опис сайту, ніж ключові слова і метатеги, що містяться в тексті сторінок;

- сайти в каталогах можуть бути знайдені за датою розміщення, за алфавітом, за оцінкою модератора, за індексом цитування, за популярністю (числу відвідувачів) тощо;

- сортування сайтів за алфавітом і датою розміщення, пошук за назвами сайтів є абсолютно об'єктивними критеріями і не залежать від модератора;

– сортування сайтів за категоріями є об'єктивним критерієм. Якщо власник сайту при додаванні сайту до каталогу вибрав відповідну категорію, а модератор з цим погодився;

– сортування сайтів за іншими критеріями не може мати 100% об'єктивний характер. В певних каталогах рейтинг може мати комерційний характер;

– сайт може займати перші позиції в каталозі, якщо назва сайту починається з перших літер алфавіту чи якщо сайт було додано в каталог нещодавно, а сортування застосовано за датою додавання;

– виявлення посилань до неіснуючих сайтів в каталозі може бути виконане автоматично, але часто потрібно додатково перевіряти існування ресурсу модератором.

Метапошукові системи

Метапошукова система дозволяє вести автоматичний пошук за запитом з використанням відразу кількох пошукових машин.

Так як різні пошуковики використовують різні алгоритми пошуку і приділяють "особливу" увагу різним ділянкам мережі, до яких вони найбільш адаптовані, логічно в багатьох випадках шукати відразу декількома пошуковими машинами. Цю можливість і пропонують searchbots, Автономні пошукові агенти. В основу роботи searchbots закладений наступний принцип: із запиту користувача генеруються запити, відформатовані в синтаксисі і логічних конструкціях кожного конкретного пошукового ресурсу.

Таким чином, з одного запиту метапошукова машина робить безліч запитів, які потім розсилаються широкому колу пошукових машин (та/або каталогів). Зібравши результати, метапошукова система видаляє дубльовані посилання і, відповідно до свого алгоритму, об'єднує/ранжує результати в загальному списку.

Безумовно, автономні пошукові агенти стали значним проривом у розвитку технологій пошуку. Але використання цих засобів приховує в собі певні проблеми. Наприклад, іноді пошук із застосуванням мета-засобів може виявитися дуже повільним, адже цим системам доводиться координувати за часом надходження результатів обробки пошукового запиту від декількох серверів. Ще одним недоліком мета-засобів є те, що вони не дозволяють повною мірою використовувати можливості мови запиту кожного з застосовуваних пошукових засобів. Більшість таких систем погано розуміє (чи не розуміє зовсім) російську мову.

Метапошуковики не ведуть власної бази Інтернет-сайтів, а лише обробляють результат, представлений іншими пошуковими машинами. Обсяг цієї обробки може бути різним. Найпростіші метапошуковики дозволяють видалити повторювані посилання та створити єдиний список сайтів, впорядкований за релевантністю.

Метапошуковики, що з'являються останнім часом, дозволяють проводити кластеризацію (об'єднання) отриманих адрес сайтів за різними критеріями. Такі системи дозволяють виявляти в списку отриманих сайтів загальні ключові фрази і групувати сайти відповідно до них. Автономні пошукові агенти продовжують розвиватися – головним плюсом метапошукових систем є їх можливість шукати в базах безлічі каталогів і пошукових машин, що дозволяє швидко переробляти великі обсяги інформації й істотно економити трафік. Автономні пошукові агенти бувають двох класів. Одні, як і звичайні пошуковики, розташовуються на публічному онлайн-ресурсі, інші встановлюються на персональний комп'ютер.

Dogpile (www.dogpile.com). Це пошуковик від InfoSpace, який об'єднує результати пошуку найбільш популярних пошукових систем Google, Yahoo! Search, MSN, Ask.com, About, MIVA, LookSmart і ін. Досить цікавий метапошуковик з веселим мультяшним інтерфейсом. Він почав працювати в 2006 році. Dogpile в першу чергу виводить ті посилання, які дублюються відразу в декількох пошукових машинах (поруч з кожним посиланням вказується, де вона була знайдена). Чим нижче по списку ви опускаєтесь, тим більше унікальні посилання бачите, а це означає, що по ним міститься або інформація, що не індексується іншими пошуковими системами, які матеріали, які були додані відносно недавно.

Є зручна функція для контролю вмісту. Він передбачає три варіанти фільтрації: відсутність фільтрації, відкидання найбільш нескромних сторінок і, нарешті, максимальна фільтрація, після якої, по ідеї, повинні залишитися тільки бездоганно благопристойні посилання. Ви можете налаштувати спосіб сортування результатів пошуку зображень, новин і медіафайлів. Крім того, відповідно до своєї назви на сайті пошуковика організовуються реальні акції на допомогу тваринам, викладені різні картинки з собаками і т.д.



Рис. 6.2.1. Головна сторінка Dogpile

Copernic Agent (copernic.com). Метапошукові машини можуть встановлюватися на комп'ютери користувачів. Вони зручні і можуть налаштовуватися на коло запитів і інтересів конкретної людини. Зараз можна знайти кілька десятків систем цього класу. Одна з найпопулярніших таких систем – Copernic Agent, програма компанії Copernic Technologies. Результати пошуку зберігаються в своїй основі програми, їх можна зберігати в файлах стандартних форматів разом з гіпертекстовими посиланнями. Розміри анотацій (описів) можна змінювати, програма сама оновлює бази пошукових ресурсів при кожному підключенні до Інтернету. Є безкоштовна і кілька платних версій програми. Платні версії працюють не тільки з індексами пошукових машин. Вони вміють працювати з контентом багатьох великих видавництв і агентств. Посилають запити до CNN, MSNBC, New York Times, Forbes і т.д. Досить серйозний мінус, присутній і в ToolBar програми і в її безкоштовній версії – відсутність пошуку по базі Google.

Metacrawler (www.metacrawler.com). Метапошукова система, розроблена у 1994 році в Університеті Вашингтона аспірантом Еріком Селбергом і професором Ореном Еціоні як кваліфікаційний проект. Протягом усього свого існування вона поєднувала результати вебпошуку з джерел, зокрема Google, Yahoo!, Bing (раніше Live Search), Ask.com, About.com, MIVA, LookSmart та інші пошукові системи. Результати запитів до пошуковика можна переглядати за декількома категоріями, зокрема Новини, Зображення та Відео.

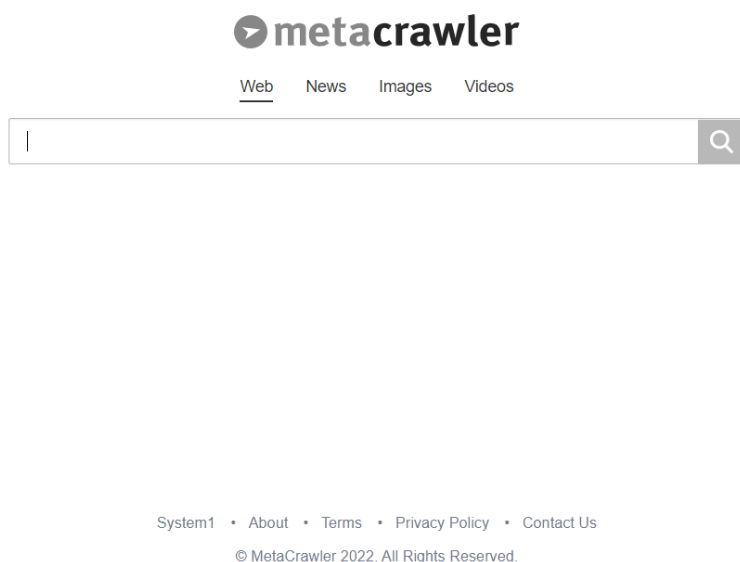


Рис. 6.2.1. Головна сторінка Metacrawler

У тому, що Google і Yandex – це незамінні інструменти для пошуку в мережі, сумніватися не доводиться – пошук в цих системах зручний, гнучкий і дуже точний. Але це не означає, що альтернативні пошукові системи не мають права на існування. У більшості випадків вони індексують меншу кількість сторінок, ніж стандартні пошукові системи. Але у таких пошукових систем є одна незаперечна перевага – вони пропонують щось нове, відмінне від прийнятих стандартів. Оскільки альтернативні пошукові сервіси використовують інший підхід до підбору ресурсів, відповідних запиту, результат пошуку буде інший, ніж у випадку з звичними пошуковими системами. Корисно використовувати метапошуковики або вузькоспеціалізовані пошукові системи. Знайдені результати фільтруються, відбираються найкращі і часто для кращого сприйняття візуалізуються за допомогою діаграми, карти сайтів, хмари тегів і т.д. Розробники альтернативних пошукових систем часом так далеко заходять у своїх пошуках нового універсального інтерфейсу, що, іноді важко визнати у вебсторінці пошукову машину. Головне, щоб візуальні ефектів не замінили собою основну задачу, для вирішення якої і були створені пошукові системи (збір та систематизація максимальної кількості інформації та її подальша обробка).

Інтелектуальні пошукові системи

Як відзначають дослідники Інтернету, результати пошуку в великій мірі залежать від правильно сформованого запиту. Багато людей, намагаючись отримати будь-яку інформацію, формулюють

свої питання в досить розмовній формі у вигляді питань типу, «яка завтра буде погода». Звичайний пошуковик видає посилання у відповідь на таке питання відповідно до принципу максимально пересічних слів в питанні і в змісті тексту на сайті. Тому, щоб отримати відповідь на своє питання, користувач повинен сам переглянути видану інформацію і знайти в ній шукане рішення задачі. У створенні інтелектуальних пошукових систем реалізується ідея про комп'ютер, який зможе спілкуватися з людиною на зрозумілій йому мові, володіючи при цьому набагато більшими знаннями, ніж звичайна людина. Передбачається, що така система буде відповідати на найрізноманітніші запитання користувачів, задані на звичайній розмовній мові. І відповідати не посиланнями на сторінки або цитатами з них, а фактами. В останні кілька років такі пошукові системи з'явилися. Це Nokia, Powerset і WolframAlpha.

Wolfram Alpha (www.wolframalpha.com). Wolfram | Alpha – база знань з елементами штучного інтелекту, в якій закладено величезну кількість математичних моделей та інформації.

Її творець Стівен Вольфрам після навчання в Ітоні і Оксфорді отримав ступінь доктора наук в галузі теоретичної фізики в Каліфорнійському технологічному інституті в 20 років. У 1981 році йому був вручений "грант геніїв", стипендія фонду Макартур. Його книга «Нова наука» (*A New Kind of Science*), в якій він заявляє, що в основі всіх наук можуть лежати прості алгоритми, а не складні структури і правила, була зустрінута із захопленням з одного боку і несхваленням з іншого.

В основі Wolfram Alpha лежить "движок" Mathematica. У нього вбудований потужний логічний механізм, який може робити висновки, маючи дані і математичну модель. Wolfram Alpha може, наприклад, не тільки відповідати на запитання на кшталт "Де знаходиться Єгипет?" (Прості пошуковики з таким навчилися справлятися), а й "Де завтра опівдні буде знаходитися МКС?". Для того щоб відповісти на останнє запитання, потрібно зрозуміти, коли настане завтра, а також мати математичну модель руху МКС або розрахувати її, виходячи з наявних про станції даних. Звичайні пошуковики цим не займаються. Wolfram Alpha для цього створено.

Пошуковиком Wolfram Alpha можна назвати з великою натяжкою. Він у вкрай рідкісних випадках звертається до зовнішніх джерел даних. Команда Стівена Вольфрама витратила кілька років на створення бази даних фактів, а також великої кількості математичних

моделей. Пошуковик займається пошуком в бібліотеці, заснованій на програмних продуктах Стефана Вольфрама – Mathematica і A New Kind of Science. Розроблені автором алгоритми дозволять надати цим даним зрозумілу для користувача форму. Коли користувач задає питання, дані для відповіді будуть братися саме з цієї бази. Мова не йде про запити на природній мові, хоча сама мова запитів дуже схожа на англійську мову. У той же час система розуміє короткі запити, схожі з тими, до яких звикли користувачі звичайних пошуковиків. Відповідь виводиться у вигляді діаграм, таблиць і графіків.

Запуск першої публічної версії системи відбувся 16 травня 2009 року.

Спеціалізовані пошуковики

Пошук зображень

Picsearch (www.picsearch.com). Сервіс працює дуже швидко, виводить результати пошуку в зручній формі і майже ніколи не показує битих посилань. Під час пошуку не забувайте користуватися додатковими настройками (кнопка Advanced Search), де можна задати розмір шуканого зображення, його кольоровість і наявність на картинках анімації.

Gettyimages (<https://www.gettyimages.com/>). Це один з найбільших фотобанків (з можливістю пошуку картинок в інших галереях), а головне в ньому – найбільший відсоток безкоштовних зображень.

Пошук музики та звуків

FindSounds (www.findsounds.com). Пошуковик створений для пошуку звуків (не плутати з музикою), різних звукових ефектів і т.д. Ресурс дозволяє шукати звукові файли різних форматів – wav, mp3, aiff, au. У базі даних ресурсу є найрізноманітніші звуки – крики тварин, скрегіт машин, дзвін, стукіт, сирени, дзижчання комах, гуркіт вибухів і стрілянини, сплеск води і т.д. У результатах пошуку ресурс показує не тільки посилання на знайдені файли, але і їх основні характеристики, а також показує графік амплітуди звуку, по якій можна судити про характер звучання даного прикладу. Шукає за аудіо форматами – AIFF, AU, MP3 і WAVE, за кількістю каналів – mono і stereo, і ще за кількома параметрами, включаючи максимальний розмір файлу. Шукає швидко і багато, що в зв'язці зі зручним і інтуїтивно-зрозумілим інтерфейсом дозволяє використовувати його в професійних потребах і не тільки. Він схожий на Google, але

сфокусований саме на звуках. Даний пошуковик надає потужні можливості, але в той же час є простим, зрозумілим і зручним в користуванні. Є дуже зручна функція для батьків: в цьому пошуковику фільтрується музика, яка містить в собі ненормативну лексику. Отримавши запит в командному рядку – наприклад, Rocket, – пошуковик видає кілька сотень посилань на файли, в яких хтось його вимовляє, або лунає звук ракети, або звук старту космічного човника, а то і зовсім звук пострілу з ракетниці в Quake або DOOM. База звукових ефектів FindSounds може знайти застосування в самих різних областях – від розробки комп'ютерних ігор та інших додатків, до створення презентацій і всіляких кліпів.

Allmusic (www.allmusic.com). Велика онлайн база даних, що належить організації All Media Guide, поряд з All Movie Guide і All Game Guide. Заснована в 1991 році Майклом Ерлевайном і математиком Володимиром Богдановим. Інтернет-портал працює з 1995 року. На ньому надана інформація по жанрах музики, музикантах і музичних групах, а також інформація про вихід альбомів і професійні рецензії. У цій основі містяться 1,695,140 альбомів, 317,552 класичних композицій, 1,207,634 виконавців, 282,732 композиторів, 98,384 біографій, 930 музичних стилів, 77,241 фотографій виконавців і т.д.

Новинні пошуковики

CuteStat (<https://www.cutestat.com/>). Пошуковик пропонує відмінну заміну ранковій газеті. Вводите в поле пошуку будь-яке слово, пов'язане з подіями або відомими людьми, і отримуєте результат – огляд найпопулярніших сервісів зі згадуванням цікавить вас інформації на них. Серед сайтів, за якими проводиться пошук і збір інформації: новини live.com, повідомлення з Twitter, відео з YouTube, Google Blogs, картинки з Flickr і багато іншого.

Наукові пошукові системи

В даному розділі зібрані посилання на спеціалізовані наукові пошукові системи, електронні архіви, засоби пошуку статей та посилань.

Google Scholar (scholar.google.com). Пошукова система по науковій літературі. Включає статті великих наукових видавництв, архіви препринтів, публікації на сайтах університетів, наукових товариств та інших наукових організацій. Що не маловажно, розраховує індекс цитування публікацій і дозволяє знаходити статті, які містять посилання на ті, що вже знайдені.

Scholar (www.scholar.ru). Пошуковик був створений для спрощення пошуку документів наукової тематики російською мовою. Основна мета проекту – збір інформації щодо вільно скачуваних публікацій. Проект не розрахований на зберігання повних текстів публікацій в тому чи іншому вигляді. Замість цього, використовується база посилань на тексти документів з інформацією про самі публікації (анотація, автори і т.д.).

PubMed (<https://pubmed.ncbi.nlm.nih.gov/>). Це пошукова система розроблена в Національному Центрі Біотехнологічної Інформації (National Center for Biotechnology Information – NCBI), який є підрозділом Національної Медичної бібліотеки США (National Library of Medicine – NLM), яка є частиною Національного Інституту Здоров'я США (National Institutes of Health – NIH). MEDLINE – база даних медичної інформації, що включає бібліографічні описи з більш ніж 4800 медичних періодичних видань з усього світу, починаючи з 1949р. В даний час MEDLINE доступна безкоштовно для пошуку через Інтернет як для фахівців, так і для широкого загалу. Плюси цієї системи очевидні: величезна база даних і велика кількість функцій для управління пошуком. Зараз це універсальний пошуковик літератури з медичної тематики.

HighWire Press (<https://www.highwirepress.com/>). Це електронний підрозділ бібліотека Стенфордського Університету. Ресурс містить повнотекстові статті, частина з яких доступна безкоштовно, частину можна купити. Це найбільше у світі сховище безкоштовних повнотекстових статей з галузі life science.

Інтернет бібліотеки

Це, звичайно, не зовсім пошукові системи. Але при їх правильному використанні допоможуть зекономити багато часу і привести до бажаного результату набагато швидше, ніж за допомогою власне пошукової машини. Тим більше, що серед посилань будь-якого пошукача якась із таких бібліотек обов'язково буде вказана. Електронних бібліотек існує багато як загального, так і вузькоспеціалізованого призначення. На деяких сайтах є додаткові сервери, на зразок «розгадувача кросвордів» або онлайн конвертування одиниць вимірювання. У деяких випадках разом з самими енциклопедіями дається досить повний каталог онлайн словників. Крім того, існують спеціалізовані пошукові машини по електронним бібліотекам. В даному огляді наведені приклади найбільш цікавих і повних з бібліотек.

Wikipedia (www.wikipedia.org). Вікіпедія (англ. Wikipedia) – вільна загальнодоступна багатомовна універсальна енциклопедія, підтримувана некомерційною організацією «Фонд Вікімедіа». У ній є інформація більш ніж на 50 мовах світу. Запущена в січні 2001 року Вікіпедія зараз є найбільшим і найбільш популярним довідником в Інтернеті. За обсягом відомостей і тематичним охопленням вважається найповнішою енциклопедією з коли-небудь створених. Однією з основних переваг Вікіпедії як універсальної енциклопедії є можливість представити інформацію рідною мовою, зберігаючи її цінність в аспекті культурної приналежності.

Світ енциклопедій (www.encyclopedia.ru). Величезна добірка енциклопедій як по загальних поняттях, так і по різним науковим галузям. До більшості сайтів зі змістом статей доступ безкоштовний. Існує каталог друкованих енциклопедій. Є внутрішня пошукова система [14].

6.3. Налаштування роботи вебсервера

Програмним засобом для розміщення гіпертекстових навчальних систем та інформаційним центром навчального мережного комплексу повинен бути вебвузол, який створюється на базі вебсервера. У вузькому значенні **вебсервер** – це набір програм, який забезпечує обмін даними засобами протоколу передачі гіпертексту (HTTP – Hyper Text Transfer Protocol). У широкому розумінні під вебсервером розуміють набір апаратних і програмних засобів, що забезпечують функціонування вебвузла. Серед вебсерверів найбільш поширеними є Apache та Microsoft Internet Information Server. Загальними вимогами до вебсерверів є: робота з мовами серверних скриптів (PHP, Perl, ASP), робота із серверами СУБД.

Програмний комплекс «Денвер»

Враховуючи популярність Apache, PHP, MySQL, компанією Dklab розроблений програмний комплекс «Денвер» (Джентльменський набір веброзробника). За допомогою цього комплексу можна організувати вебсервер Apache з підтримкою мов PHP, Perl та сервер СУБД MySQL на комп'ютері, що працює під управлінням будь-якої ОС Windows. Як правило, при встановленні комплексу не потрібно проводити жодних додаткових налаштувань, і тому його можна використовувати навіть недостатньо підготовленим

користувачем. Простота встановлення та налаштування комплексу дають змогу використовувати Денвер у процесі самостійного створення вебсайтів учнями і вдома. Серед переваг комплексу слід відзначити його модульність, можливість розширення, кирилізований інтерфейс.

Окремо слід відзначити повну автономність комплексу Денвер, яка полягає у тому, що:

- комплекс встановлюється в одну папку і не записується жодних даних в іншу папку або реєстр операційної системи;
- системі не потрібна спеціальна програма вилучення (деінсталяції) комплексу;
- для запуску комплексу не встановлюються додаткові сервіси.

У випадку встановлення комплексу існує можливість його запуску на іншому комп'ютері, виконавши лише копіювання його папки.

Базову конфігурацію можна завантажити за покликанням <http://www.denwer.ru/>. У разі необхідності існує можливість завантаження додаткових складових, які містять інтерпретатор мови Perl з модулями, бібліотеки, використання яких розширюють можливості мови PHP і забезпечують роботу з архівами, графікою, базами даних, відмінними від MySQL.

Компоненти комплексу вже сконфігуровані для роботи за замовчуванням. Звичайно, для підвищення ефективності роботи та використання додаткових можливостей необхідно редагувати конфігураційні файли, проте основні, базові можливості є доступними відразу після встановлення.

До комплексу Денвер входять такі програмні складові:

- сервер Apache, до складу якого входять виконувані файли, дистрибутивні та адаптовані конфігураційні файли;
- інтерпретатор мови PHP, що містить виконувані файли, модуль для вебсервера Apache, дистрибутивний і адаптований конфігураційний файл. Інтерпретатор, подібно до ОС Linux, працює як модуль вебсервера Apache, що дає змогу відлагодження програм;
- сервер СУБД MySQL, до складу якого входять виконувані файли, файли повідомлень про помилки, база даних MySQL;
- `phpmyadmin` – вебінтерфейс для управління базами даних;
- інтерпретатор мови Perl, що містить виконувані файли без додаткових модулів;

- програма для імітації роботи поштового сервера Sendmail, яка не відправляє листи, а лише записує їх у файл;
- система пошуку віртуальних вебвузлів.

Після запуску програми інсталяції здійснюється перевірка наявності необхідних для встановлення драйверів та утиліт операційної системи. Наступним кроком є задання папки, у якій будуть розміщені сервери (Рис. 6.3.1).

Оскільки програми встановлення додаткових модулів комплексу здійснюють перегляд корневих папок дисків, то не варто вказувати папки дуже глибокого вкладення. Програма встановлення створює віртуальний диск, який є необхідним для функціонування компонент системи. Окремий диск спрощує роботу з вебінструментарієм, формуючи структуру папок, схожу до Unix-систем. *Віртуальний диск* – це диск, кореневий каталог якого збігається з однією з папок на фізичному диску. Після його створення всі дії з віртуальним диском насправді здійснюватимуться із вказаною папкою. Для уникнення конфліктів з назвами реальних дисків операційної системи віртуальному диску слід виділити одну з останніх літер латинського алфавіту, наприклад Z.

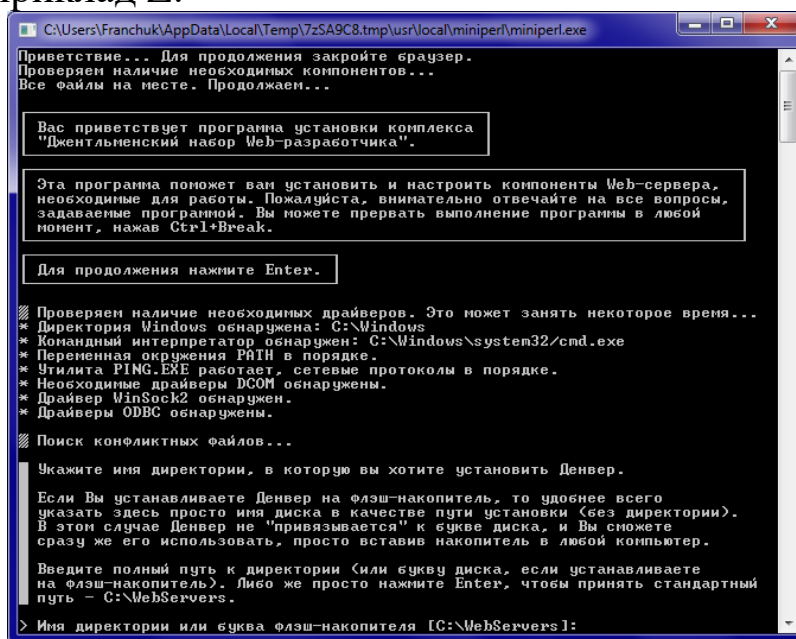


Рис. 6.3.1. Встановлення комплексу Денвер

Після копіювання файлів необхідно вказати режим роботи віртуального диска:

- віртуальний диск створюється у процесі завантаження ОС Windows. У випадку завершення роботи комплексу віртуальний диск

не від'єднується. Такий режим можна використовувати за необхідності роботи з віртуальним диском без запуску серверів;

– віртуальний диск створюється тільки після завантаження комплексу.

Найдоцільнішим є другий спосіб створення віртуального диска, оскільки це не сприятиме випадковому доступу до файлів комплексу.

Для зручності запуску та зупинки програм комплексу на робочому столі створюються ярлики.

Структура папок системи подібна до Unix-систем (Рис. 6.3.2):

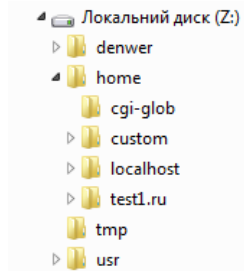


Рис. 6.3.2. Структура папок комплексу «Денвер»

Z:\etc – містить файли налаштувань комплексу та програми запуску і зупинки комплексу (run.exe, stop.exe, restart.exe). Слід звернути увагу на обов'язковість використання цих програм, оскільки в протилежному випадку існує небезпека втрати відомостей в базах даних;

Z:\home – призначена для розміщення вебсторінок. HTML-документи слід розміщувати у папці /home/<ім'я віртуального вузла>/www, який є кореневою папкою вебвузла. Побудова нового віртуального вузла полягає у створенні у папці /home папки з іменем вузла із вкладеною папкою www, та перезапуску комплексу. Усі зміни система автоматично фіксує у конфігураційних і системних файлах;

Z:\tmp – містить тимчасові дані. У цю папку поміщаються також листи, відправлені за допомогою програми для імітації роботи поштового сервера Sendmail;

Z:\usr – містить програмні компоненти – виконувані та конфігураційні файли Apache, PHP, MySQL.

Після встановлення і запуску комплексу слід відкрити його індексну сторінку за адресою <http://localhost> (Рис. . 6.3.3).

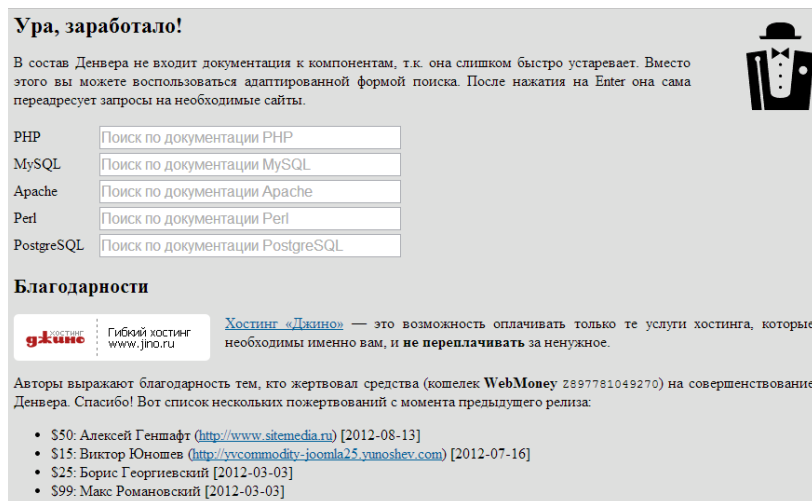


Рис. 6.3.3. Початкова сторінка комплексу «Денвер»

На цій сторінці пропонується провести тестування функціонування віртуальних вузлів різних рівнів, інтерпретаторів мов PHP та Perl, відлагоджуваного сценарію-імітатора сервера Sendmail, вебінтерфейсу для роботи із сервером MySQL phpMyAdmin.

Слід звернути увагу на думку авторів проекту про те, що «Денвер» не призначений для використання в умовах локальної мережі. Така позиція пояснюється тим, що відкритий у межах локальної мережі чи мережі Інтернет вебсервер несе потенційну небезпеку для комп'ютера, на якому він працює. Проблема ускладнюється і тим, що комплекс, як правило, завантажується від імені адміністратора, тобто з максимальними правами. Проте ці недоліки має і сервер Apache. Автори ж комплексу зовсім не проти його використання як у локальних мережах, так і в мережі Інтернет [16].

Програмний комплекс «Open Server»

Open Server – це портативна серверна платформа і програмне середовище, створене спеціально для веброзробників з урахуванням їх рекомендацій та побажань.

До складу програмного комплексу входить великий набір серверного програмного забезпечення, багатofункціональний та зручний інтерфейс, системи адміністрування та налаштування компонентів. Програмний комплекс широко використовується для розробки, налагодження і тестування вебпроектів, а так само для надання вебсервісів в локальній мережі.

Хоча спочатку програмні засоби, що входять до складу комплексу, не розроблялись спеціально для роботи один з одним, таке

поєднання програмних засобів стало популярним серед користувачів операційної системи Windows, в першу чергу через те, що вони отримували безкоштовний комплекс програм з надійністю роботи на рівні Linux серверів.

Завдяки зручності і простоті управління програмний комплекс Open Server зарекомендував себе як першокласний і надійний інструмент необхідний кожному вебмайстру.

Властивості програмного комплексу:

- Непомітна робота в області повідомлень ОС Windows;
- Швидкий старт і зупинка;
- Кілька режимів управління доменами;
- Монтування віртуального диска;
- Підтримка управління через командний рядок;
- Підтримка профілів налаштувань;
- Зручний перегляд логів всіх компонентів;
- Перемикання HTTP, MySQL і PHP модулів;
- Детальна документація;
- Доступ до доменів в «один клік»;
- Швидкий доступ до шаблонів конфігурації;
- Багатомовний інтерфейс;
- Автозапуск програм при старті Open Server;

Особливості комплексу:

- Не потребує встановлення (портативність);
- Можливість роботи з USB накопичувача;
- Одночасна робота з програмними комплексами Denwer, Xampp і т.д. ;
- Робота на локальній (мережевій) зовнішній IP-адресі;
- Підтримка SSL без додаткових налаштувань;
- Створення домену шляхом створення звичайної папки;
- Підтримка кирилических доменів;
- Підтримка аліасів (доменних покажчиків);
- Захист сервера від зовнішнього доступу;
- Runuscode конвертер доменних імен;
- Пакет з понад 40 портативних програм;
- Планувальник завдань (cron);
- Створення локального піддомена без втрати видимості основного домену в мережі Інтернет;

Програмні засоби в комплекті:

Офісні програмні засоби:

- CintaNotes – Нотатки;
- FoxitReader – PDF переглядач;
- Notepad ++ – Текстовий редактор;
- Notepad2 – Записник;
- STDUViewer – Мультипереглядач;
- SumatraPDF – Перегляд PDF;
- Sublime Text 2 – Редактор коду;
- WinMerge – Порівняння файлів і папок;

Утиліти:

- InfraRecorder – Запис дисків;
- Peazip – Архіватор;
- Punto Switcher – Автопереключення мов;
- Everything – Миттєвий пошук файлів;
- VirtuaWin – Віртуальні робочі столи;

Інтернет:

- Dmaster – Менеджер завантажень;
- FileZilla – FTP клієнт;
- Firefox – Браузер;
- Google Chrome – Браузер;
- Kitty – SSH клієнт;
- Skype – Спілкування;
- TeamViewer – Віддалений помічник;
- uTorrent – Торрент клієнт;
- WinMTR – Ping і Traceroute;
- WinSCP – SCP клієнт;

Система:

- AutoRuns – Менеджер автозавантаження;
- Sports – Моніторинг відкритих портів;
- Doublecmd – Файлменеджер;
- MyEventViewer – Системний журнал;
- ProcessExplorer – Менеджер процесів;
- ProcessMonitor – Моніторинг процесів;
- WinDirStat – Аналіз розміру папок і файлів;

Робота з графікою

- CamStudio – Запис відео;
- EYE3 – Інструмент «піпетка»;
- IcoFX – Редактор іконок;
- PicPick – Мультимастер;
- PixBuilder – Фоторедактор;

– PngOptimizer – Оптимізатор графіки.

Ідея проекту Open Server полягає в тому, щоб користувач (розробник) не був залежним від робочого місця. Звичайний розробник часто залежить від роботи за конкретним комп'ютером, від операційної системи та програм встановлених на цьому комп'ютері. Використання програмного комплексу Open Server дозволить звільнити розробника від подібних незручностей.

Користувач отримує набір портативних (що не вимагають установки) програм для веброзробника (Рис. 6.3.4). Якщо Open Server необхідний тільки як заміна таким програмам як Denwer, Vertrigo, Xampp і т.д., то можна скористатися версією «Mini», яка містить тільки серверну частину платформи.

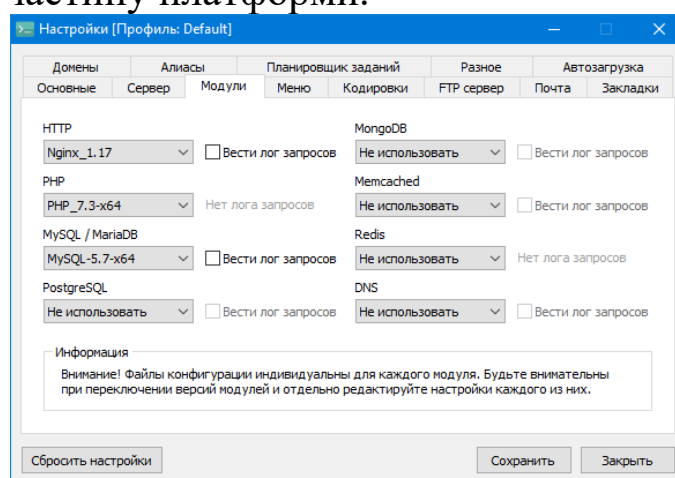


Рис. 6.3.4. Програмний засіб Open Server

Старт проекту Open Server відбувся 1 серпня 2010 року. Ідея створити нову і більш досконалу WAMP платформу народилася після багаторічного очікування оновлення іншої відомої WAMP збірки – Denwer.

Перші експериментальні версії програми були написані у співпраці з талановитим програмістом з України. Пізніше розробка велася на мові PHP в середовищі швидкої розробки додатків (RAD) для Microsoft Windows під назвою PHP Devel Studio. Надалі, з причини того, що використання середовища розробки були сильно обмежені і не дозволяло проекту розвиватися далі, програма була переписана з нуля і починаючи з версії 4.0 розробка ведеться на мові Delphi.

Програма поширюється як Freeware, знаходиться у фазі активної розробки і розвитку. Ліцензійною угодою не накладаються обмеження на використання програми і не передбачає будь-яких гарантій [16].

Питання для самоконтролю

1. Які інструменти пошуку інформації в мережі Інтернет Вам відомі?
2. Чи завжди інформації, знайденої в мережі Інтернет, можна довіряти?
3. Що забезпечують тематичні колекції посилань?
4. Що таке релевантність пошуку?
5. Дайте означення точності пошуку.
6. Опишіть принцип роботи пошукової машини.
7. З чого складаються пошукові машини?
8. Назвіть рекомендації щодо пошуку інформації.
9. Охарактеризуйте основні компоненти пошукової системи.
10. Які можливості надає використання фільтрів при пошуку інформації в мережі Інтернет?
11. Що таке браузер?
12. Що являє собою вебсторінка?
13. Яка мова використовується для написання вебсторінок?
14. Що таке гіперпосилання?
15. Які спеціалізовані пошукові системи Ви знаєте?
16. Наведіть приклади метапошукових систем.
17. Що таке вебсервер?
18. Наведіть приклади програмних засобів, які можна використати як вебсервер.
19. З чого складається пошукова машина?
20. Які функції виконує пошукова машина?
21. З яких елементів складається вікно браузера?
22. Які існують правила введення ключових слів для пошуку даних в глобальній мережі Інтернет?
23. Назвіть логічні оператори мови запитів пошукової служби. Наведіть приклади.
24. Які властивості браузера можна налаштовувати?

Тема 7. Основи мови HTML. Технології створення вебсторінок

7.1. Основні поняття мови HTML. Способи створення вебсторінок

Найпростішим компонентом Web є HTML (Hyper Text Markup Language). **Вебсторінка** (документ HTML) – текстовий файл мовою HTML формату *.htm або *.html розміщений в World Wide Web (WWW). **WWW** – всесвітня павутина, розподілена система доступу до гіпертекстових документів, що існує в мережі Інтернет. Вебсторінка, крім тексту, може містити гіпертекстові посилання, за допомогою яких можна переходити до інших вебсторінок і переглядати їх, а також містити вставки у вигляді графіки, анімації, відеокліпів і музики. Для перегляду вебсторінок можна використовувати будь-який браузер.

Мова HTML дозволяє:

- створювати й редагувати вебсторінки, у тому числі свою домашню вебсторінку, яку можна потім розмістити в Інтернеті;
- редагувати документи HTML, отримані з Інтернету, так щоб функціонували всі впроваджені в документ об'єкти (картинки, анімації тощо);
- створювати мультимедійні презентації, слайдшоу, демонстраційні проекти завдяки гіпертекстовим посиланням і можливості вставляти в документ HTML малюнки, діаграми, анімації, відеокліпи, музичний і мовний супровід, текстові спецефекти (наприклад рядок, що біжить).

Існують три способи створення вебсторінок (або документів HTML):

1. Використання текстового редактора Блокнот (Notepad), вбудованого в Windows, і перегляд результатів за допомогою браузера. Технологія створення вебсторінки така:

– у редакторі Блокнот створюється файл вебсторінки, який зберігається з розширенням *.htm. Потім цей файл завантажується і переглядається браузером.

2. Використання спеціальних редакторів документів HTML, наприклад Hot Metal Light, Hot Dog Professional, MS Front Page, HtmIpad та ін.

3. Використання Word, де створюється текст документа, який потім конвертується в HTML-формат.

Основні поняття мови HTML.

1. **Елемент** – це конструкція мови HTML або контейнер, що містить дані. Вебсторінка являє собою набір елементів.

2. **Тег** – це стартовий і кінцевий маркери елемента. Теги визначають границі дії елементів і відокремлюють елементи один від одного. Практично усі теги (за виключенням деяких) парні, тобто є «відкриваючий» і «закриваючий» теги. У тексті вебсторінки теги записують у кутові дужки, наприклад: <HTML>. Кінцевий тег завжди задається з косою рисою: </HTML>.

3. **Гіперпосилання** – фрагмент тексту, який є покажчиком на інший файл або об'єкт. Гіперпосилання дозволяють переходити від одного документа до іншого.

4. **Фрейм** – область гіпертекстового документа зі своїми смугами прокручування.

5. *Аплет* – програма, передана на комп'ютер клієнта у вигляді окремого файлу, яка запускається під час перегляду вебсторінки.

6. *Скрипт* – програма, включена до складу вебсторінки для розширення її можливостей.

7. *Завантаження (Download)* – копіювання документа з вебсервера на комп'ютер клієнта.

8. *Upload* – копіювання документа з комп'ютера клієнта на вебсервер – використовується під час створення власної вебсторінки.

Загальна структура типового найпростішого документа

HTML:

<COMMENT>Коментар</COMMENT>

<HTML>

<HEAD>

<TITLE>Назва документа</TITLE>

</HEAD>

<BODY>

Тут розташований текст самого документа HTML.

</BODY>

</HTML>

7.2. Теги мови HTML

Теги структури документа

<HTML> . . . </HTML>	визначає початок і кінець документа
<HEAD> . . . </HEAD>	визначає початок і кінець заголовка документа

<TITLE> . . . </TITLE>	вказується заголовок поточної сторінки
<BODY> . . . </BODY>	визначає початок і кінець тіла документа

Атрибути тегу BODY

BGCOLOR=« . . . »	визначає колір фона
BACKGROUND=« . . . »	вказує адресу файлу для Background (рекомендується підбирати колір bgcolor, максимально наближений до кольору background)
TEXT=« . . . »	задає колір тексту
LINK=« . . . »	задає колір гіперзв'язків (посилань)
VLINK=« . . . »	задає колір посилань, де користувач уже побував
ALINK=« . . . »	задає колір активного гіперзв'язку
<! . . . >	використовується для вставки коментаря

Приклад використання:

```
<BODY bgcolor=«f0ffff» BACKGROUND=«31.jpg»
LINK=«880000»
VLINK=«0a00ab» ALINK=«ffff00»>
```

Параграфи, роздільники, заголовки

<i>Параграфи і абзаци</i>	
 	початок нового рядка
<P>	початок нового абзацу, з відступом
<i>Заголовки</i>	
<H1> . . . </H1>	заголовок першого рівня
<H6> . . . </H6>	заголовок шостого рівня
<i>Роздільники</i>	
<HR>	горизонтальна лінія, що йде через весь екран

Атрибути тегу HR

SIZE=« . . . »	визначення товщини лінії (у пікселях)
WIDTH=« . . . »	визначення довжини лінії (у пікселях або у відсотках від ширини сторінки)
COLOR=« . . . »	визначення кольору лінії
ALIGN=left ALIGN=right ALIGN=center	визначення способу вирівнювання лінії (ліворуч, праворуч, по центру)
NOSHADE	без використання тривимірних ефектів

Приклади використання:

<HR COLOR=«008000» SIZE=«1» NOSHADE>

<HR SIZE=«3» WIDTH=«50» ALIGN=right>

<HR SIZE=«5» WIDTH=«50%» ALIGN=center>

Відображення тексту

<BASEFONT SIZE= . . . >	Встановлення розміру базового шрифту. Діапазон – від 1 до 7, (за замовчуванням встановлюється 3)
<BIG> ... </BIG>	текст буде відображатися шрифтом більшого розміру
<SMALL> . . . </SMALL>	текст буде відображатися шрифтом меншого розміру
 ... 	напівжирний текст
<I> ... </I>	курсив
<U> ... </U>	підкреслений текст
<STRIKE> . . . </STRIKE>	перекреслений текст
<S> ... </S>	перекреслений текст
<TT> ... </TT>	шрифт фіксованої ширини
<BLINK> . . . </BLINK>	миготливий текст
_{...}	текст зміщується вниз
^{...}	текст зміщується нагору
 ... 	визначення кольору, розміру і гарнітури тексту

Атрибути тегу FONT

COLOR=«...»	визначення кольору тексту
FACE=гарнітура	визначення типу гарнітури
SIZE=«...»	визначення розміру шрифту від 1 до 7 або відносно базового розміру за допомогою значень +n або -n

Приклади використання:

Відображення інформації

<ADDRESS> ... </ADDRESS>	відображає інформацію про адресу
--------------------------	----------------------------------

<code><BLOCKQUOTE> ... </BLOCKQUOTE></code>	виділяє текст відступами ліворуч і праворуч. Використовується для цитування
<code><CITE> ... </CITE></code>	використовується для посилань (назви книг, фільмів тощо) і цитат
<code><CODE> . . . </CODE></code>	використовується для частин початкового тексту програм
<code><SAMP> . . . </SAMP></code>	використовується для розміщення у документі друку програм тощо
<code> ... </code>	використовується для виділення тексту
<code><DFN> ... </DFN></code>	дає приклад для обговорюваного терміна або поняття
<code><KBD> ... </KBD></code>	використовується для виділення тексту, що набирається користувачем на клавіатурі
<code><VAR> ... </VAR></code>	використовується для позначення змінних, переданих з командами
<code><DIV> ... </DIV></code>	використовується для створення розділу в тексті

Атрибути тегу DIV

ALIGN=left ALIGN=right ALIGN=center	використовується для визначення способу вирівнювання блоку (ліворуч, праворуч, по центру)
---	---

Гіпертекстові зв'язки

<code> </code>	Створює гіперпосилання на іншу вебсторінку, інтернет ресурс або файл
<code> </code>	Створює гіперпосилання на мітку поточної сторінки
<code> </code>	Зазначає частину тексту, як позначку для гіперпосилань на сторінці
<code> </code>	Створює гіперпосилання на мітку іншої сторінки
<code> </code>	Створює гіперпосилання виклику поштової програми для написання листа за вказаною адресою
<code><A HREF = «Адреса» </code>	Гіперзв'язок по зображенню з іншою сторінкою сервера (активне зображення)

Атрибути тегу A HREF

TARGET=«_BLANK»	Відкриття документа в новому вікні
-----------------	------------------------------------

TARGET=«_TOP»	Відкриття документа в поточному вікні
TARGET=«_SELF»	Відкриття документу у поточній вкладці вікна браузера
TARGET=«_PARENT»	Відкриття документа в батьківському фреймі
TARGET=«Ім'я_фрейму»	Відкриття сторінки у зазначеному фреймі

Списки

<i>Елементи списку</i>	
aaabbbccc	для того щоб задати список, кожний елемент його позначається (можна в рядок)
<i>Атрибути тегу LI</i>	
TYPE=«CIRCLE/DISK/SQUARE»	тип мітки для цього і наступних елементів у неупорядкованому списку
TYPE=«A/a/I/i/1»	тип нумерації для цього і наступних елементів;
VALUE=«n»	вказується номер, з якого починати відлік
<i>Неупорядковані списки</i>	
 ... 	неупорядкований список
<i>Атрибути тегу UL:</i>	
<UL COMPACT> ... 	атрибут COMPACT указує, що список буде представлений у більш компактному вигляді
<UL TYPE=«CIRCLE»> ... 	атрибут TYPE=«CIRCLE» указує, що мітка буде у вигляді кружка
<UL TYPE=«DISK»> ... 	атрибут TYPE=«DISK» вказує, що мітка буде у вигляді диску (за замовчуванням)
<UL TYPE=«SQUARE»> ... 	атрибут TYPE=«SQUARE» вказує, що мітка буде у вигляді квадрата
<i>Упорядковані списки:</i>	
 ... 	упорядкований список
<i>Атрибути тегу OL:</i>	
<OL COMPACT> ... 	атрибут COMPACT вказує, що список буде представлений у більш компактному вигляді

<OL TYPE=«A»> . . . 	атрибут TYPE=«A» указує, що мітки будуть у вигляді прописних букв
<OL TYPE=«a»> . . . 	атрибут TYPE=«a» указує, що мітки будуть у вигляді малих букв
<OL TYPE=«I»> . . . 	атрибут TYPE=«I» вказує, що мітки будуть у вигляді великих римських цифр
<OL TYPE=«i»> . . . 	атрибут TYPE=«i» указує, що мітки будуть у вигляді маленьких римських цифр
<OL TYPE=«1»> . . . 	атрибут TYPE=«1» указує, що мітки будуть у вигляді арабських цифр (за замовчуванням)
<OL START=«3»> . . . 	атрибут START=«3» визначає, що список буде пронумерований із зазначеної цифри
<i>Меню</i>	
<MENU> . . . </MENU>	список у вигляді меню
<MENU COMPACT> . . . </MENU>	атрибут COMPACT указує, що список буде представлений у більш компактному вигляді
<i>Списки визначень</i>	
<DL> . . . </DL>	список визначень (словник термінів)
<DT> . . .	цей тег стоїть перед терміном;
<DD>. . .	цей тег стоїть перед визначенням зазначеного вище Терміна

Приклад використання:

<DL>

<DT>Internet

<DD>Всесвітня комп'ютерна мережа мереж, що надає доступ до інформації з усього світу.

<DT>Гіпертекст.

<DD>Система гіпертексту встановлює зв'язки між невеликими фрагментами інформації, спрощуючи пошук необхідних даних.

<DL>

Таблиці

<TABLE> . . . </TABLE>	визначає початок і кінець таблиці
<i>Атрибути тегу TABLE</i>	
BGCOLOR=« . . . »	визначає колір фону всієї таблиці
BACKGROUND=« . .	указує адресу файла для Background

. »	(рекомендується підбирати колір bgcolor, максимально наближений до кольору background)
BORDER=«n»	задає обрамлення, утворене лініями шириною n пікселів
WIDTH=«...»	задає ширину таблиці (у пікселях або відсотках);
ALIGN=left ALIGN=right ALIGN=center	задає спосіб вирівнювання таблиці по ширині сторінки (ліворуч, праворуч, по центру)
CELLSPACING=«...»	задає у пікселях відстань між границями комірки та її вмістом;
CELLPADDING=«...»	задає інтервал між комірками у пікселях
Приклад використання: < TABLE CELLSPACING=2 CELLPADDING=3 BORDER=1 ALIGN=«Center» BGCOLOR=«Gray» BACKGROUND=«Bk10.gif» WIDTH=100>	
<TR>... </TR>	визначає початок і кінець рядка в таблиці
<i>Атрибути теги TR</i>	
BGCOLOR=«...»	визначає колір фону даного рядка
BACKGROUND=«...»	указує адресу файла для Background даного рядка
BORDER=«n»	задає обрамлення, утворене лініями шириною n пікселів
ALIGN=left ALIGN=right ALIGN=center	задає спосіб вирівнювання вмісту комірок у даному рядку (ліворуч, праворуч, по центру)
VALIGN=top VALIGN=bottom VALIGN=center VALIGN=baseline	задає спосіб розміщення вмісту комірок у даному рядку по вертикалі (зверху, знизу, по центру, по базовій лінії)
Приклад використання: < TR ALIGN=«center» VALIGN=«middle»>	
<TD>... </TD>	визначає початок і кінець комірки в таблиці
<i>Атрибути теги TD</i>	
BGCOLOR=«...»	визначає колір фону даної комірки

BACKGROUND=« . . . »	указує адресу файла для Background даної комірки
WIDTH=« . . . »	задає ширину комірки (у пікселях або відсотках від ширини таблиці);
ALIGN=left ALIGN=right ALIGN=center	задає спосіб вирівнювання вмісту комірки (ліворуч, праворуч, по центру)
VALIGN=top VALIGN=bottom VALIGN=center VALIGN=baseline	задає спосіб розміщення вмісту комірки по вертикалі (зверху, знизу, по центру, по базовій лінії);
ROWSPAN=«n»	вказує, що дана комірка охоплює n сусідніх рядків
COLSPAN=«n»	указує, що дана комірка охоплює n сусідніх стовпців
NOWRAP	вказує, що у даній комірці відключається режим автоматичного розподілу тексту по всій комірці. Тобто буде відображатися лише та частина тексту, яка вміщається у комірці по довжині
Приклад використання: <TD ALIGN=«Center» VALIGN=«Baseline» COLSPAN=«2» WIDTH=«30» HEIGHT=«30» NOWRAP>	
<TH> . . . </TH>	визначає початок і кінець комірки із заголовком у таблиці
<i>Атрибути тегу TH</i>	
BGCOLOR=« . . . »	визначає колір фону даної комірки
BACKGROUND=« . . . »	указує адресу файла для Background даної комірки
WIDTH=« . . . »	задає ширину комірки (у пікселях або відсотках від ширини таблиці);
ALIGN=left ALIGN=right ALIGN=center	задає спосіб вирівнювання вмісту комірки (ліворуч, праворуч, по центру)
VALIGN=top VALIGN=bottom VALIGN=center VALIGN=baseline	задає спосіб розміщення вмісту комірки по вертикалі (зверху, знизу, по центру, по базовій лінії);
ROWSPAN=«n»	вказує, що дана комірка охоплює n сусідніх

	рядків
COLSPAN=«n»	указує, що дана комірка охоплює n сусідніх стовпців
NOWRAP	вказує, що у даній комірці відключається режим автоматичного розподілу тексту по всій комірці. Тобто буде відображатися лише та частина тексту, яка вміщається у комірці по довжині
Приклад використання: < TH ALIGN=«Center» VALIGN=«Baseline» COLSPAN=«4» WIDTH=«50»>	

Приклад взаємного розташування тегів при описі таблиці:

```
<TABLE BORDER=0>
<TR>
<TD></TD>
</TR>
</TABLE>
```

Фрейми

<FRAMESET> ... </FRAMESET>	визначає початок і кінець документа із фреймами. Використовується замість тегу BODY
<i>Атрибути тегу FRAMESET</i>	
BORDER=«n»	задає обрамлення фреймів, утворене лініями шириною n пікселів
BORDERCOLOR=« . . . »	задає колір обрамлення
COLS=« . . . , . . . , . . . »	задає горизонтальне розташування фреймів. У списку вказується ширина кожного фрейму в пікселях або відсотках. Ширину останнього фрейму можна вказати як *, тобто увесь інший простір
ROWS=« . . . , . . . , . . . »	задає вертикальне розташування фреймів. У списку вказується висота кожного фрейму в пікселях або відсотках. Висоту останнього фрейму можна вказати як *, тобто увесь інший простір
Приклад використання: <FRAMESET COLS=«179, *»>	
<FRAME>	визначає фрейм у наборі фреймів

<i>Атрибути тегу FRAME</i>	
BORDER=«n»	задає обрамлення фрейму, утвореного лінією шириною n пікселів
BORDERCOLOR=« . . . »	задає колір обрамлення
SRC=« . . . »	задає адресу (URL) файла, яка буде відображатися у цьому фреймі
NAME=« . . . »	задає ім'я цього фрейму
SCROLLING=yes SCROLLING=no SCROLLING=auto	описує наявність і тип лінії прокручування (є завжди, немає ніколи, додається при необхідності)
NORESIZE	забороняє змінювати розміри фреймів
MARGINHEIGHT=« . . . »	задає розміщення над фреймом і під ним областям вільного простору висотою по n пікселів
MARGINWIDTH=« . . . »	задає розміщення ліворуч і праворуч від фрейму областей вільного простору висотою по n пікселів
<p>Приклад використання:</p> <pre><FRAME NAME=«Framea» SRC=«Begin.htm» MARGINHEIGHT=1 MARGINWIDTH=1 SCROLLING=«Auto» NORESIZE></pre>	

Приклад взаємного розташування тегів при описі набору фреймів:

```
<FRAMESET ROWS=«200,*»>
<FRAME NAME=«Frame 1» SRC=«A1.htm»
SCROLLING=«AUTO»>
<FRAME NAME=«Frame 2» SRC=«B1.htm»
SCROLLING=«AUTO»>
</FRAMESET>
```

Відтворення звуку

Для відтворення звуку (файл *.mid) після завантаження документа HTML у браузер (тобто у фоновому режимі) потрібно записати наступну команду: `<bgsound src=«/windows/canyon.mid» loop=1>`.

Можна також використовувати файл формату *.wav. Число відтворень музики loop можна збільшити з 1 до n.

Список кольорів символів HTML

16 базових кольорів:

aqua – бірюзовий; black – чорний; blue – синій; gray – сірий; green – зелений; lime – яскраво-зелений; maroon – темно-червоний; white – білий; navy – темно-синій; olive – маслиновий; purple – фіолетовий; red – червоний; silver – яскраво-сірий; teal – яскраво-блакитний; yellow – жовтий; fuchsia – яскраво-фіолетовий.

Назва	Колір	Rgb-коди
aquamarine	Аквамарин	#7FFFD4
beige	Бежевий	#F5F5DC
black	Чорний	#000000
blue	Синій	#0000FF
brown	Коричневий	#A52A2A
chocolate	Шоколадний	#D2691E
coral	Кораловий	#FF7F50
crimson	Малиновий	#DC143C
cyan	Ціан	#00FFFF
gold	Золотий	#FFD700
gray	Сірий	#808080
green	Зелений	#00FF00
indigo	Індиго	#4B0082
khaki	Хакі	#F0E68C
magenta	Ліловий	#FF00FF
orange	Оранжевий	#FFA500
pink	Розовий	#FFC0CB
purple	Пурпурний	#800080
Red	Червоний	#FF0000
seagreen	Морської хвилі	#2E8B57
silver	Срібний	#C0C0C0
turquoise	Бірюзовий	#40E0D0
violet	Фіолетовий	#EE82EE
white	Білий	#FFFFFF
yellow	Жовтий	#FFFF00

Міняючи Rgb-коди, можна підбирати бажані кольори тексту і фону. Для базових кольорів можна використовувати приставку dark для задавання темних кольорів: darkblue, darkred і т.п.; або light – для світлих: lightgreen, lightyellow.

Форми

Практично всі сучасні браузерери дозволяють користувачеві, заповнивши спеціальну форму, виконувати деякі дії на вашому вебсервері. Коли форма інтерпретується веббраузером, створюються спеціальні екранні елементи GUI, такі як поля введення, кнопки підтвердження (check-boxes), кнопки вибору альтернатив (radio-buttons), меню, що випадають, «скроліруючі» списки, кнопки тощо. Коли користувач, заповнивши форму, натискає кнопку «Підтвердження» (SUBMIT – спеціальний тип кнопки, який задається при описі документа), інформація, введена користувачем у форму, посиляється вебсерверу для обробки й передачі іншим програмам, що працюють під сервером, відповідно до CGI (Common Gateway Interface) інтерфейсом. За допомогою мови написання сценаріїв Javascript (Vbscript) можна перевірити правильність заповнення форми до її відправлення серверу, обробити натискання інших (відмінних від SUBMIT) кнопок і багато інших подій.

При написанні сценарію використовуються мови VBScript та JavaScript (JScript). Написання сценарію здійснюється таким чином:

На мові VBScript	На мові JavaScript
<code><SCRIPT language= «VBScript» type= «text/vbscript»> ...текст сценарію на мові VBScript </SCRIPT></code>	<code><SCRIPT language=«JavaScript» type= «text/javascript»> ...текст сценарію на мові JavaScript </SCRIPT></code>

Для забезпечення сумісності з усіма браузерами необхідне використання атрибуту «language» і атрибуту «type».

Для того щоб старі браузерери, які не підтримують сценаріїв, не відображали код програми, необхідно текст сценарію узяти в тег `<!-- -->` як це показано нижче.

На мові VBScript	На мові JavaScript
<code><SCRIPT language= «VBScript» type= «text/vbscript»> <!-- 'це коментар на мові VBScript MsgBox «Привіт!»</code>	<code><SCRIPT language= «JavaScript» type= «text/javascript»> <!-- //це коментар на мові JavaScript /*Цей коментар займає</code>

--> </SCRIPT>	декілька строк*/ alert(«Привіт!»); --> </SCRIPT>
------------------	---

При створенні сторінки тег <SCRIPT> можна розміщувати у будь-якому місці сторінки. Стандарти HTML рекомендують розміщувати тег <SCRIPT> у заголовку сторінки, тобто між тегами <HEAD> та </HEAD>.

Для браузерів, які не підтримують сценаріїв, або при відключенні підтримки сценаріїв, можна скористатися тегом <noscript>.

<NOSCRIPT>

Ваш браузер не підтримує JavaScript
або його підтримка відключена

</NOSCRIPT>

Роздільником операторів у VBScript є символ «:», а в JavaScript – «;». Наприклад:

На мові VBScript	На мові JavaScript
<SCRIPT language= «VBScript» type= «text/vbscript»> MsgBox «Привіт!»: MsgBox «Ще привіт!» </SCRIPT>	<SCRIPT language= «JavaScript» type=»text/javascript»> alert(«Привіт!»); alert(«Ще привіт!»); </SCRIPT>

Якщо текст сценарію досить великий, його можна помістити в окремий текстовий файл.

Підключення файла сценарію до вебсторінки слід здійснювати за допомогою атрибуту «src» тегу <SCRIPT>.

Підключення файла сценарію «myvb.vbs», написаного на мові VBScript	Підключення файла сценарію «myjs.js», написаного на мові JavaScript
<SCRIPT src= «myvb.vbs» language= «VBScript» type= «text/vbscript»> </SCRIPT>	<SCRIPT src= «myjs.js» language= «JavaScript» type= «text/javascript»> </SCRIPT>

Слід зазначити, що в такому випадку недоречно писати текст сценарію між тегами <SCRIPT> та </SCRIPT>, тому що браузер у такому випадку проігнорує код програми.

У наведених прикладах файли із текстами сценаріїв повинні знаходитись у тій же папці, що і сторінка. Якщо, наприклад, файл знаходиться у підпапці «script», то використовується такий код:

```
<SCRIPT src= «script/myjs.js» language= «JavaScript» type=
«text/javascript»></SCRIPT>
```

Можна задати також файл, що знаходиться на іншому вузлі.

Наприклад:

```
<SCRIPT src= «http://www.host.com/myscripts.js» language=
«JavaScript» type= «text/javascript»></SCRIPT>
```

Тег FORM

Усі форми починаються тегом <FORM> і закінчуються тегом </FORM>. Між цими тегами й розташовуються елементи форми, такі як кнопки, поля введення тощо. Крім елементів форм, там можна розташовувати будь-які тексти й майже будь-які теги HTML.

Синтаксис тегу FORM:

```
<FORM
ACTION= «URL»
METHOD= «get | post»
NAME= «ім'я»
ENCTYPE=          «application/x-www-form-urlencoded
multipart/formdata»
TARGET= «вікно, за замовчуванням _self»
TITLE= «текст підказки»
LANG= «мова (ru, en, de, і т.д.)»
LANGUAGE= «Javascript | Jscript | Vbscript | VBS»
CLASS= «ім'я класу CSS»
STYLE= «безпосередній стиль»
ID= «ідентифікатор»
ONRESET= «оброблювач події»
ONSUBMIT= «оброблювач події»
.....
Елементи форми й інші елементи HTML
.....
</FORM>
```

Жоден з перерахованих параметрів тегу FORM не є обов'язковим. Розглянемо параметри:

ACTION

Указує URL, куди слід передати інформацію, занесену у форму. Звичайно, це адреса CGI програми або e-mail, оформлений через mailto:. Однак досить часто можна зустріти й інші речі, типу javascript:0, коли форма обробляється сценарієм. Якщо ACTION не зазначений, то інформація передається тому документу, у якому перебуває форма.

METHOD

При використанні методу GET (за замовчуванням), інформація з форми додається у кінець URL, який був зазначений в ACTION. Оскільки максимальна довжина URL обмежена, цей метод варто використовувати тільки для найпростіших форм. Метод POST кращий, тому що даний метод передає всю інформацію про форму негайно після звертання до зазначеного URL. CGI-програма одержує дані із форми в стандартний потік вводу.

NAME

Задає ім'я форми для використання у програмах-сценаріях.

ENCTYPE

Завжди слід використовувати значення за замовчуванням (просто опустити цей параметр), крім тих випадків, коли потрібно передати файл від клієнта до сервера.

TARGET

Також як і в тегу <A...>, указує вікно, в яке потрапить уся вихідна інформація форми. За замовчуванням – у те ж вікно, де й сама форма.

TITLE

Підказка, яка буде з'являтися, коли мишка вказує на форму (MSIE 4.0+).

LANG

Мова – «en» – англійська, «de» – німецька і т.д.

LANGUAGE

Мова, яка використовується для інтерпретації елементів сценарію для даної форми (таких, як оброблювачі подій).

CLASS

Ім'я CSS-стилю.

STYLE

Пряме включення стилю.

ID

Ідентифікатор, за яким можна посилатися на форму з мови написання сценаріїв тощо.

Елементи форми: TEXTAREA.

Тег <TEXTAREA> використовується для того, щоб дозволити користувачеві вводити більше одного рядка інформації (вільний текст). Область введення починається тегом <TEXTAREA> і закінчується тегом </TEXTAREA>.

Параметри тут також не є обов'язковими, але все-таки слід указати хоча б NAME, COLS і ROWS.

Параметри:

ROWS

Кількість рядків у полі.

COLS

Кількість стовпців у полі.

WRAP

Чи розривати довгі рядки. OFF – не розривати. VIRTUAL – розривати на екрані, але всередині, залишати їх цілими. PHYSICAL – «чесно» розривати.

ACCESSKEY

Задає клавішу, яку потрібно натиснути разом з Alt, щоб відразу потрапити в це поле.

ALIGN

Задає вирівнювання елемента щодо навколишнього тексту.

TABINDEX

Задає номер даного елемента в послідовності проходження елементів форми за допомогою клавіші Tab.

DISABLED

Поле не одержує користувацького введення й ні на що не реагує. Служить тільки для показу початкової (за замовчуванням) інформації.

READONLY

Схоже на попереднє, але поле одержує фокус і в ньому може перебувати курсор. Але його заборонено редагувати.

Інші параметри аналогічні відповідним параметрам тегу FORM.

Приклад використання тегу <TEXTAREA>:

<TEXTAREA NAME= «addr» ROWS=6 COLS=32>

Уманський державний
педагогічний університет
імені Павла Тичини

Кафедра інформатики і ІКТ

</TEXTAREA>

Зверніть увагу, що усередині цього тегу усі переклади рядків викликають переклади рядків і при виведенні на екран. Для деяких браузерів (наприклад, MSIE 3.0) це правильно й для першого рядка. Тег </TEXTAREA> необхідний навіть тоді, коли поле введення спочатку порожнє.

Елементи форми: SELECT.

Тег <SELECT> використовується для організації вибору з фіксованого числа заздалегідь певних значень. За допомогою цього тегу можна представити список, список з можливістю вибрати більш, ніж один варіант і спадаюче меню.

Як завжди, параметри необов'язкові. Майже всі параметри мають такий зміст, як раніше. Ми розглянемо тільки нові:

SIZE

Кількість елементів списку видимих на екрані. Якщо цей параметр більше 1, то маємо справу зі списком, інакше – зі спадаючим меню.

MULTIPLE

Якщо зазначене, то користувач може вибирати кілька елементів списку одночасно.

SINGLE

Якщо зазначене, то користувач не може вибирати кілька елементів списку одночасно. Якщо не зазначене ні SINGLE, ні MULTIPLE, то мається на увазі SINGLE.

Елемент <OPTION>, згаданий при описі синтаксису, задає один елемент списку. Текст, що знаходиться між <OPTION ...> і </OPTION>, буде показаний користувачеві як значення даного елемента списку. Якщо зазначений параметр SELECTED, то цей елемент буде спочатку «обраний». Інші параметри використовуються досить рідко. Параметр VALUE задає значення для даного поля «за замовчуванням».

Елементи форми: INPUT

Тег <INPUT> визначається його параметром TYPE.

TYPE

Визначає, що являє собою даний елемент. Можливі такі значення:

BUTTON – кнопка. Має сенс тільки, якщо ви обробляєте її натискання з програми сценарію.

<input type="button" value="Button"/>

CHECKBOX – елемент, який можна включити й виключити.	<input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
HIDDEN – Схований елемент. Його не видно, але він має значення. У такий спосіб часто передають параметри CGI програмі.	
IMAGE – це не просте зображення. Воно працює як кнопка SUBMIT, тобто, якщо навести на зображення мишу й нажати кнопку, інформація форми негайно передається за адресою зазначеному в ACTION. При цьому передаються й координати того місця, де була миша.	
PASSWORD – поле введення пароля, символи, що вводяться, не відображаються.	Password: <input type="password"/>
RADIO – радіокнопка. Її, звичайно, поєднують у групи (група – кілька кнопок з тим самим NAME). У включеному стані завжди перебуває тільки одна із кнопок групи.	<input type="radio"/> <input checked="" type="radio"/> <input type="radio"/>
RESET – кнопка скидання. Очищає форму.	<input type="button" value="Скинути"/>
SUBMIT – кнопка, при натисканні на яку інформація з форми передається за адресою, зазначеною в ACTION, для обробки.	<input type="button" value="Надіслати"/>
TEXT – просте поле для введення одного рядка тексту	<input type="text"/>
FILE – дає можливість користувачеві вибрати файл, що перебуває на його локальному диску. Надалі цей файл можна переслати на сервер.	

SIZE

Задає розмір видимого поля там, де це має сенс. Наприклад, для полів типу TEXT, PASSWORD, FILE тощо.

MAXLENGTH

Задає максимальний розмір поля там, де це має сенс. Визначає кількість символів, яку користувачі можуть ввести в поле введення. При перевищенні кількості припустимих символів браузер реагує на спробу введення нового символу звуковим сигналом і не дає його ввести. Не плутати з атрибутом SIZE. Якщо MAXLENGTH більше ніж SIZE, то в полі здійснюється скролінг.

За замовчуванням значення MAXLENGTH дорівнює нескінченності.

SRC

Для типу IMAGE задає URL файла зображення.

BORDER

Для типу IMAGE задає розмір бордюру навколо зображення.

CHECKED

Для типів CHECKBOX і RADIO вказує, що даний елемент з самого початку «включений».

VALUE

Значення елемента. Звичайно, це текст на кнопці або початковий текст у полі введення.

ALIGN

Для типу IMAGE повністю аналогічно ALIGN у тегу .

Відправлення файлів на сервер

Форми можна також використовувати для відправлення файлів на сервер. Усе, що потрібно від форми, це:

– вказати ENCTYPE=«multipart/form-data» у тегу FORM;

– вказати METHOD=«POST» у тегу FORM;

– використовувати <INPUT TYPE=«FILE» ...> для введення імені файла.

Наприклад:

```
<FORM ACTION=«myURL»
```

```
ENCTYPE=«multipart/form-data»
```

```
METHOD=«POST»>
```

```
Оберіть файл:<BR>
```

```
<INPUT NAME=«fn» «TYPE=«FILE»><P>
```

```
<INPUT TYPE=«SUBMIT» VALUE=«Надіслати»>
```

```
</FORM> [15]
```

Питання для самоконтролю

1. Що таке гіпертекст?
2. Чим гіпертекст відрізняється від звичайного тексту?
3. Які програмні продукти реалізують технологію гіпертексту?
4. Для чого потрібні браузерери?
5. Назвіть поширені браузерери.
6. Що таке «домашня сторінка»?
7. Як налаштувати домашню сторінку?
8. Що дозволяють робити команди навігації?
9. Які є команди навігації?
10. Що таке HTML?
11. Коли почала розвиватися HTML?
12. Що таке тег?
13. Яке розширення мають текстові документи, що містять код мовою HTML?
14. З чого складається тег?
15. Що визначають атрибути тегів?
16. Чи можна використовувати коментарі в HTML-документах?
17. Назвіть поширені HTML-редактори.
18. Що таке скрипт?
19. Які теги для створення таблиці Ви знаєте?
20. Які типи списків можна задавати в HTML?
21. Які основні теги задають структуру HTML-документа?
22. Які атрибути використовуються для тега BODY?
23. Чи можна додавати до HTML-документів зображення?
24. Яким тегом задають гіперпосилання в HTML-документі?
25. Що таке фрейми?
26. Чи можна розмістити на вебсторінці декілька фреймів?
27. Як задати наявність смуг прокручування у фреймі?
28. Який тег задає початок і кінець HTML-документа?
29. Які теги переважають у мові HTML?
30. Для чого використовують скрипти при створенні вебсторінок.

Тема 8. Основи програмування мовою РНР

РНР – мова написання скриптів, які вбудовуються безпосередньо в гіпертекстові файли й виконуються на вебсервері [15, с. 83].

РНР розшифровується як Hypertext PreProcessor. Що це означає? Почнемо небагато здалека: є два типи мов. Один тип називається "клієнтським", а інший - "серверним".

Це означає, що клієнтські мови працюють в браузері кожної конкретної людини. Типовим представником клієнтських мов є JavaScript. Усі дії і команди, які ми задаємо на цій мові, виконуються браузером (для спрощення тут ми не розглядаємо серверний JavaScript). Один і той же код, написаний нами, обробляється в одному випадку браузером Microsoft Edge, в іншому – Firefox, в третьому – Opera, в четвертому – Google Chrome, тобто тим оглядачем, який використовує кожна конкретна людина для перегляду нашої сторінки. Браузер отримує альтернативну назву – клієнт.

У випадку з серверними мовами (до яких і відноситься РНР) ми спостерігаємо іншу картину. Створений нами сайт завжди розташований на якому-небудь сервері і усі команди і скрипти, написані на мові РНР, виконуються саме на стороні сервера. Після того, як РНР-скрипт виконається, сервер "віддає" результат своєї роботи браузеру, який перетворить цей результат в зручний для користувача вид. Тут важливо розуміти наступний момент: за початковим кодом вебсторінки, який можна подивитися в будь-якому браузері через опцію аналогічну "Початковий код сторінки" не можна визначити, чи використовувалася мова РНР для створення цієї сторінки, чи ні.

Неможливо це зробити якраз тому, що РНР-скрипти обробляються на сервері, а в браузер передається вже готовий, оброблений варіант, по суті, тільки HTML-код.

РНР прекрасно працює в парі з HTML. У HTML-код можна робити вставки РНР-коду, а за допомогою РНР виводити HTML-розмітку. Важливе наступне: не має значення, наскільки складний РНР-код, в звичайно підсумку "на виході" він стане звичайним HTML.

HTML на 100% статичний. Вбудовувавши ж у вебсторінки РНР-код, ми можемо добитися того, щоб вміст однієї і тієї ж сторінки був

різний залежно від певних умов. Такі сторінки називаються динамічними.

За довгі роки свого існування мова PHP зарекомендувала себе як прекрасне рішення для створення динамічних вебсайтів.

Для повноцінної роботи з PHP на комп'ютері користувача має бути встановлене наступне програмне забезпечення:

1. Вебсервер.
2. Система Управління Базами Даних (СУБД) MySQL (у базі даних зберігається наповнення сайту).
3. Інтерпретатор PHP.
4. Текстовий редактор, в якому записується код.
5. Браузер.

Вебсервер призначений для імітації на Вашому комп'ютері того самого сервера, на якому потім розміщуватиметься Ваш сайт вже на хостингу в мережі Інтернет. Це треба для того, щоб у себе на комп'ютері Ви могли писати будь-які PHP-скрипти і дивитися, як вони працюють, вносити до них зміни і правки.

СУБД MySQL потрібна для зберігання інформації, яка буде на Вашому сайті. У випадку з HTML-сторінками увесь вміст сайту знаходиться безпосередньо в них. Кожна сторінка містить певний об'єм інформації (контенту).

При використанні PHP для цілей зберігання корисного інформаційного наповнення сайту використовується, як правило, база даних. У переважній більшості випадків це MySQL.

Інтерпретатор PHP – це своєрідна програма, яка здійснює обробку PHP-коду на вебсервері. Без нього ми ніяк не зможемо виконати наші PHP-скрипти і побачити результат їх роботи.

Для того, щоб повідомити вебсервер про те, що має бути здійснена обробка PHP-коду, необхідно використовувати наступний синтаксис при додаванні PHP в HTML-документ:

```
<?php  
.....тут йде PHP-код  
?>
```

В цьому випадку вебсервер, дійшовши до відкриваючого тега PHP (<?php) передасть управління PHP-інтерпретатору. Після ж закриття блоку PHP-коду (?>) інтерпретатор PHP припиняє свою роботу.

Простий статичний сайт, написаний на мові HTML, може складатися з ряду сторінок і є просто набором незмінних файлів. На

такому сайті немає можливості створити форму зворотного зв'язку, коментарі, реєстрацію користувачів, пошук. У такого сайту не може бути панелі управління, з якої ми могли б додавати нові сторінки, або редагувати і видаляти вже існуючі.

При використанні тільки мови HTML розробник сильно обмежений в створенні зручних і функціональних сайтів, оскільки не має ніяких "важелів управління" кожною конкретною сторінкою. Усе, що можливо - це вручну відкрити окрему HTML-сторінку в редакторі коду і якось її модифікувати.

Використання мови PHP дозволяє створювати потужні сайти завдяки тому, що є можливість продумувати і реалізовувати "поведінку" створюваного сайту, тобто можна зробити так, щоб сайт поведився по-різному в різних ситуаціях, діючи по прописаних алгоритмах залежно від виконання або невиконання певних умов. Можливість завдання потрібної логіки "поведінки" сайту є однією з найважливіших і найбільш цінних можливостей в мові PHP.

PHP надає безліч інструментів для роботи з даними різних типів. Найчастіше доводиться мати справу з текстовими даними і даними, представленими у вигляді масивів. Для полегшення цих завдань в PHP передбачені численні функції, що дозволяють з легкістю маніпулювати будь-якою інформацією.

Окрім цього, мова PHP дозволяє працювати з файлами і теками: створювати, редагувати і видаляти, переміщати і перейменовувати їх.

Для організації зручнішої взаємодії користувача з сайтом, PHP відкриває можливості:

- створення форм зворотного зв'язку;
- коментування;
- реєстрації користувачів;
- пошуку по сайту;
- запам'ятовування введеної користувачем інформації.

PHP активно використовується для створення найрізноманітніших панелей адміністратора, які дають можливість просто і зручно керувати своїм сайтом.

Додавання нових матеріалів, редагування категорій, модерація коментарів, зміна пунктів меню сайту, управління налаштуваннями – усе це і багато іншого можна реалізувати засобами PHP [14].

Арифметичні оператори

Таблиця 8.1. Арифметичні оператори

Приклад	Назва	Результат
$a + b$	Додавання	Сума a і b
$a - b$	Віднімання	Віднімає b від a
$a * b$	Множення	Добуток a і b
a / b	Ділення	Ділення a на b
$a \% b$	Залишок ділення	Залишок від ділення a на b

Оператор ділення ($\langle\langle/\rangle\rangle$) повертає цілу величину, якщо обидва оператори – цілі (або рядок перетворено в ціле). Якщо кожний операнд є величиною із плаваючою комою, виконується розподіл із плаваючою комою.

Оператори рядків

Насправді є тільки один оператор – оператор конкатенації ($\langle\langle.\rangle\rangle$).

$a = \langle\langle\text{Hello} \rangle\rangle$;

$b = a . \langle\langle\text{World!}\rangle\rangle$; // тепер $b = \langle\langle\text{Hello World!}\rangle\rangle$

Оператори присвоювання

Основним оператором присвоювання є $\langle\langle=\rangle\rangle$. Це означає, що лівий операнд одержує значення виразу з права (збірне присвоювання).

Значенням виразу присвоювання є величина, що привласнюється. Тому, якщо $\langle\langle a = 3 \rangle\rangle$, то це 3.

Крім основного оператора присвоювання ще є додаткові «комбінаційні оператори», для всіх арифметичних і строкових операторів, що дозволяє використовувати значення у виразах і потім установлювати своє значення у результаті цього виразу.

Наприклад:

$a = 3$;

$a += 5$; // тепер a дорівнює 8, нібито сказали: $a = a + 5$;

$b = \langle\langle\text{Hello} \rangle\rangle$;

$b .= \langle\langle\text{There!}\rangle\rangle$; // тепер b дорівнює $\langle\langle\text{Hello There!}\rangle\rangle$, неначе написали $b = b . \langle\langle\text{There!}\rangle\rangle$;

Бітові оператори

Бітові оператори дозволяють вам змінювати біти в цілих числах (таблиця 8.2).

Таблиця 8.2. Бітові оператори

Приклад	Назва	Результат
$\$a \& \b	I	Будуть установлені біти, які були встановлені і в $\$a$ і в $\$b$. *Приклад $\$a=5; /* 0101 */$ $\$b=12; /* 1100 */$ $\$c=\$a \& \$b; /* \c буде дорівнювати 4 (0100) */
$\$a \b	Або	Будуть установлені біти, установлені в $\$a$ або $\$b$. *Приклад: $\$a=5; /* 0101 */$ $\$b=12; /* 1100 */$ $\$c=\$a \$b; /* \c буде (1101) */
$\sim \$a$	Не	Будуть установлені не_присутні в $\$a$ біти (реверс) *Приклад: $\$a=5; /* 0101 */$ $\sim \$a /* \a буде рівно x (1010) */
$\$a \wedge \b	Xor	Установлюються біти, які встановлені в $\$a$ або $\$b$, але не в обох
$\$a \ll \b	Зсув вліво	Зрушує біти змінної $\$a$ на $\$b$ кроків уліво (кожний крок/зсув означає «помножити на 2»)
$\$a \gg \b	Зсув вправо	Зрушує біти змінної $\$a$ на $\$b$ кроків вправо (кожний крок/зсув означає «розділити на 2»)

Логічні оператори

Таблиця 8.3. Логічні оператори

Приклад	Назва	Результат
$\$a \text{ and } \b	AND	TRUE, якщо і $\$a$ і $\$b$ TRUE
$\$a \text{ or } \b	OR	TRUE, якщо $\$a$ або $\$b$ TRUE
$\$a \text{ xor } \b	XOR	TRUE, якщо $\$a$ або $\$b$ TRUE, але не обоє
$! \$a$	NOT	TRUE, якщо $\$a$ не TRUE
$\$a \&\& \b	AND	TRUE, якщо і $\$a$ і $\$b$ TRUE
$\$a \b	OR	TRUE, якщо $\$a$ або $\$b$ TRUE

Різниця у двох різних варіантах операторів «and» і «or» – у відмінності пріоритетів операцій.

Оператори порівняння

Оператори порівняння дозволяють вам порівнювати дві величини.

Таблиця 8.4. Оператори порівняння

Приклад	Назва	Результат
<code>\$a == \$b</code>	Дорівнює	TRUE, якщо \$a еквівалентно \$b
<code>\$a != \$b</code>	Не дорівнює	TRUE, якщо \$a не дорівнює \$b
<code>\$a <> \$b</code>	Не дорівнює	TRUE, якщо \$a не дорівнює \$b
<code>\$a < \$b</code>	Менше	TRUE, якщо \$a строго менше ніж \$b
<code>\$a > \$b</code>	Більше	TRUE, якщо \$a строго більше \$b
<code>\$a <= \$b</code>	Менше або дорівнює	TRUE, якщо \$a менше або дорівнює \$b
<code>\$a >= \$b</code>	Більше або дорівнює	TRUE, якщо \$a більше або дорівнює \$b

Умовні оператори

IF

Структура IF – це одна з найважливіших можливостей багатьох мов, включаючи PHP. Вона дозволяє організувати виконання фрагментів коду за умовою. Можливості PHP щодо використання виразу IF схожі на C:

```
if (expr) statement
```

Тут обчислюється логічний результат «expr». Якщо expr рівно TRUE, то PHP виконає «statement», а якщо FALSE – проігнорує. Наступний приклад виведе фразу ‘a is bigger than b’ якщо \$a більше \$b:

```
if ($a > $b)
    print «a is bigger than b»;
```

Найчастіше потрібно виконати більше ніж один вираз за умовою. Звичайно, не треба оточувати кожний вираз конструкцією IF. Замість цього можна згрупувати кілька виразів у блок виразів. Наприклад, такий код не тільки виведе фразу, але і привласнить значення \$a змінній \$b:

```
if ($a > $b) {print «a is bigger than b»; $b = $a;}
```

Вираз IF може мати необмежений ступінь вкладеності в інші вирази IF, що дозволяє вам ефективно використовувати виконання за умовою різних частин програми.

ELSE

Найчастіше потрібно виконати один вираз, якщо дотримується яка-небудь умова, і інший вираз – якщо не дотримується. Для цього застосовується ELSE. ELSE розширює можливості IF по частині обробки варіантів виразів, коли воно дорівнює FALSE.

Даний приклад виведе фразу ‘a is bigger than b’, якщо \$a більше \$b, і ‘a is NOT bigger than b’ – якщо ні:

```
if ($a > $b) {  
    print «a is bigger than b»;  
} else {  
    print «a is NOT bigger than b»;  
}
```

Вираз ELSE виконується тільки, якщо вираз IF дорівнює FALSE, а якщо є конструкції ELSEIF – то якщо і вони також дорівнюють FALSE.

ELSEIF

ELSEIF є комбінацією IF і ELSE, ELSEIF. Як і ELSE дозволяє виконати вираз, якщо значення IF дорівнює FALSE, але на відміну від ELSE воно виконається тільки, якщо вираження ELSEIF дорівнює TRUE. Наприклад такий код виведе ‘a is bigger than b’, якщо \$a>\$b, ‘a is equal to b’, якщо, \$a==\$b, і ‘a is smaller than b’ – якщо \$a<\$b:

```
if ($a > $b) {  
    print «a is bigger than b»;  
} elseif ($a == $b) {  
    print «a is equal to b»;  
} else {  
    print «a is smaller than b»;  
}
```

Усередині одного виразу IF може бути декілька ELSEIF. Перший вираз ELSEIF, який буде дорівнюватиме TRUE, буде виконано. У PHP ви можете написати ‘else if’ (два слова), що буде означати те ж саме, що й ‘elseif’ (одне слово).

Вираз ELSEIF буде виконано, тільки якщо вираз IF і всі попередні ELSEIF дорівнюють FALSE, а даний ELSEIF дорівнює TRUE.

Інший синтаксис для оператора IF: IF(): ... ENDIF;

PHP пропонує інший шлях для групування операторів з оператором IF. Найбільш часто це використовується, коли ви впроваджуєте блоки HTML всередину оператора IF, але взагалі може використовуватися де завгодно. Замість використання фігурних дужок за «IF(вираження)» повинна стояти двокрапка, один або кілька виразів і завершальний ENDIF. Розгляньте такий приклад:

```
<?php if ($a==5): ?> A = 5 <?php endif; ?>
```

У цьому прикладі блок «A = 5» впроваджений всередині виразу IF, що використовується альтернативним способом. Блок HTML буде видний, тільки якщо \$a дорівнює 5.

Цей альтернативний синтаксис застосуємо і до ELSE і ELSEIF (expr). Приклад подібної структури:

```
if ($a == 5):  
    print «a equals 5»;  
    print «...»;  
elseif ($a == 6):  
    print «a equals 6»;  
    print «!!!»;  
else:  
    print «a is neither 5 nor 6»;  
endif;
```

SWITCH

Оператор SWITCH схожий на групу операторів IF з однаковим виразом. У багатьох випадках вам потрібно порівняти змінну (або вираз) з багатьма різними значеннями і виконати різні фрагменти коду залежно від того, чому буде дорівнювати значення виразу. Це саме те, для чого призначено оператор SWITCH.

Наступні 2 приклади – це 2 різних шляхи для досягнення однієї мети, але один використовує серію операторів IF, а інший – оператор SWITCH.

```
/* приклад 1 */  
if ($i == 0) {  
    print «i equals 0»;  
}  
if ($i == 1) {  
    print «i equals 1»;  
}  
if ($i == 2) {  
    print «i equals 2»;
```



```

}
/* приклад 2 */
switch ($i) {
case 0:
print «i equals 0»;
break;
case 1:
print «i equals 1»;
break;
case 2:
print «i equals 2»;
break;
}

```

Цикли **WHILE**

Цикл WHILE – найпростіший тип циклу в PHP. Основна форма оператора WHILE: `WHILE(expr) statement`

Зміст оператора WHILE простий. Він пропонує PHP виконувати вкладений оператор доти, доки `expr` дорівнює `TRUE`. Значення виразу перевіряється щоразу на початку циклу, тому, якщо значення виразу зміниться усередині циклу, то він не перерветься до кінця поточної ітерації (виконання всього блоку вкладених операторів це одна ітерація). Іноді, якщо значення `expr` дорівнює `FALSE` із самого початку, цикл не виконується жодного разу.

У IF ви можете згрупувати трохи операторів усередині фігурних дужок або використовувати альтернативний синтаксис:

`WHILE(expr): вираз... ENDWHILE;`

Наступні приклади ідентичні, вони виводять числа від 1 до 10:

```
/* example 1 */
```

```
$i = 1;
while ($i <= 10) {
print $i++; }

```

```
/* example 2 */
```

```
$i = 1;
while ($i <= 10):
print $i;
$i++;
endwhile;

```

DO..WHILE

Цикл DO..WHILE досить схожий на WHILE за винятком того, що значення логічного виразу перевіряється не до, а після закінчення ітерації. Основна відмінність у тому, що DO..WHILE гарантовано виконається хоча б один раз, що у випадку WHILE не обов'язково.

Для циклів DO..WHILE існує тільки один вид синтаксису:

```
$i = 0;
do {
print $i;
} while ($i>0);
```

Цей цикл виконається один раз, оскільки після закінчення ітерації буде перевірене значення логічного виразу, а воно дорівнює FALSE (\$i не більше 0), і виконання циклу завершиться.

Досвідчені програмісти на С знайомі з іншим використанням DO..WHILE, що дозволяє припинити виконання блоку операторів у середині шляхом впровадження його в цикл DO..WHILE(0) і використання оператора BREAK. цей код демонструє таку можливість:

```
do {
if ($i < 5) {
print «i is not big enough»;
break;
}
$i *= $factor;
if ($i < $minimum_limit) {
break;
}
print «i is ok»;
...process i...
} while(0);
```

FOR

Цикли FOR – найбільш потужні цикли в PHP. Синтаксис циклу FOR:

```
FOR (expr1; expr2; expr3) statement
```

Перший вираз (expr1), безумовно, обраховується (виконується) на початку циклу.

На початку кожної ітерації обчислюється expr2. Якщо воно дорівнює TRUE, то цикл триває і виконуються вкладені оператори. Якщо воно дорівнює FALSE, то цикл закінчується.

Наприкінці кожної ітерації обчислюється (виконується) `expr3`. Кожен із цих виразів може бути порожнім. Якщо `expr2` порожній, то цикл триває нескінченно (PHP за замовчуванням вважає його рівним TRUE, як і C).

Розглянемо такі приклади. Усі вони виводять номери з 1 по 10:

```
/* приклад 1 */
for ($i = 1; $i <= 10; $i++) {
    print $i;
}
/* приклад 2 */
for ($i = 1;;$i++) {
    if ($i > 10) {
        break;
    }
    print $i;
}
/* приклад 3 */
$i = 1;
for (;;) {
    if ($i > 10) {
        break;
    }
    print $i;
    $i++;
}
/* приклад 4 */
for ($i = 1; $i <= 10; print $i, $i++) ;
```

Звичайно, перший варіант видається кращим (або четвертий), але можливість використання порожніх виразів у циклі FOR найчастіше виявляється корисною.

PHP також підтримує альтернативний синтаксис FOR:

```
FOR (expr1; expr2; expr3): вирази; ...; endfor;
```

Інші мови використовують оператор `foreach` для того, щоб обробляти масиви або списки. PHP використовує для цього оператор `while` і функції `list()` і `each()` .

BREAK

BREAK перериває виконання поточного циклу.

```
$i = 0;
while ($i < 10) {
```

```

if ($arr[$i] == «stop») {
break;
}
$i++;
}

```

CONTINUE

CONTINUE переходить на початок найближчого циклу.

```

while (list($key,$value) = each($arr)) {
if ($key % 2) { // skip even members
continue;
}
do_something_odd ($value);
}

```

Математичні функції довільної точності

Математичні функції довільної точності: `bcadd`, `bccomp`, `bcdiv`, `bcmod`, `bcmul`, `bcpow`, `bcscale`, `bcsqrt`, `bcsub`.

Дані функції задіяні тільки за умови, що PHP був скомпільований у режимі `--enable-bcmath`, тобто при включених у конфігурацію `bcmath` функціях.

bcadd

`bcadd` – додавання двох чисел довільної точності.

Опис

`string bcadd (string лівий операнд, string правий операнд, int [масштаб]);`

Додає лівий операнд до правого операнда і повертає суму типу `string` (строкова змінна). Факультативний параметр `масштаб` використовується, щоб установити кількість розрядів після десяткової оцінки в результаті.

bccomp

`bccomp` – порівняння двох чисел довільної точності.

Опис

`int bccomp (string лівий операнд, string правий операнд, int [масштаб]).`

Порівнює лівий операнд із правим операндом і повертає результат типу `integer` (ціле). Факультативний параметр `масштаб` використовується для установки кількості цифр після десяткової оцінки, що використовуються при порівнянні. При рівності двох операндів повертається значення 0. Якщо лівий операнд більший

правого операнда, повертається +1, і якщо лівий операнд менший правого операнда – повертається –1.

bcdiv

bcdiv – операція розподілу для двох чисел довільної точності.

Опис

string bcdiv (string лівий операнд, string правий операнд, int [масштаб]).

Ділить лівий операнд на правий операнд і повертає результат. Факультативний параметр масштаб установлює кількість цифр після десяткової оцінки в результаті.

bcmod

bcmod – отримання модуля числа довільної точності.

Опис

string bcmod (string лівий операнд, string модуль).

Одержання модуля лівого операнда, використовуючи операнд модуль.

bcmul

bcmul – операція множення для двох чисел довільної точності.

Опис

string bcmul (string лівий операнд, string правий операнд, int [масштаб]).

Множить лівий операнд на правий операнд і повертає результат. Факультативний параметр масштаб установлює кількість цифр після десяткової оцінки в результаті.

bcpow

bcpow – зведення одного числа довільної точності в степінь іншого.

Опис

string bcpow (string x, string y, int [масштаб]).

Зведення x у степінь y. Параметр масштаб може використовуватися для установки кількості цифр після десяткової оцінки в результаті.

bcscale

bcscale – встановлює масштаб за замовчуванням для всіх математичних функцій.

Опис

string bcscale (int масштаб).

Ця функція встановлює заданий за замовчуванням параметр масштабу для всіх наступних математичних функцій, які явно не визначають параметр масштабу.

bcsqrt

bcsqrt – одержання квадратного кореня числа довільної точності.

Опис

string bcsqrt (string операнд, int масштаб).

Повертає квадратний корінь операнда. Факультативний параметр масштаб установлює кількість цифр після десяткової оцінки в результаті.

bcsub

bcsub – віднімає одне число довільної точності від іншого.

Опис

string bcsub (string лівий операнд, string правий операнд, int [масштаб]).

Віднімає правий операнд із лівого операнда й повертає результат типу string. Факультативний параметр масштаб установлює кількість цифр після десяткової оцінки в результаті.

Функції Дати/Часу

Функції Дати/Часу: checkdate, date, strftime, getdate, gmdate, mktime, gmmktime, time, microtime.

checkdate

checkdate – перевіряє правильність дати/часу.

Опис

int checkdate(int month, int day, int year).

Повертає true, якщо дана дата правильна, інакше false. Перевіряє правильність дати, заданої аргументами. Дата вважається правильною, якщо:

- рік між 1 900 і 32 767 включно;
- місяць між 1 і 12 включно;
- день знаходиться в діапазоні дозволених днів даного місяця.

Високосний рік враховується.

date

date – формат локального часу/дати.

Опис

string date (string format, int timestamp).

Повертає рядок, відформатований згідно з даним рядком і використовуючи дану часову мітку або поточний локальний час,

якщо не задана тимчасова мітка. У форматному рядку повинні використовуватися такі символи:

- a – «am» або «pm»;
- A – «AM» або «PM»;
- d – день місяця, цифровий, 2 цифри (на першому місці нуль);
- D – день тижня, текстовий, 3 букви; тобто «Fri»;
- F – місяць, текстовий, довгий; тобто «January»;
- h – година, цифровий, 12-часовий формат;
- H – година, цифровий, 24-часовий формат;
- i – хвилини, цифровий;
- j – день місяця, цифровий, без початкових нулів;
- l (рядкова 'L') – день тижня, текстовий, довгий; тобто «Friday»;
- m – місяць, цифровий;
- M – місяць, текстовий, 3 букви; т.е. «Jan»;
- s – секунди, цифровий;
- S – англійський порядковий суфікс, текстовий, 2 символи; тобто «th», «nd»;
- U – секунди з початку століття;
- Y – рік, цифровий, 4 цифри;
- w – день тижня, цифровий, 0 означає неділю;
- y – рік, цифровий, 2 цифри;
- z – день року, цифровий; тобто «299».

Нерозпізнані символи у форматного рядка будуть друкуватися як є.

Приклад функції date()

```
print(date(«l dS of F Y h:i:s A»));
```

```
print(«July 1, 2000 is on a « . date(«l», mktime(0,0,0,7,1,2000))»).
```

Функції date () і mktime () можливо використовувати разом для того, щоб знайти дати в майбутньому або минулому.

Приклад функцій date() і mktime()

```
$tomorrow = mktime(0,0,0,date(«m»),date(«d»)+1,date(«Y»));
```

```
$lastmonth = mktime(0,0,0,date(«m»)-1,date(«d»), date(«Y»));
```

```
$nextyear = mktime(0,0,0,date(«m»), date(«d»), date(«Y»)+1).
```

Для того, щоб відформатувати дати на інших мовах, потрібно використовувати функції setlocale() і strftime().

strftime

strftime – форматує локальний час згідно з налагодженням locale.

Опис

string strftime (string format, int timestamp).

Повертає рядок, відформатований згідно з даним форматним рядом і використовуючи дану часову мітку або поточний локальний час, якщо позначка не задана. Назви місяців і тижнів та інші, залежні від мови рядки, залежать від поточного locale, що встановлюється за допомогою setlocale().

У форматному рядку слід використовувати такі специфікатори перетворень:

- %a – скорочена назва дня тижня згідно з поточним locale;
- %A – повна назва дня тижня згідно з поточним locale;
- %b – скорочена назва місяця згідно з поточним locale;
- %B – повна назва місяця згідно з поточним locale;
- %c – переважне уявлення дати і часу для поточного locale;
- %d – день місяця як десяткове число (у діапазоні від 0 до 31);
- %H – час як десяткове число в 24-годинному форматі (в діапазоні від 00 до 23);
- %I – час як десяткове число в 12-годинному форматі (в діапазоні від 01 до 12);
- %j – день року як десяткове число (у діапазоні від 001 до 366);
- %m – місяць, день року як десяткове число (у діапазоні від 1 до 12);
- %M – хвилини як десяткове число;
- %p – `am` або `pm` згідно з поточним часом або відповідні рядки для поточного locale;
- %S – секунди як десяткове число;
- %U – номер тижня поточного року як десяткове число, починаючи з першої неділі як перший день першого тижня;
- %W – номер тижня поточного року як десяткове число, починаючи з першого понеділка як перший день першого тижня;
- %w – день тижня як ціле число, неділя – 0-й день;
- %x – бажане уявлення дати для поточного locale, що не включає час;
- %X – бажане представлення часу для поточного locale, що не включає дату;
- %y – рік як десяткове число без сторіччя (у діапазоні від 00 до 99);
- %Y – рік як десяткове число, включаючи сторіччя;
- %Z – тимчасова зона чи назва або скорочення;
- %% – символ `%`.


```

Приклад функції strftime()
setlocale («LC_TIME», «C»);
print(strftime(«%A in Finnish is «));
setlocale («LC_TIME», «fi»);
print(strftime(«%A, in French «));
setlocale («LC_TIME», «fr»);
print(strftime(«%A and in German»));
setlocale («LC_TIME», «de»);
print(strftime(«%A.\n»));

```

Приклад буде працювати, якщо у вас установлені відповідні locale.

getdate

getdate – одержує інформацію про дату/час.

Опис

```
array getdate (int timestamp).
```

Повертає асоціативний масив, що містить інформацію про дату з такими елементами:

- «seconds» – секунди;
- «minutes» – хвилини;
- «hours» – годинник;
- «mday» – день місяця;
- «wday» – день тижня, цифровий;
- «mon» – місяць, цифровий;
- «year» – рік, цифровий;
- «yday» – день року, цифровий; тобто «299»;
- «weekday» – день тижня, текстовий, повний; тобто «Friday»;
- «month» – місяць, текстовий, повний; тобто «January».

gmdate

gmdate – форматує GMT/CUT час/дату.

Опис

```
string gmdate (string format, int timestamp).
```

Аналогічна функції date() за винятком того, що час вертається в Гринвічському форматі Greenwich Mean Time (GMT). Наприклад, при запуску у Фінляндії (GMT +0200) перший рядок нижче надрукує «Jan 01 1998 00:00:00», у той час, як другий рядок надрукує «Dec 31 1997 22:00:00».

Приклад функції gmdate()

```

echo date(«M d Y H:i:s»,mktime(0,0,0,1,1,1998));
echo gmdate(«M d Y H:i:s»,mktime(0,0,0,1,1,1998));

```

mktime

`mktime` – одержує тимчасову мітку UNIX для дати.

`int mktime(int hour, int minute, int second, int month, int day, int year)`.

Попередження. Зверніть увагу на незвичайний порядок аргументів, який відрізняється від порядку аргументів у виклику функції `mktime()` з UNIX і який погано поводить себе при неправильно заданих параметрах. Помилка досить часто зустрічається у скриптах.

Повертає тимчасову мітку Unix згідно з даними аргументами. Ця тимчасова мітка є цілим числом, рівним кількості секунд між епохою Unix (1 Січня 1970) і зазначеним часом.

Аргументи можуть бути опущені з права на ліво; кожний опущений у такий спосіб аргумент буде встановлений у поточну величину згідно з локальною датою й часом.

`Mktime` корисна при арифметичних діях з датою і її перевіркою, вона буде автоматично обчислювати коректну величину для тих, що вийшли за межі параметрів.

Наприклад, кожна з наступних строк повертає строку «Jan-01-1998».

Приклад функції `mktime()`

```
echo date(«M-d-Y», mktime(0,0,0,12,32,1997));
```

```
echo date(«M-d-Y», mktime(0,0,0,13,1,1997));
```

```
echo date(«M-d-Y», mktime(0,0,0,1,1,1998));
```

gmmktime

`gmmktime` – одержує тимчасову мітку UNIX для дати в GMT.

`int gmmktime(int hour, int minute, int second, int month, int day, int year)`.

Ідентична `mktime()` за винятком переданих параметрів, що представляють дату в GMT.

time

`time` – повертає поточну тимчасову мітку UNIX.

`int time (void)`.

Повертає поточний час, обмірюване в числі секунд із епохи Unix (1 Січня 1970 00:00:00 GMT).

microtime

`microtime` – повертає поточну тимчасову мітку UNIX у мікросекундах.

`string microtime (void)`;

Повертає рядок «msec sec», де sec поточний час, обмірюваний у числі секунд із епохи Unix (0:00:00 1 Січня, 1970 GMT), а msec – це частина в мікросекундах. Ці функції доступні тільки в операційних системах, що підтримують системний виклик `gettimeofday()`.

Функції для роботи з каталогами

Функції для роботи з каталогами: `chdir`, `dir`, `closedir`, `opendir`, `readdir`, `rewinddir`.

chdir

`chdir` – зміна каталога.

`int chdir(string directory).`

Змінює поточний PHP каталог на `directory`. Повертає `FALSE`, якщо не можна змінити, `TRUE`, якщо зміна відбулась.

dir

`dir` – клас каталога (псевдо-об'єктно орієнтований механізм).

`new dir (string directory).`

Псевдо-об'єктно орієнтований механізм для читання каталогу. Відкриває каталог із `directory`. Два реквізити доступні, якщо каталог був відкритий. Реквізит `handle` може бути використаний разом з іншими функціями роботи з каталогом типу `readdir()`, `rewinddir()` і `closedir()`. Реквізит `path` встановлює шлях каталогу, який був відкритий. Три методи доступні: читання, повернення до початку і закриття.

Приклад функції `Dir()`

```
$d = dir(«/etc»);  
echo «Handle: «.$d->handle.»<br>\n»;  
echo «Path: «.$d->path.»<br>\n»;  
while($entry=$d->read()) {  
echo $entry.»<br>\n»;  
}  
$d->close();
```

closedir

`closedir` – закрити дескриптор (`handle`) каталогу.

`void closedir (int dir_handle).`

Закриває потік каталогу, позначений як `dir_handle`. Потік попередньо повинен бути відкритий функцією `opendir()`.

opendir

`opendir` – відкрити дескриптор (`handle`) каталогу.

`int opendir (string path).`

Повертає дескриптор (handle) каталогу, який надалі використовується в closedir(), readdir(), і rewinddir() обігах.

readdir

readdir – читання даних з каталогу по дескриптору (handle).

string readdir (int dir_handle).

Повертає ім'я наступного файлу з каталогу. Імена не вертаються в будь-якому специфічному порядку.

Приклад виводу усіх файлів у поточному каталозі.

```
<?php
$handle=opendir('.');
echo «Directory handle: $handle\n»;
echo «Files:\n»;
while ($file = readdir($handle)) {
echo «$file\n»;
}
closedir($handle);
?>
```

rewinddir

rewinddir – повернення до початку даних каталогу по дескриптору (handle).

void rewinddir (int dir_handle).

Скидає потік каталогу, позначений як dir_handle у початок даних.

Поштові функції

Функція **mail()** дозволяє відсилати пошту.

bool mail(string to, string subject, string message, string additional_headers).

Mail() автоматично посилає повідомлення, що міститься в message, адресатові, зазначеному в to. Декілька одержувачів можуть бути зазначені в полі to у вигляді рядка з адресами, розділеними пробілами.

Приклад 1:

```
mail(«rasmus@lerdorf.on.ca», «Моя тема»,
«Рядок 1\nстрока 2\nстрока 3»);
```

Якщо заданий четвертий строковий аргумент, він автоматично вставляється у кінець заголовка. Звичайно, це використовується при додаванні додаткових полів у заголовок. Кілька додаткових полів розділяються символом нового рядка. [15].

Питання для самоконтролю

1. Що таке PHP?
2. Яке основне призначення мови PHP?
3. Що таке змінні у PHP?
4. Яка функція використовується у PHP для перевірки типу даних будь-якої змінної?
5. Як можна встановити з'єднання з сервером MySQL за допомогою PHP?
6. Які можливості надає використання скриптів PHP при створенні вебсторінок?
7. Наведіть приклади арифметичних операторів.
8. Які існують оператори присвоювання?
9. Який синтаксис операторів AND і OR?
10. Які умовні оператори використовуються у PHP?
11. Для чого використовується оператор ELSE?
12. Для чого призначений оператор SWITCH?
13. Які функції дати та часу Ви знаєте?
14. Для чого призначена функція `vscomp`?
15. Яка функція дозволяє отримати модуль числа довільної точності?
16. Які математичні функції використовуються у PHP?
17. Яка функція перевіряє правильність дати/часу?
18. За допомогою якої функції можна задати формат локального часу/дати?
19. Для чого використовується функція `getdate`?
20. Які функції для роботи з каталогами використовуються у PHP?
21. Що виконує функція `mail`?
22. Який оператор використовується для об'єднання рядкових даних у PHP?
23. Назвіть прості типи даних, які підтримуються у PHP.

Тема 9. Робота з електронною поштою, групами новин та розсиляннями

9.1. Електронна пошта (E-mail)

Електронна пошта (E-mail) – один із найпоширеніших інформаційних сервісів Інтернет, засіб комунікацій, що забезпечує передавання та прийом електронних повідомлень (рис. 9.1.1).



Рис.9.1.1. Електронна пошта (E-mail)

Сьогодні майже всі користувачі Інтернет мають свою поштову скриньку в мережі.

Поштова скринька (mailbox) – це інформаційний простір, виділений у вузлі мережі з унікальним ім'ям в межах сайта, для зберігання поштових повідомлень.

Засобами пошти можна отримати доступ до інформаційних ресурсів інших мереж. Для роботи електронної пошти в мережі Інтернет розроблено спеціальний протокол Simple Mail Transfer Protocol (SMTP), який є протоколом прикладного рівня і використовує протокол TCP.

SMTP забезпечує обмін поштовими повідомленнями між користувачами однієї і тієї ж або різних комп'ютерних мереж. Система підтримує:

- надсилання одиночних повідомлень одному або більше одержувачам;
- надсилання повідомлень, що містять текст, голосові повідомлення, відео або графічні матеріали;
- надсилання повідомлень користувачам мереж, що не підключені до мережі Інтернет.

Багатоцільове розширення пошти засобами мережі Інтернет (Multipurpose Internet Mail Extension – MIME) – це розширення SMTP, яке дозволяє передавання мультимедійних повідомлень (текст, аудіо,

нерухоме зображення, відео тощо). Поштовий протокол (Post Office Protocol, version 3 – POP3) і протокол доступу до пошти в Інтернеті, версія 4 (Internet Mail Access Protocol – IMAP4) використовують для з'єднання SMTP і приймача та утримання пошти в хості.

Основою будь-якої поштової служби є система адрес. Без точної адреси неможливо доставити пошту адресату. У мережі Інтернет прийнята система адрес, що базується на доменній адресі машини, підключеної до мережі.

Адреса складається із двох частин: ідентифікатора користувача, який записується перед знаком-символом електронної пошти «@» (означає англ. прийменник «at»), і доменної адреси машини, яка записується після знака «@». Форма адреси: localname@domainname.

Для електронної пошти першим застосував знак «@» інженер компанії BBN Technologies Рей Томлінсон в 1971 році при відправці першого електронного листа мережею ARPANET.

Передати поштове повідомлення можна за допомогою **поштового клієнта** – спеціальної програми для електронної пошти, що дозволяє користувачу створювати і відправляти повідомлення, а також управляти отриманими повідомленнями (переглядати, впорядковувати, друкувати, видаляти і т. п.) із власної поштової скриньки або засобами вебтехнологій.

У поштового клієнта є ряд переваг:

- можливість одночасної роботи з кількома адресами електронної пошти у разі їх наявності;
- інформація про листування зберігається на комп'ютері користувача і застрахована від збоїв бази даних на вебсервері, що призводять до зникнення листів з поштової скриньки;
- для читання або написання листа не обов'язково перебувати в онлайні.

Недолік поштового клієнта – доступність пошти лише з того комп'ютера, на якому він встановлений.

Базові функції поштових клієнтів – фільтрація повідомлень, підтримка HTML, маркування повідомлень, використання віртуальних папок, застосування шаблонів вітань і підписів, застосування шаблонів повідомлень, використання швидких шаблонів (для введення часто вживаних фраз), автоматичне резервування поштових баз, синхронізація з PDA та ін.

Сучасні поштові клієнти пропонують своїм користувачам багато додаткових функцій, наприклад, зручні поштові книги, планувальники, автовідповідачі і т. п. Деякі подібні функції є й у вебсервісів, але там їх не настільки зручно налаштовувати.

Для передавання поштового листа потрібно:

- встановити зв'язок з поштовим сервером;
- передати повідомлення;
- завершити з'єднання.

Сеанс зв'язку з поштовим сервером розбито на три етапи: авторизація, транзакція й оновлення. Авторизація користувача відбувається при з'єднанні з поштовим сервером, для чого може використовуватися будь-який поштовий клієнт, що підтримує роботу з протоколом POP3. На етапі транзакції клієнт запитує у сервера виконання необхідних дій, наприклад отримання інформації про кількість повідомлень, здобуття самих повідомлень або їх видалення. Процес оновлення призначений для виконання запиту клієнта. Після закінчення оновлення сеанс зв'язку завершується до надходження наступного запиту на з'єднання.

При проходженні етапу авторизації може використовуватися будь-який з існуючих протоколів шифрування, наприклад SSL або TLS, що робить процес отримання електронної кореспонденції більш захищеним.

Протокол POP3 дозволяє лише отримувати електронні повідомлення, а для їх відправлення використовується інший протокол, зокрема, найчастіше застосовується SMTP, або його вдосконалена версія – ESMTP (Extended SMTP). Як і POP3, протокол SMTP працює на прикладному рівні, тому йому необхідні послуги транспортного протоколу, яким слугує протокол TCP.

Найбільш поширеними поштовими клієнтами, що заслужили визнання користувачів, є: Microsoft Outlook, The Bat!, Mozilla Thunderbird та Opera Mail. Існує ще багато поштових клієнтів, серед них: Pegasus Mail, Foxmail, KMail, Becky!, Pocomail та ін [1].

9.2. Телеконференції та групи новин (Usenet)

Послуга Usenet – найвідоміша система телеконференцій; підтримує більше 8000 телеконференцій різноманітної тематики. Призначена насамперед для передавання простих текстових повідомлень, що виводяться на екран. Для передавання файлів

необхідно спершу перетворити двійкові дані на текстовий формат і навпаки програмами UUEncode чи UUDecode. Можна декодувати і за допомогою програми WinCode. Існують також інші програми кодування, зокрема Macintosh BINHEX.

Більшість сучасних програм роботи з телеконференціями дозволяють проводити пошук телеконференцій за заданою як ключове слово частиною імені. Система імен телеконференцій будується за ієрархічним принципом. Ім'я телеконференції складається з кількох частин, кожна з яких несе певну інформацію про тематику. Перший рівень в імені – основний ідентифікатор, що визначає належність конференції до якоїсь категорії (табл. 9.2.1).

Таблиця 9.2.1. Перший рівень в імені телеконференцій

Перший рівень в імені телеконференції	Інформація про тематику Телеконференції
biz	бізнес
comp	комп'ютери
news	новини загального характеру
rec	розваги (хобі та мистецтво)
sci	наука
soc	соціальні теми
talk	з орієнтацією на дискусію
misk	теми поза вищевказаними категоріями
alt	альтернативні

За основним ідентифікатором телеконференції йде ідентифікатор другого рівня, який визначає деяку, досить широку, тематичну ділянку.

Третій рівень зазвичай вводиться, коли телеконференція отримує занадто багато повідомлень, що стосуються різноманітних конкретних тем цієї тематичної ділянки. У кожній телеконференції може бути лише одне ім'я – після основного ідентифікатора або складатися з досить великої кількості частин, хоч останнє зустрічається рідко і переважно серед категорій comp, sci.

Для роботи з телеконференціями використовують спеціальні програми читання телеконференцій (news reader), які дозволяють вибрати телеконференцію і переглянути її зміст. Більшість із цих

програм належать до категорії вільно розповсюджуваних програмних продуктів та підтримують усі операційні системи [1].

Популярні програми-клієнти, що підтримують роботу сервісу телеконференцій:

- NewsLeecher – <http://www.newsleecher.com>;
- NewsBin – <http://www.newsbin.com>;
- Usenet Explorer – <http://www.netwu.com/ue>;
- Forte Agent – <http://www.forteinc.com>.

9.3. Списки розсилок

Під списком розсилання розуміють послугу, за допомогою якої розсилають повідомлення деякій групі передплатників (на зразок звичайної преси на пошті).

Списки розсилання (mail lists) – найбільш зручний спосіб регулярного одержання нових повідомлень на тему, що цікавить. На розсилання можна підписатися практично на будь-якому досить великому й цікавому сайті – і новини сайту будуть приходити електронною поштою, з різним ступенем регулярності. Цей сервіс надається тільки через протокол електронної пошти. Існують спеціальні сервери розсилок.

Розсилання можуть бути публічними і закритими. Закриті розсилання – це інструмент спільної роботи, звичайно вони створюються і використовуються в межах якоїсь компанії. Для публічних розсилок, як правило, визначені правила передплати і відмови від них.

Існують сайти, на яких пропонується відповідний вебсервіс – на таких сайтах будь-хто може створити власне розсилання. Звичайно розсилання на цих сайтах не використовуються для групового спілкування, а лише в режимі віщання.

Списки розсилання існують у формі передплати: користувач робить замовлення електронною поштою на матеріали на обрану тему і отримує всю кореспонденцію, що її стосується.

В списки розсилок не допускається втручання користувачів. Тому не слід плутати списки розсилок з групами новин. Скоріш за все розсилання нагадують електронні газети [16].

Питання для самоконтролю

1. Що таке поштовий сервер?
2. Які існують поштові сервери?
3. Для чого призначені протоколи POP3 та SMTP?
4. Адреса електронної пошти.
5. Що таке адресна книга, контакти?
6. Для чого створюються групи контактів?
7. Створення та відправлення листа. Приєднання файлів.
8. Правила етикету при використанні електронної пошти.
9. Що таке телеконференції?
10. Які групи телеконференцій існують?
11. Які програмні засоби для роботи з телеконференціями Ви знаєте?
12. Що таке розсилання?
13. Як можна передати поштове повідомлення?
14. Що таке поштовий клієнт?
15. Які поштові клієнти Ви знаєте?
16. Які є відомі безкоштовні сервери електронної пошти?
17. Які мультимедійні елементи можна додавати до електронного листа?

Тема 10. Хмарні технології

10.1. Основні характеристики хмарних технологій

Хмарні технології – це парадигма, що передбачає віддалене опрацювання та зберігання даних.

Хмара – це деякий центр опрацювання даних (дата-центр, сервер) або їх мережа, де зберігаються дані та програми, що з'єднуються з користувачами через Інтернет.

Використання хмарних технологій дозволяє споживачам використовувати програми без установки і доступу до особистих файлів з будь-якого комп'ютера, що має доступ до мережі Інтернет. За допомогою цих технологій можна вести значно ефективніше управління підприємством за рахунок централізації управлінських та облікових відомостей, опрацювання, пропускну здатності та надійності зберігання даних. Простим прикладом хмарних технологій є сервіси електронної пошти, наприклад, Gmail, Meta і т.д. Потрібно всього лише підключення до Інтернет, і можна відправити пошту, при цьому додаткового програмного забезпечення або сервера не потрібно. Хмарні технології являють собою загальний термін для всього, що включає в себе надання послуги хостингу через Інтернет. Ці послуги, в цілому, можна розділити на три категорії (Рис. 10.1.1):

- Програмне забезпечення як послуга (SaaS – англ. software as a service).
- Платформа як послуга (PaaS – англ. Platform as a Service).
- Інфраструктура як послуга (IaaS – англ. Infrastructure as a Service) [16, с. 138].



Рис. 10.1.1. Хмарні технології

Хмарні технології в навчанні надають ряд переваг:

- економія засобів на придбання програмного забезпечення;
- зниження потреби в спеціалізованих приміщеннях;

- виконання багатьох видів учбової роботи, контролю і оцінки online;
- економія дискового простору;
- антивірусна, безрекламна, антихакерська безпека та відкритість освітнього середовища для вчителів і для учнів [17, с. 3].

10.2. Хмарні технології для навчальних закладів

Прикладами використання хмарних технологій можуть бути Google Workspace for Education та Office 365 Education.

Google Workspace for Education. Сервіс Google Workspace for Education об'єднує окремі служби, за допомогою яких співробітникам одного навчального закладу можна ефективніше спілкуватися та співпрацювати з співробітниками свого або іншого навчального закладу (Рис. 10.2.1). Ці служби є простими в налаштуванні, не потребують додаткового обслуговування і ними можна користуватися безкоштовно. Адміністратори вебсайтів (вебпорталів) можуть за власним вибором поєднувати окремі служби для розміщення даних на сайті, а також для спілкування та співпраці.

Окремо слід звернути увагу на службу Google Диск, за допомогою якої користувачі можуть створювати документи, таблиці, форми, презентації та малюнки, співпрацюючи з іншими користувачами в реальному часі безпосередньо у вікні веббраузера. Всі ресурси (документи, таблиці, форми, презентації та малюнки) розміщені в службі Google Диск можна опублікувати в глобальній мережі Інтернет.

Google Workspace Рішення Продукти Ціни Ресурси Консоль адміністратора Зв'язок із відділом продажів Безкоштовна пробна версія

Як команди будь-яких розмірів підтримують зв'язок, створюють матеріали та працюють разом.

Гнучке інноваційне рішення, яке допомагає людям і організаціям досягати ще більше.

Безкоштовна пробна версія

Рис. 10.2.1. Google Workspace for Education

Office 365 Education. За допомогою Office 365 Education можна надати для викладачів, інших співробітників та студентів можливість безкоштовно працювати з електронною поштою, створювати вебсайти, редагувати та зберігати документи в глобальній мережі Інтернет, обмінюватися миттєвими повідомленнями та проводити вебконференції (Рис. 10.2.2). Як і в сервісі Google Workspace for Education для навчальних закладів в Office 365 Education є аналогічна служба для роботи з різними документами – це служба OneDrive (SkyDrive). У службі OneDrive кожному користувачу надається певний обсяг вільного місця для зберігання даних в «хмарі», які синхронізуються з даними на персональному комп'ютері користувача. Використовуючи службу OneDrive можна надавати спільний доступ до документів для інших користувачів [16].

Хмарні служби для створення сучасних занять

Створіть умови для співпраці у класі, приєднуйтеся до спільнот професійної освіти та спілкуйтеся із персоналом школи за допомогою ефективних інструментів, таких як OneDrive, Teams, тощо — усе в єдиному інтерфейсі Office 365 Education.

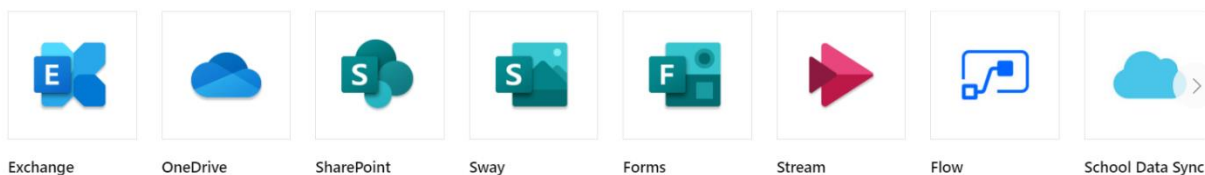
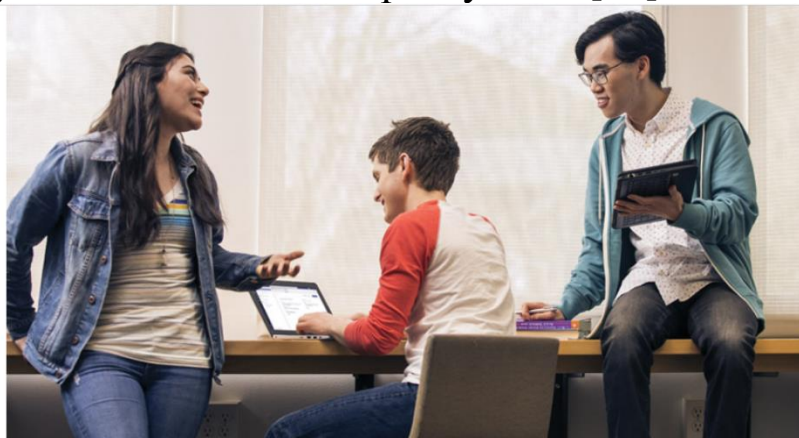


Рис. 10.2.2. Office 365 для навчальних закладів

Питання для самоконтролю

1. Що таке хмарні технології?
2. Які служби пропонуються в Google?
3. Які документи можна створювати за допомогою служби Google Диск?
4. Які сервіси (служби) використовуються в обліковому записі Microsoft?
5. Які документи можна створювати за допомогою служби OneDrive?
6. Які хмарні сервіси для організації дистанційного навчання Ви знаєте?
7. Які хмарні сервіси для організації освітнього процесу Вам відомі?
8. Які переваги надає використання Google Workspace for Education в освітньому процесі?
9. Яке основне призначення облікового запису Google?
10. Чи можна ввійти в обліковий запис Google, використовуючи сторонню електронну адресу (не Gmail)?
11. Як створити власний Google акаунт?

Список використаних джерел

1. Азарова А. О., Лисак Н. В. Комп'ютерні мережі та телекомунікації : навч. посіб. Вінниця : ВНТУ, 2012. 293 с.
2. Буйницька О. П. Інформаційні технології та технічні засоби навчання. Київ : Центр учб. літ., 2012. 240 с.
3. Воробієнко П. П., Нікітюк Л. А., Резніченко П. І. Телекомунікаційні та інформаційні мережі : Підруч. [для ВНЗ]. Київ : САММІТ-Кн., 2010. 708 с.
4. Горlach В. М., Макар В. М. Побудова та адміністрування INTRANET-мереж. Ч. 1. Основи мережних технологій : Тексти лекцій. Львів : Вид. центр Львів, 1999. 45 с.
5. Комп'ютерні мережі / уклад.: О. Гордєєв, Д. Гордєєва, М. Колдовський. Суми : ДВНЗ «УАБС НБУ», 2010. 310 с.
6. Комп'ютерні мережі / уклад. Ю. А. Зав'ялець. Чернівці, 2015. 183 с.
7. Марченко Е. К. Електронний посібник «Комп'ютерні мережі». *Комп'ютерні мережі*. URL: <https://km.ptngu.com/> (дата звернення: 18.11.2021).
8. Методичні вказівки до виконання лабораторних робіт (комп'ютерного практикуму) з дисципліни «Технології створення Web-сторінок» для студентів напряму підготовки 6.051501 «Видавничо-поліграфічна справа» спеціальності «Технології електронних мультимедійних видань» для студ. / уклад.: Б. Кушлик, О. Кушлик-Дивульська. Київ : НТУУ «КПІ», 2015. 58 с.
9. Микитишин А. Г., Митник М. М., Пасічник В. В. Комп'ютерні мережі : навч. посіб. Львів : Магнолія 2006, 2013. 256 с.
10. Навчальна практика з інформатики та комп'ютерної техніки : Робоч. зошит. Спец. 076 Підприємництво, торгівля та бірж. діяльність / уклад. П. Матушкіна. Київ : НМЦ «Укоопосвіта», 2019. 106 с.
11. Олещенко Л. М. Організація комп'ютерних мереж: конспект лекцій : Навч. посіб. Київ : КПІ ім. Ігоря Сік., 2018. 225 с.
12. Рижов О. А., Іванькова Н. А., Андросов О. І. Сучасні мережеві технології : Навч.-метод. посіб. для студентів-провізорів оч., заоч. та дистанц. форм навчання. Запоріжжя : [ЗДМУ], 2018. 68 с.

13. Струтинська О. В. Інформаційні системи та мережеві технології : навч. посіб. для дистанц. навчання / ред. М. І. Жалдак. Київ : Ун-т «Україна», 2008. 211 с.
14. Сучасні інформаційно-комунікаційні технології : навч. посіб. / Г. Г. Швачич та ін. Дніпро : НМетАУ, 2017. 230 с.
15. Сьомка Н. К., Філіппова Л. Л. Практикум з WEB-програмування : навч. посіб. Ірпінь : Ун-т ДФС України, 2016. 352 с.
16. Франчук В. М. Комп'ютерні мережі та Інтернет : Навч.-метод. пос. для студ. фізико-мат. та інформат. спец. вищ. пед. навч. закл. Київ : НПУ ім. М.П. Драгоман., 2015. 141 с.
17. Хмарні технології та Google сервіси : Лаборатор. практикум для вчителів ЗНЗ / уклад. Ю. Лимар. 2017. 64 с.
18. Хоменко В. Г., Павленко М. П. Комп'ютерні мережі : навч. посіб. Донецьк : ЛАНДОН-XXI, 2011. 316 с.

СУЧАСНІ МЕРЕЖНІ ТЕХНОЛОГІЇ

Навчальний посібник

Укладачі Стеценко В. П., Тітова Л. О.

Видається в авторській редакції

Підписано до друку 21.01.2022 р. Формат 60x84/16.

Папір офсетний. Ум. друк. арк. 9,06

Тираж 200 прим. Замовлення № 039

Видавець і виготівник «Сочінський М. М.»

20300, м. Умань, вул. Тищика, 18/19, вул. Небесної Сотні 2/1

Свідоцтво суб'єкта видавничої справи

ДК № 2521 від 08.06.2006.

тел. (04744) 4-64-88, (067) 104-64-88, (068) 854-66-19

vizavi-print.jimdo.com

e-mail: vizavi008@gmail.com

vizavi003@gmail.com