

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт

з дисципліни

„INTERNET-ТЕХНОЛОГІЇ“

для студентів спеціальності 7(8).05080101

„Мікро- та наноелектронні прилади і пристрої“

денної і заочної форм навчання

Методичні вказівки до виконання лабораторних робіт з дисципліни „Internet-технології“ для студентів спеціальності 7(8).05080101 „Мікро- та наноелектронні прилади і пристрої“ денної і заочної форм навчання / Укл.: А. В. Бабіч, Г. В. Сніжної. – Запоріжжя: ЗНТУ, 2014. – 82 с.

Укладач: А. В. Бабіч, доц., канд.фіз.-мат.наук
Г. В. Сніжної, доц., канд.фіз.-мат.наук

Рецензент: О. В. Томашевський, доц., канд.техн.наук

Відповідальний за випуск: Є.Л.Жавжаров, доц., канд. фіз.-мат. наук

Затверджено
на засіданні кафедри
мікро- та наноелектроніки

Протокол №2
від „18“ серпня 20 14 р.

ЗМІСТ

1 ЛАБОРАТОРНА РОБОТА №1 «ПОБУДОВА СКЛАДНОЇ ГІБРИДНОЇ МЕРЕЖІ»	5
1.1 Загальні теоретичні відомості	5
1.2 Завдання до лабораторної роботи	8
1.2.1 Побудова складної мережі Ethernet	8
1.2.2 Побудова бездротової ділянки мережі і її підключення до мережі Ethernet.....	10
1.3 Контрольні запитання	14
1.4 Перелік посилань	14
2 ЛАБОРАТОРНА РОБОТА №2 «НАЛАШТУВАННЯ ІР-АДРЕСАЦІЇ І МАРШРУТИЗАЦІЇ».....	15
2.1 Загальні теоретичні відомості	15
2.2 Завдання до лабораторної роботи	15
2.2.1 Аналіз правильності параметрів протоколу ІР	15
2.2.2 Налаштування ІР-адрес і перевірка працездатності протоколу TCP/IP.....	16
2.3 Контрольні запитання	17
2.4 Перелік посилань	18
3 ЛАБОРАТОРНА РОБОТА №3 «ПРОТОКОЛ TCP/IP. АНАЛІЗ МЕРЕЖЕВИХ КАДРІВ»	18
3.1 Загальні теоретичні відомості	18
3.1.1 Стек протоколів TCP/IP	18
3.1.2 Фізичний і канальний рівень	20
3.1.3 Міжмережевий рівень і протокол ІР	20
3.1.4 Транспортний рівень	21
3.2 Завдання до лабораторної роботи	22
3.3 Контрольні запитання	24
4 ЛАБОРАТОРНА РОБОТА № 4 “ВИДАЛЕНЕ УПРАВЛІННЯ ПО ПРОТОКОЛУ SSH З ВИКОРИСТАННЯМ УТИЛІТИ PUTTY”	24
4.1 Загальні теоретичні відомості	24
4.2 Завдання до лабораторної роботи	26
4.3 Контрольні запитання	28

5 ЛАБОРАТОРНА РОБОТА № 5 “СТВОРЕННЯ WEB-СТОРИНОК. ВИВЧЕННЯ МОВИ HTML”	29
5.1 Загальні теоретичні відомості	29
5.2 Завдання до лабораторної роботи	32
5.3 Контрольні запитання	34
6 ЛАБОРАТОРНА РОБОТА №6 “СТВОРЕННЯ ФОРМ. ВІКНА ДІАЛОГУ. ПЕРЕВІРКА ПРАВИЛЬНОСТІ ЗАПОВНЕННЯ ФОРМ ЗА ДОПОМОГОЮ JAVASCRIPT”	35
6.1 Загальні теоретичні відомості	35
6.1.1 Вікна діалогу. Об’єкт Window	37
6.1.2 Застосування JavaScript у веб-сторінках	38
6.2 Завдання до лабораторної роботи	40
6.3 Контрольні запитання	46
7 ЛАБОРАТОРНА РОБОТА № 7 “ТЕХНОЛОГІЯ CSS. РОЗМІТКА СТОРИНОК ЗА ДОПОМОГОЮ ТАБЛИЦЬ, ШАРІВ (DIV) і CSS”	46
7.1 Загальні теоретичні відомості	46
7.1.1 Каскадні таблиці стилів	46
7.1.2 Основні принципи табличної розмітки	54
7.1.3 Верстка за допомогою шарів	60
7.2 Завдання до лабораторної роботи	65
7.3 Контрольні запитання	66
8 ЛАБОРАТОРНА РОБОТА № 8 “СТВОРЕННЯ WEB-СТОРИНОК ЗА ДОПОМОГОЮ PHP”	67
8.1 Загальні теоретичні відомості	67
8.2 Завдання до лабораторної роботи	69
8.3 Контрольні запитання	74
9 ЛАБОРАТОРНА РОБОТА № 9 “РОБОТА З БАЗОЮ ДАНИХ MYSQL”	74
9.1 Загальні теоретичні відомості	75
9.2 Завдання до лабораторної роботи	76
9.3 Контрольні запитання	80
РЕКОМЕНДОВАНА ЛІТЕРАТУРА	81

1 ЛАБОРАТОРНА РОБОТА №1 «ПОБУДОВА СКЛАДНОЇ ГІБРИДНОЇ МЕРЕЖІ»

Мета роботи: навчитися об'єднувати сегменти мережі, які використовують стандарти Ethernet 10Base-2 і 10Base-T, як один з одним, так і з сегментами, побудованими на базі Fast- або Gigabit Ethernet; налаштовувати і підключати до мережі Ethernet сегменти, в яких використовуються бездротові точки доступу і бездротові пристрої.

1.1 Загальні теоретичні відомості

Оскільки реальні мережі постійно зростають, важливою їх характеристикою є використана в них архітектура, від вибору якої багато в чому залежить розширюваність мережі і можливість її подальшої модернізації.

У цій лабораторній роботі демонструється сумісність різних стандартів архітектури Ethernet і показано, як за допомогою мережевих шлюзів до мережі Ethernet можна підключати пристрої, що використовують іншу архітектуру (наприклад, пристрої Wi-Fi).

Ethernet - сімейство технологій пакетної передачі даних для комп'ютерних мереж (рис.1.1). Стандарти Ethernet визначають дріт'яні з'єднання і електричні сигнали на фізичному рівні, формат кадрів і протоколи управління доступом до середовища — на канальному рівні моделі OSI. Ethernet в основному описується стандартами IEEE групи 802.3. Ethernet став найпоширенішою технологією ЛВС у середині 1990-х років, витіснивши такі застарілі технології, як Arcnet і Token ring.

У стандарті перших версій (Ethernet v1.0 і Ethernet v2.0) вказано, що передаючим середовищем є коаксіальний кабель, надалі стало можливим використовувати виту пару і оптичний кабель.

Переваги використання витої пари порівняно з коаксіальним кабелем такі:

- можливість роботи в дуплексному режимі;
- низька вартість кабелю «витої пари»;
- вища надійність мереж при несправності в кабелі (з'єднання точка-точка (при з'єднанні більше 2 комп'ютерів потрібні концентратор або комутатор, які забезпечують роботу решти частини мережі):

обрив кабелю позбавляє зв'язку два вузли; у коаксиалі використовують топологію «шина», обрив кабелю розбиває сегмент на 2 половини, і обидві половини не працюють через відсутність резисторів (термінаторів) (вони мають бути на обох кінцях кабелю) навантажень, хоча застосування концентраторів або комутаторів на коаксиалі повністю вирішувало цю проблему);

- мінімально допустимий радіус вигину менше;
- велика завадостійкість завдяки використанню диференціального сигналу;
- можливість живлення по кабелю малопотужних вузлів, наприклад IP-телефонів (стандарт Power over Ethernet, POE);
- гальванічна розв'язка трансформаторного типу (в умовах СНД, де, як правило, немає заземлення комп'ютерів, застосування коаксіального кабелю часто супроводжувалося пробоем мережевих карт і іноді навіть повним «вигоранням» системного блоку).

Причиною переходу на оптичний кабель була необхідність збільшити довжину сегменту без повторювачів.

У 1995 році прийнятий стандарт IEEE 802.3u Fast Ethernet із швидкістю 100 Мбіт/с і з'явилася можливість роботи в режимі *повний дуплекс*. У 1997 році був прийнятий стандарт IEEE 802.3z Gigabit Ethernet із швидкістю 1000 Мбіт/с для передачі по оптичному волокну, і ще через два роки для передачі по витій парі. На рис. 1.1 наведено формат кадру Ethernet.

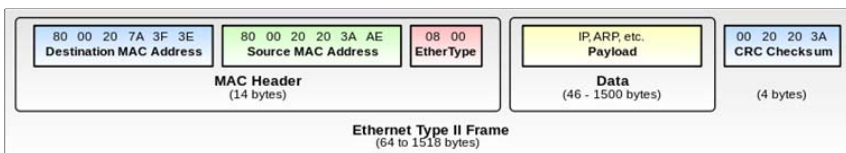


Рисунок 1.1 – Найпоширеніший формат кадру Ethernet II

Стандарти створення сегменту мережі

10BASE2, IEEE 802.3a («Тонкий Ethernet») — використовується кабель RG-58 з максимальною довжиною сегменту 185 метрів, комп'ютери приєднувалися один до одного, для підключення кабелю до мережевої карти потрібен T-коннектор (рис.1.2), а на кабелі має бути

BNC-коннектор. Потрібна наявність термінаторів на кожному кінці. Багато років цей стандарт був основним для технології Ethernet.

10BASE-T, IEEE 802.3i - для передачі даних використовують 4 дроти кабелю витої пари (дві скручені пари) категорії-3 або категорії-5, і стандарт RJ-45 (рис.1.3). Максимальна довжина сегменту 100 метрів.



Рисунок 1.2 – Т-коннектор формату BNC



Рисунок 1.3 – Коннектор RJ-49S

Wi-Fi - торгова марка Wi-Fi Alliance для бездротових мереж на базі стандарту IEEE 802.11. Під аббревіатурою Wi-Fi (англійською Wireless Fidelity, яку можна дослівно перекласти як «бездротова якість» або «бездротова точність») на сьогодні розвивається ціле сімейство стандартів передачі цифрових потоків даних по радіоканалам.

Зазвичай схема Wi-Fi мережі містить не менше однієї точки доступу і не менше одного клієнта. Також можливе підключення двох клієнтів в режимі точка-точка (Ad-hoc), коли точка доступу не використовується, а клієнти з'єднуються за допомогою мережевих адаптерів «напрямку». Точка доступу передає свій ідентифікатор мережі (SSID) за допомогою спеціальних сигнальних пакетів на швидкості 0,1 Мбіт/с кожні 100 мс. Тому 0,1 Мбіт/с — найменша швидкість передачі даних для Wi-Fi. Знаючи SSID мережі, клієнт може з'ясувати, чи можливе підключення до даної точки доступу. При попаданні в зону дії двох точок доступу з ідентичними SSID приймач може вибирати між ними на підставі даних про рівень сигналу. Стандарт Wi-Fi дає клієнту повну свободу при виборі критеріїв для з'єднання.

Переваги Wi-Fi:

– дозволяє розвернути мережу без прокладки кабелю, що зменшує вартість розгортання і/або розширення мережі. Місця, де не мож-

на прокласти кабель, наприклад, поза приміщеннями і в будовах, що мають історичну цінність, обслуговуються бездротовими мережами;

- дозволяє мати доступ до мережі мобільним пристроям;

- Wi-Fi пристрої широко поширені на ринку (гарантується сумісність устаткування завдяки обов'язковій сертифікації устаткування з логотипом Wi-Fi);

- мобільність. Ви більше не прив'язані до одного місця і можна користуватися Інтернетом в комфортних для вас умовах;

- в межах Wi-Fi-зони в мережу Інтернет можуть виходити декілька користувачів з комп'ютерів, ноутбуків, телефонів і т. п.;

- випромінювання від Wi-Fi пристроїв у момент передачі даних на порядок менші, ніж у стільникового телефону.

Недоліки Wi-Fi:

- у діапазоні 2,4 GHz працює безліч пристроїв, таких як пристрої, що підтримують Bluetooth, й ін., і навіть мікрохвильові печі, що погіршує електромагнітну сумісність;

- виробниками устаткування вказується швидкість на L1 (OSI), внаслідок чого створюється ілюзія, що виробник устаткування завищує швидкість, але насправді в Wi-Fi вельми високі службові «накладні витрати». Виходить, що швидкість передачі даних на L2 (OSI) в Wi-Fi мережі завжди нижче заявленої швидкості на L1 (OSI). Реальна швидкість залежить від частки службового трафіку, яка залежить вже від наявності між пристроями фізичних перешкод (меблі, стіни), наявності перешкод від інших бездротових пристроїв або електронної апаратури, розташування пристроїв щодо один одного і т. п.;

- частотний діапазон і експлуатаційні обмеження в різних країнах не однакові. У багатьох європейських країнах дозволені два додаткові канали, які заборонені в США; в Японії є ще один канал у верхній частині діапазону, а інші країни, наприклад Іспанія, забороняють використання низькочастотних каналів. Більш того, деякі країни, наприклад, Росія, Білорусь, Італія вимагають реєстрації всіх мереж Wi-Fi, які працюють зовні приміщень, або вимагають реєстрації Wi-Fi-оператора.

1.2 Завдання до лабораторної роботи

1.2.1 Побудова складної мережі Ethernet

При складанні завдань частково використані ресурси [1].

У цьому завданні Ви повинні ознайомитися з принципами об'єднання в єдину мережу сегментів, які використовують пристрої різних стандартів Ethernet.

Налаштування комп'ютера для роботи в мережі

1 Увійдіть до системи з обліковим записом, вказаним викладачем (наприклад, як користувач User1 з паролем Password). Цей користувач входить до локальної групи «Адміністратори» вашого комп'ютера, тому при виконанні цього і подальших завдань ви зможете змінювати настройки свого комп'ютера.

2 У меню **Пуск** виберіть пункт **Панель управління**.

3 У вікні **Панель управління**, що відкрилося, клацніть мишею на посиланні **Мережа і підключення до Інтернету**.

4 У вікні **Мережа і підключення до Інтернету** клацніть мишею на рядку **Майстер настройки мережі**.

5 На першій сторінці **Майстра настройки мережі** клацніть мишею на кнопці **Далі**.

6. На сторінці **Майстра** перед тим, як продовжити, клацніть мишею на кнопці **Далі**.

7 На сторінці **Виберіть метод підключення** виберіть радіокнопку **Інше** і клацніть мишею на кнопці **Далі**.

8 На сторінці **Інші способи підключення до Інтернету** виберіть радіокнопку **Цей комп'ютер належить мережі, що не має підключення до Інтернету**, і клацніть мишею на кнопці **Далі**.

9 На сторінці **Задайте ім'я і опис для цього комп'ютера** переконайтеся, що в полі **Ім'я комп'ютера** міститься правильна назва вашого комп'ютера (якщо ви сумніваєтеся, то запитайте у викладача) і клацніть мишею на кнопці **Далі**.

10 На сторінці **Задайте ім'я для вашої мережі** введіть в поле **Робоча група** назву **CLASS** і клацніть мишею на кнопці **Далі**. Назва робочої групи має бути однаковою для всіх комп'ютерів класу, інакше можуть виникнути проблеми з відображенням значків комп'ютерів у мережі.

11 На сторінці **Загальний доступ до файлів і принтерів** виберіть радіокнопку **Включити загальний доступ до файлів і принтерів** і клацніть мишею на кнопці **Далі**.

12 На сторінці **Все готово** для застосування мережевих параметрів переконайтеся, що всі настройки вказані правильно, і клацніть мишею на кнопці **Далі**.

13 На сторінці *Все майже готово* виберіть радіокнопку *Просто завершити роботу майстра, немає потреби запускати його на інших комп'ютерах* і клацніть мишею на кнопці *Далі*.

14 На сторінці *Завершення роботи майстра* настройки мережі переконайтеся, що настройка завершена успішно, і клацніть мишею на кнопці *Готово*.

15 У вікні, що з'явилося, *Зміна параметрів системи* клацніть мишею на кнопці *Так*, щоб перезавантажити комп'ютер.

Підключення сегменту мережі, який використовує стандарти 10Base-2 і 10Base-T, до комутатора Fast Ethernet

1 Візьміть прямий кабель на основі «витої пари». Пам'ятаючи про правила каскадування концентраторів (комутаторів), підключіть один з його кінців до перехресного порту (Uplink) гібридного концентратора.

2 Візьміть комутатор Fast- або Gigabit Ethernet і підключіть другий кінець кабелю до одного з його портів.

3 Включіть комп'ютери, підключені до комутатора Fast Ethernet.

4 Спробуйте звернутися до загальних ресурсів комп'ютерів, підключених до концентратора 10Base-T.

5 Успішне звернення до загальних ресурсів означає, що взаємодія між сегментами 10Base-T і Fast Ethernet встановлена.

1.2.2 Побудова бездротової ділянки мережі і її підключення до мережі Ethernet

У цьому завданні Ви повинні ознайомитися з принципами підключення до мережі Ethernet бездротової точки доступу, яка працює за одним із стандартів Wi-Fi, і налаштувати мережеву взаємодію між дротовими і бездротовими клієнтами.

Налаштування бездротової точки доступу

Ця частина завдання виконується на одному з комп'ютерів, за допомогою якого буде налаштована точка доступу. Оскільки налаштування бездротової точки доступу звичайно здійснюється через веб-інтерфейс з використанням стандартного мережевого Ethernet-з'єднання, обраний комп'ютер потрібно підключити до точки доступу,

а потім правильно конфігурувати параметри протоколу IP на цьому комп'ютері.

1 Візьміть бездротову точку доступу і, використовуючи роз'єм для коннектора RJ-45, з'єднайте за допомогою кабелю «вита пара» точку доступу з одним із комп'ютерів класу.

2 Включіть комп'ютер і увійдіть до системи з обліковим записом, що входить до локальної групи «**Адміністратори**».

3 В меню Пуск клацніть правою кнопкою миші на пункті **Мережеве оточення** і в контекстному меню, що з'явилося, виберіть пункт **Властивості**.

4 У вікні **Мережеві підключення**, що відкрилося, клацніть правою кнопкою миші на значку **Підключення по локальній мережі**, і в контекстному меню, що з'явилося, виберіть пункт **Властивості**.

5 У вікні властивостей мережевого підключення клацніть мишею на рядку **Протокол Інтернету (TCP/IP)** в списку **Компоненти, які використовуються цим підключенням**, а потім клацніть мишею на кнопці **Властивості**.

6 У вікні налаштування параметрів протоколу IP виберіть радіокнопку **Використовувати наступну IP адресу** і введіть такі параметри:

- IP-адреса — 192.168.1.200;
- маска підмережі — 255.255.255.0.

7 Двічі клацніть мишею на кнопках **ОК**, щоб закрити вікна налаштування мережевого підключення. Закрийте вікно **Мережеві підключення**.

8 У меню **Пуск** виберіть пункт **Інтернет**.

9 У вікні програми Microsoft Internet Explorer в полі **Адреса**, що відкрилося, введіть рядок http://IP-адрес вашої точки доступу (наприклад, http://192.168.1.1) і клацніть мишею на значку **Перехід**.

10 У вікні авторизації введіть пароль, вказаний в документації до вашої точки доступу. Подальші дії залежать від конкретної точки доступу, тому нижче наведено лише загальні кроки налаштування, які дозволяють добитися мережевої взаємодії з бездротовими клієнтами. Слід звернути увагу, що для спрощення підключення тут наведені такі параметри налаштування, які не рекомендується застосовувати в реальній роботі (зокрема, тут включається режим сповіщення і відключається захист при бездротовому доступі).

11 Знайдіть в меню управління точкою доступу розділ **Wireless Settings** («налаштування бездротової взаємодії») або аналогічний, і налаштуйте наступні (або аналогічні) параметри:

– **Channel** («канал») — залиште встановленим за умовчанням;

– **SSID** («ідентифікатор») — введіть назву вашої точки доступу, наприклад, *ClassAP*;

– **SSID Broadcast** («сповіщення») — для спрощення виявлення вашої точки доступу бездротовими клієнтами цей режим краще включити (*Enable*);

– **Wireless Mode** («стандарт, використовуваний точкою доступу») — для сумісності краще вказати змішаний режим *802.11b/g*.

12 Клацніть мишею на кнопці **Apply** («застосувати»). Перейдіть до розділу Encryption («шифрування») або аналогічному, і виберіть наступний (або аналогічний) параметр:

– **Security Mode** («режим захисту») — відключений (*Disabled*). Відключення захисту здійснюється в цьому завданні тільки для спрощення налаштування! На практиці такі параметри бездротової взаємодії застосовувати не можна.

13 Клацніть мишею на кнопці **Apply** («застосувати») і закрийте вікно програми Internet Explorer.

Налаштування бездротового адаптера і підключення до точки доступу

Ця частина завдання виконується на комп'ютері, що не має дротового підключення до мережі.

1 Візьміть бездротовий мережевий адаптер і встановіть його в один з комп'ютерів мережі.

2 Включіть комп'ютер і увійдіть до системи з обліковим записом, що входить до локальної групи «**Адміністратори**».

3 На сторінці **Майстер нового устаткування** виберіть радіокнопку **Ні, не на цей раз** і клацніть мишею на кнопці **Далі**.

4 На сторінці **Якщо з пристроєм поставляється настановний диск, вставте його** і переконайтеся, що вибрана радіокнопка **Автоматична установка (рекомендується)**, вставте компакт-диск (з комплекту бездротового адаптера) з драйвером і клацніть мишею на кнопці **Далі**.

5 Майстер нового устаткування має знайти на диску відповідний для вашого адаптера драйвер. На сторінці **Виберіть найбільш відповідне програмне забезпечення для вашого пристрою** клацніть мишею на кнопці **Далі**.

6 На сторінці **Майстер завершив установку програм для** клацніть мишею на кнопці **Готово**.

7 Клацніть правою кнопкою миші на значку **Бездротове мережеве з'єднання в Панелі задач** і виберіть в меню пункт **Проглядання доступних бездротових мереж**.

8 На сторінці **Виберіть бездротову мережу** виберіть мережу з назвою, вказаною в полі **SSID** при настройці вашої точки доступу (наприклад, ClassAP), і клацніть мишею на кнопці **Підключити**.

9 Виконайте подвійне клацання мишею на значку **Бездротове мережеве з'єднання** (ClassAP) в Панелі задач.

10 У вікні **Стан Бездротового мережевого з'єднання** клацніть мишею на кнопці **Властивості**.

11 У вікні властивостей бездротового мережевого підключення клацніть мишею на рядку **Протокол Інтернету (TCP/IP)** в списку **Компоненти, використовувані цим підключенням**, а потім клацніть мишею на кнопці **Властивості**.

12 У вікні настройки параметрів протоколу IP виберіть радіо-кнопку **Використовувати наступну IP адресу** і введіть параметри:

- IP-адреса — 192.168.1.150;
- маска підмережі — 255.255.255.0.

13 Двічі клацніть мишею **ОК** і закрийте всі вікна.

Підключення точки доступу до мережі Ethernet і перевірка взаємодії в гетерогенній мережі

Ця частина завдання виконується на комп'ютерах, підключених до мережі Ethernet, побудованої в ході виконання завдання 2 п.1.2.1 даної лабораторної роботи.

1 Використовуючи кабель «вита пара», підключіть бездротову точку доступу до порту RJ-45 комутатора Fast Ethernet.

2 Настройте на комп'ютерах мережі параметри протоколу IP, сумісні з тими, які Ви використовували при настройці бездротової точки доступу і комп'ютера з бездротовим адаптером, наприклад, такі:

- IP-адреса — 192.168.1.10x, де x — номер комп'ютера в класі;
- маска підмережі — 255.255.255.0.

Тобто, перший комп'ютер повинен мати IP-адресу 192.168.1.101, другий — 192.168.1.102 і т.д.

3 Спробуйте звернутися до загальних ресурсів на комп'ютерах мережі (особливо цікаво перевірити, чи працює звернення до ресурсів комп'ютерів з бездротовими адаптерами).

4 Закрийте всі вікна, оформіть звіт і завершіть роботу з комп'ютером.

Зміст звіту:

- мета роботи;
- завдання до лабораторної роботи;
- теоретичні основи;
- результати виконання завдань;
- висновки.

1.3 Контрольні запитання

- 1 Що таке Ethernet?
- 2 Що використовують як передавальне середовище в Ethernet?
- 3 Які переваги використання витой пари перед коаксіальним кабелем?
- 4 Виділіть основні стандарти створення сегменту мережі по Ethernet.
- 5 Якщо використовується прямий кабель, то в який порт іншого концентратора (комутатора) — звичний або перехресний — слід вставити другий коннектор RJ-45?
- 6 Чи можливо по мережі звернутися з одного комп'ютера, що належить до робочої групи **WORK1**, до ресурсів, які знаходяться на комп'ютері, що належить до робочої групи **WORK2**?
- 7 Що таке Wi-Fi?
- 8 Що використовують як передавальне середовище в Wi-Fi?
- 9 Назвіть переваги і недоліки Wi-Fi.

1.4 Перелік посилань

- 1 Побудова складної гібридної мережі [Електронний ресурс] / Режим доступу: www.ntgpk.com/content/student/polyakova/5.doc.

2 ЛАБОРАТОРНА РОБОТА №2 «НАЛАШТУВАННЯ IP-АДРЕСАЦІЇ І МАРШРУТИЗАЦІЇ»

Мета роботи - навчитися аналізувати правильність вибору масок підмереж і IP-адрес вузлів, мереж, широкомовлення; вручну налаштувати протокол TCP/IP і перевіряти його працездатність за допомогою різних утиліт, що входять до складу ОС Windows XP Professional.

2.1 Загальні теоретичні відомості

Оскільки переважна більшість сучасних мереж, не говорячи вже про глобальну мережу Інтернет, використовує стек протоколів TCP/IP, виключно важливим є чітке знання основ IP-адресації і маршрутизації. Без правильної настройки таких параметрів, як IP-адреса, маска підмережі, основний шлюз та інших, взаємодія комп'ютерів в мережі TCP/IP буде неможлива. Задача ще більш ускладнюється, якщо мережа складається з декількох сегментів, об'єднаних маршрутизаторами.

2.2 Завдання до лабораторної роботи

2.2.1 Аналіз правильності параметрів протоколу IP

При складанні завдань частково використані ресурси [1].

1 Перетворіть наступні IP-адреси і маски підмережі з десятичного в двійковий.

192.168.1.200	.	.	.
83.149.247.4	.	.	.
131.107.2.200	.	.	.
255.255.255.0	.	.	.
255.255.255.240	.	.	.

2 Проаналізуйте наведені нижче IP-адреси і відзначте ті з них, які можна використовувати для адресації вузлів. У третьому стовпці вкажіть, чому решта адрес є некоректними.

10.15.256.30		
169.254.25.38		
244.13.56.91		
0.0.0.0		
92.107.30.0		

127.16.25.30		
255.255.255.255		

3 Проаналізуйте наведені нижче маски підмережі і визначте правильні. У третьому стовпці вкажіть, чому решта масок є некоректними.

255.208.0.0		
255.255.255.0		
255.240.0.0		
255.255.255.252		
127.0.0.0		

4 Проаналізуйте наведені нижче IP-адреси і маски підмережі. Визначте адресу мережі і адресу вузла в кожному з вказаних випадків.

85.13.32.201	255.0.0.0		
131.13.32.201	255.255.0.0		
192.168.1.150	255.255.255.0		
192.168.1.150	255.255.255.128		

2.2.2 Налаштування IP-адрес і перевірка працездатності протоколу TCP/IP

1 Включіть комп'ютер і увійдіть до системи з обліковим записом, що входить до локальної групи «**Адміністратори**».

2 У меню **Пуск** клацніть правою кнопкою миші на пункті **Мережеве оточення** і виберіть в контекстному меню, що з'явилося, пункт **Властивості**.

3 У вікні **Мережеві підключення**, що відкрилося, виконайте подвійне клацання мишею значка **Підключення по локальній мережі**.

4 У вікні **Стан Підключення по локальній мережі** клацніть мишею на кнопці **Властивості**.

5 У вікні властивостей мережевого підключення в списку **Компоненти, використовувані цим підключенням** клацніть мишею на рядку **Протокол Інтернету (TCP/IP)**, а потім клацніть мишею на кнопці **Властивості**.

6 У вікні настройки параметрів протоколу IP переконайтеся, що радіокнопки **Одержати IP-адресу автоматично** і **Одержати адресу DNS-серверу** автоматично вибрані, після чого закрийте всі вікна.

7 У меню **Пуск** виберіть пункт **Виконати**.

8 У полі **Відкрити вікна Запуск програми** введіть команду **CMD** і клацніть мишею на кнопці **ОК**.

9 У вікні **Командний рядок** уведіть команду **IPCONFIG /ALL** і натисніть на клавіатурі клавішу **Enter**.

10 Уважно вивчіть відомості, виведені на екран командою **IPCONFIG**.

11 З'ясуйте у вашого партнера IP-адресу його комп'ютера, після чого виконайте команду **PING W.X.Y.Z**, де **W.X.Y.Z** — IP-адреса комп'ютера вашого партнера.

12 Якщо у видачі команди **IPCONFIG /ALL** (див. п. 10) є параметр Основний шлюз, то виконайте команду **PING W.X.Y.A**, де **W.X.Y.A** — IP-адреса основного шлюзу.

13 Виконайте команду **NETSH INT IP SET ADDR "Підключення по локальній мережі" STATIC 192.168.X.100 255.255.255.0 192.168.X.1 1**, де **X** — номер вашого комп'ютера (наприклад, 1 — для комп'ютера Comp1). Будьте уважні при її введенні (команду слід набрати в один рядок, дотримуючи пропуски). Добийтеся появи результуючого повідомлення ОК.

14 Виконайте команду **PING 192.168.Y.100**, де **Y** — номер комп'ютера вашого партнера (наприклад, 2 — для комп'ютера Comp2). Чому цього разу комп'ютер вашого партнера не відповідає?

15 Виконайте команду **PING 192.168.X.1**. Чому не відповідає основний шлюз?

16 Виконайте команду **NETSH INT IP SET ADDR "Підключення по локальній мережі" DHCP**, а потім перевірте взаємодію з іншими комп'ютерами в мережі.

17 Закрийте всі вікна, оформіть звіт і завершіть роботу.

Зміст звіту:

- мета роботи;
- завдання до лабораторної роботи;
- теоретичні основи;
- результати виконання завдань;
- висновки.

2.3 Контрольні запитання

- 1 Вкажіть 3 типи адрес в мережі Інтернет.
- 2 Що таке MAC-адреса?

- 3 Що таке IP-адреса?
- 4 Що таке DNS-ім'я?
- 5 Що називають маршрутизацією?
- 6 Що таке маска підмережі?
- 7 Що означає параметр IP-адреса автонастройки?
- 8 Що означають одержані від IP-адреси вашого партнера відповіді на команду PING?
- 9 Що означають відповіді на команду PING, одержані від IP-адреси основного шлюзу?

2.4 Перелік посилань

1. Настройка IP-адресації і маршрутизації [Електронний ресурс] / Режим доступу: www.ntgpk.com/content/student/polyakova/8.doc

3 ЛАБОРАТОРНА РОБОТА №3 «ПРОТОКОЛ TCP/IP. АНАЛІЗ МЕРЕЖЕВИХ КАДРІВ»

Мета роботи - навчитися аналізувати вміст переданих по мережі кадрів, виділяючи в них пакети протоколів різних рівнів.

3.1 Загальні теоретичні відомості

Взаємодія комп'ютерів в мережі підкоряється безлічі наборів правил (*протоколів*), які прийнято зіставляти з семирівневою еталонною моделлю OSI.

3.1.1 Стек протоколів TCP/IP

TCP/IP - збірна назва для набору (стека) мережеских протоколів різних рівнів, використовуваних в Інтернет. Особливості TCP/IP:

- відкриті стандарти протоколів, які розробляються незалежно від програмного і апаратного забезпечення;
- незалежність від фізичного середовища передачі;
- система унікальної адресації;
- стандартизовані протоколи високого рівня для поширених сервісів для користувачів.

Стек протоколів TCP/IP ділиться на 4 рівні (рис.3.1):

- прикладний,
- транспортний,
- міжмережевий,
- фізичний і канальний.

Пізніше була прийнята 7-ми рівнева модель ISO, але її не використовують. Дані передаються в пакетах. Пакети мають заголовок і закінчення, які містять службову інформацію. Дані більш верхніх рівнів вставляються в пакети нижніх рівнів (рис.3.2).

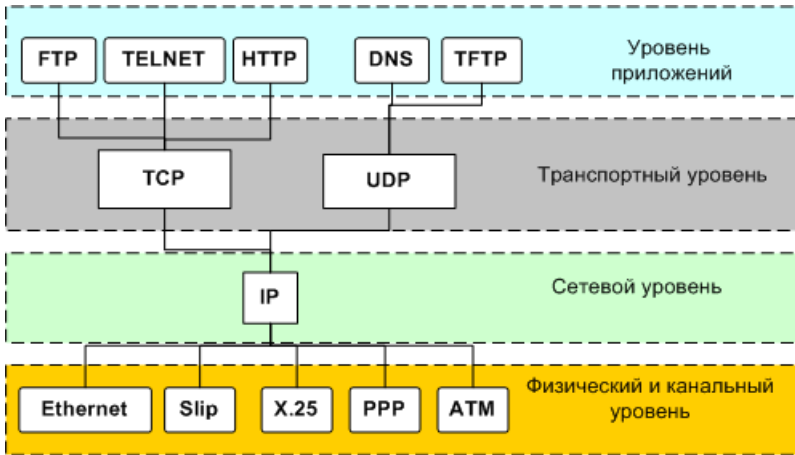


Рисунок 3.1 - Стек протоколів TCP/IP

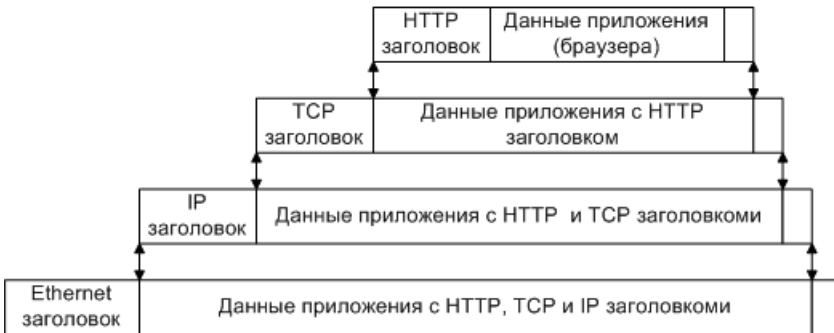


Рисунок 3.2 - Приклад інкапсуляції пакетів в стеку TCP/IP

3.1.2 Фізичний і каналний рівень

Стек TCP/IP не має на увазі використання будь-яких певних протоколів рівня доступу до середовища передачі і фізичних середовищ передачі даних. Від рівня доступу до середовища передачі потрібна наявність інтерфейсу з модулем IP, який забезпечує передачу IP-пакетів. Також необхідно забезпечити перетворення IP-адреси вузла мережі, на який передається IP-пакет, в MAC-адресу. Часто як рівень доступу до середовища передачі можуть виступати цілі протокольні стеки, тоді говорять про IP поверх ATM, IP поверх IPX, IP поверх X.25 і т.п.

3.1.3 Міжмережевий рівень і протокол IP

Основу цього рівня складає IP-протокол. **IP (Internet Protocol)** – інтернет протокол.

Перший стандарт IPv4 визначений в RFC-760 (DoD standard Internet Protocol J. Postel Jan-01-1980). Остання версія IPv4 - RFC-791 (Internet Protocol J. Postel Sep-01-1981). Перший стандарт IPv6 визначений в RFC-1883 (Internet Protocol, Version 6 (IPv6) Specification S. Deering, R. Hinden December 1995). Остання версія IPv6 - RFC-2460 (Internet Protocol, Version 6 (IPv6) Specification S. Deering, R. Hinden December 1998).

Основні задачі:

- адресація;
- маршрутизація;
- фрагментація датаграм;
- передача даних.

Протокол IP доставляє блоки даних від однієї IP-адреси до іншої.

Програму, яка реалізує функції того або іншого протоколу, часто називають модулем, наприклад, “IP-модуль”, “модуль TCP”.

Коли модуль IP одержує IP-пакет з нижнього рівня, він перевіряє IP-адресу призначення.

Якщо IP-пакет адресований даному комп'ютеру, то дані з нього передаються на обробку модулю вищого рівня (якому конкретно - вказано в заголовку IP-пакету).

Якщо ж адреса призначення IP-паketу - чужа, то модуль IP може ухвалити два рішення: перше - знищити IP-паket, друге - відправити його далі до місця призначення, визначивши маршрут проходження - так поступають маршрутизатори.

Також може бути потрібно на межі мереж з різними характеристиками розбити IP-паket на фрагменти (**фрагментація**), а потім зібрати в єдине ціле на комп'ютері-одержувачі.

Якщо модуль IP з будь-якої причини не може доставити IP-паket, він знищується. При цьому модуль IP може відправити комп'ютеру-джерелу цього IP-паketу повідомлення про помилку; такі повідомлення відправляються за допомогою протоколу ICMP, що є невід'ємною частиною модуля IP. Більш ніяких засобів контролю коректності даних, підтвердження їх доставки, забезпечення правильного порядку проходження IP-паketів, попереднього встановлення з'єднання між комп'ютерами протокол IP не має. Ця задача покладена на транспортний рівень.

3.1.4 Транспортний рівень

Протоколи транспортного рівня забезпечують прозору доставку даних між двома прикладними процесами. Процес, який одержує або відправляє дані за допомогою транспортного рівня, ідентифікується на цьому рівні номером, який називають номером порту. Отже, роль адреси відправника і одержувача на транспортному рівні виконує номер порту (або простіше - порт).

Аналізуючи заголовок свого паketу, одержаного від міжмережевого рівня, транспортний модуль визначає за номером порту одержувача, якому з прикладних процесів направлені дані, і передає ці дані відповідному прикладному процесу. Номери портів одержувача і відправника записуються в заголовок транспортним модулем, який відправляє дані; заголовок транспортного рівня містить також й іншу службову інформацію; формат заголовка залежить від використовуваного транспортного протоколу.

На транспортному рівні працюють два основні протоколи: UDP і TCP.

Протокол надійної доставки повідомлень TCP (Transfer Control Protocol) – протокол контролю передачі, його застосовують у тих випадках, коли потрібна гарантована доставка повідомлень. Основні особливості: встановлюється з'єднання; дані передаються сегмен-

тами; посилає запит на наступний пакет, указуючи його номер в полі **“Номер підтвердження” (AS)**, тим самим підтверджуючи отримання попереднього пакету; робить перевірку цілісності даних, якщо пакет битий - посилає повторний запит.

Протокол UDP (Universal Datagram Protocol) - універсальний протокол передачі даних, більш полегшений транспортний протокол, ніж TCP. Основні відмінності від TCP: немає з'єднання між модулями UDP; не розбиває повідомлення для передачі; при втраті пакету запит для повторної передачі не посилається.

3.2 Завдання до лабораторної роботи

У цій лабораторній роботі за допомогою програми — аналізатора пакетів демонструється, як мережева взаємодія (на прикладі обігу веб-браузера до веб-вузла) забезпечується узгодженою роботою цілого ряду протоколів.

1 Запустіть програму **Ethereal** (наприклад, подвійним клацанням миші на ярлику на вашому Робочому столі).

2 У меню вікна програми **The Ethereal Network Analyzer** виберіть в меню **File** пункт **Open**.

3 У вікні **Ethereal**, що відкрилося: **Open Capture File** виконайте подвійне клацання мишею в лівій частині вікна на значку жорсткого диска C:, а коли в правій частині вікна відкриється вміст диска, знайдіть і виконайте подвійне клацання мишею на папці **Setup**.

4 У папці **Setup** знайдіть файл **ToMS** і відкрийте його подвійним клацанням миші.

Примітка. У цьому файлі містяться перші 50 кадрів, захоплені при обігу за допомогою браузера Internet Explorer до веб-вузла корпорації Майкрософт (www.microsoft.com).

5 У вікні програми **Ethereal** (рис 3.3) зверніть увагу на три області: у верхній частині вікна наведено загальний список всіх захоплених кадрів, в середній частині — розшифровка кадру, виділеного у верхньому списку (на рис. 3.3 — першого кадру), а в нижній частині вікна вміст кадру представлений у вигляді так званих «необроблених даних» (у байтовому уявленні).

6 Клацніть мишею на значку «+» поряд з написом **Frame 1** в середній частині вікна, щоб розкрити докладну інформацію про кадр № 1. Зверніть увагу, що в даному кадрі містяться пакети протоколів *Ethernet*, *IP*, *UDP* і *DNS* (це видно в полі **Protocols in frame**).

7 Закрийте (клацанням миші на значку «-») розділ **Frame 1** і відкрийте розділ **Ethernet II**. Тут в *заголовку Ethernet* міститься інформація про *MAC-адреси* джерела і одержувача, а також про тип вкладеного протоколу (в даному випадку — *протоколу IP*).

8 Закрийте розділ **Ethernet II** і відкрийте розділ **Internet Protocol**. Зверніть увагу на такі найважливіші дані в *IP-заголовку*, як *IP-адреси* джерела і одержувача.

9 Закрийте розділ **Internet Protocol** і відкрийте розділ **User Datagram Protocol**. Відзначте інформацію, що міститься тут, про *но-*

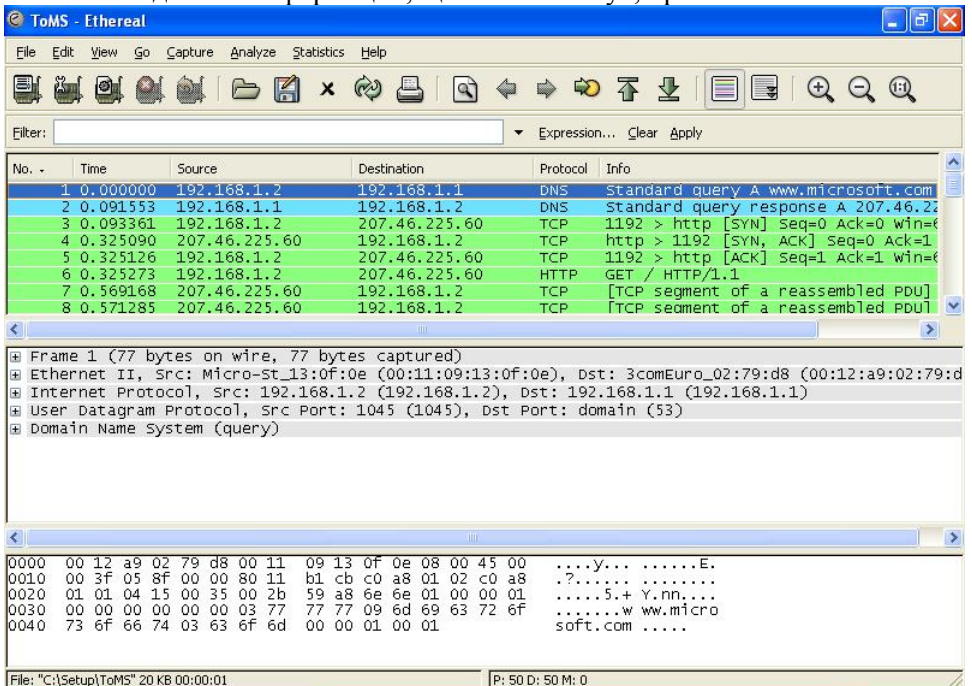


Рисунок 3.3 - Вікно програми Ethereal з відкритим файлом захоплення

мери портів, використаних протоколом UDP при обміні інформацією між комп'ютерами.

10 Закрийте розділ **User Datagram Protocol** і відкрийте розділ **Domain Name System (query)**. Зверніть особливу увагу на підрозділ **Queries (Запити)**, де міститься *ім'я веб-вузла*, яке потрібне перетворити в IP-адресу.

11 Проаналізуйте аналогічним способом другий кадр.

3.3 Контрольні запитання

- 1 Що таке стек протоколів TCP/IP? Які його особливості?
- 2 Назвіть 4 рівні стека протоколів TCP/IP.
- 3 Охарактеризуйте фізичний і канальний урівень стека протоколів TCP/IP.
- 4 Охарактеризуйте міжмережевий рівень і протокол IP. Які основні задачі даного рівня?
- 5 Які протоколи відносяться до транспортного рівня? Яке їх призначення?
- 6 Охарактеризуйте протокол TCP.
- 7 Що таке датаграма?
- 8 Охарактеризуйте протокол UDP.
- 9 Які протоколи і програми відносяться до рівня додатків?

4 ЛАБОРАТОРНА РОБОТА № 4 “ВИДАЛЕНЕ УПРАВЛІННЯ ПО ПРОТОКОЛУ SSH З ВИКОРИСТАННЯМ УТИЛІТИ PUTTY”

Мета роботи - ознайомитися з мережевим протоколом SSH і утилітою PuTTY. Навчитися дистанційно управляти операційною системою і передавати файли.

4.1 Загальні теоретичні відомості

SSH (англ. Secure Shell — «безпечна оболонка») — мережевий протокол прикладного рівня, який дозволяє здійснювати дистанційне управління операційною системою і передачу файлів. Схожий за функціональністю з протоколами **Telnetu rlogin**, але, на відміну від них, шифрує весь трафік, включаючи і передавані паролі. **SSH** допускає

вибір різних алгоритмів шифрування. **SSH-клієнти** і **SSH-сервери** є для більшості операційних систем.

PuTTY - це популярний **SSH-клієнт** і **Telnet** (**Telnet** той же **SSH**, тільки без зашифрованої передачі даних (пакетів)), тобто програма для безпечного підключення до віддаленого комп'ютера (або до серверу) і виконання на ньому різних команд. **PuTTY** веде log-файли, дозволяє налаштовувати шрифти, кольори і дозвіл консолі, допускає збереження в своїй пам'яті ключів авторизації, підтримує роботу через проксі-сервер. При цьому утиліта є безкоштовною в розповсюдженні.

Розповісти про всі команди Unix буде складно, тому напишемо лише декілька корисних команд:

man [ім'я команди] - видасть докладну інформацію по команді, наприклад: **man mv**

Для виходу з **man**, тобто із інструкції по команді, натисніть **q** (**Quit** - Вихід).

[ім'я команди] --help - також дозволить подивитися опис команди.

ls - список файлів;

ls -la - покаже всі файли (включаючи приховані), розмір файлів, власника і групу власника, права на них, дату останньої зміни;

ls -lha - те саме, що і попередня команда, тільки розмір файлів буде показаний в Кб.

ls -lha | less - дозволить проглядати файли посторінково (якщо їх багато);

cd [ім'я директорії] - перехід у вибрану директорію;

cd ../ - перехід на директорію вище;

cd ~ - перехід в кореневу директорію.

mv - перейменувати і перенести.

rm - видалити.

cp - копіювати.

> - очищення файлу. Наприклад, можна застосувати до файлів лівг. (**> access.log**, **> error.log**, **> combined.log**)

mc - запуск Midnight Commander, щось подібне до Нортон Командера, в якому зручно працювати з файлами, а також можливо працювати з ними по ftp (ftp усередині ssh)

chmod - установка прав на файл або директорію.

cat - об'єднує файл або декілька файлів, або введення із стандартного пристрою введення і виводить результат на стандартний пристрій виводу.

cat [ім'я файлу] - виведий на екран вміст файлу;

cat [ім'я файлу] | grep [шуканий рядок] - виведе на екран рядки файлу, включаючи шуканий рядок.

mkdir [ім'я директорії] - створення директорії (папки).

4.2 Завдання до лабораторної роботи

1 Для початку роботи запустіть файл **putty.exe**. Перед вами з'явиться вікно, представлене на рис. 4.1.

У полі **Host Name (or IP address)** уведіть ім'я вашого домена (наприклад, beget.ru) або **IP** залежно від висланих Вам реквізитів доступу по **SSH-протоколу**.

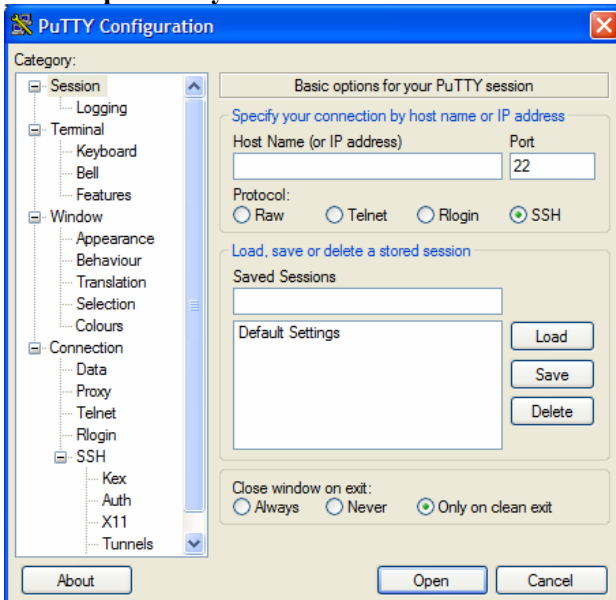


Рисунок 4.1 – Вікно програми **putty.exe**

Порт залишайте за умовчанням **22**. У полі **Saved Sessions** уведіть будь-яке ім'я сесії (конекта), наприклад, **beget.ru**, і натисніть **Save**.

2 Після цього увійдіть в SSH в меню зліва, і перед Вами з'явиться вікно на рис. 4.2. Виберіть протокол SSH версії **2 only**. Щоб з'єднатися з сервером, натисніть **Open**.

Навіть якщо Ви не виставили **2 only** і не написали назву конекта **Saved Sessions**, перед Вами з'явиться вікно вигляду, представленого на рис. 4.3.

3 У полі **login as** (рис. 4.3) введіть виданий викладачем Вам логін для доступу по SSH і тисніть **Enter**. Після чого з'явиться напис **Password**. Уведіть висланий Вам пароль для доступу по SSH.

Не лякайтеся, що під час введення пароля на екрані нічого не відображається (ні зірочок, нічого подібного). Просто після того, як Ви закінчили вводити пароль, натисніть Enter.

Якщо логін і пароль введені вірно, відбудеться підключення до серверу, і Ви потрапите в командний рядок Unix.

4 Перейдіть за допомогою SSH в директорію, вказану викладачем.

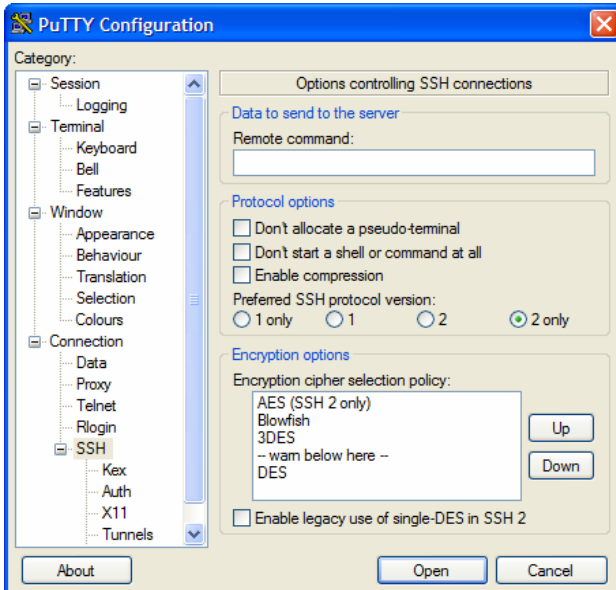


Рисунок 4.2 – Вікно програми **putty.exe**

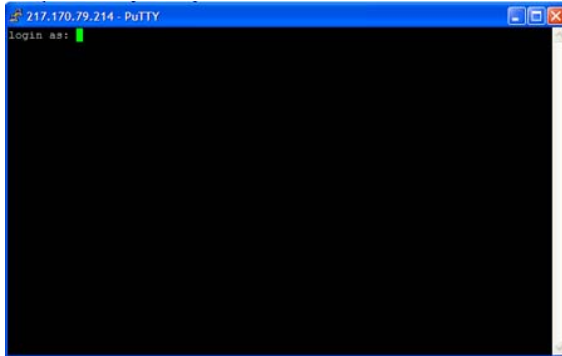


Рисунок 4.3 – Вікно програми **putty.exe**

- 5 Виведіть список файлів цієї директорії.
- 6 Створіть в цій директорії піддиректорію з Вашим прізвищем.
- 7 Скопіюйте в створену директорію текстовий файл з будь-яким змістом з Вашого робочого місця на видалений комп'ютер по SSH.
- 8 Виведіть зміст скопійованого файлу на екран.

Зміст звіту:

- мета роботи;
- завдання до лабораторної роботи;
- теоретичні основи;
- результати виконання завдань;
- висновки.

4.3 Контрольні запитання

- 1 Що таке SSH?
- 2 Що таке Telnet?
- 3 Опишіть основні настройки утиліти PuTTY.
- 4 Розкажіть про основні команди Unix, які можна використати з SSH.

5 ЛАБОРАТОРНА РОБОТА № 5 “СТВОРЕННЯ WEB-СТОРІНОК. ВИВЧЕННЯ МОВИ HTML”

Мета роботи - освоїти основні поняття мови HTML. Навчитися створювати WEB-сторінки і основні елементи.

5.1 Загальні теоретичні відомості

HTML - Hyper Text Markup Language.

Web-сторінка (документ HTML) є текстовим файлом на мові HTML формату *.htm або *.html, розміщеним у World Wide Web (WWW). WWW - Всесвітня Павутина - розподілена система доступу до гіпертекстових документів, існуюча в Інтернеті. Web-сторінка окрім тексту може містити гіпертекстові посилання, за допомогою яких можна переходити до інших Web-сторінок і проглядати їх. Web-сторінка може містити вставки у вигляді графіки, анімації, відеокліпів і музики. Для проглядання Web-сторінок можна скористатись, наприклад, MicroSoft Internet Explorer, Opera (переглядач або браузер).

Мова HTML дозволяє: 1) Створювати і редагувати Web-сторінки, зокрема свою домашню Web-сторінку, яку можна потім розмістити в Інтернеті; 2) Редагувати документи HTML, одержані з Інтернету, так щоб функціонували всі упроваджені в документ об'єкти (картинки, анімації і т.д.); 3) Створювати мультимедійні презентації, слайд-шоу, 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 Web-сторінок (або документів HTML):

– використання текстового редактора Блокнот (NotePad), вбудованого в Windows, і проглядання результатів за допомогою браузера. Цей найпростіший спосіб рекомендується початківцям. Технологія створення Web-сторінки така. У редакторі Блокнот створюється файл Web-сторінки, який зберігається з розширенням *.htm. Потім цей файл завантажується і є видимим програмою Internet Explorer. Для виклику редактора Блокнот з метою редагування файлу Web-сторінки під час її перегляду в Internet Explorer використовується пункт меню

Вид/Джерело. Після збереження файлу і виходу з Блокнота для переглядання відредагованої сторінки треба натиснути клавішу F5 або кнопку "Відновити" в панелі інструментів Internet Explorer;

– використання спеціальних редакторів документів HTML, наприклад, Hot Metal Light, Hot Dog Professional, MS Front Page, HTMLPad та ін.;

– використання WinWord, де створюється текст документа, який потім конвертується в HTML-формат.

Розглянемо основні поняття мови HTML.

Елемент – це конструкція мови HTML, або контейнер, що містить дані. Web-сторінка є набором елементів.

Тег - це стартовий і кінцевий маркери елемента. Теги визначають межі дії елементів і відділяють елементи один від одного. У тексті Web-сторінки теги полягають в кутові дужки, наприклад:

`<HTML>` Кінцевий тег завжди забезпечується косою межею: `</HTML>`.

Атрибути (параметри) тега – слідує за ім'ям тега і відділяються один від одного одним або декількома знаками табуляції або пропусками. Порядок запису атрибутів в тезі значення не має. Значення атрибуту слідує за знаком рівності, що стоїть після імені атрибуту. Якщо значення атрибуту – одне слово або число, то його можна просто вказати після знаку рівності, не виділяючи додатково. Всю решту значень необхідно брати в одинарні або подвійні лапки, особливо якщо вони містять декілька розділених пропусками слів.

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

Фрейм - область гіпертекстового документа із своїми смугами прокрутки.

Аплет - програма, яка передається на комп'ютер клієнта у вигляді окремого файлу і запускається при перегляданні Web-сторінки.

Скрипт - програма, включена до складу Web-сторінки для розширення її можливостей.

Завантаження (Download) - копіювання документа з Web-серверу на комп'ютер клієнта. *Upload* - копіювання документа з комп'ютера клієнта на Web-сервер - використовується при створенні власної Web-сторінки.

Розглянемо загальну структуру типового простого документа HTML:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
<TITLE>Назва документа</TITLE>
</HEAD>
<BODY>
Тут розташований текст самого документа HTML.
</BODY>
</HTML>

```

Згідно специфікацій HTML і XHTML тег DOCTYPE (що означає "оголошення типу документа") повідомляє валідатору, яку саме версію (X) HTML ви використовуєте в своїй сторінці. Цей тег має завжди знаходитися в першому рядку кожної сторінки. Тег DOCTYPE - ключовий компонент web-сторінок, що претендують на відповідність стандартам: без нього ваш код і CSS не пройдуть перевірку валідатором.

Тег DOCTYPE також важливий для правильного відображення і роботи сторінки в браузерях, відповідних стандартам (Mozilla, IE5/Mac, і IE6/Win).

Тег DOCTYPE, в атрибутах якого вказується повний URI (повна web-адреса), повідомляє браузерам, що сторінку потрібно вивести з дотриманням певного стандарту або підвиду цього стандарту.

Нижче перелічені повні варіанти тега DOCTYPE для різних потреб:

Стандарт HTML 4.01 Strict (строгий), Transitional (перехідний), Frameset (з фреймами)

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">

```

Стандарт XHTML 1.0 Strict (строгий), Transitional (перехідний), Frameset (з фреймами)

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Стандарт XHTML 1.1 DTD

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

5.2 Завдання до лабораторної роботи

1 Створіть простий html-документ **index_ВашеІмя.html** із заголовком «Мій перший html-документ», фон документа – кораловий (coral), колір тексту – жовтий (yellow), колір посилань – індиго (indigo). Між тегами <BODY> і </BODY> введіть будь-який текст для проглядання колірних рішень (колір посилань Ви поки не побачите).

2 Уведіть заголовки і текст.

2.1 Встановіть розмір базового шрифту рівний трем.

2.2 Створіть центральний заголовок другого рівня «Операційні системи».

2.3 Потім уведіть визначення:

«Операційна система – це програма, яка управляє апаратними і програмними засобами комп'ютера, призначеними для виконання задач користувача».

2.4 До і після визначення накресліть горизонтальні лінії сірого кольору.

2.5 Створіть заголовок третього рівня «Операційна система Linux» з вирівнюванням по центру.

2.6 Прокресліть червону горизонтальну лінію завтовшки 20 пікселів, яка займає 30% ширини екрану.

2.7 Уведіть абзаци:

Перший абзац – вирівнювання по правому краю: колір тексту – базовий, розмір тексту – на 1 більше базового, всі англійські слова – курсивом, всі заголовні літери – напівжирним:

Linux – це операційна система для IBM-сумісних персональних комп'ютерів і робочих станцій. Це розрахована на багато користувачів ОС з мережевою віконною графічною системою X Window System.

Другий абзац – вирівнювання по лівому краю: колір тексту – синій, гарнітура тексту – Comic Sans MS, всі англійські слова зміщені

вниз, всі заголовні літери – різними кольорами, решта ефектів показана в тексті:

ОС Linux підтримує стандарти відкритих систем і протоколи мережі **Інтернет** і сумісна з системами Unix, DOS, ~~MS Windows~~. Всі компоненти системи, включаючи початкові тексти, розповсюджуються з ліцензією на вільне копіювання і установку для *необмеженого числа користувачів*.

3 Створіть гіперпосилання:

3.1 Створіть новий документ **lesson3.html**, що містить посилання на вже створений вами раніше документ **index_ВашеІм'я.html**. Посиланням є текст «Перейти на головну сторінку «Операційні системи». Сторінка, задана посиланням, відкривається в новому вікні.

3.2 Тепер в **lesson3.html** створіть посилання «ОС LINUX», яке перейде на закладку, розміщену на сторінці index_ВашеІм'я.html (до цього необхідно створити закладку у файлі index_ВашеІм'я.html). При натисненні на посилання має відкритися документ index_ВашеІм'я.html відразу на заголовку «Операційна система Linux».

3.3 За допомогою створення маркірованого і впорядкованого списків, уведіть наступний текст:

Тема 2. Проектування інформаційних систем в економіці

1. Принципи проектування ІСЕ:

- визначення мети;
- визначення вимог до системи (визначення меж об'єкту);
- визначення функціональних підсистем, їх структури і задач у загальній системі управління;
- виявлення і аналіз зв'язків між підсистемами;
- встановлення порядку функціонування і розвитку всієї системи в цілому.

2. Поняття відкритої системи.

3. Поняття життєвого циклу ІС. Етапи проектування.

4. Методи і способи проектування ІС.

3.4 Накресліть таблицю: колір фону першого рядка – салатовий, другого рядка – бузковий.

ХАРАКТЕРИСТИКА ОС			
ПРОЕКТИВНІ ОС		ПРОЕКТИВНІ ОС	
+	+	+	+
Активність людини	Активність людини	Активність людини	Активність людини
	Необхідно освоювати	Орієнтація на новачка	

3.5 У документ **index_ВашеІм'я.html** додайте Рисунок розміром 100×100 пікселів, розташований по центру сторінки на відстані 20 по горизонталі. Альтернативний текст: «Тут має бути Рисунок Ім'я_Рисунка».

3.7 Створіть біжучий рядок червоного кольору «РОЗРОБКА МОЄЇ ПЕРШОЇ ВЕБ-СТОРІНКИ ЗАКІНЧЕНА» заввишки 20, яка пробігає зліва направо 5 разів.

3.8 Створіть фоновий звук при завантаженні файлу.

Зміст звіту:

- мета роботи;
- завдання до лабораторної роботи;
- теоретичні основи;
- початкові тексти і скріншоти створених WEB-сторінок;
- висновки.

5.3 Контрольні запитання

- 1 Що таке Web-сторінка (документ HTML)?
- 2 Дайте визначення поняттям: елемент, тег, атрибути тега, гіперпосилання, фрейм, аплет, скрипт, завантаження.
- 3 Як можна представити загальну структуру типового простого документа HTML?
- 4 Вкажіть загальні принципи створення маркірованих і впорядкованих списків.
- 5 Вкажіть загальні принципи створення таблиць.
- 6 Що таке фрейм? У яких випадках його використовують?

6 ЛАБОРАТОРНА РОБОТА №6 “СТВОРЕННЯ ФОРМ. ВІКНА ДІАЛОГУ. ПЕРЕВІРКА ПРАВИЛЬНОСТІ ЗАПОВНЕННЯ ФОРМ ЗА ДОПОМОГОЮ JAVASCRIPT”

Мета роботи: формування умінь створення різних форм на WEB-сторінці, використання JavaScript для перевірки правильності заповнення форм і створення вікон діалогу.

6.1 Загальні теоретичні відомості

Практично всі сучасні WWW браузеры дозволяють користувачу, заповнивши спеціальну форму, виконувати деякі дії на вашому WWW-сервері. Коли форма інтерпретується WEB-браузером, створюються спеціальні екранні елементи GUI, такі, як поля введення, кнопки підтвердження (check-boxes), кнопки вибору альтернатив (radio-buttons), випадні меню, скрольовані списки, кнопки і т.д. Коли користувач, заповнивши форму, натискає кнопку «Підтвердження» (SUBMIT – спеціальний тип кнопки, який задається при описі документа), інформація, введена користувачем у форму, посилається WWW-серверу для обробки і передачі іншим програмам, які працюють під сервером, відповідно до CGI (Common Gateway Interface) інтерфейсу. За допомогою мови написання сценаріїв JavaScript (VBScript) можна перевірити правильність заповнення форми до її відправки серверу, обробити натиснення інших (відмінних від SUBMIT) кнопок і багато інших подій.

Всі форми починаються тегом **<FORM>** і завершуються тегом **</FORM>**. Між цими тегами і розміщуються елементи форми, такі як кнопки, поля введення і т.д. Окрім елементів форм там можна розмішувати будь-які тексти і майже будь-які теги **HTML**.

Елементи форми:

- тег **<TEXTAREA>** використовують для того, щоб дозволити користувачу вводити більше одного рядка інформації (вільний текст). Область введення починається тегом **<TEXTAREA>** і закінчується тегом **</TEXTAREA>**;

- тег **<SELECT>** використовують для організації вибору з фіксованого числа наперед певних значень. За допомогою цього тега мо-

жна представити список, список з можливістю вибрати більш, ніж один варіант і спадаюче меню;

– тег **<INPUT>** використовують для багатьох речей в HTML.

Що конкретно він означає, визначається його параметром **TYPE**.

Форми можна також використовувати для відправки файлів на сервер. Чудес, проте, не буває, і оскільки дана можливість вимагає підтримки отримання файлів WEB-сервером, то, відповідно, необхідно, щоб сервер підтримував отримання файлів.

Об'єктна модель браузера містить 12 об'єктів, які дозволяють звертатися до частини браузера або частини сторінки за допомогою мови JavaScript:

– Document – надає можливість доступу до компонентів документа HTML;

– Event – надає можливість доступу до властивостей подій, коли вони відбуваються;

– History – надає інформацію про адреси, які клієнт відвідав;

– Location – надає інформацію про адресу поточного документа;

– MimeType – надає інформацію про тип MIME;

– Navigator – дозволяє звертатися до властивостей браузера;

– Selection – відображає поточне виділення документа;

– Style – представляє конкретний елемент стилю в таблиці стилів;

– TextRange – відображає розділи тексту, які формує документ HTML;

– Screen – надає інформацію про монітор і систему виводу інформації клієнта;

– Window – надає властивості, методи і події, пов'язані з вікном браузера;

– StyleSheet – представляє всі елементи стилю усередині таблиці стилів.

Окрім об'єктів в об'єктній моделі вводяться наступні поняття:

– **методи** – способи роботи з об'єктами, наприклад: закрити вікно. По суті це функція, асоційована з об'єктом *object.methodname*;

– **події** – об'єкт повідомляє нас, що щось відбулося, наприклад: елемент стає активним;

– **властивості** – властивості об'єкту, наприклад: ім'я і розміри вікна.

6.1.1 Вікна діалогу. Об'єкт Window

Відкриття вікна

Синтаксис:

`window.open` (“URL або URI”, “ім'я вікна”, “властивості вікна”).

Наступний оператор створює вікно, яке відображає вміст сторінки <http://zntu.edu.ua>: `window.open("http://zntu.edu.ua")`.

При створенні вікна ви можете також надати ім'я, в даному випадку – `zntuWindow`, для звернення до вікна як до цілі/target при відправці форми або при переході по гіперпосиланню:

`window.open("http://zntu.edu.ua", "zntuWindow")`.

Ім'я вікна не потрібне при створенні вікна. Але вікно обов'язково має мати ім'я, якщо Ви хочете звернутися до нього з іншого вікна.

При відкритті вікна Ви можете специфікувати атрибути, такі як висота/height і ширина/width, панель утиліт/toolbar, адресний рядок/location field або смуги прокрутки/scrollbars. Наступний оператор створює вікно без панелі утиліт, але із смугами прокрутки:

`window.open ("http://zntu.edu.ua", "zntuWindow", "toolbar=no, scrollbars=yes")`

Для того, щоб функція відпрацьовувалася при натисненні мишкою на елементі документа, використаємо подію **onClick** об'єкту Document.

Приклад: відкриємо посилання в новому вікні

```
<a href=""onClick="window.open('http://zntu.edu.ua', 'zntuWindow', 'left=300,top=300,width=200,height=400,toolbar=no,menubar=no,location=no,directories=no')">
```

Відкрити нове вікно

```
</a>
```

Теж саме можна зробити за допомогою кнопки:

```
<input type="button" value="Відкрити нове вікно"
```

```
onClick="window.open('http://zntu.edu.ua', 'zntuWindow', 'left=300,top=300,width=200,height=400,toolbar=no,menubar=no,location=no,directories=no')">
```

Закриття вікна. Наступний оператор закриває поточне вікно:

`window.close()`

6.1.2 Застосування JavaScript у веб-сторінках

Перевірка правильності заповнення форм за допомогою JavaScript. За допомогою JavaScript можна контролювати правильність заповнення форм. Приклад перевірки нижченаведеної форми на наявність порожніх полів при натисненні на кнопку “Відправити”:

Ім'я		Імена полів <input name="name" size="20" type="text"/> <input name="email" size="20" type="text"/> <input cols="20" name="Comment" rows="3" type="text"/>
E-mail		
Коментарій		

```
<script language="JavaScript">
<!--
function CheckForm(UserForm)
{
    var is_ok = true;
    if (UserForm.name.value == "")
    {
        is_ok = false;
        alert("Уведіть ваше ім'я!!");
        UserForm.name.focus();
    }
    if (UserForm.Comment.value == "")
    {
        is_ok = false;
        alert("Уведіть ваші коментарі!");
        UserForm.Comment.focus();
    }
    if (UserForm.email.value == "")
    {
        is_ok = false;
        alert("Уведіть ваш e-mail!");
        UserForm.email.focus();
    }
    return is_ok;
}
```

```

}
// -->
</script>

```

Виклик функції в тезі <FORM> при натисненні кнопки “Відправити”

```
<form onSubmit = "return CheckForm(this) ">
```

Можна виділити наступні етапи спільної роботи скриптів і форм:

а) створюємо форму з вказівкою атрибутів NAME або ID в самій формі і у всіх її елементах, через які функції скрипта взаємодіють з елементами форми;

б) створюємо функцію, яка виконує необхідні дії, використовуючи тег <script> (див. [A13](#))

```

<script language="JavaScript">
function name_function()
{
    //тіло функції
}

```

```
</script>
```

Якщо необхідно, щоб функція name_function() скрипта викликала періодично, то в тіло функції необхідно додати функцію setTimeout('name_function()',ms), де ms – число мілісекунд, через яке необхідно викликати функцію name_function().

в) викликаємо функцію в тілі документа

```

<script language="JavaScript">
    name_function();
</script>

```

Виведення системного часу в документ:

```

<SCRIPT LANGUAGE="JavaScript">
    current_date = new Date();
    document.write("<b>Время:</b> " + current_date.getHours() +
        "." + current_date.getMinutes() + "." + current_date.getSeconds() +
        ".");
</SCRIPT>

```

6.2 Завдання до лабораторної роботи

1 Для всіх варіантів: на сторінці мають бути форми кнопок “Відправити поштою” і “Очистити форму”.

Варіант 1. Створити форму для введення інформації про людину. Повинна вводитися наступна інформація:

Прізвище	Просте текстове введення
Ім'я	Просте текстове введення
По батькові	Просте текстове введення
Вік	Вибір з варіантів "до 7", "від 7 до 17", "від 17 до 33", "від 33 до 65" і "65 і старше"
Стать	Має бути "радіо-кнопка"
Освіта	Має бути "чек-бокс"

Варіант 2. Створити форму для введення інформації про людину. Повинна вводитися наступна інформація:

ПІБ	Просте текстове введення
Рік народження	Уводиться текстом. Заборонити введення більш ніж чотирьох символів.
Стать	Має бути "радіо-кнопка"
Громадянство	Вибір із списку держав
Інтереси	Має бути "чек-бокс"

Варіант 3. Створити форму для введення адреси. Має вводитися наступна інформація:

Країна	Вибір із списку. За умовчанням - Україна
Поштовий індекс	Просте текстове введення, не більше шести символів
Місто	Просте текстове введення
Вулиця номер будинку	Просте текстове введення
Вказівка на те, що адреса службова/домашня	Має бути "радіо-кнопка"

Варіант 4. Створити форму для введення замовлення на комп'ютер. Має вводитися наступна інформація:

Процесор	Вибір із списку. За умовчанням Intel
-----------------	--------------------------------------

	Pentium IV
Пам'ять	Просте текстове введення
Необхідний мережевий адаптер	Має бути "радіо-кнопка"
Необхідний модем	Має бути "чек-бокс"
Додаткові вимоги	Багаторядкове поле введення

Варіант 5. Створити форму для замовлення товарів. Форма має пропонувати замовити відразу декілька товарів. Для кожного слід указувати кількість. Один раз слід вказати адресу доставки. Має вводитися наступна інформація:

Найменування товару	Вибір із списку. Зробіть декілька однакових списків, щоб можна було замовити відразу декілька різних товарів.
Кількість	Вводиться текстом. Слід зробити окреме поле для введення кількості, напроти кожного списку товарів.
Адреса доставки	Текст.

Варіант 6. Створити форму для запиту пароля. Має вводитися наступна інформація:

Ім'я користувача	Вводиться текстом.
Ресурс	Вибір із списку. Передбачте декілька ресурсів. Наприклад, "Проглядання бази даних", "Редагування бази даних" і т.д. За умовчанням хай буде "Повний доступ".
Пароль	Стандартне поля для введення пароля.
Файл	Можливість вибрати необхідний файл

Варіант 7. Створити форму для замовлення на ремонт комп'ютера. Має вводитися наступна інформація:

Ім'я користувача	Вводиться текстом.
Адреса	Вводиться текстом.
Процесор	Вибір із списку. За умовчанням Intel Pentium IV
Гарантія	«Радіо-кнопки» - Так, Ні
Характер несправно-	Багаторядкове поле введення

сті	
-----	--

Варіант 8. Створити форму для замовлення на ремонт автомобіля. має вводиться наступна інформація:

Ім'я власника	Вводиться текстом.
Адреса	Вводиться текстом.
Модель	Вибір із списку. За умовчанням "ВАЗ 2101"
Коробка передач	Автомат, Механічна (радіо-кнопка)
Характер несправності	Багаторядкове поле введення

Варіант 9. Створити форму для замовлення автомобіля. Увести наступну інформацію:

Ім'я клієнта	Уводиться текстом.
Адреса	Уводиться текстом.
Модель	Вибір із списку. Можна вибрати відразу декілька моделей.
Колір	Вибір із списку. Можливо вибрати відразу декілька кольорів.
Тюнінг	Три чек-бокси

Варіант 10. Створити форму для пошуку в каталозі бібліотеки. Увести наступну інформацію:

Тема	Вибір із списку. Можна вибрати відразу декілька тем.
Автор	Уводиться текстом.
Назва	Уводиться текстом.
Роки видання	До 2000, від 2000 до 2002, від 2003 до 2005 (радіо-кнопки)

Варіант 11. Створити форму для реєстрації нового користувача в системі. Увести наступну інформацію:

Ім'я	Текст.
e-mail	Текст.
Стать	Радіо-кнопки
Місто	Спадаючий список
Пароль	Поле введення пароля.
Яку інформацію хочете одержувати	Чек-бокси

Варіант 12. Створити форму для введення інформації про те, які предмети хотів би вивчати студент. Предмети мають бути перераховані поодиночці на рядку і перед назвою кожного має стояти "checkbox". Спочатку два основні предмети мають бути вибрані і доступ до них заборонений.

Варіант 13. Створити форму, в якій є дві незалежні серії радіокнопок. Наприклад, для введення улюбленого телеканалу. Увести наступну інформацію:

ПІБ	Просте текстове введення
Місто	Спадаючий список
Канал	Серія радіо-кнопок.
Передача	Серія радіо-кнопок. Наприклад, "Новини", "Серіали", "Мультфільми", і т.д.

Варіант 14. Створити форму для введення інформації про студента. Увести наступну інформацію:

Прізвище	Просте текстове введення
Ім'я	Просте текстове введення
По батькові	Просте текстове введення
Стать	Має бути серія "радіо-кнопок"
Курс	Вибір із списку
Прислати резюме	Огляд і вибір файлу

Варіант 15. Створити форму для введення відомості нарахування зарплати. Відомість розрахована на бригаду з 3 чоловік. Для кожної людини ввести наступну інформацію:

ПІБ	Просте текстове введення
Оклад	Просте текстове введення. Обмежте розмір рядка, що вводиться, розумним числом.
Чи основне місце роботи	Чек-бокс
Відділ	Серія радіо-кнопок

Варіант 16. Створити форму для введення бухгалтерського проведення. Увести наступну інформацію:

Рахунок-Дебет	Вибір із спадаючого меню.
Рахунок-Кредит	Вибір із спадаючого меню.
Сума	Просте текстове введення. Обмежте розмір рядка, що вводиться, розумним числом.
Особистий код бухгалтера	Вибір із спадаючого меню.
Пароль	Поле для введення паролів

Варіант 17. Створити форму для введення інформації про бажаний для клієнта варіант міжміського обміну. Увести наступну інформацію:

ПІБ	Просте текстове введення
Адреса	Просте текстове введення
Кількість кімнат у клієнта	Спадаюче меню.
Бажані міста	Список з можливістю вибору декількох варіантів.
Бажана кількість кімнат	Спадаюче меню.
Поверховість	Серія радіо-кнопок. Наприклад, "Будь-яка", "Окрім першого", "Окрім останнього", "Окрім першого і останнього", і т.д.
Додаткові умови	Багаторядкове текстове поле.

Варіант 18. Створити форму для введення інформації про бажаний для клієнта варіант внутрішньоміського обміну квартири. Увести наступну інформацію:

ПІБ	Просте текстове введення
Адреса	Просте текстове введення
Кількість кімнат у клієнта	Спадаюче меню.
Бажані райони	Список з можливістю вибору декількох варіантів.
Бажана кількість кімнат	Спадаюче меню.

Поверховість	Список з можливістю вибору декількох варіантів.
Додаткові умови	Багаторядкове текстове поле.

Варіант 19. Створити форму для пошуку відповідного будинку в базі даних ріелтерської фірми. Увести наступну інформацію:

Площа	Уводиться текстом.
Поверховість	Спадаюче меню.
Район	Список з можливістю вибору декількох варіантів
Наявність комунікацій	Серія чек-боксів, наприклад, "Водопровід", "Телефон", і т.д.
Додаткові вимоги	Багаторядкове поле введення

Варіант 20. Створити форму для пошуку відповідних матеріалів в бібліотеці. Увести наступну інформацію:

Ключові слова	Уводиться текстом.
Рік публікації	Вибір діапазонів із спадаючого меню.
Характер публікації	Список з можливістю вибору декількох варіантів, наприклад, "Стаття в журналі", "Дисертація", "Монографія" і т.д.
Мова	Список з можливістю вибору декількох варіантів.
Доставка	Серія радіо-кнопок, наприклад, "Підняти в читальний зал", "Доставити на абонемент", і т.д.

2 Зробіть, щоб форми завдання 1 відкривалися в новому діалоговому вікні. Потрібно обов'язково задати: відступ зліва, відступ зверху, ширину, висоту, не відображення панелі інструментів, не відображення панелі меню, відображення рядка стану, не відображення адресного рядка, не відображення кнопок браузера, відображення смуг прокрутки.

Зробіть перевірку заповнення для кожної з форм завдання 1.

Використовуючи JavaScript, зробіть на першій сторінці виведення поточної дати і виведення поточного часу.

Виведіть у форму годинник, що йде.

Зміст звіту:

- мета роботи;
- завдання до лабораторної роботи;
- теоретичні основи;
- початкові тексти і скріншоти, створених WEB-сторінок з формами;
- висновки.

6.3 Контрольні запитання

- 1 Що таке форми і для чого їх використовують?
- 2 Як створюють форми?
- 3 Форма TEXTAREA.
- 4 Форма SELECT і OPTION.
- 5 Форма INPUT.
- 6 Відправка файлів на сервер.
- 7 Що таке об'єктна модель браузера?
- 8 Дайте визначення основним поняттям об'єктної моделі: методи, дії, властивості.
- 9 Вкажіть синтаксис написання JavaScript.

7 ЛАБОРАТОРНА РОБОТА № 7 “ТЕХНОЛОГІЯ CSS. РОЗМІТКА СТОРІНОК ЗА ДОПОМОГОЮ ТАБЛИЦЬ, ШАРІВ (DIV) і CSS”

Мета роботи: вивчити основи технології CSS на прикладі розробки власного сайту. Зробити розмітку сторінок за допомогою шарів (DIV) і технології CSS.

7.1 Загальні теоретичні відомості**7.1.1 Каскадні таблиці стилів**

Каскадні таблиці стилів або CSS (від англійської Cascading Style Sheets), є наслідком подальшого розвитку HTML і дають можливість перейти на наступний рівень подання інформації. Таблиці стилів дозволяють розділити смисловий зміст сторінки та її оформлення.

У перших версіях стандарту HTML не було передбачено ніяких засобів для управління зовнішнім виглядом інформації. Загальна концепція гіпертексту була направлена на доступність інформації для будь-яких пристроїв, здатних відтворювати текст. Для розмітки рекомендувалося використовувати тільки логічні теги, визначаючи заголовки, підзаголовки, списки, абзаци, цитати і т.д., тобто ті елементи, які й складають структуру документа. Інтерпретація ж зовнішнього вигляду залишалася повністю на совісті останнього терміналу.

З того часу багато що змінилося, і стандарт HTML втратив первинну гнучкість. Спочатку Netscape додав “поліпшені теги”, які дозволили ширше управляти зовнішнім виглядом інформації, яка подається. Нововведення прижилося, і всі розширення Netscape стали стандартом de facto. Потім точно так вчинила Microsoft. Коли спохопилися, то HTML був жахливою сумішшю логічних і оформлювальних тегів, несумісних розширень і повністю перестав відповідати первинній концепції - представляти інформацію на будь-якому пристрої незалежно від його характеристик з виводу інформації.

Тоді було зроблено широкомасштабну стандартизацію, внаслідок чого на світ з'явився стандарт HTML 3.2. Він не був революційним, а лише розставив по місцях всі нововведення і узагальнив рекомендації для виробників браузерів. Революційні зміни були введені в новому стандарті - HTML 4.0, або, як його стали називати, Dynamic HTML. В обіг були введені шари, таблиці стилів і універсальна об'єктна модель браузера.

Як же була розв'язана проблема з поданням зовнішнього вигляду інформації? У цьому і є революційність підходу. Все оформлення рекомендується винести в зовнішній стилевий файл. Основна ж сторінка міститиме тільки інформацію і посилання на необхідні стилі.

При показі сторінки конкретному пристрою має задіюватись відповідна випадку таблиця стилів. Для стільникового телефону і монітора комп'ютера вони, зрозуміло, мають бути різними. У першому випадку ми використовуємо мінімальне оформлення, яке дозволить представити інформацію найбільш оптимально і компактно. У другому ж випадку в нашому розпорядженні є все багатство шрифтового і кольорового оформлення.

Таблицю стилів потрібно написати всього один раз при створенні сайту для кожного з пристроїв, на якому планується виведення інформації. До того ж таблиця стилів може бути єдиною для цілого

сайту, і, отже, не потрібно буде повторювати одні й ті самі описи стилів на кожній із сторінок.

Розміщення всієї стильової інформації в одному зовнішньому файлі відкриває нам й інші корисні можливості - адже змінивши вміст тільки одного (!) стильового файлу, ми можемо за лічені секунди змінити весь дизайн сайту. Причому ніяких інших переробок не знадобиться. Зрозуміло, це вірно лише в тому разі, якщо спочатку сайт був спроектований вірно.

Підключення таблиць стилів

Для здійснення цієї задачі ми можемо скористатися одним з 3-х пропонованих методів:

- зовнішній файл,
- inline-опис,
- опис в секції заголовка.

inline-опис або описи, вбудований в тег:

```
<p style="color:red; text-align:center;">
```

Цей текст перевизначений стилем

```
</p>
```

За допомогою додаткового атрибуту style ми можемо визначити потрібні нам стильові параметри в будь-якому тезі. Це найлегший спосіб, і діє він в межах лише одного тега. Але уявіть, наскільки збільшиться розмір файлу, і наскільки незручно буде його виправляти, якщо ми вказуватимемо стиль у кожного тега. Цей спосіб не дуже відрізняється, наприклад, від прямого опису зовнішнього вигляду за допомогою тега .

Опис в секції заголовка

Його дія розповсюджується на всю сторінку. Визначення стилів відбувається за допомогою класів, які є списками з визначенням всіх необхідних параметрів оформлення.

При використанні цього методу опис стилів необхідно розмістити в секції заголовка:

```
<head> ....
```

```
<style type="text/css">
```

```
<!--
```

```
.header {
text-align :center;
font-size : 27pt;}
```

```
.red { color : red; }
-->
</style>
</head>
```

Тепер ці стилі можна застосовувати в будь-якому місці html-коду. Для цього використовують наступну конструкцію:

```
<p class=header>Цей текст написаний стилем header<p>
  <p class=red> Цей текст написаний червоним кольором<p>
```

Окрім визначення нових класів ми також маємо нагоду перевизначати стандартні теги. Наприклад, тег <p>:

```
<style type="text/css">
<!-- p { text-align : center; font-size :12pt;}
--> </style>
```

Тепер весь текст, укладений в теги <p></p>, виглядатиме так, як визначено даним стилем. Це дуже зручно і дозволяє легко адаптувати вже існуючі сторінки до використання стилів. Крім того, це декілька зменшує об'єм файлу за рахунок відсутності зайвих атрибутів class.

Винесення опису стилів у зовнішній файл

Діапазон його дії тягнеться на всі файли, в які включено опис. Це спосіб найбільш відповідає концепції HTML 4.0. У разі, якщо нам потрібно буде змінити зовнішній вигляд сайту, то буде достатньо скоректувати лише один цей файл. Порівняйте його з попередніми способами. У одному з них доведеться міняти опис на кожній сторінці, а в іншому навіть більше того - біля кожного тега, що, зрозуміло, все ускладнює.

Як же здійснюється упровадження зовнішнього файлу? Спершу створюється стильовий файл з описом всіх потрібних нам класів (mystyle.css):

```
.header { text-align : center; font-size : 27pt;}
.red { color :red; }
p { text-align : center; font-size : 12pt;}
```

А потім посилання на нього впроваджується в документ за допомогою тега <link>:

```
<head> .... <link rel="stylesheet" type="text/css"
href="css/mystyle.css" title="MyStyleSheet"> .... </head>
```

Це найзручніший спосіб, і для основної таблиці стилів рекомендується користуватися саме ним.

Каскадність стилів

Каскадність полягає у тому, що стилі можуть перевизначатися. Наведений вище список способів упровадження стилів відповідає порядку перевизначення. Розташований нижче спосіб може перевизначати вищерозташований.

Наприклад, ми визначили в зовнішньому стильовому файлі, що текст у тезі <p> має бути написаний за допомогою шрифту заввишки 10 пунктів. Але якщо в заголовку сторінки ми додатково вкажемо, що той самий текст у тезі <p> має бути написаний шрифтом у 12 пунктів, то текст буде виведений саме таким кеглем - тобто стиль в заголовку сторінки перевизначив стиль у зовнішньому файлі.

Синтаксис CSS

Опис кожного класу робиться за допомогою конструкції, подібній цієї:

```
.small { font-size: 9pt; }
```

Спочатку вказується ім'я класу - воно може бути довільним, але бажано все-таки давати осмислену назву. Далі у фігурних дужках перераховуються всі необхідні параметри для даного класу. Параметри відділяються один від одного крапкою з комою.

Зазначимо, що ім'я класу починається з крапки або символу #, і в такий спосіб визначає універсальний клас, тобто такий, який може бути застосований до будь-якого тегу. І робиться це за допомогою наступної конструкції:

```
<p class=small> Накладаємо стиль на цей текст</p>
```

Існують універсальні класи і, так звані, тегові класи:

```
p.small { font-size: 9pt; }
```

Клас, визначений в такий спосіб, спрацює тільки в тому тезі, для якого він призначений, а для всіх інших буде проігнорований.

Ми можемо визначати параметри не тільки для одного тега, але і відразу для декількох. Для цього у визначенні стилю достатньо перерахувати їх через кому:

```
p, td { font-size: 9pt; color:green; }
```

Такий прийом називають угрупованням, і в даному випадку ми визначили і для <p>, і для <td> однаковий розмір і колір тексту.

У разі перевизначення існуючих тегів, в описі стилю можна вказувати не всі параметри, а лише ті, які ми хочемо змінити. Решта параметрів прийме значення за умовчанням, які для різних тегів різні.

Псевдокласи

У CSS є таке поняття як псевдоклас. На відміну від звичного класу, дія псевдокласу розповсюджується не на весь текст, до якого застосований даний стиль, а лише на його частину, і можливо лише в певному стані. Щоб було зрозумілішим, давайте розберемо ефект, при якому посилання підкреслюються лише при наведенні на них курсора. Ось фрагмент таблиці стилів, який дозволяє досягати вищеописаного ефекту:

```
a { text-decoration: none; }
a:hover { text-decoration: underline; }
```

Верхній рядок - це перевизначення стандартного тега <a>, яке забороняє підкреслювати посилання, а ось нижній - це визначення стилю для псевдокласу hover, який описує стиль посилання в мить, коли курсор знаходиться над ним.

А ось і інший приклад псевдокласу - визначення буквиці спочатку абзацу:

```
p:first-letter { font-size: 200%; font-weight: bold; }
```

Помітьте, що і в тому, і в іншому випадку дія стилю розповсюджується або на певний стан (користувач збирається клацнути по посиланню), або на фрагмент тексту (змінюється тільки перша буква абзацу). У цьому і полягає значення псевдокласів.

Примітка: як і в будь-якій достатньо складній мові, при створенні таблиці стилів можна користуватися коментарями. Їх формат аналогічний класичному C:

```
/* Цей текст є коментарем */
```

Для невеликих сайтів ця можливість Вам навряд стане в нагоді, а ось при створенні складних, багаторівневих таблиць стилів коментарі можуть стати в нагоді. До речі, тут буде доречно навести золоте правило - чим зрозуміліше названа змінна (в даному випадку ім'я класу), тим менше коментарів необхідно.

Основні параметри CSS

Всі параметри, використовувані для визначенні стилю, умовно можна розділити на декілька великих груп, які керують:

- видом шрифту (гарнітура, кегль, колір, нахил, жирність і т.д.);
- форматуванням абзацу (вирівнювання, інтерліньяж, відстань між словами, відступ червоного рядка і т.д.);

- властивостями блоку (відступи зліва-зверху-справа-знизу, місцеположенням блоку на сторінці, видимістю блоку і т.д.);
- інші, які не вписуються ні в одну з перерахованих вище груп (колір посилань сторінки, зміна зовнішнього вигляду курсора і т.д.).

Розглянемо докладніше параметри, використовувані для управління зовнішнім виглядом тексту і форматування абзаців - які найчастіше вживаються.

Основні параметри шрифту:

- font-weight: [bold|normal|number] - жирність шрифту;
- font-style: [normal|italic|oblique] - нахил шрифту;
- font-size: number - розмір шрифту;
- font-family: name - гарнітура шрифту;
- color:number - колір шрифту;
- background-color: number - колір підкладки;
- background: url - текстурна підкладка.

Псевдокласи посилань:

- A:active{} Таблиця стилів для активних посилань (при натисненні);
- A:link{} Таблиця стилів для власне посилань;
- A:visited{} Таблиця стилів для відвіданих посилань;
- A:hover {} Таблиця стилів для посилань при наведенні покажчика миші.

Основні параметри абзацу (і Елементів типу "Box")

- text-align: [left|right|center|justify] – вирівнювання;
- text-indent: number - відступ червоного рядка;
- line-height: number - інтерліньяж;
- letter-spacing: number - трекінг;
- padding-left: number - відступ від тексту зліва;
- padding-right: number - відступ від тексту справа;
- padding-top: number - відступ від тексту зверху;
- padding-bottom: number - відступ від тексту знизу;
- margin-left: number - відступ від межі зліва;
- margin-right: number - відступ від межі справа;
- margin-top: number - відступ від межі зверху;
- margin-bottom: number - відступ від межі знизу

Одиниці вимірювання в CSS

У властивостях, яким потрібна вказівка розмірів, можна використовувати декілька способів для їх завдання:

- відносний розмір у відсотках (%);
- відносний розмір за допомогою словесного опису (larger, smaller, xx-small, x-small, small, medium, large, x-large, xx-large);
- абсолютний розмір у друкарських одиницях - розмір може задаватися в пунктах (pt), піках (pc), пікселях (px), середньою шириною букви "m" (em), середньою шириною букви "x" (ex);
- абсолютний розмір в стандартних одиницях довжини - розмір може задаватися в сантиметрах (cm), міліметрах (mm), дюймах (in), абсолютний в пікселях (px).

Завдання кольору в CSS

Колір для тих властивостей, де це потрібно, може бути визначений одним з трьох способів:

- за допомогою назви кольору: yellow, red, green, grey,..
- шістнадцятковим завданням кольору у форматі #RRGGBB: #ff0000, #883490, #ffffff,..
- десятковим завданням складових кольору у форматі rgb(red, green, blue): rgb(255,0,0), rgb(100,23,78),..

Приклади опису таблиці стилів:

```
.epigraph {
    font-size: 12pt;
    font-style: italic;
    text-align: right;
    color: rgb(127,127,0);
}

p.big {
    font-size: 16px;
    font-weight: bold;
    color: #ff0000;
}

.menu {
    font-weight: bold;
    font-size: 9pt;
    font-family: arial, helvetica, sans-serif;
}
```

```
a:hover {
  color: #b63a3a;
  text-decoration: none;
}
```

Декілька простих правил

Фон. Текст повинен читатися. Дикий, неладний фон сильно заважає сприйняттю змісту. При виборі фону враховуйте також, що роздруковувати вдасться лише темний текст на світлому фоні.

Картинки. Текст з ілюстраціями виглядає краще, ніж просто текст. Потрібно враховувати, що в Росії користувачі Інтернету страждають від повільного зв'язку, графіка завантажується довго, тому краще, якщо ви помістите невеликі за розміром і кількістю кілобайт картинки. **Зкачавши вподобану картинку з чужого сайту, ви порушуєте закон про авторські права.** Добре, коли ви самі малюєте ілюстрації до своєї сторінки.

Шрифти. Дуже багато різних шрифтів на сторінці - поганий тон. Краще виділяти слова розміром, курсивом або жирним стилем. Підкреслення сприймаються як посилання, краще цього уникати. Пам'ятайте, ніщо не дратує читача так, як миготливий текст.

Стиль. Не важливо, в якому стилі буде оформлений ваш сайт, чи помістите ви сердечка, квіточки і вензельочки, або використовуєте в оформленні строгу лінійну графіку, головне, щоб всі сторінки сайту були оформлені одноманітно, підлеглі одній ідеї.

7.1.2 Основні принципи табличної розмітки

До появи CSS єдиним засобом, який мали в своєму розпорядженні дизайнери для розробки макету сторінки, були таблиці. Лише прості варіанти дизайну могли бути створені без таблиць. При цьому єдиним варіантом розташування елементів на сторінці було їх відображення у вигляді вертикальної послідовності. Про розміщення елементів в різних областях сторінки залишалося тільки мріяти. Намагаючись знайти рішення цієї задачі, дизайнери знайшли, що для оформлення виведення можна використовувати структурний елемент `<TABLE>`, якщо встановити значення його атрибуту **BORDER** рівним нулю (`<table border="0">`), і, отже, зробити таблицю невидимою. Розміщуючи вміст сторінки в різних елементах таблиці, можна оформити розділи так, щоб елементи знаходилися в різних областях сторі-

нки. Застосування таблиць для оформлення призвело до зміни основних характеристик Мережі. Цей штучний спосіб виявився настільки успішним, що більшість дизайнерів використовує саме таблиці при створенні Web-сайтів.

Найочевидніша перевага структурованих документів полягає у тому, що їх можна занести в список - і згодом знайти - за допомогою пошукових програм. Проте значущість цієї переваги знижується, якщо документи не забезпечені структурною розміткою або мають невправильну структурну розмітку.

Якщо для оформлення виведення сторінок використовують таблиці, програмне забезпечення, яке прочитує Web-сторінки, стикається з розміткою таблиці, і програма намагається інтерпретувати сторінку в контексті таблиці. Програма не в змозі зрозуміти, що таблиця насправді не є таблицею - вона призначена для візуального оформлення виведення, а не для представлення даних – і здебільшого її структура не представляє інтересу для користувача з обмеженнями зору.

Приклади макетів на основі HTML-таблиць. Два стовпці з шапкою і нижнім колонтитулом

У прикладі використано три основні таблиці — верхня і нижня - це прості таблиці, а ось те, що знаходиться між ними, і представляє для нас інтерес (рис. 7.1).

Спочатку створюємо таблицю з двома колонками шириною **25%** і **75%**. Для вирівнювання вмісту комірок по верхньому краю додано параметр **valign** із значенням, рівним **top**.

За умовчанням вертикальне вирівнювання відбувається по центру вертикалі, що в даному разі не потрібне. Висота колонок може бути різною залежно від інформації, яка в них міститься, тому, щоб уникнути зсуву блоків щодо один одного, необхідно додавати атрибут **valign**.

Після чого в кожному комірку додаємо ще по одній таблиці. Множинне вкладення однієї таблиці в іншу — один із давніх прийомів верстки, який використовують розробники сайтів.

Ширину внутрішніх таблиць треба задати **100%**, щоб вони заповнювали всю комірку цілком, а значення відступів через параметри **cellspacing** і **cellpadding** встановлюємо за бажанням.

Незважаючи на громіздкість коду, швидкість завантаження тексту буде дещо вищою, ніж при використуванні однієї таблиці. Хоча при такому малому об'ємі інформації це не дуже помітно.

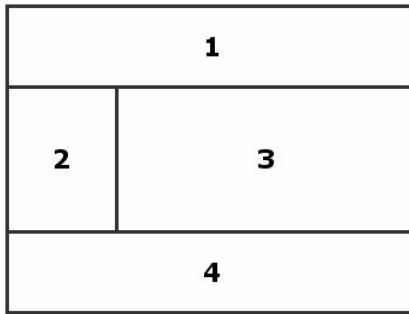


Рисунок 7.1 – Приклад макету: два стовпці з шапкою і нижнім колонтитулом

Приклад №1: двоколоночний макет з плаваючою шириною, з шапкою і нижнім колонтитулом.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="content-type" content="text/html; charset=windows-
1251">
```

```
<title>Основні типи макетів на основі HTML-таблиць | Двоколоноч-
ний макет з плаваючою шириною, з шапкою і нижнім колонтиту-
лом</title>
```

```
<style>
```

```
<!--
```

```
body {
```

```
    margin: 0;
```

```
    padding: 0;
```

```
    font-family: Verdana, Tahoma, arial, geneva, Helvetica, sans-serif;
```

```
    font-size: 11px;
```

```
    color: #000000;
```

```
    background-color: #ffffff; }
```

```
div#menunav { border: 1px solid #000000;
```

```
    width: 160px;
```

```
    font-size: 11px;
```

```
    margin: 20px; }
```

```
div#menunav ul { border: 0; margin: 15px 20px 15px 15px; padding:
```

[illegible]

```
align="center">НИЖНІЙ КОЛОНТИТУЛ</p></td> ° °</tr> </table>
</body> </html>
```

Три стовпці з шапкою і нижнім колонтитулом

1		
2	3	4
5		

Рисунок 7.2 – Триколоночний макет з плаваючою шириною, з шапкою і нижнім колонтитулом

Приклад №2: триколоночний макет з плаваючою шириною, з шапкою і нижнім колонтитулом.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="content-type" content="text/html; charset=windows-
1251">
```

```
<title>Основні типи макетів на основі HTML-таблиць | Триколоноч-
ний макет з плаваючою шириною, з шапкою і нижнім колонтиту-
лом</title>
```

```
<style>
```

```
<!--
```

```
body {
```

```
margin: 0;
```

```
padding: 0;
```

```
font-family: Verdana, Tahoma, arial, geneva, Helvetica, sans-serif;
```

```
font-size: 11px;
```

```
color: #000000;
```

```
background-color: #ffffff; }
```

[illegible]

```

2-го рівня</h2> <p>Абзац...”1...</p> <p>Абзац...”2...</p> <p>Абзац...”3...</p> <p>Абзац...”4...</p></td></tr></table>  

<td valign="top" width="20%"> <table  

width="100%" border="0" cellpadding="0" cellspacing="5"> <tr>  

<td><h2>Заголовок 2-го рівня</h2> <p><a href="#">розділ ”1</a></p> <p><a  

href="#">розділ ”2</a></p> <p><a href="#">розділ  

”3</a></p></td></tr></table>  

<table width="100%" border="0" cellpadding="0"  

cellspacing="5"> <tr> <td align="center">НИЖНІЙ КО-  

ЛОНТИТУЛ</td></tr></table> </body> </html>

```

7.1.3 Верстка за допомогою шарів

Незважаючи на те, що термін «шар» широко використовують при розробці сайту, розшифровці цього поняття в літературі з сайтобудування практично не приділяється увага. Надалі ми активно використовуватимемо термін «шар», тому спочатку необхідно визначити, що ж під ним розуміють.

Спочатку шари ввела компанія Netscape, включивши в свій браузер підтримку тега `<layer>`. Цей тег дозволяв приховувати/показувати поточний вміст, встановлювати положення щодо вікна браузера, накладати один шар поверх інших і завантажувати дані у вміст шару з файлу. Що характерно, всі ці параметри легко змінювалися за допомогою JavaScript, і це розширювало можливості по створенню дійсно динамічного контенту на сторінці. Не зважаючи на такий вражаючий набір можливостей, тег `<layer>` не був включений в специфікацію HTML, і так і залишився лише розширенням браузера Netscape.

Проте необхідність у вказаних можливостях, які б застосовувалися на сайтах, вже назріла, і в кінці 1996 року синтаксис для роботи з шарами був розроблений і схвалений в робочому проекті консорціуму «CSS Positioning (CSS-P)». Основне навантаження лягало на стилі, з їх допомогою можна управляти видом будь-якого елемента, зокрема міняти значення динамічно через JavaScript. На жаль, об'єктні моделі браузерів для доступу до елементів різнилися, тому доводилося писати достатньо складний код, який би враховував ці відмінності.

В даний час розробники популярних браузерів стали дотримуватися специфікацій HTML і CSS, що значно полегшило життя творцям

сайтів, оскільки знизився час на відладку сайту в різних браузерях. Проте, відмінності в підходах у браузерів існують, і при їх виникненні розробники дотримуються наступних форм роботи:

– якщо спостерігаються невеликі відмінності у відображенні одного сайту в різних браузерах, то на ці відмінності закривають очі, просто кажучи, ніяк не виправляють. Тут слід зробити обмовку, що сайт у будь-якому разі має відображатися коректно і без помилок;

– якщо у сайту є істотні відмінності при його показі в одному й іншому браузері, то для їх усунення застосовують хаки.

Хак - це набір прийомів, коли певному браузеру «підсовують» код, який розуміється тільки цим браузером, а іншими ігнорується.

Не зважаючи на те, що хаки працюють, використовувати їх слід обмежено або взагалі обходитися без них. Річ у тому, що хаки знижують універсальність коду, і для модифікації параметрів одного елементу доводиться вносити зміни одночасно в різних місцях.

Є й інший, перспективний шлях - дотримуватися специфікації CSS. Не зважаючи на те, що браузери не повною мірою самі її підтримують, вони прогресують саме в цьому напрямку, а саме, повної підтримки різних специфікацій (HTML, CSS, DOM). Отже виходить, що майбутні версії браузерів будуть уніфіковані, і один і той самий сайт відображатиметься коректно.

Знову повернемося до шарів. Зрозуміло, що вони безпосередньо пов'язані із стилями. Раз так, то чи не виходить, що кожен елемент HTML-коду, до якого додаються стилі, є шаром? У якомусь значенні так і є. Проте це внесло б неабияку плутанину, якщо замість «таблиця» або «параграф» ми б говорили «шар». Тому домовимось відносити цей термін тільки до тегів <div>.

Шар — це елемент веб-сторінки, створений за допомогою тега <div>, до якого застосовують стильове оформлення.

Отже, верстка за допомогою шарів полягає в конструктивному використанні тегів <div> і стилів. При цьому дотримуються наступних принципів.

Розділення вмісту і оформлення

Код HTML має містити тільки теги розмітки і теги логічного форматування, а будь-яке оформлення виноситься за межі коду в стилі. Такий підхід дозволяє незалежно управляти видом елементів

сторінки та її вмістом. Завдяки цьому над сайтом можуть працювати декілька осіб, при цьому кожен виконує свою функцію самостійно від інших. Дизайнер, верстальник і програміст працюють над своїми задачами автономно, знижуючи час на розробку сайту.

Активне застосування тега <div>

Використовуючи шари, істотне значення приділяють універсальному тегу <div>, який виконує безліч функцій. Фактично це основа, на яку «навішують» стилі, перетворюючи її то в іграшку, то в звірятко. Але це не значить, що застосовують тільки один цей тег, бо потрібно вставляти малюнки і оформляти текст. Адже при верстці за допомогою шарів тег <div> є цеглинкою верстки, її базовим фундаментом.

Завдяки цьому тегу HTML-код розпадається на ряд чітких наочних блоків, за рахунок чого верстку шарами називають також блоковою версткою. Код при цьому виходить компактнішим, ніж при табличній верстці, до того ж пошукові системи його краще індексують.

Таблиці застосовуються тільки для представлення табличних даних. При верстці шарами, звичайно ж, використовують таблиці, але тільки в тих випадках, коли вони потрібні, наприклад, для наочного відображення чисел й інших табличних даних. Варіант, коли від таблиць пропонується відмовитися взагалі, є недоцільним і, більш того, шкідливим.

Використання стилів не є обов'язковою характеристикою верстки шарами, і для табличної верстки стилі можуть застосовуватися достатньо активно. Інша справа, що це стає стандартом де-факто, і без стилів тепер ніяк не обійтися.

Одним з найпопулярніших варіантів верстки веб-сторінок виступає двоколонний макет, де в лівій колонці розташовуються посилання, а в правій колонці - основний контент. Така схема досить традиційна і звична для користувачів, тому також користується визнанням серед розробників (рис. 5.1).

Дві колонки створюють за допомогою стильової властивості **float** із значенням **left**, яке застосовується для лівої колонки. При цьому одночасно слід встановити відступ зліва для правої колонки, значення якого дорівнює ширині лівої колонки плюс відстань між ними (приклад 3).

Приклад № 3: створення двох колонок

```
#leftcol { /* Ліва колонка */
  float: left; /* Обтікання справа */
  width: 200px; /* Ширина колонки */}
#rightcol { /* Права колонка */
  margin-left: 225px; /* Відступ зліва */}
```

Наведений спосіб створення колонок хороший тим, що ширину першого шару можна задавати як в пікселях, так й у відсотках. При цьому зміняться тільки значення властивостей **width** і **margin-left** (приклад 4).

Приклад №4: ширина колонки у відсотках

```
#leftcol { /* Ліва колонка */
  float: left; /* Обтікання справа */
  width: 20%; /* Ширина колонки */}
#rightcol { /* Права колонка */
  margin-left: 21%; /* Відступ зліва */}
```

Остаточний варіант створення макету наведено в прикладі 5.

Приклад №5: повний лістинг

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Байки з склепа</title>
<style type="text/css">
body {
  font-family: Arial, Helvetica, sans-serif; /* Рубаний шрифт тексту */
  margin: 0; /* Обнуляємо відступи браузера */ }
h1 {
  font-size: 36px; /* Розмір шрифту */
  font-weight: bold; /* Жирне зображення */
  margin: 0; /* Прибираємо відступи */
  color: #fc6; /* Колір тексту */ }
h2 {
  margin-top: 0; /* Прибираємо відступ зверху */ }
#head { /* Верхній блок */
  background: #0080c0; /* Колір фону */
  padding: 10px; /* Поля навколо */ }
#menu { /* Ліва колонка */
```

```

float: left; /* Обтікання справа */
border: 1px solid #333; /* Параметри рамки навколо */
width: 200px; /* Ширина колонки */
padding: 5px; /* Поля навколо тексту */
margin: 10px 10px 20px 5px; /* Значення відступів */ }
#menu div {
padding: 2px; /* Поля навколо посилань */ }
#menu a {
display: block; /* Відображати посилання як блок */
padding: 2px; /* Поля навколо посилань */
margin: 1px; /* Відступи навколо */
font-size: 0.9em; /* Розмір тексту */
text-decoration: none; /* Прибираємо підкреслення тексту */ }
#menu a:hover {
background: #faf3d2; /* Колір фону */
border: 1px dashed #634f36; /* Додавання пунктирної рамки */
margin: 0; /* Прибираємо поля */ }
#content { /* Права колонка */
margin: 10px 5px 20px 225px; /* Значення відступів */
padding: 5px; /* Поля навколо тексту */
border: 1px solid #333; /* Параметри рамки */ }
.line {
border-bottom: 1px groove #333; /* Лінія між посиланнями */ }
#foot { /* Нижній блок */
background: #333; /* Колір фону */
padding: 5px; /* Поля навколо тексту */
color: #fff; /* Колір тексту */
clear: both; /* Відміняємо обтікання */ }
</style>
</head>
<body>
<div id="head">
<h1>Байки із склепу</h1>
</div>
<div id="menu">
<div><a href="1.html">Всі байки</a></div>
<div><a href="2.html">Байки по автору</a></div>
<div><a href="3.html">Байки з теми</a></div>

```

```

<div class="line"></div>
<div><a href="4.html">Популярні розділи</a></div>
<div><a href="text.html">Останні байки</a></div>
</div>
<div id="content">
  <h2>Алтарь демона </h2>
  <p>Вранці, при яскравому сонячному світлі, все виглядало зовсім не так і похмуро, як планувалося, а навіть навпаки. Від свічок залишилися одні потьоки, «кров» виглядала як фарба, а пір'я майже цілком розлетілося від вітру. Добре збереглися тільки малюнки крейдою, але й вони були швидше прикольні, ніж злі і загадкові. Діти з непідробленим інтересом розглядали зображення, але без тіні тих відчуттів, які були у дорослих вночі при місяці.</p>
  <p>Тим не менше, виявилася одна людина, на яку робота справила велике враження, - сторож табору. Вдень він підійшов до автора &laquo;алтаря&raquo;.</p>
  <p>- Ваша робота? - почав сторож, киваючи у бік площі.<br />
  - А що таке? <br />
  - Ну, як же... Тут кров..., пір'я птаха мертвого..., знаки якісь страшні намальовані..., а у вас все ж таки діти маленькі, вони злякатися можуть...</p>
</div>
<div id="foot">
  Copyright &copy; Влад Мержевич
</div>
</body>
</html>

```

При використанні відступів і полів пам'ятаємо, що браузер встановлює ширину шару з урахуванням властивостей width, border, padding і margin.

7.2 Завдання до лабораторної роботи

1 Реалізуйте приклади №1, 2 і 5. Поясніть їх.

2 Створіть стильовий файл, який містить оформлення web-сайту. Обов'язкове визначення оформлення наступних елементів:

- заголовки 1-4 рівнів;
- пункти меню 1-4 рівнів;

- гіперпосилання;
- таблиці і комірки таблиць;
- основний текст.

3 Для оформлення сторінки використати стандартні графічні елементи. Зручно використати спеціалізовані редактори, наприклад Xara WebStyle, але цілком підійде простий графічний редактор, наприклад PhotoShop. Розробіть:

- банер;
- роздільники;
- спискові елементи (буллети);
- піктограми й ікони (додому, назад, і т.д.).

Тематика сайтів уточнюється з викладачем.

Зміст звіту:

- мета роботи;
- завдання до лабораторної роботи;
- теоретичні основи;
- початкові тексти і скріншоти створених WEB-сторінок і стильових файлів;
- висновки.

7.3 Контрольні запитання

- 1 Що таке таблиці стилів CSS і для чого їх використовують?
- 2 Які існують методи підключення таблиць стилів?
- 3 У чому полягає каскадність стилів?
- 4 Синтаксис CSS.
- 5 Що таке псевдокласи?
- 6 Перерахуйте основні параметри CSS.
- 7 Назвіть основні правила оформлення web-сторінок.
- 8 У чому суть табличної розмітки сторінки?
- 9 Назвіть переваги і недоліки табличної розмітки сторінки.
- 10 У чому суть верстки сторінки за допомогою шарів?
- 11 Які стильові властивості використовують для верстки сторінки за допомогою шарів?
- 12 Які типи макетів набули найбільшого поширення?

8 ЛАБОРАТОРНА РОБОТА № 8 “СТВОРЕННЯ WEB-СТОРІНОК ЗА ДОПОМОГОЮ PHP”

Мета роботи - ознайомитися з мовою скриптів PHP; навчитися створювати різні форми на WEB-сторінці, використовуючи мову HTML.

8.1 Загальні теоретичні відомості

PHP – мова написання скриптів, які вбудовуються безпосередньо в гіпертекстові файли і виконуються на Web-сервері. Програма на PHP вміщується в теги `<?php>...<?>`, а інтерпретатор обробляє команди між цією парою тегів і формує остаточний файл, який передається на локальну машину. Як вже згадувалося, мова PHP досить проста, і основні її конструкції (цикли, умови, функції) нагадують C.

В даний час випущена четверта версія PHP з поліпшеними продуктивністю і можливостями. Так, в PHP 4 з'явилися сесії, подібні Active Server Page, які дозволяють маніпулювати даними протягом всього сеансу доступу користувача, і розширені вбудовані функції системи. Кожна конструкція PHP закінчується крапкою з комою, блоки програм виділяються фігурними дужками, для операцій привласнення використовується знак рівності (“=”), а для порівняння – подвійна рівність (“==”). Мова підтримує унарні операції, на зразок “++”, “==”, “+=” й ін. Крім великого числа вбудованих функцій для роботи з рядками, числами, датами, файлами і масивами можна оголошувати додаткові функції за допомогою ключового слова `function`. До особливостей PHP, мабуть, відноситься формат запису змінних і масивів – кожен ідентифікатор змінної починається із значка \$.

Масиви в PHP служать для зберігання даних різних типів, і на відміну від синтаксично строгих мов програмування в них допускається привласнення елементам масиву чисел, рядків, дат й інших типів даних. Змінні можуть автоматично конвертуватися в масиви і назад. Масиви автоматично створюються при завантаженні вмісту файлу або при читанні рядка таблиці, але їх можна оголошувати і за допомогою функції `array()`, наприклад:

```
$a=array();
$a[1]=«січень»;
$a[2]=«лютий»;
```

```
...
```

```
$a[12]=«грудень»;
```

У квадратних дужках указується індекс масиву. Якщо він опускається, то масив вважається динамічним.

Для виводу значень в PHP використовуються оператори **echo**, **print** і функція **printf()**. Як правило, конструкції **echo** і **print** роблять одне і те саме, а функція **printf()**, крім видачі рядкових значень, виконує їх форматування, наприклад:

```
$name=«Петров»;
$name1=«Петро»;
print «<table>«;
printf(«<tr><td>%s</td><td>%s</td></tr>\n», $name, $name1);
print «</table>«
```

PHP тісно інтегрований з Web. Це, зокрема, означає, що значення, які вносяться у форми, і параметри, збережені Web-браузером, автоматично конвертуються в змінні. Так, описавши у формі поле введення **<input type=«text» name=«pdate» size=«12» value=«дата»>**, в скрипті можна маніпулювати змінною **\$pdate**. Інтеграція з Web дозволяє створювати динамічні Web-сторінки, залежні від будь-якого параметра, наприклад, гіперпосилання, по якому перейшов користувач, або введенного у форму значення. Скажімо, задавши у файлі список тем опитів, можна вивести їх в традиційній для сайтів формі.

```
<form method=«POST» action=«dovotes.php3»>
<?
$votefile=File(«votes.txt»);
for ($i=0; $i<count
($votefile); $i++)

print «<p><input type=radio name=iselect value= $votefile[$i]>«;
}
?>
<center><input type=«submit» value=«Готово»
name=«B1»></center>
</form>
```

Будь-який скрипт PHP складається з послідовності операторів. Оператор може бути привласненням, викликом функції, циклом, умовним виразом або порожнім виразом (який нічого не робить). Оператори звичайно закінчуються крапкою з комою. Також оператори можуть бути об'єднані в групу виведенням групи операторів у фігурні дужки. Група операторів також є оператором.

У PHP є чудова можливість використовувати локальні змінні оточення, до яких відносять \$_POST, \$_GET й інші змінні. З погляду безпеки php скриптів, в ідеалі мають використовуватися тільки \$_POST і \$_GET для отримання параметрів, переданих скрипту через рядок браузера або командний рядок. Використовувати \$_POST і \$_GET безпечно з однієї простої причини – ви самі визначаєте, якими змінними із запиту до php скрипта ви скористаєтеся, а які залишите без уваги.

Одержані змінні \$_POST і \$_GET є простим асоційованим масивом, де ключем масиву є ім'я змінної, а відповідним значенням – її значення.

8.2 Завдання до лабораторної роботи

1 Запустіть сервер PHP, натиснувши на ікону **StartDenwer**. Перевірка HTML. у директорії Z:\home\localhost\www\Група\Прізвище\ (це директорія, де зберігаються html-документи серверу Apache) створіть файл **index.html** з будь-яким текстовим змістом. Тепер запустіть браузер і наберіть:

<http://localhost/Група/Прізвище/index.html>.

Повинен завантажитися Ваш файл. Якщо все пройшло успішно, то сервер Apache працює коректно, і Ви можете далі виконувати роботу.

2 Перевірка PHP. Створіть файл **test.php** з наступним змістом:

```
<html>
<head>
<title>Example</title>
</head>
<body>
<?php
echo "Hi, I'm a PHP script!<br>\n";
phpinfo();
```

```
?>
</body>
</html>
```

Тепер наберіть в браузері: <http://localhost/Група/Прізвище/test.php>. Має відобразитися сторінка з різноманітною інформацією про РНР, яка генерується функцією **phpinfo()**. Ознайомтеся з вмістом.

3 Робота з формами. Створіть файл **request.html**:

```
<html>
<head>
<title>Запит інформації</title>
<body>
<center>
Хочете більше знати про наші товари?
<p>
<table width=400><TR><TD align=right>
<form action="email.php" method="post">
Ваше ім'я:<br>
<input type="text" name="name" size="20" maxlength="30">
<p>
Ваш email:<br>
<input type="text" name="email" size="20" maxlength="30">
<p>
Мене цікавлять:
<select name="preference">
<option value="Яблука">Яблука
<option value="Апельсини">Апельсини
</select>
<p>
<input type="submit" value="Відправити запит! ">
</form>
</td></tr></table></center>
</body>
</html>
```

У файлі Ви вказали, що дані форми оброблятимуться файлом **email.php**. Наведемо його зміст:

```
<?
/* Цей скрипт одержує змінні з request.html */
print "<CENTER>";
print "Привіт, ".$_POST['name'];print "<BR><BR>";
print "Спасибі за Ваш інтерес.<BR><BR>";
print "Вас цікавлять ".$_POST['preference'].". Інформацію про них ми
пошлемо вам на email: ".$_POST['email'].".";
print "</CENTER>";?>
```

4 Відправка email. Для цього в PHP є функція MAIL. Допишемо в кінець файлу **email.php** наступний код:

```
<?
if (mail($_POST['email'], "Запит на інформацію",
$_POST['name']."\nСпасибі за ваш інтерес!\nВас цікавлять
".$_POST['preference']."\nМи їх поширюємо безкоштовно. Зверніться
в найближчий філіал нашої компанії і отримайте ящик цього продукту.\n",
"From: \"PHP mail()\" <mail@php.net>"))
{
    echo "ОК, функція mail() спрацювала коректно.<br>\n";} else {
    echo "При виклику mail() відбулася помилка.<br>\n";}
?>
```

Кожен лист зберігається в директорії /tmp/sendmail у окремому файлі з розширенням .eml. Це дуже зручно, оскільки дозволяє відкрити такий файл в Outlook і переглянути, як лист виглядає з урахуванням всіх перекодувань і перетворень.

Поясніть, як працює даний код.

5 Обробка рядків в php. Створіть файл для роботи з рядками. Введіть текст.

```
<?php
$strarray=explode(":", "Понеділок:Вівторок:Середа:Четвер:П'ятни
ця:Субота:Неділя");?>
<select name="dayofweek" size="1">
<?php for($i=0;$i<7;$i++){ ?>
<option><?php echo($strarray[$i]);?></option>
<?php } ?>
</select>
```

Поясніть, як працює даний код.

6 Робота з масивами. Створіть файл для роботи з масивами. Введіть текст.

```
<?php
// ----- визначаємо масив для місяців -----
$q[]="";
$q[]="січня";
$q[]="лютого";
$q[]="березня";
$q[]="квітня";
$q[]="травня";
$q[]="червня";
$q[]="липня";
$q[]="серпня";
$q[]="вересня";
$q[]="жовтня";
$q[]="листопада";
$q[]="грудня";
// ----- визначаємо масив для днів тижня -----
$e[]="неділя";
$e[]="понеділок";
$e[]="вівторок";
$e[]="середа";
$e[]="четвер";
$e[]="п'ятниця";
$e[]="субота";
$m=date('m'); // ---- зчитуємо місяць
if ($m=="01") $m=1;
if ($m=="02") $m=2;
if ($m=="03") $m=3;
if ($m=="04") $m=4;
if ($m=="05") $m=5;
if ($m=="06") $m=6;
if ($m=="07") $m=7;
if ($m=="08") $m=8;
if ($m=="09") $m=9;
$we=date('w'); // ---- зчитуємо день тижня
```

```

$chislo=date('d'); // ---- зчитуємо число
$den_nedeli = $e[$we]; // ---- витягуємо з масиву відповідне значення дня тижня
$mesyac = $q[$m]; // ---- витягуємо з масиву відповідне значення місяця
echo "Сьогодні ".$chislo." ".$mesyac." ".$den_nedeli;
?>

```

Поясніть, як працює даний код.

7 Створіть шаблон для формування відомостей користувача про себе. Для цього необхідно:

а) створити файл *.html, при запуску якого з'являється наступне:
1 Текст, який виводитиметься вгорі сторінки:

Вибери параметри тексту: ☐ **Жирний** ☐ *Курсив* ☐ Розмір

12	v	12	v	Синий	v
----	----------------------------------	----	----------------------------------	-------	----------------------------------

2 Підзаголовок:

Вибери параметри тексту: ☐ **Жирний** ☐ *Курсив* ☐ Розмір

12	v	12	v	Синий	v
----	----------------------------------	----	----------------------------------	-------	----------------------------------

3 Основний текст резюме:

4 Вибери параметри тексту:

☐ **Жирний** ☐ *Курсив* ☐ Розмір

5	v	5	v	Синий	v
---	----------------------------------	---	----------------------------------	-------	----------------------------------

5 Якщо у тебе є фотографія, то помісти її на свою сторінку:

Огляд...

6 Якщо у тебе є файл тексту резюме, то тут ти можеш поставити посилання до нього:

Огляд...

7 Вибери колір для фону:

Синий Темный



або файл

фону

Огляд...

8 Кнопка

СФОРМУВАТИ СТОРІНКУ

б) написати файл, який містить php-скрипт, що обробляє введену інформацію і виводить особисту сторінку із заданими параметрами.

Зміст звіту:

- мета роботи;
- завдання до лабораторної роботи;
- теоретичні основи;
- початкові тексти і скріншоти створених WEB-сторінок і скриптів PHP;
- висновки.

8.3 Контрольні запитання

- 1 Що таке мова PHP?
- 2 В який тег вміщується програма PHP?
- 3 Як відбувається передача даних з форми в скрипт PHP?
- 4 Як відправити дані на адресу електронної пошти?
- 5 Які команди використовують для обробки рядків в PHP?
- 6 Які команди використовують для роботи з масивами в PHP?
- 7 Для чого використовують змінні оточення \$_POST і \$_GET?

9 ЛАБОРАТОРНА РОБОТА № 9 “РОБОТА З БАЗОЮ ДАНИХ MYSQL”

Мета роботи - овоїти основи роботи з базою даних MySQL.

9.1 Загальні теоретичні відомості

Управління сервером звичайно здійснюється з командного рядка. У Windows відкрийте командний рядок і виконайте наступні команди:

```
cd mysql\bin  
mysql\bin>mysqld --standalone
```

Ця команда запустить mysql у фоновому режимі.

Тепер можна спробувати увійти до серверу. Для цього використовують команду mysql. Спочатку існує єдиний користувач, якому надається право входу-**root**, яка не має пароля. Перше, що потрібно зробити - увійти під ім'ям **root** і встановити для нього пароль. Команда mysql може використовувати наступні опції:

```
\mysql\bin>mysql -u root mysql  
Welcome to the MySQL monitor.Commands end with ; or \g.  
Your MySQL connection id is 1 to server version: 3.23.19  
  
Type 'help' for help.  
  
mysql>
```

Якщо ви це одержали, отже ви успішно увійшли до бази даних mysql, яку використовують для адміністрування серверу.

Спершу створюємо базу даних і таблицю. Входимо в MySQL і виконуємо команди:

```
>CREATE DATABASE products;  
>CREATE TABLE clients (name VARCHAR(25), email VARCHAR(25),  
preference VARCHAR(8));
```

Для спілкування з MySQL з PHP знадобляться наступні функції.
int mysql_connect(string hostname, string username, string password);

Створити з'єднання з MySQL.

Параметри:

Hostname – ім'я хоста, на якому знаходиться база даних.

Username – ім'я користувача.

Password – пароль користувача.

Функція повертає параметр типу `int`, який більше 0, якщо з'єднання пройшло успішно, і дорівнює 0 в іншому випадку.

int mysql_select_db(string database_name, int link_identifier);

Вибрати базу даних для роботи.

Параметри:

Database_name – ім'я бази даних.

link_identifier – ID з'єднання, яке одержане в функції `mysql_connect`. (параметр необов'язковий, якщо він не вказується, то використовується ID від останнього виклику `mysql_connect`)

Функція повертає значення `true` або `false`.

int mysql_query(string query, int link_identifier);

Функція виконує запит до бази даних.

Параметри:

Query – рядок, що містить запит

link_identifier – див. попередню функцію.

Функція повертає ID результату або 0, якщо сталася помилка.

int mysql_close(int link_identifier);

Функція закриває з'єднання з MySQL.

Параметри:

link_identifier – див. вище.

Функція повертає значення `true` або `false`

int mysql_num_rows(int result);

Функція повертає кількість рядків в результаті запиту.

Параметр **result** – містить ID результату запиту.

int mysql_result(int result, int i, column);

Функція повертає значення поля в стовпці **column** і в рядку **i**.

9.2 Завдання до лабораторної роботи

1 Створіть файл **email.php**, який заносить інформацію про клієнта (лабораторна робота №8, файл **request.html**) в базу даних з наступним змістом:

```

<?
/* Цей скрипт одержує змінні з request.html */
/* Деякі змінні */
$hostname = "localhost";
$username = "root";
$password = "";
$dbName = "products";
/* Таблиця MySQL, в якій зберігаються дані */
$userstable = "clients";
/* email адміністратора */
$adminaddress = "administration@me.com";
/* створити з'єднання */
MYSQL_CONNECT($hostname,$username,$password) OR DIE("Не
можу створити з'єднання ");
@mysql_select_db("$dbName") or die("Не можу вибрати базу даних
");
PRINT "<CENTER>";
PRINT "Привіт, ".$_POST['name'];
PRINT "<BR><BR>";
PRINT "Спасибі за ваш інтерес.<BR><BR>";
PRINT "Вас цікавлять ".$_POST['preference']." Інформацію про них
ми пошлемо вам на email: ". $_POST['email'];
PRINT "</CENTER>";
/* Відправляємо email */
mail($_POST['email'], " Запит на інформацію ", $_POST['name'])."\n
Спасибі за ваш інтерес!\n
Вас цікавлять ".$_POST['preference']."\n
Ми їх поширюємо безкоштовно. Зверніться в найближчий філіал
нашої компанії і отримайте ящик цього продукту.\n
");
mail("administration@me.com",
"Був запит на інформацію.",
$_POST['name']." цікавили ".$_POST['preference']."\n
email-адреса: ".$_POST['email']."\n");
$mail=$_POST['email'];
$user=$_POST['name'];
$pref=$_POST['preference'];
/* Вставити інформацію про клієнта в таблицю */

```

```
$query = "INSERT INTO $userstable VALUES('$user','$mail','$pref)";
$result = MYSQL_QUERY($query);
PRINT "Інформація про вас занесена в базу даних.";
/* Закрити з'єднання */
MYSQL_CLOSE();
?>
```

Поясніть, як працює даний код.

2 Створіть код виконання запиту, написавши скрипт **apple.php**:

```
<?/* Скрипт показує клієнтів, які яблука люблять більш ніж апель-
сини */
$hostname = "localhost";
$username = "root";
$password = "";
$dbName = "products";
/* Таблиця MySQL, в якій зберігаються дані */
$userstable = "clients";
/* створити з'єднання */
MYSQL_CONNECT($hostname,$username,$password) OR DIE("Не
можу створити з'єднання ");
@mysql_select_db("$dbName") or die("Не можу вибрати базу даних
");
$pref="preference";
/* Вибрати всіх клієнтів - яблучників */
$query = "SELECT * FROM $userstable WHERE $pref = 'Яблука'";
$result = MYSQL_QUERY($query);
/* Як багато знайшлося таких */
$number = MYSQL_NUMROWS($result);
/* Надрукувати всіх в красивому вигляді */
$i = 0;
IF ($number == 0) {
PRINT "<CENTER><P>Любителів яблук нема</CENTER>";
} ELSEIF ($number > 0) {
PRINT "<CENTER><P>Кількість любителів яблук: $num-
ber<BR><BR>";
WHILE ($i < $number){
$name = mysql_result($result,$i,"name");
$email = mysql_result($result,$i,"email");
```

```

PRINT "Клієнт $name любить Яблука.<BR>";
PRINT "Його Email: $email.";
PRINT "<BR><BR>";
$i++;
}
PRINT "</CENTER>";
}
?>

```

Пояснить, як працює даний код.

3 а) Створіть базу даних для збереження інформації, яка передається з форми, описаної в завданні 3.2.7 лабораторної роботи №3 (з7лр3);

б) Додати в php-файл, який обробляє форму, описану в з7лр3, процедуру запису інформації в базу даних і рядок «Інформація про Вас занесена в БД»;

в) Створити сторінку для переглядання будь-якого резюме (з7лр3), записаного в базі даних MySQL, використовуючи запити до БД.

4 Створити сторінку обробки замовлення - файл із записом в базу даних:

а) Створити файл **avto.html** за зразком на малюнку.

Сайт объявлений: Поиск автомобиля

Критерии поиска

Фирма	Toyota		
Модель	Carina		
Цена от	1000	до	5000
Год производства с	1980	до	2001

Отправить

б) Створити файл **avto_php.php**. Цей файл повинен одержувати змінні з avto.html. Результат натиснення на кнопку «Відправити»:

Вас цікавить машина будь-якої фірми, будь-яка модель (якщо фірма і модель будь-яка) або, наприклад,
 Вас цікавить машина фірми Toyota,
 модель Carina
 Середня ціна 3000 (якщо вибрана певна фірма і модель)

в) Запустити MySQL, створити базу даних «Poisk_avto», створити таблицю «Zakaz», що містить поля Firma, Model, SrZena;

г) Додати у файл avto_php.php процедуру запису інформації в базу даних і рядок «Інформація про Ваше замовлення занесена в БД»;

д) Занести інформацію в БД;

е) Створити сторінку для проглядання всіх замовлень, записаних в базі даних MySQL, використовуючи запити до БД.

Зміст звіту:

- мета роботи;
- завдання до лабораторної роботи;
- теоретичні основи;
- початкові тексти і скріншоти створених WEB-сторінок і скриптів PHP;
- висновки.

9.3 Контрольні запитання

- 1 Як запустити MySQL у фоновому режимі?
- 2 Як увійти до бази даних?
- 3 Як створити базу даних?
- 4 Як створити таблицю?
- 5 Які типи даних підтримуються MySQL?
- 6 Що виконує функція **mysql_connect**?
- 7 Що виконує функція **mysql_select_db**?
- 8 Що виконує функція **mysql_query**?
- 9 За допомогою якої команди створюється новий запис в таблиці?
- 10 За допомогою якої команди обновляється запис в таблиці?

- 11 За допомогою якої команди видаляється запис з таблиці?
- 12 За допомогою якої команди здійснюється запит до даних?
- 13 Які оператори можна використовувати в запиті?
- 14 Що виконує функція **mysql_close**?
- 15 Що виконує функція **mysql_num_rows**?
- 16 Що виконує функція **mysql_result**?

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

- 1 **Пек, С.** Web-сервер WebSite [Текст]: пер. англ. /С.Пек, С.Аррантс. - К.: Издательская группа BHV, 1997. - 544 с.
- 2 **Федоров, А.** Введение в HTML /А.Федоров //КомпьютерПресс.- 1996.-№6.- С. 110-115.
- 3 **Фролов, А.В.** Глобальные сети компьютеров. Практическое введение в Internet, E-Mail, FTP, WWW и HTML, программирование для Windows Sockets [Текст] /А.В.Фролов, Г.В.Фролов - М.: ДИАЛОГ-МИФИ, 1996. - 288 с. - (Библиотека системного программиста; Т. 23).
- 4 **Фролов, А.В.** Сервер Web своими руками. Язык HTML, приложения CGI и ISAPI, установка серверов Web для Windows [Текст] /А.В.Фролов, Г.В.Фролов. - М.: ДИАЛОГ-МИФИ, 1997. - 288 с. - (Библиотека системного программиста; Т. 29)
- 5 [http: // www.nsca.uiuc.edu / General / Internet / WWW / HTMLPrimer.html](http://www.nsca.uiuc.edu/General/Internet/WWW/HTMLPrimer.html)
- 6 [http: // kuhttp.cc.ukans.edu / lynx help / HTML quick.html](http://kuhttp.cc.ukans.edu/lynx/help/HTMLquick.html)
- 7 [http: // home.mcom.com / home / services docs / html-extensions.html](http://home.mcom.com/home/services/docs/html-extensions.html)
- 8 [http: // www.cs.cmu.edu /~tilt / cgh](http://www.cs.cmu.edu/~tilt/cgh)
- 9 www.php.spb.ru/php/
- 10 **Бобцов, А.** Интернет-технологии - образованию [Текст] /А.Бобцов.- М.: Питер, 2009. - 464 с.
- 11 **Боженюк, А.В.** Интеллектуальные интернет-технологии [Текст] /А.В.Боженюк, Э.М.Котов, А.А.Целых.-М.: Феникс, 2009. - 384 с.
- 12 **Герасевич, Виталий** Блоги и RSS: интернет-технологии нового поколения.- СПб: BHV, 2010. - 256 с.
- 13 **Герасевич, Виталий** Блоги и RSS: интернет-технологии нового поколения [Текст] /В.Герасевич.- СПб: BHV, 2012.-256с.

14 **Гришин, В.** Интернет и бизнес. Как использовать современные технологии и не остаться в дураках? Книга по Требованию [Текст] /В.Гришин - Москва, 2012. - 186 с.

15 **Евдокимов, Н.В.** Основы контентной оптимизации. Эффективная Интернет-коммерция и продвижение сайтов в Интернет [Текст] /Н.В.Евдокимов.- М.: Вильямс, 2013. - 160 с.

16 **Инькова, Н.А.** Современные интернет-технологии в коммерческой деятельности [Текст] /Н.А.Инькова.- М.: Омега-Л, 2014.- 192 с.