

Міністерство освіти і науки України
Сумський державний педагогічний університет
імені А.С. Макаренка

Кафедра інформатики

Методичні рекомендації

ФАХОВЕ ВСТУПНЕ ВИПРОБУВАННЯ З ІНФОРМАТИКИ

Частина I

Рівень вищої освіти – Магістр
Спеціальність – 014 Середня освіта
Предметна спеціалізація – Інформатика*

Суми – 2016

Укладачі:

Дегтярьова Н.В., кандидат педагогічних наук, викладач кафедри інформатики

Петренко С.І., викладач кафедри інформатики

Удовиченко О.М., викладач кафедри інформатики

Безуглий Д.С., лаборант кафедри інформатики

Дані методичні рекомендації створені для повторення відомих фактів та визначення напряму відповіді абітурієнта на питання іспиту з інформатики. Вказівки до кожного питання скоріше дають підказку до окремих частин, на які розбивається кожне питання. Тут НЕ пропонуються повні та вичерпні відповіді на кожне з них, але наводяться основні моменти, знання яких повинен продемонструвати вступник. Матеріал до окремого питання потребує доопрацювання та більш глибокого осмислення, тому в кінці наводяться джерела для глибшого вивчення матеріалу.

ЗМІСТ

Операційні системи.....	7
<i>Основи операційних систем. Різновиди операційних систем. ОС Windows. Загальний огляд. Організація роботи з елементами ОС. Стандартні програми. Встановлення та видалення програмного забезпечення. Налаштування ОС.....</i>	<i>7</i>
Робота з текстовими даними.....	9
<i>Основи роботи в середовищі Microsoft Office. Текстовий процесор Microsoft Word. Загальні поняття. Введення та редагування тексту.....</i>	<i>9</i>
Робота з текстовими даними.....	11
<i>Текстові редактори, процесори та видавничі системи. Форматування тексту в текстовому процесорі (процесор на вибір).....</i>	<i>11</i>
Робота з об'єктами в текстовому процесорі.....	12
<i>Робота з таблицями, графічними об'єктами, формулами в MS Word.....</i>	<i>12</i>
Електронні таблиці.....	14
<i>Табличний процесор MS Excel: призначення та можливості. Типи даних. Робота з формулами. Побудова графіків.....</i>	<i>14</i>
Електронні презентації.....	16
<i>Загальні поняття презентацій. Редактор презентацій MS PowerPoint. Принципи побудови презентацій. Оформлення слайдів. Використання зовнішніх об'єктів. Використання анімації. Додаткові можливості.....</i>	<i>16</i>
Інформаційні моделі та системи.....	18
<i>Бази даних і системи з базами даних. Трирівнева архітектура баз даних. Моделі даних. Методологія «сутність-зв'язок». Ключі та їх призначення.....</i>	<i>18</i>
Робота з текстовими даними.....	20
<i>Основи роботи в середовищі LibreOffice. Текстовий процесор Writer. Загальні поняття. Редагування та форматування тексту. Додаткові можливості.....</i>	<i>20</i>
Інтернет та комунікація.....	22
<i>Поняття мережі Інтернет. Браузери, їх приклади та особливості. Робота з електронною поштою. Outlook Express. Електронна пошта: протоколи SMTP. Організація доступу до поштової скриньки користувача за допомогою протоколу POP3. Стандарт MIME.....</i>	<i>22</i>
Безпроводні мережі.....	25
<i>Стандарт IEEE 802.11 безпроводних мереж (Wi-Fi). Управління доступом до каналу зв'язку.....</i>	<i>25</i>
Типи адрес стека TCP/IP. IP-адреси. Протоколи IP та ICMP.....	27
<i>Типи IP-адрес. Доменні імена. Формат IP-адреси, класи IP-адрес. Використання масок під час IP-адресації. Порядок призначення IP-адрес. Формат IP-пакета. Схема IP-маршрутизації.</i>	

<i>Маршрутизація з використанням масок. Фрагментація IP-пакетів. Призначення і характеристика протоколу ICMP, Формат ICMP-пакета. Типи ICMP-повідомлень.</i>	27
Протоколи TCP та UDP. Система доменних імен DNS. Протоколи DHCP.	29
<i>Адресація прикладних програм. Порти. Протокол UDP. Формат TCP-сегмента. Логічні з'єднання, порядкові номери та номери підтвердження. Управління вікном прийому. Система DNS, схема роботи DNS. Режими DHCP, алгоритм динамічного призначення адрес.</i>	29
Технологія прикладного рівня.	31
<i>Протокол HTTP. Синтаксис протоколу HTTP. Коди відповідей сервера Принцип роботи FTP. FTP-сервер та FTP-клієнт. Протокол FTP.</i>	31
Теоретичні основи інформатики. Математична логіка.	33
<i>Логіка висловлювань. Закони логіки висловлювань. Нормальні форми логіки висловлювань. Логіка першого ступеня.</i>	33
Теоретичні основи інформатики. Основи теорії множин.	35
<i>Поняття множини. Поняття кортежу. Декартів добуток множин. Операції над множинами. Доведення рівностей з множинами. Комп'ютерне зображення множин.</i>	35
Теоретичні основи інформатики. Теорія графів.	37
<i>Основні означення та властивості. Деякі спеціальні класи простих графів. Способи подання графів. Шляхи та цикли, зв'язність. Ізоморфізм графів. Ейлерів цикл у графі. Гамільтонів цикл у графі. Зважені графи та алгоритми пошуку найкоротшого шляху. Обхід графів. Планарні графи.</i>	37
Теоретичні основи інформатики. Дерева та їх застосування.	39
<i>Основні означення та властивості. Обхід дерев. Префіксна та постфіксна форми запису. Бінарне дерево пошуку. Дерева прийняття рішень. Алгоритм бектрекінг.</i>	39
Теоретичні основи інформатики. Відношення.	40
<i>Відношення та їх властивості. Відношення еквівалентності. Відношення часткового порядку. Операції над відношеннями.</i>	40
Алгоритми стиснення даних.	41
<i>Загальна характеристика методів кодування та стиснення даних. Статистичне кодування Хофмана і Шенона-Фано. Арифметичне кодування. Метод «стосу книг». Архіватори. Стиснення графічних файлів.</i>	41
Проблеми захисту інформації.	43
<i>Можливі канали витоку інформації. Функції систем захисту інформації. Основні принципи проектування систем захисту інформації. Захист інформації.</i>	43
Захист від несанкціонованого доступу.	47
<i>Ідентифікація об'єктів і механізми підтвердження справжності. Паролі. Надання повноважень і механізми контролю доступу до інформації. Принцип мінімальних привілей. Розділення користувачів.</i>	47

Архітектура комп'ютерів. Визначення, класифікація и етапи розвитку.....	51
<i>Поняття архітектури комп'ютерів. Архітектура апаратних засобів. Архітектура неймановського комп'ютера. Архітектура постнеймановських комп'ютерів. Системний інтерфейс і архітектура системної плати. Архітектура системної плати. Принципи проектування сучасних комп'ютерів.</i>	
	<i>51</i>
Архітектура мікропроцесорів сімейства i80x86.	57
<i>Загальні характеристики мікропроцесорів і еволюція розвитку. Логічна структура розвинутого мікропроцесора. Програмна модель мікропроцесора (8-бітні, 16-бітні та 32-бітні). Регістри. Типи даних. Організація захищеного режиму. Сегментна організація пам'яті. Сторінкова організація пам'яті. Суміщення сегментації і сторінкової організації пам'яті. Переривання і виключення. Організація мультизадачності.....</i>	
	<i>57</i>
Кешування пам'яті.	68
<i>Принципи кешування. Кеш прямого відображення. Асоціативний кеш. Архітектура і характеристики засобів кешування. Управління кешуванням. Мікроархітектура мікропроцесорів (i8086-i80484, P5, P6, P7).....</i>	
	<i>68</i>
Системна пам'ять персонального комп'ютера.....	74
<i>Логічна організація пам'яті. Розподіл адресного простору пам'яті. Розширена пам'ять. Додаткова пам'ять (Expanded Memory). Системна пам'ять (Shadow RAM). Фізична організація пам'яті. Організація динамічного ОЗП. Організація банків пам'яті і режим роботи ОЗП. Типи динамічної пам'яті. Модулі динамічної пам'яті</i>	
	<i>74</i>
Дискові пристрої зовнішньої пам'яті.....	79
<i>Фізична будова жорстких і гнучких дисків. Типи оптичних дисків.</i>	
	<i>79</i>
Зовнішні інтерфейси.	82
<i>Послідовний інтерфейс СОМР-порт. Програмна модель СОМР-порту. Паралельний LPT-порт. Режими роботи. Універсальна послідовна шина USB. Архітектура і взаємодія елементів системи USB. Типи передач і формати інформації, яка передається через шину USB.....</i>	
	<i>82</i>
Основи побудови сайтів. Основи мови HTML.....	86
<i>Основні вимоги процесу побудови сайтів. Основи мови гіпертекстової розмітки. Основні поняття HTML. Основна структура html-документу. Робота з текстом.....</i>	
	<i>86</i>
Основи мови HTML.....	88
<i>Розміщення об'єктів на Web-сторінці: списки, таблиці, зображення, відео та аудіо-об'єкти. Форматування об'єктів.....</i>	
	<i>88</i>
Основи мови HTML.....	92
<i>Розбиття сторінки на частини (блоки, фрейми). Створення та робота з гіперпосиланнями. Створення закладки (якоря).....</i>	
	<i>92</i>
Основи гіпертекстової розмітки веб-сторінок.....	94
<i>Таблиці каскадних стилів. Особливості приєднання до html-документу. Переваги використання css-файлів. Вбудовані стилі, групування селекторів.....</i>	
	<i>94</i>

Основи побудови сайтів.	96
<i>Поняття CMS, функції, особливості використання, приклади. Використання шаблонів Web-сторінок, вимоги до існування сайту, створеного за допомогою шаблонів.</i>	<i>96</i>
Рекомендована література.....	97

Операційні системи.

Основи операційних систем. Різновиди операційних систем. ОС Windows. Загальний огляд. Організація роботи з елементами ОС. Стандартні програми. Встановлення та видалення програмного забезпечення. Налаштування ОС.

Персональний комп'ютер (ПК) являє собою надзвичайно складний комплекс, ручне керування ресурсами якого можливе, але недоцільне, особливо з огляду на те, наскільки швидкохідними є процеси, які виконуються в ПК. В зв'язку з цим уже давно виникла ідея написання програми самостійного узгодженого керування апаратними і програмними ресурсами самою обчислювальною системою. Таке керівництво здійснюється операційними системами.

Операційна система (ОС) – сукупність програм для узгодженого керування апаратними та програмними засобами ПК.

До основних функцій ОС відносяться:

- керування ресурсами ПК (процесором, оперативною пам'яттю (ОП), зовнішніми пристроями);
- запуск та виконання системних і користувацьких програм;
- обмін інформацією між різними пристроями ПК (процесор – ОП, процесор – співпроцесор і т.д.);
- збереження інформації у зовнішній пам'яті та обслуговування дисків;
- забезпечення діалогу користувача з ПК;
- мережеві операції.

Структуру ОС складають такі модулі:

- базовий модуль (ядро ОС) – керує роботою програми і файлової системою, забезпечує доступ до неї та обмін файлами між периферійними пристроями;
- командний процесор – розшифровує та виконує команди користувача, що надходять насамперед через клавіатуру;
- драйвери периферійних пристроїв – програмно забезпечують узгодженість роботи цих пристроїв із процесором (кожен периферійний пристрій обробляє інформацію по різному і в різному темпі);
- додаткові сервісні програми (утиліти) – роблять зручним і багатостороннім процес взаємодії користувача з комп'ютером.

Операційні системи на різних ПК класифікуються за різними ознаками. Зокрема, за ознакою втручання оператора розрізняють операційні системи пакетної обробки, які опрацьовують масштабні задачі (розрахунок прогнозу погоди, траєкторії літака тощо) з мінімальним втручанням оператора в роботу ОС, операційні системи реального часу, які керують складними обчислювальними комплексами (колайдер, запуск космічної ракети тощо) та діалогові операційні системи, які спроектовані під забезпечення діалогу з оператором. Такі операційні системи розрізняються, в свою чергу, за дуже великою кількістю параметрів (Unix та post-Unix, відкриті та пропріетарні, однозадачні та багатозадачні, текстові та графічні, 32-розрядні та 64-розрядні тощо).

Microsoft пропонує 6 різних варіантів постачання Windows 7 (Starter, Home Basic, Home Premium, Professional, Enterprise, Ultimate), проте через роздрібну мережу поширюються переважно версії Home Premium і Professional. Інші варіанти поширюються обмежено, наприклад, тільки для корпоративних клієнтів або тільки в країнах, що розвиваються. Всі варіанти, крім найбільш спрощених, працюють як на 32-бітній, так і на 64-бітній платформах.

Поняття операційної системи напряму пов'язане з такими поняттями, як файл, файлова система, програма, команда та ін.

Мінімальні вимоги до ПК для установки Windows 7: процесор – 1 ГГц для 32-bit і для 64-bit версій; оперативна пам'ять – 1 Гб для 32-bit версії і від 2Гб і вище для 64-bit версії; відеокарта – з підтримкою DirectX 9; вільне місце на жорсткому диску – 16 Гб для 32-bit і 20 Гб для 64-bit; оптичний накопичувач – DVD-ROM.

Windows 7 має підтримку мультитач-управління (технологія, за якою сенсорний екран або тачпад відстежує одночасно декілька точок натиснення). Windows 7 включає більше шрифтів для відображення нелатинських символів, ніж для відображення латинських. У панелі керування шрифтами за замовчуванням будуть відображатися тільки ті шрифти, розкладка для яких встановлена в системі. Перевагою Windows 7 можна вважати більш тісну інтеграцію з виробниками драйверів. Більшість з них визначаються автоматично, при цьому в 90% випадків зберігається зворотна сумісність з драйверами для Windows Vista.

Windows 7 підтримує три варіанти оформлення: Windows Aero, спрощений стиль, класичний. Змін зазнав інтерфейс системи, панель завдань і меню «Пуск». Програваач Windows Media Player 12 отримав новий інтерфейс і відтворює більшість форматів, на відміну від попередника, якому потрібно було встановлювати додатково кодеки. Додана можливість захисту даних на USB-накопичувачах за допомогою Enhanced Storage. Покращена робота в бездротовій мережі, після першого підключення Windows запам'ятає мережу і наступного разу встановить до неї підключення автоматично. У Windows 7 прискорені процеси переходу в сплячий режим, відновлення роботи і повторного підключення до бездротової мережі. Браузер Internet Explorer 8 встановлений за замовчуванням в системі тепер має прискорювачі, зручні можливості для швидкого пошуку, нові зручні веб-фрагменти. Також існує механізм, що дозволяє запускати програми, які були створені для більш ранніх операційних систем, в режимі сумісності.

Стандартні програми Windows 7 знаходяться в програмній групі Всі програми/Стандартні, до яких відносять графічний редактор Paint, текстові редактори Блокнот та WordPad, Калькулятор, Записки, Ножиці, Звукозапис, Командний рядок, Провідник та інші.

Зазвичай поширення програм відбувається у вигляді дистрибутива. Дистрибутив включає в себе як один, так і декілька файлів. У дистрибутиві є файл інсталяції. Процес запуску почнеться відразу після подвійного клацання мишею по ньому. Зазвичай цей інсталяційний файл називається так само, як і програма. Але буває і таке, що ім'я цього дистрибутива стандартне. Наприклад, setup.exe або install.exe. Після запуску програми установки, можливо, доведеться відповісти на декілька нескладних запитань, вибрати тип установки, а також погодитися з умовами ліцензійної угоди. Потім почнеться сам процес установки програми в систему, а по його завершенні на екрані з'явиться відповідне повідомлення.

Правильно видалити програму можна двома способами. Перший – скористатися власним «деінсталятором» програми, який можна знайти в її папці. Як правило, цей файл містить у своїй назві слово uninstall. Другий спосіб – скористатися розділом Панелі Керування системи, який в Windows 7 називається Програми (Програми та компоненти). Із списку всіх встановлених програм обрати ту, яку треба видалити, і запустити процедуру видалення. Окрім того, існує безліч програм для видалення ПЗ: Uninstall Tool, Revo Uninstaller, Final Uninstaller тощо.

Робота з текстовими даними.

Основи роботи в середовищі Microsoft Office. Текстовий процесор Microsoft Word. Загальні поняття. Введення та редагування тексту.

Microsoft Office – це пакет офісних програм від корпорації Microsoft. Даний продукт є міжнародним форматом роботи в усіх сферах діяльності людини, пов'язаних з комп'ютером.

Офісний пакет має декілька версій, від яких безпосередньо залежить наявність тих чи інших у ньому програм. Так у повну версію Microsoft Office 2010 входять Word (текстовий процесор), Excel (табличний процесор), PowerPoint (для підготовки презентацій), Access (робота з базами даних), Publisher (настільна видавнича система), OneNote (для створення швидких нотаток і організації особистої інформації), Outlook (персональний органайзер, який складається з календаря, планувальника завдань, нотаток, менеджера електронної пошти та адресної книги), InfoPath (для збору даних і керування ними), SharePoint Workspace (для спільної роботи над документами та їх бібліотеками в єдиному спеціальному середовищі SharePoint Workspace), SharePoint Designer (для веб-дизайну), Lync (для спілкування в реальному часі, використовуючи різні види комунікацій), Project (система управління проектами).

Загальноприйнятим стандартом текстового процесора давно став Microsoft Word, який має такі можливості:

1. Здійснювати всі традиційні операції над текстом:

- редагування текстової інформації;
- форматування символів, застосовуючи шрифти різних розмірів і накреслень;
- форматування сторінки (сюди також входять виноски, колонтитули);
- форматування документа в цілому (автоматичний зміст, різні покажчики);
- вибір синонімів, перевірка правопису, використання автоматичного перенесення слів.

2. У цій програмі реалізовані можливості нових технологій зв'язування та впровадження об'єктів, завдяки яким можна в документ включати таблиці, текстові фрагменти та ілюстрації.

3. Це одна з перших загальнодоступних програм, за допомогою якої можна виконувати операції верстки, які часом необхідні видавничим професійним системам, а також з її допомогою можна готувати повноцінні оригінал-макети для подальшого тиражування в друкарні.

4. Дана програма автоматизує майже всі операції на кожному етапі роботи, що прискорює підготовку документу: створення документа на основі шаблонів, пошук та заміна фрагментів тексту, використання бібліотеки стилів оформлення абзаців і стилів оформлення таблиць, копіювання формату, макрова, автозбереження та багато іншого.

Будь-який текстовий документ в процесі свого формування проходить наступні етапи:

1. Створення документа на основі вже існуючого документа або шаблону.
2. Введення тексту.
3. Редагування тексту.
4. Форматування документа.
5. Збереження текстового документа на енергонезалежному носії.
6. Публікація документа.

У Word існує два режими уведення тексту:

✓ режим вставки – під час уведення текст буде зміщувати символи праворуч, а нові будуть вставлятися в текст.

✓ режим заміни – під час уведення тексту символи, що розміщені справа від курсора, замінюються новими.

Під час уведення тексту з клавіатури потрібно дотримуватися таких правил:

– перехід на новий рядок відбудеться автоматично, як тільки буде заповнений поточний рядок. Не можна використовувати клавішу Enter для переходу до нового рядка. Для примусового переходу до нового рядка без утворення нового абзацу використовують комбінацію клавіш Shift+Enter (розрив рядка);

– перехід на нову сторінку відбудеться автоматично, як тільки буде заповнена поточна сторінка. Не можна використовувати клавішу Enter для переходу до нової сторінки. Для примусового переходу до нової сторінки використовують комбінацію клавіш Ctrl+Enter (розрив сторінки);

– між словами ставиться один пробіл. Між словами, які повинні залишатися в одному рядку, необхідно поставити так званий нерозривний пробіл, використовуючи комбінацію клавіш Ctrl+Shif+Пробіл;

– не можна використовувати клавішу Пробіл для отримання абзацного відступу («червоного рядка») або вирівнювання тексту по ширині сторінки;

– розділові знаки . , ; ! ? пишуться разом із словом, за яким йдуть;

– після розділових знаків . , ; ! ? ставиться пробіл, за винятком тих випадків, коли цими знаками закінчується абзац;

– перед знаками " ([{ ставиться пробіл. Слово, що йде за цими знаками, пишеться без пробілу;

– знаки ")] } пишуться разом із словом, за яким йдуть. Після цих знаків ставиться пробіл, за винятком тих випадків, коли ставляться розділові знаки, які пишуться разом із словом, за яким йдуть;

– до і після дефісу пробіл не ставиться;

– до і після тире пробіл ставиться;

– знак дефіса (-) пишеться разом з обома частинами слова;

Редагування – це виправлення помилок, внесення додаткових слів і речень, зміна їх розташування, вилучення зайвих фрагментів, тобто зміна змісту матеріалу. До засобі в редагування текстових документів також належать такі дії, як перевірка правопису та граматики, пошук та заміна символів і фрагментів, переклад окремих слів, речень і документів.

Робота з текстовими даними.

Текстові редактори, процесори та видавничі системи. Форматування тексту в текстовому процесорі (процесор на вибір).

Для роботи з текстом виділяють такі групи програм:

✓ Редактори текстів програм – призначені для редагування програм на мові програмування, вбудовані в систему програмування. Приклад: Turbo C, Turbo Pascal.

✓ Текстові редактори для документів простої структури – призначені для створення й зміни текстових файлів (без графічних об'єктів), а також їх перегляду на моніторі, виводу на друк, пошуку фрагментів тощо. Приклад: Блокнот, WordPad, MultiEdit.

✓ Текстові процесори для документів складної структури – дозволяють виконувати операції набору, редагування та оформлення тексту (параграфи, глави, розділи, складне оформлення). Приклад: MS Word, Corel WordPerfect, LibreOffice Writer.

✓ Настільні видавничі системи – призначені для підготовки рекламних буклетів, оформлення книг, журналів, газет. За функціональними характеристиками та можливостями настільні видавничі системи аналогічні кращим текстовим процесорам. Але видавничі системи мають більш широкі можливості управління підготовкою тексту, крім того підготовлені ними матеріали мають значно вищий рівень якості. Приклад: Adobe PageMaker, QuarkXPress, Ventura Publisher, Microsoft Office Publisher.

Загальноприйнятим стандартом текстового процесора давно став Microsoft Word і не в останню чергу завдяки широким можливостям форматування.

Під форматуванням розуміють операції, які пов'язані з оформленням тексту і зміною його зовнішнього вигляду. Операції форматування дійсні тільки для виділених об'єктів.

Розрізняють три основні типи форматування: форматування символів, форматування абзаців та форматування сторінок.

До *форматування символів* відносять зміну шрифту, його розміру, накреслення (напівжирний, курсив, підкреслення), колір тексту, ефекти (верхній і нижній індекси, перекреслений) тощо. Форматування здійснюється або у діалоговому вікні «Шрифт», яке відкривається командою Головна/Шрифт, або ж за допомогою відповідних кнопок групи «Шрифт» на вкладці «Головна».

До *форматування абзаців* відносять вирівнювання тексту (по лівому краю, по правому краю, по центру, по ширині сторінки), встановлення відступів абзаців, інтервалів між абзацами, міжрядкових інтервалів, вибір фону, контуру абзацу, контроль положення абзацу на сторінці тощо. Форматування здійснюється або у діалоговому вікні «Абзац», яке відкривається командою Головна/Абзац, або ж за допомогою відповідних кнопок групи «Абзац» на вкладках «Головна» або «Розмітка сторінки».

До *форматування сторінок* відносять зміну розмірів полів сторінки, орієнтацію (книжкова, альбомна), розмір паперу, поділ тексту на сторінки, поділ сторінок на колонки, задання кольору, тем, колонтитулів тощо. Форматування здійснюється за допомогою відповідних кнопок на вкладці «Розмітка сторінки».

Для того, щоб застосувати відразу весь набір атрибутів форматування, стандартизувати оформлення документів, скоротити термін їх створення, використовують *стилі форматування*. Розрізняють стиль символів, абзаців, заголовків, текстів, анотацій, приміток, номерів сторінок. Є вбудовані стилі та стилі користувачів. Перші призначені для форматування стандартних складових документа, їх можна модифікувати, але не вилучати. Власні стилі користувачі створюють безпосередньо або шляхом модифікування наявних. Вибір стилю виконується на вкладці «Головна».

Робота з об'єктами в текстовому процесорі.

Робота з таблицями, графічними об'єктами, формулами в MS Word.

Таблиці в документах Word використовують, здебільшого, для упорядкування представлення даних. Таблиці можна застосовувати для створення бланків документів, а табличні дані можна використовувати для створення діаграм. Комірки таблиці можуть містити текст, графічні об'єкти, вкладені таблиці.

Для вставки таблиці використовують вкладку «Вставка».

Для роботи з таблицями в Microsoft Word застосовують контекстні вкладки «Конструктор» і «Макет» групи вкладок «Робота з таблицями». Ці вкладки автоматично відображаються, коли курсор знаходиться в будь-якій комірці існуючої таблиці, і зникають, коли курсор знаходиться за межами таблиці. Не існує способів примусового відображення контекстних вкладок.

Таблицю можна створити в будь-якому місці документа. Вона може розміщуватися на декількох сторінках. Допускається змінювати розмір і положення таблиці на сторінці, призначити їй назву і визначити першу рядок таблиці в якості заголовка таким чином, що при переході таблиці на нову сторінку заголовок буде автоматично з'являтися на новій сторінці. Для створення таблиці можна використовувати вже готовий макет, що значно спрощує її форматування.

Редагування вмісту комірок проводиться за допомогою тих же прийомів, які використовуються для редагування тексту або графіки в документі. Перш ніж використовувати команди редагування таблиць, потрібно виділити потрібні комірки, рядки або стовпці за допомогою миші або відповідних команд. З елементами таблиці можна здійснювати різні маніпуляції: копіювання, переміщення, видалення, об'єднання та інше.

У таблицях Word можна виконувати нескладні обчислення з використанням формул. Формула вставляється в комірку таблиці як спеціальне поле. При зміні значень в обчислюваних комірках результат не буде автоматично змінюватися. Для перерахунку слід виділити комірку і натиснути клавішу F9.

У документ Word можна вставити такі типи графіки як малюнок, зображення, геометричні фігури, SmartArt, діаграма за допомогою кнопок «Малюнок» (малюнки з графічних файлів всіх основних графічних форматів: jpeg, bmp, gif, tiff та ін.), «Картинка» (малюнки бібліотеки графіки Microsoft), «Фігури» (прямокутники, фігурні стрілки, виноски, та ін.), «SmartArt» і «Діаграма», розташованих на вкладці «Вставка» в групі «Ілюстрації». Крім того, графічні об'єкти або векторну графіку «Напис» і «WordArt» можна вставити з групи «Текст» на вкладці «Вставка».

Для роботи з малюнками застосовують контекстну вкладку «Формат» групи вкладок «Робота з малюнками». Ця вкладка автоматично відображається при виділенні малюнка. Для виділення малюнка слід клацнути по ньому мишею. Ознакою виділення є маркери малюнка.

Після вставки графіки в документ, на Стрічці з'являються контекстні вкладки під загальною назвою, яка відображається в рядку заголовка вікна програми. Контекстні інструменти з'являються тільки тоді, коли в документі виділено об'єкт певного типу:

- ✓ «Формат» в групі «Робота з малюнками» (вставка растрових малюнків з файлу і кліпу);
- ✓ «Формат» в групі «Засоби малювання» (вставка в документ готових фігур);
- ✓ «Конструктор», «Формат» в групі «Робота з малюнками SmartArt» (вставка графічного елемента SmartArt для візуального представлення інформації);
- ✓ «Конструктор», «Макет», «Формат» в групі «Робота с діаграмами» (вставка діаграми для представлення та порівняння даних);
- ✓ «Формат» в групі «Робота з написами» (вставка попередньо відформатованих написів);
- ✓ «Формат» в групі «Робота з об'єктами WordArt» (вставка декоративного тексту в документ).

Інструменти для редагування зображень в Word 2010 дозволяють обрізати фотографії, змінювати насиченість, колірну температуру, яскравість, контрастність, оформляти їх спеціальними ефектами без звернення до додаткових графічних редакторів.

Геометричні фігури рекомендується групувати, тобто об'єднувати в єдине ціле.

Малюнки і текстові поля можна накладати один на одного в різній послідовності. Для того щоб малюнок можна було вільно переміщувати в документі, слід встановити для нього один з режимів обтікання текстом (навколо рамки, по контуру, наскрізне, зверху і знизу, за текстом, перед текстом).

Microsoft Word 2010 має можливість вставки в документ математичних формул як в довільному вигляді, так і з готових шаблонів. У попередніх версіях редактора для цього були використані надбудови Microsoft Equation 3.0 або Math Type. Варто відзначити, що Equation 3.0 є вбудованим додатком в Word, включаючи Word 2010, а ось Math Type необхідно встановлювати додатково. Щоб створити формулу, потрібно в меню «Вставка» в групі «Символи» клацнути стрілку, що знаходиться поруч з кнопкою «Формула». Далі слід обрати або одну з готових поширених математичних формул, або створити власну. Для роботи з формулою застосовують контекстну вкладку «Конструктор» групи вкладок «Робота з формулами». Щоб створити формулу за допомогою Microsoft Equation 3.0, на вкладці «Вставка» в групі «Текст» скористатися кнопкою «Об'єкт» і у діалоговому вікні «Вставка об'єкта» обрати пункт Microsoft Equation 3.0.

Електронні таблиці.

Табличний процесор MS Excel: призначення та можливості. Типи даних. Робота з формулами. Побудова графіків.

Табличними процесорами називають пакети програм, які призначені для створення електронних таблиць (ЕТ) і маніпулювання їх даними. Серед таких пакетів найбільшого поширення набули Lotus 1-2-3, Gnumeric, SuperCalc і Microsoft Excel.

Табличні процесори являють собою зручний засіб для проведення економічних, бухгалтерських, інженерних і статистичних розрахунків. Вони дозволяють не тільки створювати таблиці, але і проводити складний економічний аналіз, моделювати і оптимізувати рішення різного роду господарських ситуацій.

До основних функцій електронних таблиць відносять:

- ✓ створення, редагування, форматування і друк ЕТ;
- ✓ створення багатотабличних документів, об'єднаних формулами;
- ✓ побудова діаграм, їх модифікація і розв'язування економічних завдань графічними методами;
- ✓ робота з ЕТ як з базами даних: сортування таблиць, вибірка даних за запитом, створення підсумкових і зведених таблиць;
- ✓ використання інформації при побудові таблиць із зовнішніх баз даних;
- ✓ вирішення економічних завдань типу «що-якщо» шляхом підбору параметрів;
- ✓ розв'язування оптимізаційних завдань;
- ✓ статистична обробка даних;
- ✓ створення слайд-шоу;
- ✓ розробка макрокоманд, налаштування середовища під потреби користувача і т.д.

Microsoft Excel – програма багатовіконна, що дозволяє одночасно завантажувати декілька файлів. Вікно Excel містить безліч різних елементів, деякі з них властиві всім програмам ОС Windows, інші є тільки у вікні Excel. Документ, який створюється і обробляється даною програмою, називається Книгою і складається з листів. Листи можна додавати, видаляти, міняти місцями і перейменовувати. Кожен лист складається з рядків (нумеруються арабськими цифрами) і стовпців (іменуються літерами латинського алфавіту). Кожна комірка має ім'я (адресу), що складається з назви стовпця і номера рядка, на перетині яких вона знаходиться.

Введення і редагування даних може виконуватися по-різному в залежності від типу даних. MS Excel дозволяє вводити в комірки інформацію, одного з наступних типів:

- ✓ Текст;
- ✓ Числа;
- ✓ Дата і час;
- ✓ Формули.

Формула – це вираз, який може включати числа, функції, адреси комірок (посилання), знаки операцій. Формула починається зі знаку рівності (=)! Якщо значення в комірках, на які є посилання в формулі, змінюються, то результат зміниться автоматично. При обробці формул MS Excel дотримується основних математичних правил:

- в першу чергу обчислюються вирази всередині дужок;
- спочатку виконуються множення і ділення, потім додавання і віднімання;
- оператори з однаковим пріоритетом виконуються зліва направо;
- кількість дужок, які відкриваються, і дужок, які закриваються, має бути однаковою.

Діаграма – це графічно представлена залежність однієї величини від іншої. За допомогою діаграм взаємозв'язок між даними стає більш наочним, полегшується порівняння різних даних. Для

аналізу та наочного подання даних Excel пропонує різні типи діаграм: пелюсткову, кругову, гістограму, графік, точкову тощо. Крім кругової, усі типи діаграм можуть відображати декілька рядів даних. Створення діаграми починається із введення на листі числових даних для діаграми. Потім ці дані необхідно нанести на діаграму, вибравши потрібний тип діаграми на вкладці «Вставка», група «Диаграммы». Можна побудувати впроваджені діаграми і діаграми на окремих листах. Діаграма складається з багатьох елементів. Деякі з цих елементів відображаються за замовчуванням, інші можна додавати в разі необхідності. Вигляд елементів діаграми можна змінювати, переміщаючи їх на інше місце на діаграмі, змінюючи розмір або формат. Також можна видаляти непотрібні елементи з діаграми. Окрім того, MS Excel пропонує багато готових макетів діаграм.

Електронні презентації.

Загальні поняття презентацій. Редактор презентацій MS PowerPoint. Принципи побудови презентацій. Оформлення слайдів. Використання зовнішніх об'єктів. Використання анімації. Додаткові можливості.

Комп'ютерна презентація – це файл, в який зібрані матеріали виступу, підготовлені у вигляді комп'ютерних слайдів. Презентація зазвичай містить в собі текст, ілюстрації до нього і витримана в єдиному графічному стилі. Слайд – це окремий кадр презентації, що може містити в собі заголовок, текст, графіку, діаграми, відео-фрагмент тощо.

Етапи створення презентації:

1. Збір необхідної інформації.
2. Планування виду презентації, можливо з використання шаблонів.
3. Редагування та оформлення слайдів.
4. Налаштування спецефектів для демонстрації презентації.
5. Попередній перегляд та редагування у разі необхідності.
6. Збереження на енергонезалежному носії.

Рекомендації щодо створення презентації:

✓ Перший слайд презентації має містити назву проекту (роботи), прізвище, ім'я, по батькові виконавця, номер групи, а також прізвище, ім'я, по батькові, науковий ступінь, посаду керівника.

✓ Кожен слайд повинен містити заголовок. В кінці заголовків крапка не ставиться. Використовуйте слова, написані великими літерами, тільки для коротких заголовків.

✓ Текст пояснень має бути коротким і складеним з простих речень. Всього на слайді має бути не більше 6–8 рядків тексту. Дієслова мають бути використані в одній часовій формі.

✓ Інформацію краще розташовувати горизонтально. Найбільш важлива інформація повинна розташовуватися в центрі екрану.

✓ Для виділення інформації слід використовувати жирний шрифт або курсив; підкреслення застосовувати не бажано, оскільки даний спосіб виділення тексту співпадає з гіперпосиланням.

✓ Для заголовків використовують кегль не менше 24 пт, для основного тексту – не менше 18 пт.

✓ Текст варто підсилувати доречною графічною інформацією (схеми, діаграми, фотографії).

Підписи до ілюстрацій доцільно розміщувати знизу.

✓ Для заголовків застосовують шрифти без засічок (наприклад, Arial), для основного тексту – гарнітури з засічками (наприклад, Times New Roman).

✓ Для наукових презентацій максимальна кількість різних шрифтів на одному слайді – два, для інших видів дозволяють не більше трьох.

✓ На одному слайді рекомендується використовувати не більше трьох кольорів: один для фону, один для заголовка, один для основного тексту.

✓ Для фону і тексту слід використовувати контрастні кольори.

✓ Особливу увагу звертають на колір гіперпосилань (до і після використання).

✓ Презентація повинна бути виконана в єдиному стилі. Один і той же елемент на різних слайдах повинен бути одного кольору.

✓ Не перевантажуйте слайди анімаційними ефектами. Для зміни слайдів використовуйте один і той же анімаційний ефект.

✓ Музичний супровід має бути доцільним і ненав'язливим.

Перевагою програми MS PowerPoint 2010 є використання в її інтерфейсі стрічки і вкладок команд, оформлених в єдиному стилі з іншими програмами пакету Microsoft Office.

Спільна робота програми з SharePoint Workspace і SharePoint Server забезпечує швидкий обмін інформацією.

Вбудовані теми в PowerPoint пропонують широкий асортимент макетів і параметрів форматування, що дозволяють створювати однаковий професійний дизайн для всіх презентацій (вкладка «Дизайн»). При форматуванні презентацій з використанням тем макети слайдів, фони, шрифти, кольори і ефекти форматування зображень оновлюються відразу, завдяки чому всі елементи презентації доповнюють один одного. При створенні в презентації зображень Office Art (таких, як діаграми SmartArt і графіки Excel) форматування автоматично узгоджується з обраною темою.

На вкладці «Переходи» знаходяться інструменти для настройки ефектів плавного переходу між слайдами.

Елементи, розташовані на вкладці «Анімація», дозволяють зробити матеріал більш живим і насиченим. Схеми анімації – це вбудовані в PowerPoint способи, що задають появу на екрані вмісту слайда під час демонстрації презентації.

Вкладка «Показ слайдів» містить елементи, призначені для перегляду готової презентації в повноекранному режимі і виконання необхідної підготовки до показу.

На вкладці «Вид» можна обрати найбільш зручний режим представлення слайдів на екрані в поточний момент роботи над проектом.

У PowerPoint 2010 всі слайди можна розділити на логічні групи або розділи, що дозволяє легше переміщатися між ними і впорядковувати їх.

Макет визначає оформлення елементів слайда, тобто присутність і розташування заголовків, підзаголовків, списків, малюнків, таблиць, діаграм, автофігур і відеофрагментів, SmartArt на слайді.

PowerPoint надає безліч нових і вдосконалених засобів для роботи з зображеннями: художні ефекти, засоби корекції, засоби роботи з кольором. За допомогою засобу «Видалити фон» можна сфокусувати увагу глядачів тільки на необхідних фрагментах зображення, відкинувши все непотрібне.

PowerPoint має широкі можливості для створення інтерактивної презентації. *Інтерактивність презентації* – це можливість користувача управляти показом презентації: вибирати, на який слайд перейти, відкрити потрібний документ тощо. У PowerPoint інтерактивність можна створити трьома способами:

- 1) за допомогою гіперпосилань;
- 2) за допомогою керуючих кнопок;
- 3) за допомогою тригерів.

Для переходу з одного слайда на інший, до ресурсу в Інтернеті або до іншого файлу або програми можна скористатися *гіперпосиланнями*, в якості якого може бути текст, малюнок, фігура, кнопка та ін.

Керуюча кнопка – це кнопка, при натисканні на яку в процесі демонстрації презентації, відбувається перехід на слайд або документ. Відмінності керуючої кнопки від звичайного гіперпосилання в тому, що дія, яка відбувається може бути різною: перехід по слайдам, запуск програми, запуск макросу, запуск звуку. Крім того, для керуючої кнопки можна вибрати один з двох варіантів спрацьовування кнопки: або по клацанню мишею, або по наведенні покажчика миші.

Тригер (або перемикач) – засіб анімації, що дозволяє задати умову дії або часу виділеному елементу анімації. При цьому анімація запускається по клацанню на тригері. Тригер може запускати кілька ефектів.

Інформаційні моделі та системи.

Бази даних і системи з базами даних. Трирівнева архітектура баз даних. Моделі даних. Методологія «сутність-зв'язок». Ключі та їх призначення.

База даних (БД) – це сукупність взаємозв'язаних даних про групу об'єктів, містить інформацію необхідну для автоматизації роботи з даними.

Система управління базами даних (СУБД) – це сукупність мовних і програмних засобів, призначених для створення, ведення і спільного використання БД багатьма користувачами.

Архітектура СУБД розділена на *три основні рівні*: зовнішній, внутрішній, концептуальний.

На зовнішньому рівні користувачеві надається інтерфейс для маніпуляції даних, задаються права доступу користувачів на взаємодію з базою даних, пріоритети і т.д.

У триповерхових СУБД є завжди є внутрішній рівень, на якому СУБД і операційна система сприймають і обробляють дані.

Концептуальний рівень відповідає за зв'язок між зовнішнім рівнем (інтерфейс користувача) і внутрішнім рівнем (фізичним представленням даних у базі даних). Він призначений для відображення зовнішнього рівня на внутрішній, забезпечує незалежність між цими рівнями.

Опис структури даних на будь-якому з трьох рівнів називається схемою даних, кожна з яких має свою власну назву і залежить від рівня. На найвищому або зовнішньому рівні є кілька зовнішніх схем даних або підсхем. На концептуальному рівні опис бази даних відбувається за допомогою концептуальних схем. Внутрішній рівень СУБД описується за допомогою внутрішньої схеми даних.

Метою створення трирівневої архітектури бази даних була незалежність даних від рівнів. Під терміном незалежністю даних потрібно розуміти наступне: зміни на нижніх рівнях (концептуальний і внутрішній рівень) в ідеалі ніяк не повинні впливати на верхній рівень. Існує два типи незалежності даних: логічна незалежність від даних і фізична незалежність від даних.

Логічна незалежність від даних означає, що всі зовнішні схеми залишаться незмінні, якщо будуть вноситися зміни на концептуальному рівні СУБД (в концептуальну схему даних). Вносити зміни в концептуальну схему даних означає: додавання і видалення нових сутностей (нових таблиць), додавання атрибутів (стовпців) або створення нових зв'язків між таблицями. Всі перераховані вище операції повинні здійснюватися без необхідності внесення змін у вже існуючі зовнішні схеми даних.

Фізична незалежність від даних означає, що концептуальна схема даних буде захищена від змін і буде незалежна від змін на внутрішньому рівні (зміна внутрішньої схеми даних). Під зміною внутрішньої схеми даних треба розуміти: зміну файлової структури зберігання даних, використання різних файлових систем, використання різних запам'ятовуючих пристроїв, модифікацію індексів тощо. Всі ці зміни ніяк не повинні впливати на концептуальну і зовнішню схеми даних.

Модель даних – це деяка абстракція, що, будучи застосовна до конкретних даних, дозволяє користувачам і розроблювачам трактувати їх уже як інформацію, тобто відомості, що містять не тільки дані, але й взаємозв'язок між ними.

Моделі БД:

- 1) ієрархічна;
- 2) мережева;
- 3) реляційна;
- 4) об'єктно-орієнтована;
- 5) багатовимірна.

Ієрархічна БД – відносини між елементами записів мають деревоподібну (ієрархічну) структуру. Елементи такого дерева відносин називають вузлами. Кожен вузол підкоряється тільки одному, який стоїть вище, на найвищому рівні – вузол, що не підкоряється жодному, тобто корінь

дерева. Типовий приклад – файлова структура на диску з каталогами різного рівня. Пошук даних при цьому вельми трудомісткий – необхідно послідовно, крок за кроком, пройти всі підлеглі каталоги, переміщаючись по гілках дерева каталогів.

Мережева БД – має гнучкішу організацію, кожен елемент запису може мати довільні зв'язки з будь-яким іншим елементом, до вертикальних зв'язків додаються горизонтальні. Пошук елементів полегшується.

Реляційна БД – найпоширеніший тип. Інформація організована у вигляді двовимірної таблиці, однієї або декількох взаємозв'язаних. Зв'язок між елементами-записами здійснюється по загальних полях.

Об'єктно-орієнтовані БД – використовують об'єктно-орієнтовану технологію, коли дані і код для їх обробки упаковані разом. При цьому знімаються обмеження на типи даних і їх довжину.

Багатовимірні БД – на відміну від реляційних 2-мірних, використовують для зберігання даних 3-х, 4-х або N-мірний куб. Застосовуються для аналізу і прогнозування.

Як бачимо, є декілька моделей баз даних. Деякі з них значно ефективніше реляційної, але не набули широкого поширення із-за складності створення відповідних СУБД.

Будь-який об'єкт в таблиці унікальний, повторень бути не повинно. Серед всіх полів, що описують об'єкт має бути унікальне поле – ключ, що однозначно характеризує об'єкт. По ключовому полю здійснюється ідентифікація об'єкта. Найчастіше це поле Код-запису типу «счетчик», може бути також інвентарний номер, номер студентського квитка тощо. Ключ може бути простим або складним, який складається з декількох ключових полів. Поле або набір полів, що гарантують унікальність запису, має назву первинний ключ.

Як правило в БД не одна таблиця, а декілька зв'язаних між собою. Тому існує поле-зовнішній ключ – це поле, по якому організовується зв'язок між таблицями. У останньому випадку поле-первинний ключ однієї таблиці є також полем в іншій таблиці.

У теорії СУБД відомі 4 варіанти **зв'язків**, традиційно званих відносинами, вони встановлюють правила відповідності записів в двох таблицях, причому «дивимосся» як би з боку першої таблиці, вона є «головною», а друга «підлеглою»:

1. Відношення **один-до-одного**, одному запису першої таблиці відповідає не більше одного запису другої таблиці і навпаки. Поле – первинний ключ першої таблиці пов'язано з таким самим полем в другій таблиці, яке також є первинним ключем. У обох полях не повинно бути значень, що повторюються, і значення повинні відповідати один одному. Можна сказати, що це не дві таблиці, а одна, розділена надвоє вертикальною межею, так що частина полів опинилася в першій таблиці, а частина в другій. Розділення йде за функціональною ознакою, мета – введення записів на одному робочому місці.

2. Відношення **один-до-багатьох**, в цьому випадку одного запису першої таблиці може відповідати декілька записів другої таблиці. Поле – первинний ключ першої таблиці пов'язано з таким самим полем в другій таблиці, яке є зовнішнім ключем, значення по зовнішньому ключу можуть повторюватися.

3. Відношення **багато-до-одного**, декільком записам першої таблиці відповідає один запис з другої таблиці. Можна сказати, що це відношення один-до-багатьох навпаки, тепер як би розглядаємо з боку другої таблиці.

4. Відношення **багато-до-багатьох**, декільком записам першої таблиці відповідає декілька записів другої таблиці. В цьому випадку зв'язки між таблицями непрямі, організовані за допомогою ще однієї таблиці з використанням складного ключа, тобто набору ключових полів.

Робота з текстовими даними.

Основи роботи в середовищі LibreOffice. Текстовий процесор Writer. Загальні поняття. Редагування та форматування тексту. Додаткові можливості.

LibreOffice – програмне забезпечення, що дозволяє обробляти тексти, електронні таблиці, бази даних, презентації, графіки та багато іншого. LibreOffice працює на операційних системах Microsoft Windows, Gnu/Linux та Mac OS X. Пакет доступний на різних мовах (зокрема українською), його можна використовувати вільно для будь-яких цілей. Можливості LibreOffice можна порівняти з можливостями аналогічних комерційних програм, тому дане програмне забезпечення розглядається як гідна їм альтернатива. У пакет LibreOffice входять такі програми:

Base – система управління базами даних. За функціоналом утиліта мало чим поступається MS Access. Відносно інтерфейсу, то виглядає вона надто вже просто, проте це абсолютно не впливає на якість виконання нею повсякденних завдань. Програма підтримує всього лише один формат – ODB.

Calc – табличний процесор – управління листами і аналіз інформації в електронних таблицях, виконання обчислень. Програма підтримує практично всі функції MS Excel. Calc працює з наступними форматами файлів: ODS, SXC, STC, DIF, DBF, OTS, XLS, XLSX, XLT, XLTX, SLK, CSV, HTML, XML.

Draw – векторний графічний редактор для створення і редагування ілюстрацій, малюнків, логотипів і блок-схем. Програма навряд чи замінить графічний редактор, вона більше підходить для створення малюнків до документів – креслень, схем, діаграм. Також можна створювати прості об’ємні фігури, наприклад, куби, сфери та циліндри і навіть змінювати освітлення об’єктів. Draw підтримує такі формати файлів: ODG, OTG, SXD, STD.

Impress – програма для створення і редагування презентацій для зустрічей, слайд-шоу і веб-сторінок, що володіє простеньким та інтуїтивно зрозумілим інтерфейсом. У LibreOffice Impress включені шаблони для створення слайдів професійної якості. Для слайдів можна також призначити ряд динамічних ефектів, в тому числі анімацію і ефекти переходів, можна вибрати автоматичну або ручну зміну слайдів при показі. Працює з такими типами файлів: ODP, SXI, STI, DIF, PPS, PPT, OTP, POT, PPTX, SXD, ODG, PPTX, PPSX, POTM.

Math – редактор формул, призначений для створення і редагування рівнянь і наукових формул. Програма дозволяє будувати складні математичні рівняння для друку і презентацій в одному з форматів: ODF, SXM, MML. Серед особливостей LibreOffice Math – можливість використання без установки.

Write – текстовий процесор, який дозволяє створювати, редагувати тексти і малюнки в листах, документах, звітах або веб-сторінках. Продукт підтримує такі формати: ODT, OTT, SXW, RTF, DOCX, DOC, TXT, HTML, XML. Інтерфейс текстового процесора схожий із зовнішнім виглядом перших версій MS Word. Крім цього, LibreOffice Write імітує різні функції MS Word, такі як запис макросів, перехресні індекси і посилання, виноска, малювання і таблиці. Відмітна особливість програми – функція автозавершення – прописування найбільш ймовірного слова, ще до закінчення його набору. Можна також створювати в текстових документах різні індекси і таблиці. Є можливість визначення структури і зовнішнього вигляду індексів і таблиць в залежності від конкретних потреб користувача. Активні гіперпосилання і закладки дозволяють переходити прямо до відповідних елементів в тексті.

Writer містить різні настільні видавничі та графічні засоби, які сприяють швидкому створенню професійно оформлених документів, таких як брошури, бюлетені та запрошення. При форматуванні документів можна використовувати розмітки з декількома колонками, врізки, графічні об’єкти, таблиці та інші об’єкти.

Текстові документи в LibreOffice мають вбудовану функцію розрахунку, яка допомагає виконувати складні обчислення або створювати логічні зв'язки. В текстовому документі може бути легко створена таблиця для виконання розрахунків.

Інтернет та комунікація.

Поняття мережі Інтернет. Браузери, їх приклади та особливості. Робота з електронною поштою. Outlook Express. Електронна пошта: протоколи SMTP. Організація доступу до поштової скриньки користувача за допомогою протоколу POP3. Стандарт MIME.

Сьогодні у світі є сотні тисяч великих та малих комп'ютерних мереж. Багато з них з'єднані між собою й утворюють єдиний інформаційний простір, який складається з мільйонів комп'ютерів.

Цей єдиний віртуальний простір називають *Інтернетом*. За кількістю комп'ютерів та охоплюваною територією Інтернет є найбільшою у світі мережею. За типом вона належить до клієнт-серверних мереж, тобто в Інтернеті є комп'ютери-сервери, які зберігають інформацію та надають її комп'ютерам-клієнтам.

Обмін інформацією між серверами та клієнтами здійснюється за певними правилами, які називають *протоколами*. Всі дані, що циркулюють у глобальному інформаційному полі, розбито на невеликі блоки і вкладено в пакети. Кожний пакет окрім даних має заголовок, де зберігаються адреса відправника, адреса одержувача та інша інформація, необхідна для збирання пакетів у пункті призначення. Теоретично можливо, що різні пакети одного повідомлення пройдуть різними шляхами, але все одно досягнуть адресата і будуть зібрані в повне повідомлення.

Поділ даних на пакети та їх збирання у пункті призначення здійснюється під керуванням протоколу TCP (Transmission Control Protocol – протокол керування передаванням), а власне передавання пакетів мережею та досягнення ними адресата забезпечує протокол IP (Internet Protocol – міжмережний протокол).

У Інтернеті використовується велика кількість протоколів, завдяки чому існує широкий спектр служб, які надаються та підтримуються за допомогою цієї глобальної мережі.

Служби (сервіси) Інтернету – це набір послуг, які надаються клієнтам Інтернету. До них належать:

1. World Wide Web – *всесвітня павутина* – служба пошуку та перегляду гіпертекстових документів, що включають в себе графіку, звук і відео. Ця служба базується на протоколі HTTP (Hyper Text Transfer Protocol – протокол передавання гіпертексту).

2. E-mail – *електронна пошта* – служба передачі електронних повідомлень.

3. Usenet, News – *телеконференції, групи новин* – різновид мережевої газети або дошки оголошень.

4. FTP – служба передачі файлів.

5. ICQ – служба для спілкування в реальному часі за допомогою клавіатури.

6. Telnet – служба віддаленого доступу до комп'ютерів.

7. Gopher – служба доступу до інформації за допомогою ієрархічних каталогів.

Також до служб Інтернету належать блоги, Вікі-проекти (в тому числі й Вікіпедія), Інтернет-магазини, Інтернет-аукціони, електронні платіжні системи, Інтернет-телебачення, Інтернет-радіо, IP-телефонія.

Браузер – програмне забезпечення для комп'ютера або іншого електронного пристрою, як правило, під'єданого до Інтернету, що дає можливість користувачеві взаємодіяти з текстом, малюнками або іншою інформацією на гіпертекстовій веб-сторінці. Тексти та малюнки можуть містити посилання на інші веб-сторінки, розташовані на тому ж веб-сайті або на інших веб-сайтах.

У глобальній мережі браузери використовують для запиту, обробки, маніпулювання і відображення змісту веб-сайтів. Багато сучасних браузерів також можуть використовуватися для обміну файлами з серверами ftp, а також для безпосереднього перегляду змісту файлів багатьох

графічних форматів (gif, jpeg, png, svg), аудіо-відео форматів (mp3, mpeg), текстових форматів (pdf, djvu) та інших файлів. Загальні, первинні і вторинні функції та послуги, які пропонують браузері включають також завантаження, закладки та управління паролями. Вони також пропонують такі функції, як перевірка орфографії, панелі інструментів пошукової системи, вкладки, фільтрацію реклами, ключів доступу HTML і блокування спливаючих вікон.

Браузери поширюються, як правило, безкоштовно. Споживачам браузер може бути поставлений в формі самостійного (автономного) додатку або в складі комплектного програмного забезпечення. Наприклад, браузері Internet Explorer і Microsoft Edge поставляються в складі операційної системи Microsoft Windows; Mozilla Firefox – окремо або в складі дистрибутивів Linux (наприклад, Ubuntu); Safari – в складі операційної системи Mac OS X; Google Chrome, Opera та інші браузері – як окремі програми з безліччю варіантів для різних операційних систем.

Електронна пошта – популярний сервіс в Інтернеті, що робить можливим обмін даними будь-якого змісту (текстові документи, аудіо-відео файли, архіви, програми). Вона дуже корисна, якщо немає повноцінного доступу (on-line) до Інтернету. Електронна пошта – типовий сервіс відкладеного зчитування (off-line). Після відправлення повідомлення адресат отримує його на свій комп'ютер через деякий період часу, і знайомиться з ним, коли йому буде зручно.

Електронна пошта повторює переваги (простоту, дешевизну, можливість пересилання нетекстової інформації, можливість підписати і зашифрувати лист) та недоліки (негарантований час пересилки, можливість доступу для третіх осіб під час пересилки, неінтерактивність) звичайної пошти.

Користувач електронної пошти має електронну скриньку, розміщену на одному з серверів електронної пошти. Доступ до скриньки зумовлений введенням імені користувача (слово з латинських літер) та пароля. Відповідно надсилати листи можна конкретній людині, котра також має свою поштову скриньку.

Дана служба чи не найстаріша в інтернет. Найпопулярніші програми-клієнти електронної пошти: Eudora Mail, Evolution, KMail, Mozilla Mail, Mozilla Thunderbird, Netscape Mail, Novell GroupWise, Opera Mail (M2), Outlook, Outlook Express, TheBat!

Отримати поштову скриньку можна у свого провайдера інтернет (фірма що надає доступ до Інтернет) або зареєструвати скриньку на одному з безкоштовних серверів.

Outlook Express – програма для роботи з електронною поштою і групами новин від компанії Microsoft. Outlook Express поставляється у складі операційних систем Windows.

Outlook Express дозволяє з легкістю керувати кількома поштовими скриньками з одного вікна.

Дана програма дуже функціональна. З її допомогою кожен користувач має можливість:

- підшукувати цікаві групи новин для читання і підписуватися на них;
- блокувати небажану пошту (чорний список);
- групувати повідомлення і відповіді. Корисна функція, особливо коли в бесіді з e-mail бере участь одночасно кілька людей. Outlook Express сортує відповіді таким чином, щоб вони відображалися під відповідними повідомленнями;
- підписувати повідомлення, створюючи файли підписів, які можуть містити посилання на особисті ресурси, контактні дані або просто якусь цікаву фразу або цитату;
- відправляти повідомлення без орфографічних помилок, завдяки функції «Орфографія»;
- масово пересилати повідомлення;
- автоматично отримувати нову пошту з кожним підключенням до мережі;
- імпортувати і експортувати адресну книгу;
- використовувати Outlook Express в якості браузера;
- відправляти повідомлення великих розмірів.

Як вже зазначалося, в основі будь-якої взаємодії, яка відбувається в мережі, є спеціальні правила, які визначають послідовність обміну інформацією та її вид. Такі правила називаються протоколом. Поштові програми різні протоколи для прийому і відправки пошти.

Так, наприклад, протокол SMTP, що працює за принципом клієнт-сервер, призначений для відправки повідомлень з комп'ютера до адресата. Зазвичай доступ до сервера SMTP не захищається паролем, так що можна використовувати для відправки листів будь-який відомий сервер в мережі. На відміну від серверів для відправки листів, доступ до серверів для зберігання повідомлень захищається паролем. Тому необхідно використовувати сервер або службу, в якій існує обліковий запис. Ці сервери працюють по протоколах POP і IMAP, які розрізняються способом зберігання листів.

Відповідно до протоколу POP3 повідомлення, які надходять на певну адресу, зберігаються на сервері до того моменту, поки вони не будуть завантажені на комп'ютер протягом чергового сеансу. Після завантаження повідомлення можна відключитися від мережі і перейти до читання пошти. Таким чином, використання пошти по протоколу POP3 є найбільш швидким і зручним у використанні.

MIME (Multipurpose Internet Mail Extensions – багатоцільові розширення інтернет-пошти) – стандарт, що описує передачу різних типів даних електронною поштою, а також, в загальному випадку, специфікація для кодування інформації і форматування повідомлень таким чином, щоб їх можна було пересилати через Інтернет.

Безпроводні мережі.

Стандарт IEEE 802.11 безпроводних мереж (Wi-Fi). Управління доступом до каналу зв'язку.

Безпроводна мережа – тип комп'ютерної мережі, яка використовує бездротове з'єднання для передачі даних й підключення до мережевих вузлів.

Бездротові мережі, звані Wi-Fi-або WLAN (Wireless LAN)-мережі, мають, в порівнянні з традиційними провідними мережами чималі переваги, головною з яких є простота розгортання. Бездротова мережа не потребує прокладання кабелів, гарантує мобільність користувачів в зоні її дії і простота підключення до неї нових користувачів.

У той же час бездротові мережі на сучасному етапі не позбавлені недоліків. Перш за все, це низька, за сьогоднішніми мірками, швидкість з'єднання, яка залежить від наявності перешкод і від відстані між приймачем і передавачем.

Є кілька типів бездротових стандартів: 802.11a, 802.11b і 802.11g. У відповідності з цими стандартами існують і різні типи обладнання. Стандарти бездротових мереж сімейства 802.11 відрізняються один від одного перш за все максимально можливою швидкістю передачі. Так, стандарт 802.11b передбачає максимальну швидкість передачі до 11 Мбіт/с, а стандарти 802.11a і 802.11g - максимальну швидкість передачі до 54 Мбіт/с. Крім того, в стандартах 802.11b і 802.11g передбачено використання одного і того ж частотного діапазону - від 2,4 до 2,4835 ГГц, а стандарт 802.11a увазі застосування частотного діапазону від 5,15 до 5,35 ГГц. Слід врахувати, що стандарт 802.11g повністю сумісний зі стандартом 802.11b, тобто стандарт 802.11b є підмножиною стандарту 802.11g, тому в бездротових мережах, заснованих на обладнанні стандарту 802.11g, можуть також працювати клієнти, оснащені бездротовим адаптером стандарту 802.11b. Вірно і зворотне. Втім, у таких змішаних мережах криється один підводний камінь: якщо ми маємо справу із змішаною мережею, тобто з мережею, у якій є клієнти як з бездротовими адаптерами 802.11b, так і з бездротовими адаптерами 802.11g, то всі клієнти мережі будуть працювати за протоколу 802.11b. Клієнти 802.11b «не чують» клієнтів 802.11g. Тому для того, щоб забезпечити спільний доступ до середовища передачі даних клієнтів, що використовують різні протоколи, в подібних змішаних мережах точки доступу повинні відпрацьовувати певний механізм захисту. В результаті застосування механізмів захисту в змішаних мережах реальна швидкість передачі стає ще менше.

Тому при виборі обладнання для бездротової мережі варто зупинитися на обладнанні одного стандарту. Протокол 802.11b є вже застарілим. Так що оптимальний вибір - обладнання стандарту 802.11g.

Для того, щоб в мережі стала можливою локальна доставка фреймів, необхідна певна система адресації, тобто присвоєння імен комп'ютерам та інтерфейсам. Кожен вузол має унікальний спосіб само-ідентифікації. Ніякі дві фізичні адреси в мережі не повинні збігатись. Фізичні адреси, які в Ethernet також називають адресами керування доступом до середовища передавання (Media Access Control, MAC), записані в мережевому адаптері ПК або мережевих інтерфейсах пристроїв (маршрутизаторів, комутаторів тощо).

MAC-адреса має довжину 48 бітів і записується у вигляді дванадцяти шістнадцяткових цифр (наприклад, 00-60-2F-3A-07-BC). Перші шість цифр, що задаються IEEE, ідентифікують виробника або продавця пристрою і, містять унікальний ідентифікатор організації (Organizationally Unique Identifier – OUI). Другі шість цифр містять серійний номер інтерфейсу або інше значення, що задається конкретним виробником. MAC-адресу іноді називають прошитою, оскільки вона записана в постійній пам'яті інтерфейсу або пристрою.

Без MAC-адрес локальна мережа являла б собою лише групу ізольованих комп'ютерів, і доставка Ethernet-фреймів була б неможливою. Внаслідок цього на каналному рівні до даних

верхніх рівнів додається заголовок, що містить MAC-адресу пристрою та кінцевик. Заголовок та кінцевик містять службову інформацію, призначену для канального рівня пристрою, якому направляється фрейм. Дані верхніх рівнів перетворюються в заголовок та кінцевик канального рівня.

Типи адрес стека TCP/IP. IP-адреси. Протоколи IP та ICMP.

Типи IP-адрес. Доменні імена. Формат IP-адреси, класи IP-адрес. Використання масок під час IP-адресації. Порядок призначення IP-адрес. Формат IP-пакета. Схема IP-маршрутизації. Маршрутизація з використанням масок. Фрагментація IP-пакетів. Призначення і характеристика протоколу ICMP, Формат ICMP-пакета. Типи ICMP-повідомлень.

Розрізняють три типи адрес стека TCP/IP: локальні, IP-адреси та доменні імена.

Під *локальною* (апаратною, фізичною) адресою розуміють такий тип адреси, який використовується засобами базової технології для доставки даних в межах підмережі. В різних підмережах є допустимими різні мережеві технології, різні стеки протоколів, тому при створенні TCP/IP передбачалась наявність різних типів локальних адрес.

Для того, щоб в мережі Ethernet стала можливою локальна доставка фреймів, необхідна певна система адресації, тобто присвоєння імен комп'ютерам та інтерфейсам. Кожен вузол має унікальний спосіб само-ідентифікації. Ніякі дві фізичні адреси в мережі не повинні збігатись. Фізичні адреси, які в Ethernet також називають адресами керування доступом до середовища передавання, записані в мережевому адаптері ПК або мережевих інтерфейсах пристроїв (маршрутизаторів, комутаторів тощо).

MAC-адреса має довжину 48 бітів і записується у вигляді дванадцяти шістнадцяткових цифр. Перші шість цифр, що задаються IEEE, ідентифікують виробника або продавця пристрою і, містять унікальний ідентифікатор організації. Інші шість цифр містять серійний номер інтерфейсу або інше значення, що задається конкретним виробником.

IP-адреси є основним типом адрес, на основі яких мережевий рівень передає пакети між мережами. Ці адреси складаються з 4 байтів і записуються у десятково-точковій нотації (наприклад, 195.1.7.26). Кожна з частин адреси, розділених точками, називається октетом (оскільки складається з 8 біт). IP-адреса призначається адміністратором під час конфігурування комп'ютерів та маршрутизаторів.

Символьні доменні імена. Вся мережа Internet побудована на ієрархічній системі адресації. Такий підхід дозволяє здійснювати маршрутизацію, засновану на класах адрес, а не на індивідуальних адресах. Однак використання IP-адрес не дуже зручно для користувачів. Ймовірність того, що користувач може помилитися і ввести неправильну IP-адресу, досить висока, оскільки числа IP-адреса ніяк не пов'язана з тематикою ресурсу.

Для прив'язування вмісту Web-сторінки та її адреси була розроблена спеціальна система доменних імен (DNS). Служба DNS призначена для трансляції IP-адрес в імена і навпаки. Домен – це група вузлів, що розташовані в одній географічній області, або ж вузлів, що використовуються зі спільною метою. Доменним іменем називають рядок символів і/або цифр, і зазвичай таке ім'я відповідає цифровій IP-адресі Web-вузла в мережі Internet.

Сервер доменних імен – мережевий пристрій, який за запитом користувача перетворює доменні імена у відповідні IP-адреси і повертає результат клієнту. Система доменних є строго ієрархічною, тому існує декілька рівнів імен і відповідно серверів DNS.

З метою отримання можливості опису мереж різного розміру та полегшити їх класифікацію, IP-адреси було розділено на групи, які називають класами. Така схема адресації називається класовою. Кожна повна 32-бітна IP-адреса поділяється на дві частини, що описують мережу та вузол. Біт або послідовність бітів на початку кожної адреси задають її клас. Є п'ять класів IP-адрес.

Адреса класу А призначена для дуже великих мереж. В ній використовується тільки перший октет як ідентифікатор мережі. Три октети, що залишились, ідентифікують адресу вузлів. Перший біт в адресі класу А завжди нульовий.

Адреса класу В використовується для мереж середнього та великого розмірів. В IP-адресі класу В два перших октети використовується для мережевої адреси, а два других являють собою адресу вузла.

Адреси класу С – це найчастіше використовувані адреси, призначені для використання в малих мережах. Адреса даного класу починається з двійкової комбінації 110.

Адреси класу D були створені для реалізації в IP-адресах механізму багатоадресної розсилки..

Адреси класу E також були описані в стандартах та виділені в окремий блок. Однак вони були для власних потреб і не використовувались в мережі Internet.

Традиційна схема поділу IP-адрес на номер мережі та номер вузла базується на понятті класу, який визначається значеннями декількох перших бітів адреси.

Альтернативою цієї традиційної схеми є використання іншої ознаки, за допомогою якої можна більш гнучко встановлювати межу між номером мережі та номером вузла. Такою ознакою є маска – 32-розрядне число, яке використовується в парі з IP-адресою. Двійковий запис маски містить одиниці у тих розрядах, які повинні в IP-адресі інтерпретуватись як номер мережі. Оскільки номер мережі є цільною частиною адреси, одиниці у масці повинні являти собою неперервну послідовність.

Для стандартних класів IP-адрес маски мають такі значення:

Клас А – (255.0.0.0);

Клас В – (255.255.0.0);

Клас С – (255.255.255.0).

Механізм масок широко використовується в IP-маршрутизації, причому маски можуть використовуватись з різною метою. За їх допомогою адміністратор може структурувати свою мережу, не вимагаючи від постачальника послуг додаткових номерів мереж. На основі цього ж механізму постачальники послуг можуть об'єднати адресні простори декількох мереж з метою зменшення об'єму таблиць маршрутизації та підвищення за рахунок цього продуктивності роботи маршрутизаторів.

Протокол ICMP (Internet Control Message Protocol RFC-792, 1256) виконує діагностичні функції, а також повідомляє про збої та помилки. Перенесення ICMP-пакета відбувається за допомогою IP-пакета. ICMP посідає особливе місце в стеку протоколів TCP/IP. З одного боку, він використовує IP, і, отже, є на вищому рівні, проте, з іншого, його система адресації та функції не відповідають нормам транспортного рівня, тому зачислити його до цього рівня не можна. ICMP є обов'язковим доповненням до IP.

Головні функції ICMP:

- передавання відповіді на пакет або луна-відповіді;
- контроль часу наявності данограм у системі;
- переадресація пакета;
- видавання повідомлень про недосяжність адресата або некоректність параметрів;
- формування та пересилання часових позначок.

Протоколи TCP та UDP. Система доменних імен DNS. Протоколи DHCP.

Адресація прикладних програм. Порти. Протокол UDP. Формат TCP-сегмента. Логічні з'єднання, порядкові номери та номери підтвердження. Управління вікном прийому. Система DNS, схема роботи DNS. Режими DHCP, алгоритм динамічного призначення адрес.

Протокол TCP (Transmission Control Protocol) є транспортним протоколом стека протоколів TCP/IP, що забезпечує гарантовану доставку даних зі встановленням віртуального з'єднання.

Протокол надає програмам, що використовують його, можливість передачі безперервного потоку даних. Дані, що підлягають відправці в мережу, розбиваються на порції, кожна з яких забезпечується службовою інформацією, тобто формуються пакети даних. У термінології TCP пакет називається сегментом.

Відповідно до функціонального призначення протоколу структура TCP-сегменту припускає наявність наступних інформаційних полів:

- номер порту-відправника і номер порту-одержувача — номери портів, що ідентифікують програми, між якими здійснюється взаємодія;
- поля, призначені для забезпечення гарантованої доставки: розмір вікна, номер послідовності і номер підтвердження (див. Реалізація режиму гарантованої доставки);
- прапори, що управляють, — спеціальні бітові поля, що управляють протоколом.

Протокол UDP (User Datagram Protocol) – протокол транспортного рівня, що входить в стек протоколів TCP/IP, котрий забезпечує негарантовану доставку даних без встановлення віртуального з'єднання.

Оскільки на протокол не покладається завдань по забезпеченню гарантованої доставки, а лише потрібно забезпечувати зв'язок між різними програмами, то структура заголовка дейтаграми UDP (так називається пакет протоколу) виглядає достатньо просто – вона включає всього чотири поля. Перші два поля містять номери UDP-портів програми-відправника і програми-одержувача. Два інших поля в структурі заголовка дейтаграми призначені для управління обробкою – це загальна довжина дейтаграми і контрольна сума заголовка.

Порт – натуральне число, що записується в заголовках протоколів транспортного рівня моделі OSI (TCP, UDP, SCTP, DCCP). Це 16-бітові поля містять номери портів - числа, які визначаються за спеціальним списком. Використовується для визначення процесу-отримувача пакету в межах одного хосту.

Порт джерела ідентифікує додаток клієнта, з якого відправлено пакети. Відповідні дані передаються клієнту на підставі цього номера. Порт призначення ідентифікує порт, на який відправлено пакет.

DNS (Domain Name System,) – розподілена комп'ютерна система для отримання інформації про домени. Основна область застосування даної системи — перетворення імені хоста в IP-адресу і надання даних про маршрутизацію пошти.

Схематичне представлення процесу визначення IP-адреси по запроваджуваному імені домену.

Коли користувач запускає веб-браузер і вводить назву домену сайту, його ПК надсилає запит до DNS-сервера інтернет-провайдера для отримання IP-адреси, на якому знаходиться домен.

Якщо DNS-сервери провайдера не виявляють в своєму кеші інформації про запитуваний сайт, відправляють запит на кореневі DNS-сервери. Кореневий DNS-сервер шукає в своїй базі даних інформацію про сервери імен хостинг-провайдера, на яких присутній цей сайт. Далі, він повідомляє їх DNS-серверу провайдера. Після того, як DNS-сервер інтернет-провайдера отримує інформацію про сервери імен хостинг-провайдера він опитує кожен з них і, у разі отримання позитивного

результату отримання IP-адреси, поміщає в кеш. Після цього DNS-сервер провайдера передає IP-адреса браузеру користувача, що зробив запит сайту. Після цього браузер, отримавши IP-адресу потрібного сайту, переходить на сам сайт.

DHCP (Dynamic Host Configuration Protocol – протокол динамічної конфігурації вузла) – це стандартний протокол, який дозволяє комп'ютерам автоматично одержувати IP-адресу та інші параметри, необхідні для роботи в мережі. Для цього комп'ютер звертається відповідно – до DHCP-сервера. Мережевий адміністратор може задати діапазон адрес, які будуть розподілені між комп'ютерами. Це дозволяє уникнути ручного налаштування комп'ютерів мережі й зменшує кількість помилок. Протокол DHCP використовується в більшості великих мереж TCP/IP.

Технологія прикладного рівня.

Протокол HTTP. Синтаксис протоколу HTTP. Коди відповідей сервера Принцип роботи FTP. FTP-сервер та FTP-клієнт. Протокол FTP.

HTTP – це протокол передачі гіпертексту між розподіленими системами. HTTP є фундаментальним елементом сучасного Web.

Клієнт надсилає запит, який складається з 3-х частин: стартова рядок – у ній вказується тип повідомлення; заголовок – описують тіло повідомлення, визначаючи його параметри; тіло повідомлення – повідомлення, повинно відділятися порожнім рядком. HTTP забезпечує спілкування між безліччю хостів і клієнтів, а також підтримує цілий ряд мережових налаштувань.

Спілкування між хостом і клієнтом відбувається в два етапи: запит і відповідь. Клієнт формує HTTP запит, відповідь на який сервер дає відповідь (повідомлення). Основою веб-спілкування є запит, який надсилається через Єдиний покажчик ресурсу (URL). Протокол https призначений для більш безпечного обміну даними.

Відповідаючи на запит від клієнта, сервер посилає відповідь, яка містить повідомлення про стан код. Даний код несе в собі інформацію яка дає клієнту можливість зрозуміти, як інтерпретувати відповідь:

- 1xx: Інформаційні повідомлення
 - 2xx: Повідомлення про успіх
 - 3xx: Перенаправлення
 - 4xx: Клієнтські помилки
 - 5xx: Помилки сервера
- Служба архівів FTP.

FTP-архіви це розподілений депозитарій текстів, програм, фотографій та іншої інформації, що зберігається у вигляді файлів на різних комп'ютерах у всьому світі.

Інформація в FTP-архівах розділена на три категорії:

Захищена інформація, режим доступу до якої визначається її власниками і дозволяється за спеціальною угодою із споживачем.

Інформаційні ресурси обмеженого використання, до яких відносяться, наприклад, програми класу shareware.

Вільно розповсюджені інформаційні ресурси або freeware, якщо мова йде про програмне забезпечення.

Технологія FTP була розроблена в рамках проекту ARPA і призначена для обміну великими обсягами інформації між машинами з різною архітектурою. Головним у проекті було забезпечення надійної передачі, тому з сучасної точки зору FTP здається перевантаженим зайвими, що рідко використовуються можливостями. Стрижень технології становить FTP-протокол.

FTP-протокол.

FTP - один з найстаріших протоколів Internet і входить в його стандарти. FTP призначений для вирішення завдань поділу доступу до файлів на віддалених хостах, прямого чи непрямого використання ресурсів віддалених комп'ютерів, забезпечення незалежності клієнта від файлових систем віддалених хостів, ефективною і надійною передачею даних. Обмін даними в FTP відбувається по TCP-каналі. Обмін побудований на технології "клієнт-сервер".

FTP з'єднання ініціюється інтерпретатором протоколу користувача. Керування обміном здійснюється по каналу керування в стандарті протоколу TELNET. Команди FTP генеруються інтерпретатором протоколу користувача і передаються на сервер. Відповіді сервера відправляються користувачеві також по каналу керування.

Команди FTP визначають параметри каналу передачі даних і самого процесу передачі. Вони також визначають і характер роботи з віддаленою та локальною файловими системами.

Сесія управління ініціалізує канал передачі даних. При організації каналу передачі даних послідовність дій інша, відмінна від організації каналу управління. У цьому випадку сервер ініціює обмін даними у відповідності з узгодженими в сесії керування параметрами.

Канал даних встановлюється для того ж хоста, що і канал керування, через який ведеться настройка каналу даних. Канал даних може бути використаний як для прийому, так і для передачі даних.

Теоретичні основи інформатики. Математична логіка.

Логіка висловлювань. Закони логіки висловлювань. Нормальні форми логіки висловлювань. Логіка першого ступеня.

Основним (первісним) поняттям математичної логіки є поняття простого висловлення. Під висловленням розуміють будь-яку оповідальну пропозицію, що стверджує щось про що-небудь, і при цьому ми можемо сказати, істинна вона чи хибна в даному місці і часу. Логічними значеннями висловлень є «істина» і «хибність». Вважається, що кожне висловлення або істинне, або хибне, і жодне висловлення не може бути одночасно істинним і хибним. Висловлення, що являє собою одне твердження, прийнято називати простим або елементарним. Висловлення, що виходять з елементарних за допомогою граматичних сполучників «не», «і», «або», «якщо ..., то ...», «тоді і тільки тоді», прийнято називати складними або складеними. Аналогічно складні висловлення можуть бути отримані з простих висловлень.

Рівносильності алгебри висловлень можна виразити законами логічних операцій.

Закони ідемпотентності

$$1. x \wedge x \equiv x$$

$$2. x \vee x \equiv x$$

Закони нуля та одиниці

$$3. x \wedge 1 \equiv x$$

$$4. x \vee 1 \equiv 1$$

$$5. x \wedge 0 \equiv 0$$

$$6. x \vee 0 \equiv x$$

Закон протиріччя

$$7. x \wedge \bar{x} \equiv 0$$

Закон виключення третього

$$8. x \vee \bar{x} \equiv 1$$

Закон зняття подвійного заперечення

$$9. \bar{\bar{x}} \equiv x$$

Закони поглинання

$$10. x \wedge (x \vee y) \equiv x$$

$$11. x \vee (x \wedge y) \equiv x$$

Закони розкриття імплікації та еквівалентності

$$12. x \leftrightarrow y \equiv (x \rightarrow y) \wedge (y \rightarrow x)$$

$$13. x \rightarrow y \equiv x \vee \bar{y}$$

Закони де Моргана

$$14. \overline{x \wedge y} \equiv \bar{x} \vee \bar{y}$$

$$15. \overline{x \vee y} \equiv \bar{x} \wedge \bar{y}$$

$$16. \overline{\bar{x} \wedge \bar{y}} \equiv x \vee y$$

$$17. \overline{\bar{x} \vee \bar{y}} \equiv x \wedge y$$

Заперечення. Запереченням висловлення x називається нове висловлення, що є істинним, якщо висловлення x хибне, і хибним, якщо висловлення x істинне. Заперечення висловлення x позначається як \bar{x} і читається «не x » або «не вірно, що x ». Заперечивши істинне висловлення, логіки вказують на те, що одержане в результаті заперечення висловлення є хибним.

Кон'юнкція (логічне множення). Кон'юнкцією двох висловлень x , y називається нове висловлення, що вважається істинним, якщо обоє висловлення x , y істинні, і помилковим, якщо хоча

б одне з них хибне (тобто, в інших випадках). У природній мові аналогом кон'юнкції є вирази « x і y », « x разом з y », «як x , так і y », «не тільки x , а й y ».

Кон'юнкція висловлень x , y позначається символом $x \wedge y$ (або $x \& y$, або ux), і читається як « x і y ». Висловлення x , y називаються членами кон'юнкції. Усі можливі логічні значення кон'юнкції двох висловлень x і y описуються таблицею істинності. З визначення операції кон'юнкції видно, що сполучник «і» в алгебрі логіки вживається в тому ж змісті, що й у повсякденній мові.

Диз'юнкція (логічне додавання) Диз'юнкцією двох висловлень x , y називається нове висловлення, що вважається істинним, якщо хоча б одне з висловлень x , y істинне, і хибним, якщо вони обоє хибні. Диз'юнкція висловлень x , y позначається символом $x \vee y$, читається як « x або y ». Висловлення x , y називаються членами диз'юнкції.

Розділова (сувора) диз'юнкція – складне висловлення, яке буде істинним тоді і тільки тоді, коли одне з простих висловлень істинне, а інше – обов'язково хибне (x або y , але не обидва; x , якщо не y).

Виключна (антикон'юнкція) диз'юнкція – складне висловлення, яке буде істинним тоді і тільки тоді, коли у крайньому разі одне з простих висловлень, що його складає, хибне. Виключна диз'юнкція має також назву операція „штрих Шеффера”.

Імплікація. Імплікацією двох висловлень x , y називається нове висловлення, що вважається помилковим, якщо x істинне, а y – хибне, і істинним у всіх інших випадках. Імплікація висловлень x, y позначається символом $x \rightarrow y$ (або $x \Rightarrow y$), читається як «якщо x , то y » або «з x випливає y ». Висловлення x називають умовою або посилкою, висловлення y – наслідком або висновком, висловлення $x \rightarrow y$ - імплікацією. Слід зазначити, що прості висловлення, які об'єднуються імплікацією, не можна переставляти місцями. Вживання слів “якщо..., то...” в алгебрі висловлень відрізняється від вживання їх у повсякденній мові, де ми, як правило, вважаємо, що, якщо висловлення x хибне, то висловлення “якщо x , то y ” узагалі не має змісту.

Еквіваленція (подвійна імплікація). Еквіваленцією двох висловлень x , y називається нове висловлення, що вважається істинним, коли обоє висловлення x , y або одночасно істинні, або одночасно хибні. І хибним у всіх інших випадках. Еквіваленція висловлень x , y позначається символом $x \leftrightarrow y$ (або \Leftrightarrow , рідше \sim), і читається як “для того, щоб x , необхідно і досить, щоб y ”, або “ x тоді і тільки тоді, коли y ”, “якщо x , то y , і навпаки”. Висловлення x , y називаються членами еквіваленції.

Теоретичні основи інформатики. Основи теорії множин.

Поняття множини. Поняття кортежу. Декартів добуток множин. Операції над множинами. Доведення рівностей з множинами. Комп'ютерне зображення множин.

Під множиною розуміють деяку сукупність різних поміж собою об'єктів, які розглядаються як єдине ціле.

Об'єкти, з яких складено множину, називають її елементами. Порядок елементів множини не має значення. Належність елемента до множини позначають символом. Якщо b не є елементом A , то пишуть: $b \notin A$.

Множина може мати скінчену кількість елементів або бути нескінченною.

Кортеж - упорядкований набір фіксованої довжини.

Множина впорядкованих пар елементів, з яких перший належить до A , а другий – до B , називається декартовим (прямим) добутком множин A та B .

Існує множина, яка не містить жодного елемента. Така множина називається порожньою і позначається символом \emptyset .

Множина як об'єкт може бути елементом іншої множини.

Універсальною множиною (універсумом) називається множина, що містить всі елементи з деякою заданою властивістю. Позначається така множина через U .

Множина A , всі елементи якої належать і до множини B , називається підмножиною (частиною) множини B . Таке співвідношення поміж множинами називається включення і позначається символом \subset , тобто $A \subset B$ (A включене до B).

Дві множини A та B називаються рівними (позначається $A=B$), якщо $A \subset B$ та $B \subset A$. Це є визначення рівності двох множин за допомогою операції включення.

Множина всіх кортежів довжини n на множинах A_1, \dots, A_n називають *декартовим добутком* множин A_1, \dots, A_n і позначають $A_1 \times \dots \times A_n$.

Зазвичай розглядають п'ять основних операцій над множинами: доповнення, об'єднання, переріз, різницю та симетричну різницю.

Елементи множини U , які не входять до A , утворюють *доповнену* множину до A (позначаються \bar{A}).

Об'єднанням двох множин – A та B (позначається $A \cup B$) – називається множина C , яка складається з усіх тих елементів, які належать хоча б до однієї з цих множин.

Перерізом двох множин – A та B (позначається $A \cap B$) – називається множина C , яка складається з усіх тих елементів, які належать множені A і множені B одночасно.

Різницею двох множин – A та B (позначається $A \setminus B$) – називається множина $C = A \setminus B = \{x \mid x \in A \text{ та } x \notin B, x \in U\}$.

Симетричною різницею двох множин – A та B (позначається $A \oplus B$, $A \Delta B$ або $A - B$) – називається множина $C = A \oplus B = (A \setminus B) \cup (B \setminus A)$.

Властивості операцій над множинами:

1 $A \cup B = B \cup A$, $A \cap B = B \cap A$ – комутативність;

2 $A \cup (B \cap C) = (A \cup B) \cap C$, $A \cap (B \cup C) = (A \cap B) \cup C$ – асоціативність;

3 $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$, $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ – дистрибутивність;

4 $A \cup \emptyset = A$, $A \cap \emptyset = \emptyset$

5 $A \cup A = A$, $A \cap A = A$ – ідемпотентність;

6 $A \cup \bar{A} = U$, $A \cap \bar{A} = \emptyset$ – доповнення;

7 $A \cup U = U$, $A \cap U = A$

8 $\emptyset = U$, $U = \emptyset$

9 $A \cup (A \cap B) = A$, $A \cap (A \cup B) = A$ – поглинання;

10 $A \cap B = A \cup \bar{B}$, $A \cup B = \bar{A} \cap \bar{B}$ – правило де Моргана;

11 $\bar{\bar{A}} = A$ – подвійного доповнення;

12 $A \setminus B = A \cap \bar{B}$ – вираз для різниці.

Теоретичні основи інформатики. Теорія графів.

Основні означення та властивості. Деякі спеціальні класи простих графів. Способи подання графів. Шляхи та цикли, зв'язність. Ізоморфізм графів. Ейлерів цикл у графі. Гамільтонів цикл у графі. Зважені графи та алгоритми пошуку найкоротшого шляху. Обхід графів. Планарні графи.

Скінченний неорієнтований граф задано, якщо задано такі два об'єкти:

1) скінченна не порожня множина $X = \{x_1, x_2, \dots, x_n\}$; елементи цієї множини називають вершинами графа;

2) деяка множина неупорядкованих пар елементів з X ; ця множина позначається U , її елементи називають ребрами.

Граф означається парою множин X та U , записують у вигляді $G = (X, U)$.

За наочного подавання графа вершини зображуються точками, ребра – лініями, які з'єднують точки.

Графи, в яких одну й ту саму пару вершин з'єднує кілька ребер називають мультиграфами. Існують графи, в яких певні ребра можуть мати збіжні кінці. Такі ребра називають петлями, а граф з петлею.

Поняття орієнтованого графа (орграфу) виникає, якщо ребрам графа надати напрямок (тобто орієнтацію) в такий спосіб, що один з кінців ребра буде початком, а інший – кінцем.

Орієнтований граф задано, якщо зазначено два об'єкти:

1) не порожня скінченна множина X – вершини графа;

2) множина U , утворена з упорядкованих пар вершин.

Елементи множини U називають дугами. Дуга орієнтованого графа зображується відрізком із зазначенням напрямку.

Граф може бути задано переліком його елементів, тобто за визначенням (елементи позначаються латинськими літерами з індексами або просто натуральними числами). Такий метод не є наочний, що утруднює виявлення характеристик графа.

Геометричне задання графа. Кожен граф може бути задано геометрично у тривимірному просторі, але не завжди його можна зобразити на площині так, щоб ребра перетинались тільки в вершинах. Граф, який може бути зображено на площині, називається планарним (рис. 3.12).

Матричне задання графа. Не завжди зручно задавати граф у тому вигляді, як це зазначено вище. Наприклад, при опрацюванні графа на комп'ютері його зручно зображати в матричній формі.

Скінченний орієнтований граф задається матрицями суміжності та інцидентності.

Матрицею суміжності орграфу G називається квадратна матриця $A(G) = [a_{ij}]$ порядку n , в якій:

$$a_{ij} = \begin{cases} 1, & \text{якщо } (x_i, x_j) \in U \\ 0, & \text{якщо } (x_i, x_j) \notin U \end{cases}$$

Матрицею інцидентності графа G називається матриця $B(G) = [b_{ij}]$, порядку $n \times m$, в якій:

$$b_{ij} = \begin{cases} 1, & \text{якщо вершина } x_i \text{ є інцидента до ребра } u_j \\ 0, & \text{якщо вершина } x_i \text{ не є інцидента до ребра } u_j \end{cases}$$

За допомогою матриць зручно задавати графи (орграфи) для опрацювання на комп'ютері. Однак слід зазначити, що за великої кількості вершин матриця суміжності стає громіздкою. Те саме можна сказати і про матрицю інцидентності, причому її розміри залежать, окрім того, й від кількості ребер (дуг) графа.

Графи $G_1 = (X_1, U_1)$ та $G_2 = (X_2, U_2)$ називаються ізоморфними, якщо поміж множинами їхніх вершин можна встановити взаємнооднозначну відповідність, за якої кожні дві вершини є суміжні в одному з графів тоді й лише тоді, коли відповідні їм вершини є суміжні в іншому графі.

Скінчена послідовність вершин та ребер графа в якій кожне ребро є ребро, яке з'єднує послідовні вершини називається маршрутом на графі. Довжина маршруту – це кількість ребер, які входять до маршруту. Маршрут, в якому всі ребра є різні, називають ланцюгом. Замкнений ланцюг називають циклом. Ланцюг називають простим, якщо всі його вершини є різні. Простий цикл – це цикл, у якому всі вершини, окрім першої та останньої, є різні.

Маршрут, який не містить повторних дуг, називається шляхом, а той, що не містить повторних вершин, – простим шляхом. Замкнений шлях називається контуром, а простий замкнений шлях – простим контуром.

Першою задачею теорії графів у працях видатного математика XVIII сторіччя Л. Ейлера була задача про кенігсберзькі мости. Місто Кенігсберг розташоване на річці Прегель, через яку прокладено сім мостів: чи можна обійти всі мости, проходячи кожним не більш одного разу? Мовою графів ця задача формулюється в такий спосіб: чи можна в мультиграфі, обійти всі ребра, проходячи кожним лише один раз?

Якщо існує цикл у скінченному графі, в якому кожне ребро графа брало участь один раз, то такий цикл називається ейлеровим циклом, а граф, який містить такий цикл, – ейлеровим графом.

Гамільтоновим циклом називається простий цикл, що проходить через усі вершини розглянутого графа. Гамільтонів цикл існує далеко не в кожному графі. Більш того, через кожні дві вершини розглянутого графа може проходити простий цикл, а гамільтонів цикл при цьому може бути відсутній:

Теоретичні основи інформатики. Древа та їх застосування.

Основні означення та властивості. Обхід дерев. Префіксна та постфіксна форми запису. Бінарне дерево пошуку. Древа прийняття рішень. Алгоритм бектрекінг.

Деревом називається зв'язний граф без циклів. Дерево не може мати ані кратних ребер, ані петель, ані ізольованих вершин. Кожний ланцюг у дереві є простий, через те що в іншому разі дерево містило б цикл, що є неможливе.

При додаванні до дерева ребра утворюється цикл, а при вилученні хоча б одного ребра дерево розпадається на компоненти, кожна з яких є або деревом, або ізольованою вершиною.

Древа називаються істотно різними, якщо вони не є ізоморфні.

Теорема: Еквівалентні є такі означення дерева:

- а) дерево – це зв'язний граф без циклів;
- б) дерево – це зв'язний граф, у якому кожне ребро є перешийком;
- в) дерево – це зв'язний граф, цикломатичне число якого дорівнює нулеві;
- г) дерево – це граф, у якому для кожних двох вершин існує лише один з'єднувальний ланцюг.

Довільний незв'язний граф, який не містить циклів, називається лісом. Граф, усі компоненти зв'язності якого є деревами, називається лісом.

Задача про найкоротшу з'єднувальну мережу

Нехай $G = (X, U, L)$ – навантажений граф, у якого задано вагу L_{ij} кожного ребра $\{i, j\}$. Треба побудувати кістякове дерево T графа G , сума ваги ребер якого є мінімальна.

Цій задачі можна дати таку інтерпретацію: n пунктів на місцевості треба зв'язати мережею доріг чи то ліній телефонного зв'язку мінімальної вартості. Така мережа буде деревом, і при цьому серед усіх кістякових дерев вона матиме мінімальну суму довжин ребер, які входять до неї. Алгоритм розв'язання поставленої задачі є надто простий. Дерево будується покроково; на кожному кроці долучається одне ребро. Правило для вибору цього ребра є таке: серед ще не обраних ребер береться найкоротше, яке не утворює циклу з вже обраними ребрами.

Древа рішень є одним з найбільш популярних методів вирішення завдань класифікації та прогнозування. Древа рішень дозволяють візуально і аналітично оцінити результати вибору різних рішень і використовується в галузі статистики та аналізу даних для прогнозних моделей.

Древа рішень використовують, коли потрібно прийняти рішення в умовах невизначеності, коли кожне рішення залежить від результату попередніх результатів або деяких заданих умов, що з'являються з певною ймовірністю.

Бектрекінг. Роботу цього алгоритму можна інтерпретувати як процес обходу деякого дерева. Кожна його вершина відповідає деякій послідовності (x_1, \dots, x_i) , причому вершини, які відповідають послідовностям вигляду (x_1, \dots, x_i, y) , – сини цієї вершини. Корінь дерева відповідає порожній послідовності.

Теоретичні основи інформатики. Відношення.

Відношення та їх властивості. Відношення еквівалентності. Відношення часткового порядку. Операції над відношеннями.

Множина називається впорядкованою, якщо кожному її елементові поставлено у відповідність число n ($n \in \mathbb{N}$, n – номер цього елемента) та елементи множини розміщено в порядку зростання їхніх номерів.

Відношення позначимо літерою R , тоді запис xRy вказує на те, що поміж x та y ($x \in X$, $y \in Y$) існує зв'язок.

Відношення повністю визначається парами (x, y) , для яких воно виконується, тому кожне бінарне відношення можна розглядати як множину впорядкованих пар (x, y) . При цьому порядок вибору елементів істотний. Перший елемент завжди вибирається з першої множини, другий – з другої.

Відношення, яке визначене на одному об'єкті називається унарним, якщо ж його визначено поміж парами об'єктів, – називаються бінарним, поміж трьома об'єктами – тернарним і т. д.

Відношенням R на множинах A та B називається довільна підмножина множини декартова добутку $A \times B$. Якщо $(a, b) \in R$, то це записується як: aRb .

Якщо $A=B$, то $R \subset A \times A$ і в цьому випадку стверджують, що бінарне відношення R задано на множині A .

Зображення відношення R ($R \subset A \times B$) точками в таблиці називають графіком відношення; множину x ($x \in A$), для яких існує таке y ($y \in B$), що $(x, y) \in R$, називають областю визначення відношення R , а множину y ($y \in B$), для яких існує таке x , що $(x, y) \in R$, – множиною значень.

1. Табличний спосіб завжди можна розглядати як різновид матричного, так як таблицю можна представити у вигляді матриці. Тому відношення R можна задати також бульовою матрицею суміжності, або відношення, рядки якої позначають елементами множини A , а стовпчики – елементами множини B

2. Спосіб задання відношень стрілками.

3. Завдання відношень перерізом

Нехай $c = (a, b)$ – кортеж довжини 2 (де $c \in A \times B$).

Елемент a називається проекцією елемента c на множину A .

Нехай E – підмножина декартова добутку множин A та B ($E \subset A \times B$). Множина елементів з A , які є проекцією елементів множини E на A , називається проекцією множини E на множину A .

Бінарне відношення R , яке визначено на множині A , називається відношенням порядку, якщо воно є антисиметричне й транзитивне. Бінарне відношення R на A називається відношенням нестрогого порядку, якщо воно є рефлексивне, антисиметричне й транзитивне.

Бінарне відношення R на A називається відношенням строгого порядку, якщо воно є антирефлексивне, антисиметричне та транзитивне.

Якщо відношення порядку є повне, то воно називається відношенням повного, або лінійного порядку, а якщо воно не має властивості повноти, то називається відношенням часткового порядку. Множина, на якій визначено відношення повного порядку, називається лінійно впорядкованою.

Бінарне відношення R на множині A називається відношенням еквівалентності, якщо воно є одночасно рефлексивне, симетричне й транзитивне

Алгоритми стиснення даних.

Загальна характеристика методів кодування та стиснення даних. Статистичне кодування Хофмана і Шенона-Фано. Арифметичне кодування. Метод «стосу книг». Архіватори. Стиснення графічних файлів.

Система двійкового кодування, що використовується в комп'ютерах, зручна з точки зору надійності інформаційних процесів. Однак двійкове кодування збільшує розміри файлів порівняно з іншими системами кодування. Тому виникає потреба у зменшенні розмірів файлів для ефективнішої реалізації інформаційних процесів.

Для зменшення розміру файлів використовують спеціальні способи стиснення даних, які називають алгоритмами (методами) стиснення даних. Розрізняють алгоритми стиснення, що забезпечують стиснення без втрати даних, і алгоритми, що передбачають часткову втрату даних.

Методи стиснення без втрати даних.

Ентропійне кодування. Ентропія в даному випадку означає мінімальну кількість біт, в середньому необхідне для представлення символу. Простий ЕК комбінує статистичну модель і сам кодувальник. Вхідний файл для побудови статистичної моделі, що складається з ймовірностей появи певних символів. Потім кодувальник, використовуючи модель, визначає, які бітові або байтові кодування призначати кожному символу, щоб самі часто зустрічаються були представлені самими короткими кодуваннями, і навпаки.

Кодування Лемпеля/Зіва. Даний метод базується на алгоритмі кодування послідовності об'єктів (цифр, літер тощо), в якій є фрагменти, що повторюються. Замінюється така послідовність на іншу в якій указуватимемо, яка цифра повторюється підряд і скільки разів. Тобто відбулося стиснення даних приблизно на 43%. Цей алгоритм дуже дієвий для стиснення графічних зображень з великими ділянками однакового кольору. Однак при відсутності повторювань він призведе до збільшення розміру файлу.

Кодування Хаффмана/Шенона. Цей метод часто застосовується при стисненні текстових даних. Він враховує частоту вживання в конкретній мові певних літер. Наприклад, в українській мові найбільш вживаними є літери і, а, о, е, а літери ш, щ, ф, х використовуються набагато рідше. У таких випадках використовують не 8 бітну систему кодування, а систему кодування змінної довжини, в якій найбільш вживані символи кодуються 1-4 бітами, а ті, що зустрічаються рідше, – 7-8 бітами.

Алгоритм Барроуза-Уїлера. Алгоритм, придуманий в 1994 році, оборотно трансформує блок даних так, щоб максимізувати повторення однакових символів. Сам він не стискає дані, але готує їх для більш ефективного стиснення через інший алгоритм стиснення.

Алгоритм Шеннона — Фано. Одна з найбільш ранніх технік (1949 рік). Створює двійкове дерево для представлення ймовірностей появи кожного з символів. Потім вони сортуються так, щоб самі часто зустрічаються перебували нагорі дерева, і навпаки.

Арифметичне кодування. Був розроблений в 1979 році в ІВМ. Дані, що кодуються будуть представлені в вигляді дробу, яка підбирається таким чином, щоб можна було представити текст найбільш компактно. Досягає дуже хорошою ступеня стиснення, зазвичай більшою, ніж у Хаффмана, однак він більш складний. Замість розбиття ймовірностей по дереву, алгоритм перетворює вхідні дані в одне раціональне число від 0 до 1

Метод «стосу книг» Алгоритм працює наступним чином. Визначається місце першого символу в алфавіті. Це число записується в вихідний блок. Потім поміщається тільки що використаний символ на перше місце, і повторюється процес з другим символом, потім з третім і так далі.

Архіватор — програмне забезпечення, що використовується для об'єднання і стиснення інформації.

Методи стиснення зображень. Варіант групового кодування. Ідея методу полягає в тому, що послідовність значень, що повторюються, заміняється парою чисел: одне з них вказує на довжину групи (число повторень даного значення), а інше - на власне це значення.

Інший методом, яким користуються досить часто, - JPEG. Основна ідея методу перебуває в поділі інформації в зображенні за рівнем важливості, і потім у відкиданні менше важливої її частини, зменшуючи тим самим загальний об'єм збережених даних. Це досягається перетворенням матриці кольірних значень у матрицю амплітуд, що відповідають визначеним частотам розкладання зображення.

Проблеми захисту інформації.

Можливі канали витоку інформації. Функції систем захисту інформації. Основні принципи проектування систем захисту інформації. Захист інформації.

Канал витоку інформації – сукупність джерела інформації, матеріального носія або середовища поширення, що несе зазначену інформацію сигналу й засобу виділення інформації із сигналу або носія. Одним з основних властивостей каналу є місце розташування засобу виділення інформації із сигналу або носія, що може розташовуватися в межах контрольованої зони, або поза нею (рис. 1).



Рис. 1. Класифікація каналів витоку інформації

1) Електромагнітний канал. Причиною його виникнення є електромагнітне поле, пов'язане із проходженням електричного струму в апаратних компонентах автоматизованої системи (АС). Електромагнітне поле може індукціювати струм в близько розташованих провідних лініях (наведення). Електромагнітний канал у свою чергу поділяються на такі канали:

- радіоканал (високочастотне випромінювання);
- мережний канал (наведення на мережу електроживлення);
- канал заземлення (наведення на проведення заземлення);
- лінійний канал (наведення на лінії зв'язку між комп'ютерними системами).

2) Акустичний (віброакустичний) канал. Пов'язаний з поширенням звукових хвиль у повітрі або пружних коливаннях в інших середовищах, що виникають при роботі пристроїв відображення інформації АС.

3) Візуальний канал. Пов'язаний з можливістю візуального спостереження зловмисником за роботою пристроїв відображення інформації АС без проникнення в приміщення, де розташовані компоненти системи. Як засіб одержання інформації в цьому випадку можуть розглядатися фотокамери, відеоканали й т.п.

4) Інформаційний канал. Пов'язаний з доступом до елементів АС, носіїв інформації, програмного забезпечення (у тому числі до операційних систем), а також з підключенням до ліній зв'язку. Інформаційний канал може бути розділений на канали:

- канал ліній зв'язку;
- канал виділених ліній зв'язку;
- канал локальної мережі;
- канал машинних носіїв інформації;
- канал термінальних і периферійних пристроїв.

Система захисту інформації – взаємопов'язана сукупність організаційних та інженерно-технічних заходів, засобів і методів захисту інформації, які повинні забезпечувати безпеку інформаційної системи від несанкціонованого доступу (НСД) до інформаційних ресурсів. Система захисту інформації є інструментом адміністраторів інформаційної безпеки, які виконують функції із забезпечення захисту інформаційної системи і контролю її захищеності.

Система захисту інформації повинна виконувати такі функції:

- реєстрація і облік користувачів, носіїв інформації, інформаційних масивів;
- забезпечення цілісності системного та прикладного програмного забезпечення та інформації яка оброблюється;
- захист комерційної таємниці, включаючи використання сертифікованих засобів криптографічного захисту;
- створення захищеного електронного документообігу з використанням сертифікованих засобів криптографічного перетворення і електронного цифрового підпису;
- централізоване управління системою захисту інформації;
- управління доступом;
- забезпечення ефективного антивірусного захисту, тощо.

Захист інформації від несанкціонованого доступу (НСД) є складовою частиною загальної проблеми забезпечення захисту інформації в інформаційно-комунікаційній системі та мережі (ІКСМ). В загальному випадку комплекс програмно-технічних засобів та організаційних рішень по захисту інформації в ІКСМ реалізується в рамках системи захисту інформації від НСД, яка умовно складається з таких чотирьох підсистем:

- 1) управління доступом до ІКСМ, до її послуг та ресурсів;
- 2) реєстрація і облік користувачів, послуг, інформаційних ресурсів;
- 3) криптографічного захисту;
- 4) забезпечення цілісності інформаційних потоків, інформаційних ресурсів та програмного забезпечення.

Закриття каналів несанкціонованого отримання інформації повинно починатися з контролю доступу користувачів до ресурсів ІКСМ. Ця задача вирішується на основі ряду принципів, на яких базується система захисту інформації:

Принцип виправданості доступу – користувач повинен мати достатню «форму допуску» для отримання інформації того рівня конфіденційності, що він вимагає, і ця інформація дійсно необхідна йому для виконання його виробничих функцій.

Принцип достатньої глибини контролю доступу. Засоби захисту інформації повинні включати механізми контролю доступу до всіх видів інформаційних і програмних ресурсів ІКСМ, які у відповідності з принципом виправданості доступу слід розмежовувати між користувачами.

Принцип розмежування інформаційних потоків. Для попередження порушення інформаційної безпеки, яке, наприклад, може мати місце при запису секретної інформації на несекретні носії і в несекретні файли, її передачі програмам і процесам, які не призначені для обробки секретної інформації, а також при передачі секретної інформації по незахищених каналах зв'язку, необхідно здійснювати відповідне розмежування інформаційних потоків.

Принцип персональної відповідальності. Кожний користувач повинен нести персональну відповідальність за свою діяльність в системі, включаючи будь-які операції з конфіденційною інформацією і можливі порушення її захисту.

Принцип цілісності засобів захисту. Даний принцип передбачає, що засоби захисту інформації в ІКСМ повинні чітко виконувати свої функції у відповідності з переліченими принципами і бути ізольованими від користувачів, а для свого супроводу повинні включати спеціальний захищений інтерфейс для засобів контролю, сигналізації про спроби порушення захисту інформації і впливу на процеси в системі.

Захист інформації – сукупність методів і засобів, що забезпечують цілісність, конфіденційність і доступність інформації за умов впливу на неї загроз природного або штучного характеру, реалізація яких може призвести до завдання шкоди власникам і користувачам інформації.

Захист інформації ведеться для підтримки таких властивостей інформації як:

- 1) Цілісність – неможливість модифікації інформації неавторизованим користувачем.
- 2) Конфіденційність – інформація не може бути отримана неавторизованим користувачем.
- 3) Доступність – полягає в тому, що авторизований користувач може використовувати інформацію відповідно до правил, встановлених політикою безпеки не очікуючи довше заданого (прийнятного) інтервалу часу.

Відповідно до властивостей інформації, виділяють такі загрози її безпеці:

- загрози цілісності:
 - знищення;
 - модифікація;
- загрози доступності:
 - блокування;
 - знищення;
- загрози конфіденційності:
 - несанкціонований доступ (НСД);
 - витік;
 - розголошення.

Аспекти захисту інформації

Конфіденційність – захист від несанкціонованого ознайомлення з інформацією.

Цілісність – захист інформації від несанкціонованої модифікації.

Доступність – захист (забезпечення) доступу до інформації, а також можливості її використання. Доступність забезпечується як підтриманням систем в робочому стані так і завдяки способам, які дозволяють швидко відновити втрачену чи пошкоджену інформацію.

Види захисту інформації

Кожен вид захисту інформації забезпечує окремі аспекти інформаційної безпеки:

Технічний – забезпечує обмеження доступу до носія повідомлення апаратно-технічними засобами (антивіруси, фаєрволи, маршрутизатори, токени, смарт-карти тощо):

- попередження витоку по технічним каналам;
- попередження блокування ;

Інженерний – попереджує руйнування носія внаслідок навмисних дій або природного впливу інженерно-технічними засобами (сюди відносять обмежуючі конструкції, охоронно-пожежна сигналізація).

Криптографічний – попереджує доступ за допомогою математичних перетворень повідомлення (Ш):

- попередження несанкціонованої модифікації;
- попередження НС розголошення.

Організаційний – попередження доступу на об'єкт інформаційної діяльності сторонніх осіб за допомогою організаційних заходів (правила розмежування доступу).

Захист від несанкціонованого доступу.

Ідентифікація об'єктів і механізми підтвердження справжності. Паролі. Надання повноважень і механізми контролю доступу до інформації. Принцип мінімальних привілеїв. Розділення користувачів.

Ідентифікація об'єкта – одна з функцій підсистеми захисту. Ця функція виконується в першу чергу, коли об'єкт робить спробу увійти в мережу. Якщо процедура ідентифікації завершується успішно, даний об'єкт вважається законним для даної мережі.

Наступний крок – аутентифікація об'єкта (перевірка справжності об'єкта). Ця процедура встановлює, чи є даний об'єкт саме тим, ким він себе проголошує.

Після того як об'єкт ідентифікований і підтверджена його справжність, можна встановити сферу його дії і доступні йому ресурси. Таку процедуру називають наданням повноважень (авторизацією).

Коли користувач починає роботу комп'ютерній системі, використовуючи термінал, система запитує його ім'я та ідентифікаційний номер. Залежно від відповідей користувача комп'ютерна система проводить його ідентифікацію. Потім система перевіряє, чи є користувач дійсно тим, за кого він себе видає. Для цього вона запитує у користувача пароль. Пароль – це лише один із способів підтвердження автентичності користувача.

Перерахуємо можливі способи підтвердження автентичності.

1. Визначена інформація, що знаходиться в розпорядженні користувача: пароль, персональний ідентифікаційний номер, угода про використання спеціальних закодованих фраз.

2. Елементи апаратного забезпечення, що знаходяться в розпорядженні користувача: ключі, магнітні картки, мікросхеми тощо.

3. Характерні індивідуальні особливості користувача: відбитки пальців, малюнок сітківки ока, тембр голосу і т.п.

4. Характерні прийоми і риси поведінки користувача в режимі реального часу: особливості динаміки і стиль роботи на клавіатурі, прийоми роботи з маніпулятором і т.п.

5. Навички та знання користувача, обумовлені освітою, культурою, навчанням, вихованням, звичками і т.п.

Традиційно кожен законний користувач комп'ютерної системи отримує ідентифікаційний номер та/або пароль. На початку сеансу роботи на терміналі користувач вказує свій ідентифікаційний номер (код користувача) системі, яка потім запитує у користувача пароль.

Найпростіший метод підтвердження автентичності з використанням пароля заснований на порівнянні поданого користувачем пароля з вихідним значенням, що зберігаються на робочій станції. Оскільки пароль повинен зберігатися в таємниці, його слід шифрувати перед пересиланням по незахищеному каналу. Якщо значення користувача та системи збігаються, то пароль вважається справжнім, а користувач – законним.

Якщо хто-небудь, хто не має повноважень для входу в систему, дізнається будь-яким чином пароль і ідентифікаційний номер законного користувача, він отримує доступ до системи.

Іноді одержувач не повинен розкривати вихідну відкриту форму пароля. У цьому випадку відправник повинен пересилати замість відкритої форми пароля відображення пароля, що отримується з використанням односторонньої функції $\alpha(\cdot)$ пароля. Це перетворення має гарантувати неможливість розкриття пароля по його відображенню, так як той, хто хоче дізнатися пароль нашої задачі, на нерозв'язну числову задачу.

Наприклад, функція $\alpha(\cdot)$ може бути визначена наступним чином:

$$\alpha(P) = E_P(ID),$$

де P – пароль відправника;

ID – ідентифікатор відправника;

E_P – процедура шифрування, виконувана з використанням пароля P в якості ключа.

Такі функції особливо зручні, якщо довжина пароля і довжина ключа однакові. В цьому випадку підтвердження автентичності за допомогою пароля складається з пересилання одержувачеві відображення $\alpha(P)$ і порівняння його з попередньо обчисленим і збереженим еквівалентом $\alpha'(P)$.

На практиці паролі складаються тільки з кількох літер, щоб дати можливість користувачам запам'ятати їх. Короткі паролі уразливі до атаки повного перебору всіх варіантів. Для того щоб запобігти такій атаці, функцію $\alpha(P)$ визначають інакше, а саме:

$$t(P) = E_{P \oplus K}(ID),$$

де K і ID – відповідно ключ і код відправника.

Очевидно, значення $\alpha(P)$ обчислюється заздалегідь і зберігається у вигляді $\alpha'(P)$ в ідентифікаційній таблиці у одержувача. Підтвердження достовірності складається з порівняння двох відображень пароля $\alpha(P_A)$ і $\alpha'(P_A)$ і визнання пароля P_A , якщо ці відображення рівні. Звичайно, будь-хто, хто отримує доступ до ідентифікаційної таблиці, може незаконно змінити її вміст.

Контроль доступу – функція відкритої системи, що забезпечує технологію безпеки, яка дозволяє або забороняє доступ до певних типів даних, засновану на ідентифікації суб'єкта, якому потрібен доступ, і об'єкта даних, що є метою доступу.

Контроль доступу є однією з найважливіших елементів захисту вашого ПК і Інформації на ньому. Доступ до захищеної інформації повинен бути обмежений, щоб тільки люди, які мають право доступу (повноваження або привілеї), могли отримувати цю інформацію. Комп'ютерні програми і в багатьох випадках чужорідні комп'ютери за допомогою локальної мережі, Інтернету, бездротової мережі можуть отримати секретну інформацію, яка не призначена їм. Це може завдати як фінансові, так і інформаційні втрати. У зв'язку з цим необхідний механізм контролю доступу до захищеної інформації. Складність механізмів контролю доступу повинна бути в паритеті з цінністю інформації, тобто чим більш важливою або цінною інформація є, тим складнішими повинні бути механізми контролю доступу. Основними механізмами контролю доступу є ідентифікація і аутентифікація.

Ідентифікація – присвоєння суб'єктам і об'єктам ідентифікатора і (або) порівняння ідентифікатора з переліком привласнених ідентифікаторів. Якщо людина під час входу в обліковий запис операційної системи комп'ютера вводить ім'я користувача, він заявляє про те, хто він. Проте, його заява може і не бути правдою. Щоб визначити людину як дійсного користувача, починається Аутентифікація.

Аутентифікація є актом перевірки заяви особистості. Коли, наприклад, людина хоче увійти в систему і отримати якусь інформацію або зробити в ній певні зміни, вона вводить ім'я користувача, а потім система запитує «пароль», щоб перевірити справжність. Система звіряє зв'язку «імя-пароль» і переконавшись в достовірності заяви – виконує вхід до системи.

Існує три різних типи інформації, яка може бути використана для перевірки автентичності: те, що знаєте тільки ви, або те, що є тільки у вас. Прикладом того, що ви знаєте можуть бути такі речі, як ПІН-код, пароль, або дівоче прізвище вашої матері. Прикладами того, що у вас є, можуть бути водійські права або магнітні картки. Також це можуть бути біометричні прилади. Прикладами біометрії може бути відбиток пальця, голос і сітківка ока. Суворо аутентифікація вимагає надання інформації від двох з трьох різних типів аутентифікації інформації. Наприклад, те, що ви знаєте, а також хто ви. Це називається двофакторна аутентифікація. У комп'ютерних системах, які використовуються сьогодні, Ім'я користувача і пароль є найбільш поширеною формою аутентифікації. Імена користувачів і паролі досягли своєї мети, але в нашому сучасному світі вони

не дають нам повної впевненості. Імена користувачів і паролі поступово замінюються складнішими механізмами аутентифікації.

Після успішної ідентифікації і аутентифікації, людина або програма отримує в своє розпорядження саме ті ресурси до яких програма або людина має право доступу, а також які дії будуть допущені до виконання (запуск, перегляд, створення, видалення або зміна). Це називається дозволом.

Принцип найменших привілеїв, також відомий як принцип мінімальних привілеїв або просто мінімальні привілеї в інформаційній безпеці, інформатики та інших областях – принцип організації доступу до ресурсів, коли в той чи інший рівень абстракції від обчислювального середовища, кожен модуль (такий, як процес, користувач або програма, які ми розглядаємо) повинні мати доступ до такої інформації і ресурсів, які мінімально необхідні для успішного виконання його робочої мети.

Це означає давати користувачеві тільки ті привілеї, які є абсолютно необхідними для того, щоб зробити свою роботу. Наприклад, користувачеві який займається резервним копіюванням не потрібно встановлювати будь-яке інше програмне забезпечення, крім того, що йому необхідно для роботи. Будь-які інші привілеї, наприклад як установка програмного забезпечення і так далі повинні бути заблоковані. Принцип поширюється також на користувачів персонального комп'ютера вдома, де вони повинні працювати в обліковим записом користувача і відкривати свій обліковий запис адміністратора (захищену паролем) тільки тоді, коли ситуація вимагає цього.

Стосовно до користувачів, принцип найменшого допуску, означає надання користувачам тільки мінімально необхідних можливостей при роботі з операційною системою і додатками.

Адміністрування користувачів полягає у створенні облікової інформації користувачів (що визначає ім'я користувача, приналежність користувача до різних груп користувачів, пароль користувача), а також у визначенні прав доступу користувача до ресурсів мережі - комп'ютерів, каталогів, файлів, принтерів і т.п.

Створення облікової інформації користувачів здійснюється в мережі Windows NT утилітою User Manager для локального комп'ютера і User Manager for Domains для всіх комп'ютерів домену. Права доступу до ресурсів задаються в мережі Windows NT різними засобами, залежно від типу ресурсу. Можливість використання комп'ютерів Windows NT Workstation в якості робочих станцій - за допомогою User Manager for Domains, доступ до локальних каталогів і файлів (тільки для файлової системи NTFS, що підтримує права доступу) - за допомогою засобів Windows NT Explorer, до віддалених розділяються каталогами - за допомогою Server Manager, доступ до принтерів - з панелі Printers.

Можуть бути визначені наступні типи користувачів і груп користувачів:

- локальний інтерактивний користувач комп'ютера (користувач, який заведено в локальній обліковій базі даних комп'ютера, і який працює з ресурсами комп'ютера інтерактивно);
- локальний мережний користувач комп'ютера (користувач, який заведено в локальній обліковій базі даних комп'ютера, і який працює з ресурсами комп'ютера через мережу);
- користувач домену (користувач, який заведено в глобальній обліковій базі даних домену на PDC);
- локальна група комп'ютера (може створюватися на всіх комп'ютерах домену, крім PDC і BDC, в яких вона вироджується в локальну групу домену);
- локальна група домену - складається з користувачів домену (заводиться тільки на PDC);
- глобальна група домену - складається з користувачів домену (може входити в локальну групу домену).

Для кожного типу груп є деякий набір вбудованих груп: Administrators, Server Operators, Users, Everyone, DomainUsers і ін.

Для однозначної ідентифікації глобальної групи в багатодоменному мережі, використовується складене її ім'я, наприклад Marketing \ Managers, де Marketing - ім'я домену, Managers - ім'я глобальної групи.

Типи операцій доступу

Операції доступу – це дії об'єктів над суб'єктами. Операції можуть бути або дозволені, або заборонені, або взагалі не мати сенсу для даної пари об'єкта і суб'єкта.

Всі операції поділяються на підоперації, що мають особливі назви:

- дозволу (permissions) - це безліч операцій, які можуть бути визначені для суб'єктів усіх типів по відношенню до об'єктів типу файл, каталог або принтер;

- права (user rights) - визначаються для об'єктів типу група на виконання деяких системних операцій: створення резервних копій, вимикання комп'ютера (shutdown) і т.п. Права призначаються за допомогою User Manager for Domains;

- можливості користувачів (user abilities) - визначаються для окремих користувачів на виконання дій, пов'язаних з формуванням їх операційного середовища, наприклад, зміна складу програмних груп, які показуються на екрані дисплея, включення нових ікон в Desktop, можливість використання команди Run і т.п.

Права і дозвіл, що дані групі автоматично надаються її членам, дозволяючи адміністратору розглядати велику кількість користувачів як одиницю облікової інформації.

Архітектура комп'ютерів. Визначення, класифікація и етапи розвитку.

Поняття архітектури комп'ютерів. Архітектура апаратних засобів. Архітектура неймановського комп'ютера. Архітектура постнеймановських комп'ютерів. Системний інтерфейс і архітектура системної плати. Архітектура системної плати. Принципи проектування сучасних комп'ютерів.

Архітектура комп'ютера — це набір дисциплін, які описують функціональність, організацію та реалізацію комп'ютерних систем. Деякі визначення архітектури є описами можливостей та програмної моделі комп'ютера, але не конкретної реалізації. Інші описи комп'ютерної архітектури містять опис архітектури системи команд, логічної побудови та реалізації.

Серед таких комп'ютерних архітектур, найбільшого поширення отримали 2 типи архітектури: *прінстонська* (фон Неймана) і *гарвардська*. Обидві вони виділяють 2 основних вузли ЕОМ: центральний процесор і пам'ять комп'ютера. Різниця полягає в структурі пам'яті: в прінстонській архітектурі програми і дані зберігаються в одному масиві пам'яті і передаються в процесор одним каналом, тоді як гарвардська архітектура передбачає окремі сховища і потоки передачі для команд і даних.

У докладніший опис, що визначає конкретну архітектуру, також входять: структурна схема ЕОМ, засоби і способи доступу до елементів цієї структурної схеми, організація і розрядність інтерфейсів ЕОМ, набір і доступність регістрів, організація пам'яті та способи її адресації, набір і формат машинних команд процесора, способи представлення і формати даних, правила обробки переривань.

За перерахованими ознаками та їх поєднаннями серед архітектур виділяють:

За розрядністю інтерфейсів і машинних слів: 8 -, 16 -, 32 -, 64-розрядні (ряд ЕОМ має й інші розрядності);

За особливостями набору регістрів, формату команд і даних: CISC, RISC, VLIW;

За кількістю центральних процесорів: однопроцесорні, багатопроцесорні, суперскалярні;

багатопроцесорні за принципом взаємодії з пам'яттю: симетричні багатопроцесорні (SMP), масивно-паралельні (MPP), розподілені.

Найбільш узагальнений спосіб класифікації архітектур апаратних засобів комп'ютера базується на поняттях потоку команд L і потоку даних D в обчислювальній структурі. При цьому розрізняють одинарний потік S і множинний потік M. Відповідно до цього підходу можна визначити чотири класи структур апаратних засобів ЕОМ.

SISD - архітектура з одинарним потоком команд і одинарним потоком даних. Управління здійснює одинарна послідовність команд, кожна з яких забезпечує виконання однієї операції зі своїми даними і далі передає управління наступній команді. У комп'ютерах цього типу команди виконуються тільки послідовно в часі на одному процесорному елементі.

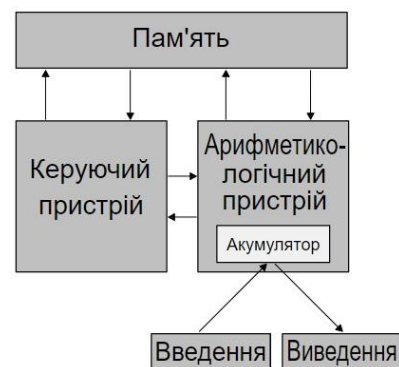
MISD - архітектура з множинним потоком команд і одинарним потоком даних, яка отримала також назву конвеєра обробки даних. Вона становить ланцюжок послідовно з'єднаних процесорів (мікропроцесорів), які управляються паралельним потоком команд. На вхід конвеєра з пам'яті подається одинарний потік даних, які проходять послідовно через всі процесори, кожен з яких робить обробку даних під керуванням свого потоку команд і передає результати наступного ланцюжком процесору, який використовує їх як вхідні дані. Конвеєрну архітектуру запропонував академік С.А. Лебедев в 1956 році.

SIMD - архітектура ЕОМ з одинарним потоком команд і множинним потоком даних. Процесор таких машин має матричну структуру, в вузлах якої включене велика кількість порівняно простих швидкодіючих процесорних елементів, які можуть мати власну або загальну пам'ять даних.

Одинарний потік команд виробляє одне загальне пристрій управління. При цьому всі процесорні елементи виконують одночасно одну і ту ж команду, але над різними операндами, які доставляються з пам'яті множинним потоком.

MIMD - архітектура з множинними потоками команд і даних. До таких структур належать багатопроесорні і багатомашинні обчислювальні системи. Вони можуть відрізнятися принципом управління (централізоване або розподілене), організацією пам'яті (загальною, розподілена або комбінована) і структурою зв'язків між комп'ютерами або процесорами. Гнучкість MIMD структур дозволяє організувати спільну роботу комп'ютерів, які входять в них, або процесорів за розпаралеленого програмою при рішенні однієї складної задачі, або роздільну роботу всіх комп'ютерів при одночасному вирішенні безлічі завдань за допомогою незалежних програм. Нижче наводиться більш докладний опис деяких найбільше поширених структур комп'ютерів.

Архітектура фон Неймана— архітектура електронних обчислювальних машин, основною відмінністю якої від інших подібних архітектур є спільне зберігання даних та машинних команд в комірках однієї й тієї ж пам'яті, що унеможлиблює їх розрізнення за способом представлення або кодування. Названа так на честь відомого математика та теоретика обчислювальної техніки Джона фон Неймана (John von Neumann), та по сьогодні залишається домінуючою схемою організації ЕОМ загального призначення.



Обчислювальна машина є машиною з архітектурою фон Неймана, якщо:

Програма та дані зберігаються в одній загальній пам'яті. Це дає можливість виконувати над командами ті ж дії, що і над даними.

Кожна комірка пам'яті машини ідентифікується унікальним номером, який називається адресою.

Різні слова інформації (команди та дані) розрізняються за способом використання, але не за способом кодування та структурою представлення в пам'яті.

Кожна програма виконується послідовно, починаючи з першої команди, якщо немає спеціальних вказівок. Для зміни цієї послідовності використовуються команди передачі управління.

Обчислювальна машина є машиною з архітектурою фон Неймана, якщо:

- Програма та дані зберігаються в одній загальній пам'яті. Це дає можливість виконувати над командами ті ж дії, що і над даними.

- Кожна комірка пам'яті машини ідентифікується унікальним номером, який називається адресою.

- Різні слова інформації (команди та дані) розрізняються за способом використання, але не за способом кодування та структурою представлення в пам'яті.

- Кожна програма виконується послідовно, починаючи з першої команди, якщо немає спеціальних вказівок. Для зміни цієї послідовності використовуються команди передачі управління.

Машина фон Неймана, як і практично кожна сучасна ЕОМ загального призначення, складається з чотирьох основних компонентів:

1. Операційний пристрій (ОП), який виконує команди з визначеного набору, який називається системою (набором) команд, над порціями інформації, яка зберігається відокремленій від операційного пристрою пам'яті (хоча сучасні архітектури мають в складі операційного пристрою додаткову пам'ять (зазвичай банк регістрів), в якій операнди зберігаються порівняно короткий час безпосередньо в процесі проведення обчислень.

2. Пристрій управління (ПУ), який організує послідовне виконання алгоритмів, розшифрування команд, які поступають із пристрою пам'яті (див. нижче), реагує на аварійні

ситуації та виконує загальні функції управління всіма вузлами обчислювальної машини. Зазвичай ОП та ПУ об'єднуються в структуру, яка називається центральним процесором. Слід звернути увагу, що вимога саме послідовного, в порядку надходження з пам'яті (в порядку зміни адрес в лічильнику команд) виконання команд є принциповою. Архітектури, які не додержуються такого принципу, взагалі не вважаються фон-нейманівськими.

3. Пристрій пам'яті (ПП) — масив комірок з унікальними ідентифікаторами (адресами), в яких зберігаються команди та дані.

4. Пристрій вводу-виводу (ПВВ), який забезпечує зв'язок ЕОМ з зовнішнім світом, різними пристроями, які передають інформацію на переробку в ЕОМ та приймають результати.

Після завантаження програми (алгоритму й даних для обробки) в запам'ятовуючий пристрій, машина фон Неймана може працювати автоматично, без втручання оператора. Кожна комірка пам'яті машини має унікальний номер — адресу, спеціальний механізм, найчастіше — лічильник команд — забезпечує автоматичне виконання необхідної послідовності команд, і визначає на кожному етапі адресу комірки, з якої необхідно завантажити наступну команду.

Перед початком виконання програми в лічильник записується адреса її першої команди. Визначення адреси наступної команди відбувається за одним з наступних сценаріїв:

1. Якщо поточна команда не є командою передачі управління (тобто це просто арифметична або логічна операція над даними), то до поточного значення лічильника додається число, яке дорівнює довжині поточної команди в мінімально адресованих одиницях інформації (зрозуміло, що це можливо за умови, якщо звичайні команди в блоках, не розділених командами передачі управління, розташовуються послідовно в пам'яті, інакше адреса наступної команди може зберігатись, наприклад, безпосередньо в команді).

2. Якщо поточна команда — команда передачі управління (команда умовного або безумовного переходу), яка змінює послідовний хід виконання програми, то в лічильник примусово записується адреса тої команди, яка була замовлена при виконанні переходу, де б вона не знаходилась.

До цього часу більшість універсальних комп'ютерів будують за принципами архітектури Джона фон Неймана. Але недоліки цієї архітектури, пов'язані з закладеним в ній послідовним характером організації обчислень, ставлять перепони в пошуку шляхів побудови швидких комп'ютерних систем. Тому крім архітектури Джона фон Неймана за час існування комп'ютерної техніки було створено цілий ряд інших архітектур. До них, зокрема, належать гарвардська архітектура, асоціативна машина, машина потоків даних, редуційна машина. Було запропоновано покласти в якості обчислювальної парадигми нейронні мережі, в яких використана ідея з моделей мозку та генетичні алгоритми, в яких застосовані ідеї з біології та еволюції архітектури комп'ютерних мереж. Останнім часом багато уваги приділяється квантовим комп'ютерам, елементи яких працюють за законами квантової механіки, біологічним комп'ютерам, а також паралельним комп'ютерам, оскільки практично всі сьогоденні комп'ютери є паралельними.

Гарвардська архітектура вперше була реалізована Ховардом Айкеном в комп'ютері Марк-1 в Гарварді. Вона передбачає розділення пам'яті на пам'ять даних і пам'ять команд. Тим самим розділяються шини передачі керуючої і оброблюваної інформації (рис. 2). При цьому підвищується продуктивність комп'ютера за рахунок суміщення в часі пересилання та обробки даних і команд.

Необхідність використання гарвардської архітектури можна пояснити так. Ядро комп'ютера Джона фон Неймана складається з процесора та основної пам'яті. Бажано, аби обидві компоненти ядра не пригальмовували одна одну, тобто працювали із рівною швидкістю. На практиці вузол



Рис. 2

пам'яті є значно (на порядок) повільнішим від процесора і цей розрив у швидкодії з прогресом інтегральних технологій лише зростає. Зменшити розрив можна структурними методами, збільшуючи розрядність інформаційного слова пам'яті. Саме цей підхід реалізує гарвардська архітектура з двома запам'ятовувальними пристроями. Зрозуміло, що тут паралельно виконуються операції вибирання команд програми, з одного боку, а з другого - вибирання та запис кодів даних і результатів обчислень.

Дуальна пристонсько-гарвардська архітектура. Швидкі комп'ютери гарвардської архітектури є складнішими щодо програмування порівняно з комп'ютерами принстонської архітектури. Зрозуміло, що бажано створити комп'ютер з дуальною архітектурою, яка водночас запозичує нову якість - швидкодію від гарвардської архітектури та стандартну парадигму розробки програм від принстонської архітектури. Злиття двох архітектур виконують на рівні кеш пам'яті шляхом її поділу на кеш даних та кеш команд (рис. 3). Злиттям архітектур програмісту надано зручність програмних технологій принстонської архітектури, а з боку процесора реалізовано гарвардську архітектуру, в результаті чого він значно менше пригальмовується з боку основної пам'яті.

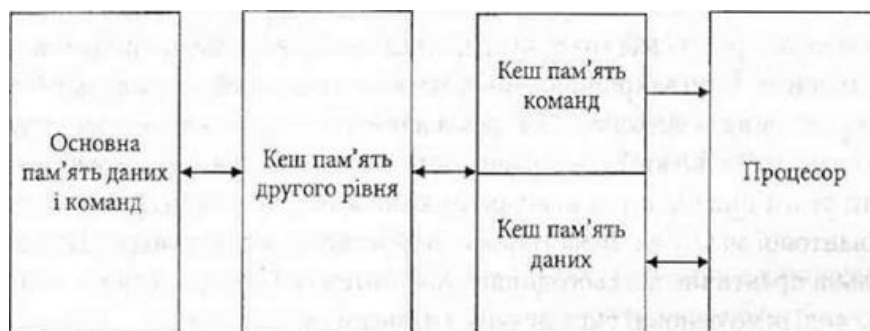


Рис. 3

Асоціативна машина

передбачає маніпуляції з даними не відповідно до їх адрес, як це є в машині Джона фон Неймана, а відповідно до значення цих даних або їх частин. Базовими тут є операції пошуку і порівняння. Основою асоціативної архітектури є асоціативна пам'ять, яка забезпечує одночасний доступ до багатьох даних, в яких співпадають значення відповідних розрядів. Тим самим за рахунок високої паралельності обробки досягається висока продуктивність на класі операцій, для виконання яких ця машина є ефективною (зокрема, логічні операції, операції пошуку та сортування). Асоціативна машина є складовою практично кожного сучасного комп'ютера.

Машина потоків даних. Керування обчислювальним процесом в машині потоків даних здійснюється даними за їх готовністю до обробки. Кожне дане в такій машині має спеціальні ознаки. За цими ознаками пристрій керування знаходить дані, які готові до обробки, і передає їх в АЛП для виконання відповідних операцій. Тим самим тут за рахунок можливого паралельного аналізу та обробки даних досягається гранично висока продуктивність. Принципи машини потоків даних використовуються в багатьох сучасних високопродуктивних комп'ютерах.

Паралельні комп'ютерні системи. Перші паралельні комп'ютерні системи, до складу яких входило лише два процесори, були побудовані в кінці 60-х років минулого століття. В 70-х роках такі системи мали в своєму складі до 64-х процесорів, в 80-х роках - до 1000, а в кінці 90-х років фірма ІВМ анонсувала конструкцію суперкомп'ютера з паралельною архітектурою, який включав понад мільйон процесорів і на даний час є найпродуктивнішим у світі. Паралельна обробка інформації є ключовим напрямком побудови високопродуктивних комп'ютерних систем. Однак і паралельні комп'ютерні системи мають обмеження. По-перше, зі збільшенням кількості процесорів ускладнюється задача розподілу завдань між процесорами. Для її вирішення використовуються додаткові процесори, кількість яких може значно перевищувати кількість процесорів, зайнятих безпосередньо виконанням алгоритму. По-друге, послідовна природа багатьох алгоритмів обмежує прискорення, якого можна досягти, використовуючи багатопроцесорну організацію.

Основою системної плати є чіпсет який визначає її архітектуру. На даний час у переважній більшості ПК використовуються два види архітектур системних плат: шинно-мостова та хабова.

Після появи шини PCI, для неї довелося будувати міст від системної шини. Спочатку її називали "прибудованою" (mezzanine bus), але незабаром вона надовго стала центральною шиною, навколо якої компонувалися всі інші елементи. Її центральне місце не оспорювалось, оскільки PCI мала високу продуктивність — 132 Мбайт/с. Традиційно на схемах шини PCI зображують посередині, як екватор. Процесор і пам'ять (разом з кеш-пам'яттю) зображують вище — "північніше", а шини ISA і всі пристрої, що підключаються до PCI і ISA, зображують нижче — "південніше" "екватора". Відповідні частини чіпсета одержали назви північного (north) і південного (south) моста. Співзвучне слово серверний відноситься до чіпсета, який орієнтований на застосування в комп'ютерах-серверах.

Шинно-мостова архітектура чіпсетів проіснувала довгий час і пережила кілька поколінь процесорів (від 2-го до 6-го). Відхід вторинного кеша із системної плати на процесор (P6 і Pentium 4 у Intel і K7 у AMD) трохи спростив північну частину чіпсета — тепер не треба керувати статичною кеш-пам'яттю, а залишається лише забезпечувати когерентність процесорного кеша з основною пам'яттю, доступ до якої можливий і з боку шини PCI.

Коли з'явився порт AGP, завдання північного моста ускладнилися: контролеру пам'яті доводиться працювати вже на три фронти — йому посилають запити процесор, майстри шини PCI (і ISA, але теж через PCI) і порт AGP. Пропускна здатність AGP у режимі 2x складає 533, а в режимі 4x — 1066 Мбайт/с, так що шина PCI за продуктивністю стала уже другорядною (нагадаємо, що AGP — не шина розширення, а спеціалізований порт підключення графічного акселератора). З уведенням високошвидкісних режимів UltraDMA (ATA/66 і ATA/100) зв'язок двохканального контролера IDE з пам'яттю через шини PCI став уже занадто сильно завантажувати цю шину. Передбачені можливості підвищення продуктивності PCI використовуються рідко: розширення розрядності до 64 біт обходиться занадто дорого (велике число провідників породжує свої проблеми), а підвищення частоти до 66 МГц для шини можливо, лише якщо всі її абоненти підтримують цю частоту. Досить установити одну "просту" карту PCI, і продуктивність центральної шини падає до початкових 133 Мбайт/с.

Відповіддю на ці зміни в розміщенні сил став перехід на *хабову архітектуру* чіпсета. У даному контексті хаби — це спеціалізовані мікросхеми, що забезпечують передачу даних між підключеними до них шинами.

Північний хаб чіпсета виконує ті ж функції, що і північний міст вищеописаної архітектури: він зв'язує шини процесора, пам'яті і порту AGP. Однак на "південній" стороні цього хаба знаходиться вже не шина PCI, а високопродуктивний інтерфейс зв'язку з південним хабом. Пропускна здатність цього інтерфейсу складає 266 Мбайт/с і вище, у залежності від чіпсета. Якщо чіпсет має інтегровану графіку, то в північний хаб входить і графічний контролер із усіма своїми інтерфейсами (аналоговими і цифровими інтерфейсами дисплея, шиною локальної пам'яті). Чіпсети з інтегрованим графічним контролером можуть мати зовнішній порт AGP, що стає доступним при відключенні убудованого графічного контролера. Є чіпсети у яких порт AGP є чисто внутрішнім засобом підключення убудованого контролера, і зовнішній графічний контролер до них може підключатися тільки по шині PCI.

Минуло вже більше двадцяти років з тих пір, як був сконструйований перші комп'ютери RISC, проте деякі принципи їх функціонування можна перейняти, з огляду на сучасний стан технології розробки апаратного забезпечення. Якщо відбувається дуже різка зміна в технології (наприклад, новий процес виробництва робить час звернення до пам'яті в 10 разів менше, ніж час звернення до центрального процесора), змінюються всі умови. Тому розробники завжди повинні враховувати можливі технологічні зміни, які могли б вплинути на баланс між компонентами комп'ютера. Існує

ряд принципів розробки, іноді званих принципами RISC, яким по можливості намагаються слідувати виробники універсальних процесорів. Через деякі зовнішні обмеження, наприклад вимоги сумісності з іншими машинами, доводиться час від часу йти на компроміс, але ці принципи - мета, до якої прагне більшість розробників.

– Всі команди повинні виконуватися безпосередньо апаратним забезпеченням. Тобто звичайні команди виконуються безпосередньо, без інтерпретації мікрокомандами. Усунення рівня інтерпретації підвищує швидкість виконання більшості команд. У комп'ютерах типу CISC більш складні команди можуть розбиватися на кілька кроків, які потім виконуються як послідовність мікрокоманд. Ця додаткова операція знижує швидкодію машини, але може використовуватися для рідше використовуваних команд.

– Комп'ютер повинен запускати якомога більше команд в секунду. У сучасних комп'ютерах використовується багато різних способів підвищення продуктивності, головний з яких - запуск якомога більшої кількості команд в секунду. Зрештою, якщо процесор зможе запустити 500 млн команд в секунду, то його продуктивність становить 500 MIPS, скільки б часу не займало виконання цих команд. (MIPS - скорочення від Millions of Instructions Per Second – мільйон команд в секунду.) Цей принцип передбачає, що паралелізм повинен стати важливим чинником підвищення продуктивності, оскільки запустити на виконання велику кількість команд за короткий проміжок часу можна тільки в тому випадку, якщо є можливість одночасного виконання кількох команд. Хоча команди будь-якої програми завжди розташовуються в пам'яті в визначеному порядку, комп'ютер може змінити порядок їх запуску (так як необхідні ресурси пам'яті можуть бути зайняті) і (або) завершення. Звичайно, якщо команда 1 встановлює значення в регістр, а команда 2 використовує цей регістр, потрібно діяти з особливою обережністю, щоб команда 2 не рахувала значення з регістра раніше, ніж воно там виявиться, щоб не допускати подібних помилок, необхідно зберігати в пам'яті велику кількість додаткової інформації, але завдяки можливості виконувати кілька команд одночасно продуктивність все одно буде вищою.

– Команди повинні легко декодуватися. Межа кількості команд, що запускаються в секунду залежить від темпу декодування окремих команд. Декодування команд дозволяє визначити, які ресурси їм необхідні і які дії потрібно виконати. Все, що сприяє спрощенню цього процесу, корисно. Наприклад, можна використовувати однакові команди з фіксованою довжиною і з невеликою кількістю полів, чим менше різних форматів команд, тим краще.

– До пам'яті повинні звертатися тільки команди завантаження і збереження. Один з найпростіших способів розбити операцію на окремі кроки - зробити так, щоб операнди здебільшого команд бралися з регістрів і поверталися туди ж. Операція переміщення операндів з пам'яті в регістри і назад може здійснюватися в різних командах. Оскільки доступ до пам'яті займає багато часу, тривалість якої неможливо спрогнозувати, виконання цих команд можуть взяти на себе інші команди, єдине призначення яких - переміщення операндів між регістрами і пам'яттю. Тобто до пам'яті повинні звертатися тільки команди завантаження і збереження (LOAD і STORE).

– Регістрів повинно бути багато. Оскільки доступ до пам'яті відбувається відносно повільно, в комп'ютері має бути багато регістрів (принаймні 32). Якщо слово було одного разу завантажено з пам'яті, при наявності великого числа регістрів воно може міститися в регістрі до тих пір, поки не буде потрібне. Повернення слова з регістра в пам'ять і нове завантаження цього ж слова в регістр небажані. Кращий спосіб уникнути зайвих переміщень - наявність достатньої кількості регістрів.

Архітектура мікропроцесорів сімейства i80x86.

Загальні характеристики мікропроцесорів і еволюція розвитку. Логічна структура розвинутого мікропроцесора. Програмна модель мікропроцесора (8-бітні, 16-бітні та 32-бітні). Регістри. Типи даних. Організація захищеного режиму. Сегментна організація пам'яті. Сторінкова організація пам'яті. Суміщення сегментації і сторінкової організації пам'яті. Переривання і виключення. Організація мультизадачності.

Під архітектурою процесора прийнято розуміти його програмне модель, тобто програмно-видимі властивості. Під мікроархітектурою прийнято розуміти внутрішню реалізація цієї програмної моделі. Для однієї і тієї ж архітектури різними фірмами і в різних поколіннях застосовуються істотно різні мікроархітектурні реалізації, при цьому, природно, прагнуть до максимального підвищення продуктивності (швидкості виконання програм).

Сучасний мікропроцесор являє собою один-два "кристала", які об'єднують CPU з різними функціональними вузлами. З ускладненням структури мікропроцесора зростають вимоги до підвищення тактової частоти для досягнення важливої швидкодії. З огляду на залежність від шляхів досягнення можливого компромісу (створення продуктивного мікропроцесора з розгалуженою структурою і обмеженою тактовою частотою) розрізняють ряд архітектур сучасних мікропроцесорів:

- RISC (Reduced Instruction Set Computer) - процесори з скороченою системою команд. Ці процесори зазвичай мають набір однорідних регістрів універсального призначення, причому їх число може бути більшим. Система команд відрізняється відносною простотою, коди інструкцій мають чітку структуру, як правило, з фіксованою довжиною. В результаті апаратна реалізація такої архітектури дозволяє з невеликими витратами декодувати і виконувати ці інструкції за мінімальну (в межах 1) число тактів синхронізації. Відповідні переваги дає і уніфікація регістрів.

- CISC (Complete Instruction Set Computer) - процесори з повним набором інструкцій. Склад і призначення їх регістрів істотно неоднорідні, широкий набір команд ускладнює декодування інструкцій, на що витрачаються апаратні ресурси. Зростає число тактів, вкрай важливе для виконання інструкцій.

- MISC (Multipurpose Instruction Set Computer): є об'єднаними RISC CPU з мікропрограмно-керованим ПЗУ, об'єднання можливе як в одному корпусі, так і в різних. Основна частина (host), в якості якої є RISC CPU, виконує звичайні команди за один такт, а розширені команди перетворюються на адресу прошивки, які вже дешифровані. Реалізація MISC CPU на двох кристалах дозволяє розширити місце для мікропроцесорних регістрів.

Покоління процесорів сімейства i80x86.

Перше покоління процесорів. (Процесори 8086 і 8088 і математичний співпроцесор 8087) Черговий революційний процесор Intel - i8086 - з'явився в 1978 р. 14 16-розрядних регістрів: 4 регістра загального призначення (AX, BX, CX, DX), 2 індексних регістра (SI, DI), 2 вказівних (BP, SP), 4 сегментні регістри (CS, SS, DS, ES), програмний лічильник або покажчик команди (IP) і регістр прапорів (FLAGS, включає в себе 9 прапорів). Основні характеристики: 16-розрядна шина даних; сегментна адресація пам'яті; розмір шини адреси був збільшений з 16 біт до 20 біт; 29 тис. Транзисторів (технологія 3 мкм); система переривань; тактова частота 4,77-10 МГц; на виконання кожної інструкції йшло в середньому по 12 тактів процесорного ядра.

Друге покоління процесорів. (80286 з співпроцесором 80287)

відмінності: тактова частота 6.0 МГц - 12.5 МГц; до 14 регістрів процесора Intel 8086 були додані 11 нових регістрів: регістр слова стану машини, 16 біт (MSW); регістр завдання, 16 біт (TR); регістри дескрипторної таблиці, один 64-бітний і два 40-бітних (GDTR, IDTR, LDTR) і 6 регістрів

розширення сегментних регістрів, 48 біт; захищений режим; розширена шина 24 біта (16 Мбайт фізичної пам'яті); 134 тис. Транзисторів (технологія 1,5 мкм); на виконання інструкції йшло в середньому по 4,5 такту; рік виготовлення: 1982 Тактова частота: 6.0 МГц - 12.5 МГц.

Третє покоління процесорів. (386/387 з суфіксами DX і SX, визначальними розрядність зовнішньої шини). По-справжньому розвиток багатозадачності почалося після виходу мікропроцесора i80386 в 1985 р Це перший 32-розрядний процесор, який поклав початок сімейству процесорів IA-32 (32-bit Intel Architecture). Крім розширення діапазону безпосередньо подаються величин (16 біт відображають цілі числа в діапазоні 0-65535 або від - 32767 до 32767, а 32 біта - більш ніж 4 мільярди) збільшився і обсяг пам'яті (до 4 Гбайт реальної, 64 Тбайт віртуальної). Тактова частота 40 МГц. Додавання 32-розрядних регістрів. Введений новий режим роботи процесора - віртуальний 8086 процесор. Сторінкова адресація пам'яті (стало можливо організувати віртуальну пам'ять). 275 тис. транзисторів (технологія 1 мкм). Введена концепція паралельного функціонування внутрішніх пристроїв процесора: шинний інтерфейс, блок попередньої вибірки, блок декодування команд, виконавчий блок, блок сегментації, блок сторінкової адресації.

На такому процесорі стала "розквітати" система MS Windows - спочатку оболонка, а потім і операційна система. У плані організації виконання інструкцій істотних змін, які спричинили за собою скорочення числа тактів на інструкцію, не відбулося - ті ж середні 4,5 такту.

Рік: 1985 Тактова частота: 16 МГц - 33 МГц

Четверте покоління процесорів. (486, знову DX і SX) в видиму архітектурну модель великих змін внесено не було, але зате був прийнятий ряд заходів для підвищення продуктивності. Частота 25 МГц - 100 МГц. В цих процесорах значно ускладнений виконавчий конвеєр - основні операції виконує RISC-ядро, "завдання" для якого готують з вхідних CISC-інструкцій x86. => Конвейеризація обчислень (5 ступенів). Такий конвеєр став здатним виконувати інструкцію в середньому за два такту. Підтримка багатопроцесорного режиму роботи

1,2 млн. транзисторів (технологія 1 мкм)

Починаючи з процесора i80486, всі наступні моделі процесорів Intel підтримують різні концепції енергозбереження

Рік: 1989 Тактова частота: 25 МГц - 100 МГц

П'яте покоління процесорів. (Процесор Pentium у Intel знаменитий своєю помилкою блоку з плаваючою точкою і K5 у AMD). Тактова частота 60МГц - 233МГц. До внутрішнього кешу команд додали 8 Кбайт для даних. Роздільне кешування програмного коду і даних. В процесорах Pentium використовується кеш-пам'ять першого рівня (кеш L1) об'ємом 16 Кб, розділена на 2 сегменти: 8 Кб для даних і 8 Кб для інструкцій. 64-бітна шина даних дозволяє процесору Pentium за один шинний цикл обмінюватися вдвічі більшим об'ємом даних з оперативною пам'яттю, ніж 486 (при однаковій тактовій частоті).

Процесор з такою архітектурою може одночасно "випускати" до двох виконаних інструкцій, але в середньому виходить 1 такт на інструкцію. Внутрішні шини стали 128 і 256 біт, зовнішня шина даних 64 біт.

Рік: 1993 Тактова частота: 60МГц - 233МГц

Шосте покоління процесорів. (Pentium Pro) Можна охарактеризувати за такими особливостями: тактова частота 150 МГц - 200 МГц; кеш L1 16 Кбайт на кристалі процесора; кеш L2 256/512 Кбайт зовнішній на материнській платі; містить близько 5,5 мільйона транзисторів (технологія виготовлення 0,35 мікрон (для процесорів 120 МГц і нижче 0,6 мікрон)); динамічне виконання, під яким розуміється виконання інструкцій не в тому порядку (out of order), як це передбачається програмним кодом, а в тому, як "зручно" процесору.

В результаті всіх цих хитрувань середнє число тактів на інструкцію у Pentium Pro скоротилося до 0,5 такту.

AMD випускає процесор K5-PR133 (реально працюючий на частоті 116,7 МГц). Цей процесор має вбудований кеш 24 Кбайт, технологія виготовлення 0,35 мікрон, близько 4,3 мільйонів транзисторів.

Рік виготовлення: 1995 Тактова частота: 150 МГц - 200 МГц

Сьоме покоління процесорів. (Pentium 4)

Процесор Pentium 4 випущений в листопаді 2000 року, являє собою абсолютно нове покоління процесорів. Якщо замість імені йому привласнити порядковий номер, це буде процесор 786, так як він є представником зовсім іншого покоління, що відрізняється від попередніх процесорів.

Далі представлені основні технічні характеристики процесора Pentium 4.

- Тактова частота процесора знаходиться в діапазоні 1,3-1,7 ГГц і вище.
- Кількість транзисторів - 42 млн, 0,18-мікрона технологія, площа кристала - 217 мм² (Willamette).
- Кількість транзисторів - 55 млн, 0,13-мікрона технологія, площа кристала - 131 мм² (Northwood).
- Тактова частота шини процесора 400 або 533 МГц.
- Кеш-пам'ять першого рівня об'ємом 20 Кбайт (кеш контролю виконання команд об'ємом 12 Кбайт, плюс 8 Кбайт кеша даних).
- Асоціативний восьмирівневого 128-розрядна кеш-пам'яті другого рівня об'ємом 256 Кбайт, що працює на частоті процесора.

Рік випуску: 2000. Тактова частота: 1300-1400 МГц.

Восьме покоління процесорів (Itanium і Itanium 2)

Процесор Itanium був представлений в травні 2001. Якби Intel все ще використовувала числа для найменування процесорів, то Itanium на повній підставі отримав би назву 886, як процесор восьмого покоління сімейства Intel.

Itanium є першим процесором сімейства IA-64 (64-розрядна архітектура Intel).

Основні технічні характеристики процесора Itanium наведені нижче.

- Тактова частота процесора - 733/800 МГц і 1 ГГц.
- Три рівня інтегрованої кеш-пам'яті:
- 64-розрядна (плюс 8 розрядів для виконання ЕСС) шина процесора з тактовою частотою 266 МГц і пропускною спроможністю 2,1 Гбайт / с (тільки процесори Itanium).
- 128-розрядна (плюс 8 розрядів для виконання ЕСС) шина процесора з тактовою частотою 400 МГц і пропускною спроможністю 6,4 Гбайт / с (тільки процесори Itanium 2).
- 25 мільйонів транзисторів, плюс до цього до 300 млн транзисторів в кеш-пам'яті третього рівня.
- Кількість транзисторів - 221 млн, включаючи вбудовану кеш-пам'ять третього рівня (тільки Itanium 2).
- 128 регістрів для роботи з цілими числами, 128 регістрів з плаваючою комою, 8 регістрів розгалуження, 64 регістра предиката.

Itanium - перший мікропроцесор, в основу якого покладена 64-розрядна архітектура IA-64

Рік випуску: 2002. Тактова частота: 773 МГц - 1.33 ГГц

Дев'яте покоління мікропроцесорів

Core 2 Duo, Core 2 Quad - засновані на мікроархітектурі нового покоління є представниками другого покоління чіпів, створених з використанням процесу з нормами 65нм.

Аналогом від AMD є покоління, спочатку зване журналістами K8L, потім K10. І те й інше назва помилково. Сама AMD відмовилася від K-образних назв, і процесори дев'ятого покоління вона називає "AMD Next Generation Processor Technology" (Phenom).

При цьому тактові частоти системної шини процесорів Core 2 Duo становлять 800 або 1066 МГц, обсяг кеша L1 кожного ядра - по 32 Кб для даних і інструкцій відповідно, обсяг загального розподіленого кешу L2 - 2 Мб або 4 Мб.

Рік випуску 2006.

Десяте покоління мікропроцесорів

Core i7 - десяте покоління, засноване на мікроархітектурі Nehalem. Процесори виготовлялися за технологією 45 нм.

Core i7 спілкується з елементами чіпсета через послідовну шину з фірмовою назвою QuickPath Interconnect (QPI). Мікроархітектура Nehalem дозволяє використовувати кілька таких зовнішніх інтерфейсів для одного процесора.

Рік випуску 2008. Частота 2,66 - 3,6 ГГц.

Мікроархітектура	Технологія	Покоління	Дата випуску
Sandy Bridge	32 нм	11	1 кв. 2011 р.
Ivy Bridge	22 нм	12	2 кв. 2012 р.
Haswell	22 нм	13	3 кв. 2013 р.
Broadwell (Rockwell)	14 нм	14	3 кв. 2014 р.
Skylake	14 нм	15	4 кв. 2015 р.
Kaby lake	14 нм	16	4 кв. 2016 р.
Cannonlake	10 нм	17	2 кв. 2017 р.

Логічна структура мікропроцесора, тобто конфігурація складових мікропроцесор логічних схем і зв'язків між ними, визначається функціональним призначенням. Саме структура задає склад логічних блоків мікропроцесора і те, як ці блоки повинні бути пов'язані між собою, щоб повністю відповідати архітектурним вимогам. Спрацьовування електронних блоків мікропроцесора в певній послідовності призводить до виконання заданих архітектурою мікропроцесора функцій, тобто до реалізації обчислювальних алгоритмів. Одні і ті ж функції можна виконати в мікропроцесорах зі структурою, що відрізняється набором, кількістю і порядком спрацьовування логічних блоків. Різні структури мікропроцесорів, як правило, забезпечують їх різні можливості, в тому числі і різну швидкість обробки даних. Логічні блоки мікропроцесора з розвинуеною архітектурою показані на рис. 4.

I - керуюча частина, II - операційна частина.

- БУПК – блок управління послідовністю команд;
- БУВОп – блок управління виконанням операцій;
- БУФКА – блок управління формуванням кодів адрес;
- БУВП – блок управління віртуальною пам'яттю;
- БЗП – блок захисту пам'яті;
- БУПРПр – блок управління перериванням роботи процесора;
- БУВВ – блок керування введенням / виведенням;
- РНОЗП – реєстри надоперативний запам'ятовуючий пристрій;
- АЛБ – арифметико-логічний блок;
- БДА – блок додаткової арифметики;
- БС – блок синхронізації.

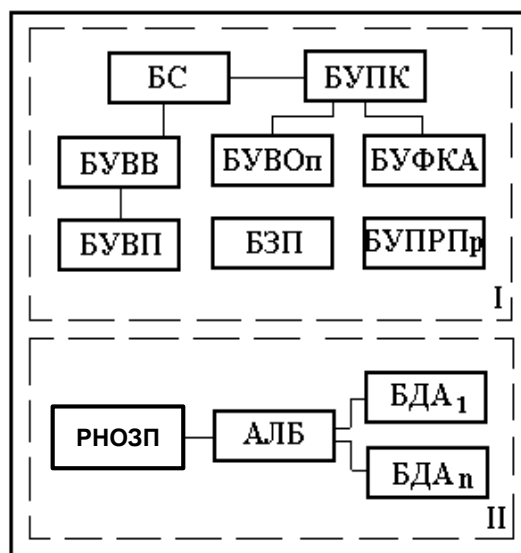


Рис. 4. Загальна логічна структура мікропроцесора

Уперше поняття архітектура ЕОМ стало згадуватися з появою машин 3-го покоління для їхньої порівняльної оцінки.

Програмна модель комп'ютера, частиною якої є програмна модель мікропроцесора, що містить 32 регістра тією чи іншою мірою доступних для використання програмістом може бути розділена на 2 великі групи:

16 Користувальницькі реєстри

16 Реєстри загального призначення.

Реєстр процесора — комірка швидкодійної внутрішньої пам'яті процесора, яка використовується для тимчасового збереження операндів, з якими безпосередньо проводяться обчислення, а також часто використовуваних даних з метою швидкого доступу до них. Крім того, в реєстрах зберігається і додаткова інформація, потрібна процесору для функціонування (зокрема, поточний контекст процесора, адреса наступної команди тощо).

Реєстри поділяються на користувачькі і системні.

1. Користувачькі реєстри.

Як випливає з назви, користувачькими реєстри називаються тому, що програміст може використовувати їх при написанні своїх програм. До цих реєстрів належать реєстри:

1. Загального призначення:

– `eax / ax / ah / al` (Accumulator register) - акумулятор.

Застосовується для зберігання проміжних даних.

– `ebx / bx / bh / bl` (Base register) - базовий реєстр.

Застосовується для зберігання базової адреси деякого об'єкту в пам'яті;

– `ecx / cx / ch / cl` (Count register) - реєстр - лічильник.

Застосовується в командах, які виробляють деякі повторювані дії.

– `edx / dx / dh / dl` (Data register) - реєстр даних.

Так само, як і реєстр `eax / ax / ah / al`, він зберігає проміжні дані.

– `esi / si` (Source Index register) - індекс джерела.

Цей реєстр у ланцюжкових операціях містить поточну адресу елемента в ланцюжку-джерелі;

– `edi / di` (Destination Index register) - індекс приймача (одержувача).

Цей реєстр у ланцюжкових операціях містить поточну адресу в ланцюжку-приймачі.

– `esp / sp` (Stack Pointer register) - реєстр покажчика стека.

Містить покажчик вершини стека в поточному сегменті стека.

– `ebp / bp` (Base Pointer register) - реєстр покажчика бази кадру стека.

Призначений для організації довільного доступу до даних усередині стека.

2. Сегментні:

– Сегмент коду. Містить команди програми.

Для доступу до цього сегменту служить реєстр `cs` (code segment register) - сегментний реєстр коду. Він містить адресу сегменту з машинними командами, до якого має доступ мікропроцесор (тобто ці команди завантажуються в конвеєр мікропроцесора).

– Сегмент даних. Містить оброблювані програмою дані. Для доступу до цього сегменту служить реєстр `ds` (data segment register) - сегментний реєстр даних, який зберігає адресу сегменту даних поточної програми.

– Сегмент стека. Цей сегмент являє собою область пам'яті, звану стеком. Роботу зі стеком мікропроцесор організує за наступним принципом: останній записаний в цю область елемент вибирається першим. Для доступу до цього сегменту служить реєстр `ss` (stack segment register) - сегментний реєстр стека, що містить адресу сегмента стека.

– Адреси додаткових сегментів даних повинні міститися в реєстрах `es`, `gs`, `fs` (extension data segment registers).

3. Стану і управління:

– `eflags / flags` (flag register) - реєстр прапорів. Розрядність `eflags / flags` - 32/16 біт. Окремі біти даного реєстра мають певне функціональне призначення і називаються прапорами. Молодша частина цього реєстра повністю аналогічна реєстру `flags` для i8086.

– eip / ip (Instruction Pointer register) - реєстр - покажчик команд. Реєстр eip / ip має розрядність 32/16 біт і містить зсув наступного підлягає виконанню команди щодо вмісту сегментного реєстра cs в поточному сегменті команд. Цей реєстр безпосередньо недоступний програмісту, але завантаження і зміна його значення виробляються різними командами управління, до яких відносяться команди умовних і безумовних переходів, виклику процедур і повернення з процедур. Виникнення переривань також призводить до модифікації реєстру eip / ip.

II. Системні реєстри.

1. Реєстри управління.

– Хоча мікропроцесор має чотири реєстри управління, доступними є тільки три з них — виключається **cr1**, функції якого поки не визначені (він зарезервований для майбутнього використання).

– Реєстр **cr0** містить системні прапори, для керування режимами роботи мікропроцесора, що відображають його стан глобально, незалежно від конкретних завдань, що виконуються.

– Реєстр **cr2** використовується при сторінковій організації оперативної пам'яті для реєстрації ситуації, коли поточна команда звернулася за адресою, що міститься в сторінці пам'яті, відсутній в даний момент часу в пам'яті.

– Реєстр **cr3** також використовується при сторінковій організації пам'яті. Це так званий реєстр каталога сторінок першого рівня. Він містить 20-бітову фізичну базову адресу каталога сторінок поточного завдання. Цей каталог містить 1024 32-бітових дескриптора, кожен з яких містить адресу таблиці сторінок другого рівня. У свою чергу кожна з таблиць сторінок другого рівня містить 1024 32-бітових дескриптора, що адресують сторінкові кадри в пам'яті. Розмір сторінкового кадру — 4 Кбайт.

2. Реєстри системних адрес.

– реєстра таблиці глобальних дескрипторів gdtr (Global Descriptor Table Register) що має розмір 48 біт і що містить 32-бітову (біти 16—47) базову адресу глобальної дескрипторної таблиці GDT і 16-бітове (біти 0—15) значення межі, що є розміром в байтах таблиці GDT;

– реєстра таблиці локальних дескрипторів ldtr (Local Descriptor Table Register) що має розмір 16 біт і що містить так званий селектор дескриптора локальної дескрипторної таблиці LDT. Цей селектор є покажчиком в таблиці GDT, який і описує сегмент, що містить локальну дескрипторну таблицю LDT;

– реєстра таблиці дескрипторів переривань idtr (Interrupt Descriptor Table Register) що має розмір 48 біт і що містить 32-бітову (біти 16—47) базову адресу дескрипторної таблиці переривань IDT і 16-бітове (біти 0—15) значення межі, що є розміром в байтах таблиці IDT;

– 16-бітового реєстра завдання tr (Task Register), який подібно до реєстра ldtr, містить селектор, тобто покажчик на дескриптор в таблиці GDT. Цей дескриптор описує поточний сегмент стану завдання (TSS — Task Segment Status). Цей сегмент створюється для кожного завдання в системі, має жорстко регламентовану структуру і містить контекст (поточний стан) завдання. Основне| призначення сегментів TSS — зберігати поточний стан завдання у момент перемикавання на інше завдання.

3. Реєстри відлагодження.

Реєстри dr0, dr1, dr2, dr3 мають розрядність 32 біти і призначені для завдання лінійних адрес чотирьох точок переривання. Реєстр dr6 називається реєстром стану відлагодження. Біти цього реєстра встановлюються відповідно до причин, які викликали виникнення останнього виключення з номером 1.

Біти і їх призначення:

b0 — якщо цей біт встановлений в 1, то останнє виключення (переривання) виникло в результаті досягнення контрольної крапки, визначеної в реєстрі dr0;

b1 — аналогічно b0, але для контрольної крапки в регістрі dr1;
b2 — аналогічно b0, але для контрольної крапки в регістрі dr2;
b3 — аналогічно b0, але для контрольної крапки в регістрі dr3;
bd (біт 13) — служить для захисту регістрів відлагодження;
bs (біт 14) — встановлюється в 1, якщо виключення 1 було викликано станом прапора tf = 1 в регістрі eflags;

bt (біт 15) встановлюється в 1, якщо виключення 1 було викликано перемиканням на завдання зі встановленим бітом пастки в TSS t = 1.

Решта всіх бітів в цьому регістрі заповнюється нулями. Обробник виключення 1 по вмісту dr6 повинен визначити причину, після якої відбулося виключення, і виконати необхідні дії.

З точки зору розмірності мікропроцесор апаратно підтримує наступні основні типи даних:

- байт - вісім послідовно розташованих бітів, пронумерованих від 0 до 7, при цьому біт 0 є наймолодшим значущим бітом;

- слово - послідовність з двох байт, що мають послідовні адреси. Розмір слова - 16 біт; біти в слові нумеруються від 0 до 15. Байт, що містить нульовий біт, називається молодшим байтом, а байт, що містить 15-й біт - старшим байтом. Мікропроцесори Intel мають важливу особливість - молодший байт завжди зберігається за меншою адресою. Адресою слова вважається адреса його молодшого байта. Адреса старшого байта може бути використаний для доступу до старшої половини слова.

- подвійне слово - послідовність з чотирьох байт (32 біта), розташованих по послідовним адресами. Нумерація цих біт виробляється від 0 до 31. Слово, що містить нульовий біт, називається молодшим словом, а слово, що містить 31-й біт, - старшим словом. Молодше слово зберігається за меншою адресою. Адресою подвійного слова вважається адреса його молодшого слова. Адреса старшого слова може бути використаний для доступу до старшої половини подвійного слова.

- у чотири рази збільшене слово - послідовність з восьми байт (64 біта), розташованих по послідовним адресами. Нумерація біт виробляється від 0 до 63. Подвійне слово, що містить нульовий біт, називається молодшим подвійним словом, а подвійне слово, що містить 63-й біт, - старшим подвійним словом. Молодше подвійне слово зберігається за меншою адресою. Адресою почетвереній слова вважається адреса його молодшого подвійного слова. Адреса старшого подвійного слова може бути використаний для доступу до старшої половини почетвереній слова.

Крім трактування типів даних з точки зору їх розрядності, мікропроцесор на рівні команд підтримує логічну інтерпретацію цих типів:

- Цілий тип зі знаком - бінарне значення зі знаком, розміром 8, 16 або 32 біта. Знак в цьому довічним числі міститься в 7, 15 або 31-м біте відповідно. Нуль в цих бітах в операндах відповідає позитивному числу, а одиниця - негативного. Негативні числа представляються в додатковому коді.

Числові діапазони для цього типу даних наступні:

- 8-розрядний ціле - від -128 до +127;
- 16-розрядний ціле - від -32 768 до +32 767;
- 32-розрядний ціле - від -231 до + 231-1.

- Цілий тип без знака - бінарне значення без знака, розміром 8, 16 або 32 біта. Числовий діапазон для цього типу наступний:

- байт - від 0 до 255;
- слово - від 0 до 65 535;
- подвійне слово - від 0 до 232-1.
- Показчик на пам'ять двох типів:

- ближнього типу - 32-розрядний логічний адресу, що представляє собою відносно зміщення в байтах від початку сегмента. Ці покажчики можуть також використовуватися в суцільній (плоскої) моделі пам'яті, де сегментні складові однакові;

- далекого типу - 48-розрядний логічний адресу, що складається з двох частин: 16-розрядної сегментної частини - селектора, і 32-розрядної зміщення.

- Ланцюжок - представляє собою деякий безперервний набір байтів, слів або подвійних слів максимальної довжини до 4 Гбайт.

- Бітове поле являє собою безперервну послідовність біт, в якій кожен біт є незалежним і може розглядатися як окрема змінна. Бітове поле може починатися з будь-якого біта будь-якого байта і містити до 32 біт.

- Неупакований двійковий-десятковий тип - байтове уявлення десяткової цифри від 0 до 9. Неупаковані десяткові числа зберігаються як байтові значення без знака по одній цифрі в кожному байті. Значення цифри визначається молодшим напівбайтів.

- Упакований двійковий-десятковий тип являє собою упаковане представлення двох десяткових цифр від 0 до 9 в одному байті. Кожна цифра зберігається у своєму напівбайті. Цифра в старшому напівбайті (біти 4-7) є старшою.

Мінімальний захист який використовується в комп'ютерних системах в основному використовує 2 режими:

- Режим повного доступу до ресурсів системи.

- Режим часткового доступу.

Такий спосіб забезпечує зняття конфліктів, що пов'язані із втручанням різних користувачів до конкретного ресурсу, а також забезпечують функціонування різного роду додатків.

Багатозадачний режим повинен забезпечуватися захистом від несанкціонованого доступу та засобами розподілу ресурсів системи. Захищений режим є основним режимом 32-розрядних процесорів і забезпечує повну незалежність виконання багатьох задач одночасно. В основному даний режим передбачає розподіл ресурсів пам'яті та різного роду системних таблиць, доступ до яких організовується з допомогою переривань, які в певний момент часу надають таблицю для користування. В основному захист пам'яті базується на програмно керованих засобах, тобто на рівні ОС. В захищеному режимі використовується сегментація пам'яті, відповідно до цього кожна задача використовує ті сегменти, які їй дозволила ОС. Інформація про це записується в дескрипторних таблицях. Захист пам'яті за допомогою сегментації не дозволяє:

- використовувати сегменти не за призначенням;

- порушувати права доступу

- отримувати доступ до елементів які виходять за рамки сегменту;

- змінювати вміст таблиць дескрипторів (не маючи на це прав).

Захищений режим має засоби переходу від 1-ї задачі до іншої. Стан кожної задачі може бути збережений в спеціальному сегменті стану.

Механізм що дозволяє будь якій задачі використовувати логічний адресний простір до 64 Гбайт називається механізмом віртуальної пам'яті.

Технологія сегментації дозволяє поділити адресний простір процесу на кілька сегментів. Довільна адреса програми містить ідентифікатор сегмента (segment ID) і зсув від його початку. Кожен сегмент може мати власні налаштування захисту. Сегменти завантажуються в оперативну пам'ять цілком, кожний з них описується за допомогою дескриптора, що містить фізичну адресу, за яким завантажений сегмент (базова адреса), розмір і інформація про захист. Апаратна частина перевіряє границі сегмента при кожній операції доступу, що виключає факти ушкодження процесом

сусіднього сегмента. Завантаження і вивантаження (свопінг) можуть бути проведені на рівні окремих сегментів, а не всієї програми.

Сегментація в інших випадках використовується разом з розбиттям на сторінки, що в результаті дає гнучкий гібридний механізм керування пам'яттю. У таких системах сегменти не обов'язково повинні розташовуватися у пам'яті неперервно.

Сторінкова пам'ять — це підхід до організації віртуальної пам'яті, при якому одиницею відображення віртуальних адрес на фізичні є регіон константного розміру (так звана сторінка). Типовий розмір сторінки — 4096 байт, для деяких архітектур — до 128 Кб.

Підтримка такого режиму присутня в більшості 32-бітних та 64-бітних процесорів. Такий режим є класичним майже для усіх сучасних ОС, в тому числі Windows та сімейства UNIX. Широке використання такого режиму почалося з процесора VAX та ОС VMS з кінця 1970-х років. В сімействі x86 підтримка з'явилася з поколінням 386, першим 32-бітним поколінням.

Вирішує наступні проблеми:

- підтримка ізоляції процесів та захисту пам'яті шляхом створення власного віртуального адресного простору для кожного процесу;
- підтримка ізоляції області ядра від коду користувальницького режиму;
- підтримка пам'яті «тільки для читання» та невиконуваної пам'яті;
- підтримка відвантаження невикористовуваних сторінок в зону підкачування на диску (див. свопінг);
- підтримка відображення файлів в пам'ять;
- підтримка роздільної між процесами пам'яті;
- підтримка системного виклику `fork()` в ОС сімейства UNIX.

Сегментація може використовуватися для сумісного використання організації захисту програмних ресурсів, сторінкова організація вирішує проблеми фрагментації і перекомпоновки. Такий комбінаційний підхід був застосований в декількох сучасних обчислювальних системах.

В зв'язку з тим, що розміри таблиць сегментів і сторінок можуть бути значними, вони звичайно поміщуються в основну пам'ять, а не в спеціальні апаратні регістри. Таким чином, апаратне виконання однієї команди може вимагати трьох звертань до основної пам'яті для запису одного слова даних.

Для подолання цієї проблеми дуже ефективним є показовий метод запам'ятовування найбільш часто використаних сегментів і сторінок в невеликій швидкодіючій асоціативній пам'яті. Цей метод використаний в обчислювальних машинах IBM модель 67 і HiS645. Але навіть використання буферів асоціативної пам'яті не виключає повністю зниження швидкодії викликаної багаторівневою схемою адресації. Звичайно це зниження швидкодії вдається зробити меншим десяти процентів, однак воно суттєво залежить від характеристик програми і ефективності роботи буферів асоціативної пам'яті.

Сегментація і робота з сторінками приводить до збільшення вартості обчислювальної системи.

Для повного використання можливостей сторінкової організації і сегментації потребується значна ускладненість програм управління даними операційної системи.

Відмінності між перериваннями і виключеннями полягають у тому, що переривання призначені для обробки запитів від зовнішніх пристроїв, а винятки для обробки помилок, що виникають при виконанні команд. Програмні переривання також відносяться до винятків. Переривання, зроблені обладнанням, виконуються після виконання поточної команди і відбуваються в результаті якихось зовнішніх асинхронних (не пов'язаних з поточним процесом) подій, натискання клавіші, наприклад. Після того, як програма обробки переривань закінчує обслуговування переривання, виконання перерваної програми триває з команди, яка йде відразу за

командою, після якої відбулося переривання. Винятки класифікуються як помилки, пастки або припинення (передчасне припинення виконання програми).

Джерелами винятків є три типи подій:

– генеруються програмою винятку, що дозволяють програмі контролювати певні умови в заданих точках програми (INT0 - перевірка на переповнення, INT3 - контрольна крапка, BOUND - перевірка меж масиву);

– виключення машинного контролю, що виникають у процесі контролю операцій всередині чіпа і транзакцій на шині процесора (справедливо для процесора Pentium 4);

– виявлені процесором помилки в програмі (поділ на нуль, порушення правил захисту, відсутність сторінки і т.п.)

Помилки - це винятки, які виявляються і виконуються до виконання команди містить помилку. Пастки-це винятки, про які повідомляється негайно після виконання тієї команди, яка викликає виключення (вироблене дію вже не виправити). Припинення - це винятки, які не дозволяють визначити точно команду, яка викликала виняток. Припинення використовуються для повідомлення про грубі помилки, таких як апаратна помилка, неправильні значення в системних таблицях. Всі можливі переривання зводяться в спеціальну таблицю (до 256 різних переривань і виключень). У таблиці кожному переривання або виключення зіставляється вектор переривання-це покажчик на відповідну програму обслуговування (вектор дозволяє знайти адресу програми, обробній переривання).

Типи переривань:

1. Програмні переривання. Програмні переривання можуть використовуватися для доступу до сервісів операційної системи. Ще один тип програмного переривання, що застосовується при налагодженні, - переривання покрокового режиму.

2. Масковані переривання. Масковані переривання - найбільш загальний спосіб відповіді на асинхронні зовнішні сигнали від апаратури.

Обробка переривань

Коли відбувається переривання, відбуваються такі дії. По-перше, адреса поточної команди і регістр прапорів зберігаються в стеку, що дозволяє відновити перервану програму. Потім, за вектором, який визначає відповідний елемент в таблиці переривань визначається початковий адресу програми обробки переривання. Виконується програма обробки. І нарешті, після команди повернення з програми обробки переривання (IRET) відновлюється колишній стан процесора і за адресою повернення (за адресою збереженому в стеку) поновлюється виконання програми.

26. 14. Організація мультизадачності

Під мультизадачністю або багатозадачністю розуміють здатність комп'ютера виконувати кілька задач одночасно. Задачею, в загальному випадку, називають послідовність взаємопов'язаних дій, що ведуть до досягнення певної мети. Під задачею (в тематиці системних наук) розуміють програму і пов'язані з нею дані, яка виконується або очікує виконання, поки виконується інша програма. Іншими словами, задачею вважаються програми, завантажені в оперативну пам'ять комп'ютера. Та задача, яка виконується в даний момент, тобто поточна програма називається процесом. У визначення задачі часто включаються і ресурси процесора, які потрібні для виконання цієї задачі, тобто для досягнення поставленої мети: обсяг необхідної для вирішення задачі пам'яті, необхідні регістри, необхідний час для її виконання і т.п. Дескриптори всіх сегментів і шлюзів однієї задачі зазвичай об'єднуються в одній локальній дескрипторній таблиці (LDT) з метою посилення захисту даної задачі від сторонніх небажаних впливів.

Звичайний процесор не може, строго кажучи, одночасно виконувати більш одного командного потоку. Однак він може виконувати кілька завдань поперемінно, швидко переключаючись з одного командного потоку на інший. Оскільки швидкодія сучасних процесорів дуже велика, для користувачів створюється враження одночасного виконання безлічі завдань.

Вперше мультизадачність була організована досить давно, ще в епоху переважання великих комп'ютерів загального призначення. Головною метою запровадження багатозадачного режиму в той час було ефективне використання дорогих комп'ютерів багатьма користувачами одночасно, в режимі поділу часу. Однак вона здійснювалася чисто програмним шляхом за допомогою спеціальних операційних систем з поділом часу. До теперішнього часу багатозадачність отримала дуже широке поширення і по іншим причинам, причому багато її функції підтримуються і апаратно.

Кешування пам'яті.

Принципи кешування. Кеш прямого відображення. Асоціативний кеш. Архітектура і характеристики засобів кешування. Управління кешуванням. Мікроархітектура мікропроцесорів (i8086-i80484, P5, P6, P7).

Принцип роботи кеш-пам'яті полягає в наступному. Процесор рідко використовує весь обсяг ОЗУ практично одночасно. Скакати з одного кута пам'яті в інший, періодично «пробігаючи» по всьому її об'єму - це не найкращий спосіб використання ресурсів комп'ютера. Найчастіше всі звернення процесора до пам'яті зосереджені в невеликій області (як показує статистика - 5-10% від загального обсягу). Якщо дані з цієї області апаратно скопіювати в кеш, а потім постійно звіряти кеш і ОЗУ на предмет цілісності даних, то можна забезпечити режим роботи, при якому процесор буде звертатися тільки до кеш-пам'яті, витрачаючи на це значно менше ресурсів і часу, ніж зазвичай.

Природно, що весь обсяг ОЗУ скопіювати в кеш немає можливості - такий об'єм кеш-пам'яті за ціною порівнюється з вартістю комп'ютера, а сенсу зменшувати обсяг ОЗУ теж немає. Було вирішено реалізувати алгоритм роботи процесора, кеш-пам'яті і ОЗУ апаратно, щоб не витрачати ресурси процесора.

Принцип полягає в наступному: коли процесор звертається до певної комірки пам'яті, сегмент пам'яті певного обсягу (цей обсяг називається об'ємом сторінки кеш) копіюється в кеш повністю. Якщо процесор далі не зробить глобальний стрибок на іншу, далеку від поточного, адресу пам'яті, то подальша робота процесора буде відбуватися безпосередньо з кеш, минаючи ОЗУ, а контролер кеш-пам'яті в проміжках, коли процесор зайнятий обчисленнями (або паралельно з роботою процесора) буде відновлювати вірні дані в ОЗУ або в кеш (у разі наявності пристроїв, що безпосередньо працюють з пам'яттю). Природно, чим більше буде сторінок і чим більше буде їх обсяг - тим вищою буде швидкість роботи процесора.

Починаючи з 386-х процесорів кеш-пам'ять встановлювалася на більшість материнських плат. Все це дозволяло підвищувати продуктивність систем не в рази, а на порядки. Але з виходом 486 процесора кеш-пам'ять була розділена: з'явився кеш, реалізований безпосередньо на кристалі процесора (у 486DX). Незважаючи на малий його обсяг - 1 Кб - це вже був стрибок у продуктивності і дуже великий: адже кеш, розташований на кристалі процесора працює на частоті процесора, що в кілька разів перевищує швидкість роботи кеш-пам'яті, встановленої на материнській платі.

Починаючи з 486 процесорів кеш-пам'ять відчуває такий поділ на кеш 1-го рівня - встановлений на кристалі (або в одному корпусі) процесора і працює з ним на одній частоті, і кеш 2-го рівня, встановлений на материнській платі, що працює швидше звичайної ОЗУ, але повільніше кеша першого рівня.

Однак цим не обмежується використання кеш-пам'яті. Адже таким чином її можна використовувати там, де існує проблема швидкодії, але є можливість впорядкувати дані. До таких застосувань відносять:

- апаратне кешування жорстких дисків (кеш-пам'ять встановлюється безпосередньо на жорсткому диску або на спеціальному контролері);
- програмне кешування CD-ROM, а також інших пристроїв зберігання інформації (програмно - за допомогою операційної системи, апаратно - на самому пристрої або на контролері).

Найбільш популярною є кеш-пам'ять першого рівня (процесорна) та другого (встановлена на материнській платі).

Адреса пам'яті, за якою відбувається звернення до кеша прямого відображення, однозначно визначає рядок кеша, де може знаходитись необхідний блок.

Сутність архітектури прямого відображення полягає в тому, що кожен рядок кеша може відображати з будь-якої сторінки кешованої пам'яті тільки відповідний йому рядок. При цьому на

кожний рядок кеша може претендувати багато сторінок пам'яті з однаковою молодшою частиною адреси, що є зміщенням всередині сторінки. Один рядок в певний момент містить копію тільки однієї з цих сторінок. Адресу (номер) рядка в кеш-пам'яті називають індексом (index). Інформацію про те, яка саме сторінка основної пам'яті займає даний рядок, тобто старша частина адреси чи номер сторінки, містить тег. Пам'ять тегів має кількість комірок, яка дорівнює кількості рядків кеша, а її розрядність повинна бути достатньою, щоб вмістити старші біти адреси кешованої пам'яті, які не потрапили на шину адреси кеш-пам'яті. Крім адресної частини тега кожний рядок кеша відображає біти ознак дійсності та модифікування даних.

На початку кожного звернення до кеш-пам'яті контролер спочатку зчитує комірку каталогу із заданим індексом, порівнює біти адреси тега зі старшими бітами адреси пам'яті та аналізує ознаку дійсності. Такий аналіз виконується в спеціальному циклі стеження (snooper cycle), який ще називають циклом запиту (inquire). Якщо в результаті аналізу з'ясується, що потрібний блок не знаходиться в кеші, то генерується або продовжується цикл звернення до основної пам'яті (випадок кеш-промаху). Коли ж є кеш-попадання, то запит обслуговується кеш-пам'яттю. У випадку кеш-промаху після зчитування основної пам'яті нові дані розміщуються в рядку кеша за умови, що він чистий, а в його тегу розташовуються старші біти адреси, і встановлюється ознака дійсності даних. З основної пам'яті рядок переписується в кеш повністю, незалежно від обсягу даних, що вимагаються, оскільки ознака дійсності належить до всіх його байтів. Якщо контролер реалізує випереджаюче зчитування (read ahead), то в наступні вільні цикли шини оновиться і наступний рядок, якщо він був чистим. Читання «із запасом» дає змогу за необхідності здійснювати пакетний цикл зчитування з кеша через межу рядка.

Недоліком такої організації кеш-пам'яті є робота «вхолосту» (cache trashing), коли в процесі виконання програми процесора по черзі будуть потрібні блоки (сторінки) пам'яті, зміщені один стосовно іншого на величину, кратну розміру сторінки. Чергове звернення замінюватиме дані, зчитані в попередньому зверненні, які будуть необхідні в наступному. Таким чином, це буде суцільна низка кеш-промахів. Зменшує кількість кеш-попадань також переключення сторінок в багатозадачних обчислювальних системах. Оскільки різні задачі претендуватимуть на одні й ті самі рядки кеша, то збільшення його розмірів за архітектури прямого відображення не дає суттєвого підвищення ефективності. Підвищити її, не збільшуючи ємність кеша, можна тільки зміною структури кеш-пам'яті.

Набірно-асоціативна архітектура кеша дає можливість кожній сторінці основної кешованої пам'яті претендувати на один з кількох рядків кеша, об'єднаних в набір (set). У кеші такої архітектури є кілька паралельно і погоджено працюючих каналів прямого відображення, де контролер кеша приймає рішення про те, в який з рядків набору помістити черговий блок даних.

У простішому випадку кожний блок з основної пам'яті поміщається в один з двох рядків (Two Way Set Associative Cache). Такий кеш містить два банки пам'яті й тегів.

Номер набору, тобто індекс, в якому відображено необхідний блок даних, однозначно визначається середньою частиною адреси, як номер рядка в кеші прямого відображення. Рядок набору, який відображає необхідний блок, визначається порівнянням тегів, паралельно виконуваним для всіх каналів кеша. З кожним набором пов'язана також ознака, що визначає рядок набору, який підлягає заміщенню новим блоком даних у випадку кеш-промаху. На заміщення обирається той рядок, до якого довше всього не зверталися. Воно виконується згідно з алгоритмом LRU (Least Recently Used). За відносно великої кількості каналів (рядків в наборі) вдаються до спрощення, а саме використовують алгоритм Pseudo LRU для чотирьох рядків (Four Way Set Associative Cache), який дає змогу приймати рішення, використовуючи лише три біти. Можливе також застосування алгоритму заміщення FIFO чи випадкового (random) заміщення. Це простіше, але менш ефективно. Набірно-асоціативна архітектура широко застосовується для первинного кеша

сучасних комп'ютерів. Ємність кешованої пам'яті визначається так само, як і у випадку прямого відображення, але для набірно-асоціативної архітектури фігуруватиме ємність одного банку, а не всього кеша, а також розрядність комірок тега, що належать до цього банку.

У повністю асоціативному кеші, на відміну від попередніх архітектур, будь-який його рядок може відображати будь-який блок пам'яті. Це суттєво підвищує ефективність використання його обмеженої ємності. Всі біти адреси кешованого блока, за винятком бітів, які визначають розташування даних у рядку, тобто зміщення, зберігаються в пам'яті тегів. При цьому для визначення наявності даних кеш-пам'яті, що вимагаються, необхідне порівняння тегів усіх рядків зі старшою частиною адреси, а не одного чи кількох, як при прямому відображенні чи набірно-асоціативній архітектурі. Таким чином, відпадає необхідність послідовного перебирання комірок пам'яті тегів. Виконується тільки паралельний аналіз усіх комірок. Але це завдання поки що вирішене тільки для невеликих ємностей первинного кеша в деяких процесорах.

Первинний кеш інструкцій тісно пов'язаний з блоком попередньої вибірки, а первинний кеш даних - з виконавчим блоком процесора. Вторинний кеш є загальним, і в процесорах P6 підключений до окремої внутрішньої шини кеш-пам'яті. У процесорах i486 і Pentium вторинний кеш є зовнішнім і підключається до зовнішньої системної шини процесора. У процесорах Celeron 266 і 300 вторинний кеш відсутня.

Будь-який внутрішній запит процесора на звернення до пам'яті направляється у внутрішній кеш. Якщо запитувана ділянка пам'яті є в рядку внутрішнього кеша, то він обслуговує цей запит. У разі промаху запит задовольняється, як тільки необхідні дані зчитуються з ОЗУ. Заповнення рядка до кінця відбувається паралельно з обробкою отриманих даних. Виділення і заміщення рядків в процесорах i486 і P5 виконуються тільки для кеш-промахів при зчитуванні. При промахах записи заповнення рядків здійснюються тільки в процесорах P6. Кешування є в будь-якому режимі процесора.

Буфер асоціативної трансляції TLB зберігає входження в каталог і в таблиці сторінок, до яких зверталися останнім часом. У i486 для даних і інструкцій використовується єдиний TLB, а в процесорах P5 і P6 ці буфери роздільні.

Великі сторінки, зокрема 2Мбайт в режимі PAE (Physical Address Extension - режим розширення фізичної адреси до 36 біт) і 4 Мбайт в режимі PSE (Page Size Extension - прапорець розширення розміру сторінки) обслуговуються роздільними TLB.

Буфери запису пов'язані з виконавчим блоком процесора. Вони дозволяють на деякий час відкласти фактичний запис у зовнішній кеш і основну пам'ять, пропонуючи шину для інших обмінів, необхідних для виконання наступних інструкцій. Запис буферизується у всіх режимах роботи процесора. Але буферизації запису в порти введення / виводу не відбувається.

Всі механізми кешування, як правило, прозорі для прикладних програм і після дозволу кешування пропускають через себе потоки інструкцій і даних без вимог явного програмного управління. Але знання особливостей механізмів кешування пам'яті дає можливість оптимізувати коди.

Процесори мають механізми зовнішнього стеження за станом свого кеша. Для підтримки узгодження даних кеша та основної пам'яті процесор відпрацьовує цикли стеження (Snoop Cycle чи Inquire Cycle), що ініціюються зовнішньою для нього системою. В цих циклах, які відбуваються при зверненні до пам'яті з боку зовнішнього абонента, процесор визначає наявність даних, що вимагаються, в своєму кеші. Якщо вони відображаються в кеші, то дії процесора залежать від стану відповідного рядка кеша і типу зовнішнього звернення. Звернення за записом призведе до анулювання даного рядка. Звернення за зчитуванням до ділянки, що відповідає модифікованому («брудному») рядку, призведе до вивантаження його вмісту в основну пам'ять перед тим, як зовнішній абонент виконає реальне зчитування. В процесорах P6 звернення до «брудного» рядка з

боку інших процесорів може спричинити вивантаження його вмісту безпосередньо в процесор, що звертався. Це відповідно збереже час, а вивантаження цього рядка в основну пам'ять відбудеться пізніше, згідно з алгоритмом оберненого запису.

У просторі основної пам'яті комп'ютера є ділянки, для яких кешування принципово недопустиме, зокрема розподільна пам'ять адаптерів. Для таких ділянок непридатний алгоритм оберненого зв'язку. Крім того, кешування інколи відключають при виконанні однократно виконуваних ділянок програми з тим, щоб з кеша не витіснити корисніші фрагменти програми.

Механізм керування кешуванням містить як програмні, так і апаратні засоби, які дозволяють кешування та обмежують його можливості. До програмних засобів належать прапорці керуючих регістрів і біти елементів каталогу та таблиць сторінок, а також спеціальні інструкції. Апаратні засоби – це вхідні сигнали дозволу кешування і керування записом та очищенням кеша, а також вихідні сигнали керування вторинним кешем. Процесори P6 містять також регістри MTRR (Memory Type Range Registers – регістри визначення типів пам'яті в складі M8R – модельно специфічних регістрів), які визначають можливості кешування на рівні ділянок фізичної пам'яті. Якщо різні механізми визначають можливості кешування конкретної ділянки пам'яті по-різному, то спрацьовує найжорсткіше обмеження – заборона кешування. Вона більш пріоритетна, ніж дозвіл, а режим WT скасовує режим WB.

Кешуванням керують на етапі заповнення рядків, а кеш-попадання зчитування пам'яті обслуговуються тільки з кеша. Існує можливість анулювання рядків шляхом оголошення їх недостовірними, а також очищення всієї кеш-пам'яті. Очищення внутрішньої кеш-пам'яті здійснюється зовнішнім сигналом FLUSH# за один такт системної шини. Те саме відбувається за сигналом RESET. При оберненому записі під час очищення відбувається також вивантаження всіх модифікованих рядків в основну пам'ять. Для цього необхідна значна кількість тактів системної шини. Крім того, існують інструкції анулювання (інвалідазації). Інструкція INVD анулює рядки внутрішнього кеша без вивантаження модифікованих рядків. Інструкція WBINVD попередньо вивантажує модифіковані рядки в основну пам'ять.

Анулювання рядків здійснюється при записі у відображувану ними ділянку пам'яті з боку зовнішніх процесорів. Якщо зовнішній процесор виконує запис у пам'ять, то головному процесору повинен бути поданий сигнал AHOLD. За цим сигналом головний процесор негайно віддає керування шиною адреси, на якій зовнішнім процесором встановлюється адреса пам'яті, що супроводжується стробом EADS#. Якщо пам'ять, що адресується, наявна в первинному кеші, то процесор складає біт достовірності цього рядка.

Загальне програмне керування кешуванням здійснюється бітами керуючого регістра CRO: CD (Cache Disable) і NW (No Write Through).

i8086

Всього в процесорі i8086 було 14 16-розрядних регістрів:

- 4 регістри загального призначення (AX, BX, CX, DX),
- 2 індексні регістри (SI, DI),
- 2 вказівні регістри (BP, SP),
- 4 сегментні регістри (CS, SS, DS, ES),
- програмний лічильник або показник команди (IP)
- регістр прапорців (FLAGS, включає 9 прапорів).

Розмір шини адреси був збільшений з 16 біт до 20 біт, що дозволило адресувати 1 Мбайт (220 байт) пам'яті. Шина даних була 16-розрядною.

Для того, щоб адресувати більший, ніж i8080, обсяг пам'яті, треба було змінити спосіб адресації пам'яті. Адже якщо використовувати старі методи, коли адресу до комірки пам'яті містився у вказівних регістрах, то довелося б збільшувати розмір тих самих регістрів, щоб мати

можливість звертатися до більшого обсягу пам'яті. Тому для адресації 1 Мбайт пам'яті застосовували наступну схему. На шину адреси подавалася фізична адреса розміром 20 біт, яка формувалася шляхом складання вмісту одного із сегментних регістрів (16 біт), помноженого на 24, з вмістом вказівного регістра: таким чином, адресація комірки пам'яті вироблялася за номером сегмента і ефективною адресою комірки в сегменті (яка також називається зсувом). Якщо результат додавання виявлявся більше ніж 220 -1, то 21-ий біт відкидався; така процедура називається «загортанням» адреси (англ. address wraparound). Цей метод згодом (після появи захищеного режиму) назвали реальним режимом адресації процесора, такий режим дозволяє адресувати до 1 Мбайт пам'яті.

i80486

Intel486 мав розташовану на кристалі кеш-пам'ять об'ємом 8 Кбайт, пізніше — 16 Кбайт, що працює на частоті ядра.

Спочатку кеш Intel486 працював за принципом наскрізного запису (англ. write-through, WT), але пізніше, в рамках сімейства Intel486, були випущені моделі з внутрішнім кешем, працює за принципом зворотного запису (англ. write-back, WB). Процесор міг використовувати і зовнішній кеш, швидкість читання-запису якого, однак, була помітно нижче ніж у внутрішнього кеша. При цьому внутрішній кеш стали називати кешем першого рівня (Level 1 Cache), а зовнішній кеш, розташований на материнській платі, кешем другого рівня (Level 2 Cache). Кеш мав 4-канальну набірно-асоціативну архітектуру і працював на рівні фізичних адрес пам'яті.

В Intel486 був вдосконалений механізм виконання інструкцій в кілька етапів. Конвеєр процесорів серії Intel486 складався з 5 ступенів: вибірка інструкції, декодування інструкції, декодування адрес операндів інструкції, виконання команди, запис результату виконання інструкції. Використання конвеєра дозволило під час виконання однієї інструкції виробляти підготовчі операції над іншою інструкцією. Це в значній мірі дозволило збільшити продуктивність процесора.

У процесорі є той же набір інструкції що і в Intel386, до якого було додано декілька додаткових регістрів, а саме, три 32-бітових тестових регістра (TR5, TR4, TR3). Також були додані нові прапори в регістрі прапорів (EFLAGS) та в інших керуючих регістрах (CR0, CR3).

Внаслідок включення співпроцесора в кристал процесора, в Intel 486 можна звертатися і до регістрів FPU: регістри даних, регістр тегів, регістр стану, покажчики команд і даних FPU, регістр управління FPU.

Набір інструкцій не зазнав істотних змін, але були додані додаткові інструкції для роботи з внутрішньої кеш-пам'яттю (INVD, INVLPG, WBINVD), одна інструкція (BSWAP) для забезпечення сумісності з процесорами Motorola, дві інструкції для атомарних операцій з пам'яттю: CMPXCHG (для порівняння з обміном — нове значення записувалося тільки якщо старе збігалося із заданим, старе запам'ятовувалося) і XADD (інструкція для складання двох операндів з приміщенням результату в другому операнді, а не в першому, як у ADD). Інструкція CPUID дозволяла вперше в сімействі x86 безпосередньо отримати детальну інформацію про версію і властивості процесора. Крім цього, до набору інструкцій додалося 75 інструкцій FPU.

Довжина черги інструкцій була збільшена до 32 байт.

P5

Процесори Pentium першого покоління. Дві (єдині) моделі анонсовані 23 березня 1993 і працювали з тактовою частотою ядра 60 і 66 МГц, частота системної шини (FSB) дорівнювала частоті ядра, тобто множник ядра дорівнював 1,0. Кеш другого рівня розміщувався на материнській платі і міг мати розмір до 1 Мб. Процесор випускався в 273-контактному корпусі CPGA, встановлювався в корпус Socket 4 і працював від напруги 5 В. Всі процесори Pentium відносяться

до класу SL Enhanced — це означає, що в них передбачена система SMM, що забезпечує зниження енергоспоживання. Ранні варіанти процесорів, з частотами 60-100 МГц (ядра P5 і P54C), мали помилку в модулі FPU (математичний співпроцесор), яка в рідкісних випадках призводила до зменшення точності операції ділення. Цей дефект був виявлений в 1994 році і став відомий як «Pentium FDIV баг». Процесори на ядрі P5 виготовлялися з використанням 800-нанометрового техпроцесу, по біполярній BiCMOS-технології. Процесор містить 3,1 млн транзисторів, а розмір кристала ядра становить 294 мм². Pentium 66 споживає струм в 3,2 А і має потужність 16 Вт, що зажадало установки додаткового вентилятора. Виробництво таких процесорів виявилось дуже складним і відсоток виходу придатних кристалів виявився занадто малий. Багато фахівців, вказуючи на численні недоліки процесорів Pentium першого покоління, не радили купувати дані моделі. Виробництво на час довелося зупинити. Однак незабаром почалося виробництво вдосконалених процесорів, заснованих на ядрі P54C.

P6

Перші процесори архітектури P6 в момент виходу значно відрізнялися від існуючих процесорів. Процесор Pentium Pro відрізняло застосування технології динамічного виконання (зміни порядку виконання інструкцій), а також архітектура подвійної незалежної шини (англ. Dual Independent Bus), завдяки чому було знято багато обмежень на пропускну здатність пам'яті, характерні для попередників і конкурентів. Тактова частота першого процесора архітектури P6 становила 150 МГц, а останні представники цієї архітектури мали тактову частоту 1,4 ГГц. Процесори архітектури P6 мали 36-розрядну шину адреси, що дозволило їм адресувати до 64 ГБ пам'яті (при цьому лінійний адресний простір процесу обмежено 4 ГБ, див. PAE).

Суперскалярний механізм виконання інструкцій із зміною їх послідовності.

Подвійна незалежна шина.

P7 (NetBurst)

Архітектура NetBurst розроблялася, в першу чергу, з метою досягнення високих тактових частот процесорів. Характерними особливостями архітектури NetBurst є гіперконвейеризація і застосування кешу послідовностей мікрооперацій замість традиційного кешу інструкцій. АЛП процесорів архітектури NetBurst також має суттєві відмінності від АЛП процесорів інших архітектур. Також можливе застосування технології Hyper-threading.

Системна пам'ять персонального комп'ютера.

Логічна організація пам'яті. Розподіл адресного простору пам'яті. Розширена пам'ять. Додаткова пам'ять (Expanded Memory). Системна пам'ять (Shadow RAM). Фізична організація пам'яті. Організація динамічного ОЗП. Організація банків пам'яті і режим роботи ОЗП. Типи динамічної пам'яті. Модулі динамічної пам'яті

Як відомо, використовуваний в IBM PC, PC / XT мікропроцесор i8088 через свої 20 адресних шин надає доступ лише до 1-Мбайтному простору пам'яті. Перші 640 Кбайт адресується простору в IBM PC-сумісних комп'ютерах називають зазвичай стандартною пам'яттю (conventional memory). Решта 384 Кбайта зарезервовані для системного використання і носять назву пам'яті у верхніх адресах (UMB, Upper Memory Blocks, High DOS Memory або UM Area - UMA). Ця область пам'яті резервується під розміщення системної ROM BIOS (Read Only Memory Basic Input Output System), під відеопам'ять і ROM-пам'ять додаткових адаптерів.

Адресний простір - це просто діапазон адрес, що позначають певне місце в пам'яті. Адресні простори поділяються на три різновиди: фізичне адресний простір, лінійне адресний простір, логічне адресний простір, відоме також як віртуальний адресний простір.

Максимально допустимий для даного процесора кількість окремо адресуються осередків пам'яті і регістрів портів або контролерів називається адресним простором (АП) даного процесора. Адресний простір традиційно поділяють на системний (основний) АП та адресний простір вводу-виводу (АПВВ), маючи на увазі під цим не тільки відмінність в схемотехнічній побудові мікросхем пам'яті і портів ВВ, а й різні форми адресації і протоколи передачі.

Адресу комірки пам'яті або регістра порту, до якого проводиться звернення, процесор вказує у вигляді коду, виставленого на шину адреси (ША). Обсяг адресного простору і потенційно можливу кількість адрес комірок в системі пов'язані з розрядністю m шини адреси процесора співвідношенням. $N_{АП} = 2^m$.

При цьому набір можливих значень адрес лежить в діапазоні $0..2^m$. Обсяг АП, допустимий за розрядності ША, може бути розширений за рахунок використання керуючих сигналів, що дозволяє одночасно мати в системі основне АП та АПВВ, діапазони адрес яких накладаються. Фізичні адреси - це реальні, апаратні адреси, доступні в системі. Якщо в системі є 64 Мб пам'яті, в ній допустимі фізичні адреси можуть перебувати в діапазоні від 0 до 0x3fffffff (у шістнадцятковому форматі). Кожен адреса відповідає одному набору транзисторів у мікросхемах і окремому поєднанню сигналів на адресній шині процесора. Процесор перетворює логічні адреси в лінійні адреси із застосуванням способу, залежного від архітектури. В архітектурі x86 це перетворення передбачає просте додавання віртуальної адреси з іншою адресою - базовою адресою сегменту процесу; оскільки цей базовий адресу встановлюється рівним 0 для будь-якої задачі, логічні адреси та лінійні адреси у цій архітектурі однакові. Потім результуючий лінійний адресу перетвориться в фізичну адресу для взаємодії з оперативною пам'яттю системи.

Комп'ютери, які використовують процесор i80286 з 24-розрядними адресними шинами, фізично можуть адресувати 16 Мбайт, а в разі процесорів i80386/486 - 4 Гб пам'яті. Така можливість є тільки для захищеного режиму роботи процесора, який операційна система MS-DOS не підтримує. Розширена пам'ять (extended) розташовується вище області адрес 1 Мбайт (не треба плутати 1 Мбайт ОЗУ і 1 Мбайт адресного простору). Для роботи з розширеною пам'яттю мікропроцесор повинен переходити з реального в захищений режим і назад. На відміну від i80286 мікропроцесори i80386/486 виконують цю операцію досить просто, саме тому для них у складі MS-DOS є спеціальний драйвер - менеджер пам'яті EMM386 (див. рис. 2).

До речі, при наявності відповідного драйвера розширену пам'ять можна емулювати як додаткову. Апаратну підтримку в цьому випадку повинен забезпечувати мікропроцесор не нижче i80386 або допоміжний набір спеціальних мікросхем (наприклад, набори NEAT фірми Chips and Technologies). Слід зауважити, що багато плати пам'яті, що підтримують стандарт LIM / EMS, можуть використовуватися також і як розширеної пам'яті.

Майже на всіх персональних комп'ютерах область пам'яті UMB рідко виявляється заповненою повністю. Пустує, як правило, область розширення системного ROM BIOS або частину відеопам'яті і області під додаткові модулі ROM. На цьому й базується специфікація додаткової пам'яті EMS (Expanded Memory Specification), вперше розроблена фірмами Lotus Development, Intel і Microsoft (тому звана іноді LIM-специфікацією). Ця специфікація дозволяє використовувати оперативну пам'ять понад стандартних 640 Кбайт для прикладних програм. Принцип використання додаткової пам'яті заснований на перемиканні блоків (сторінок) пам'яті. В області UMB, між відеобуфером і системним ROM BIOS, виділяється незайняте 64-Кбайтних "вікно", яке розбите на сторінки. Програмні та апаратні засоби дозволяють відображати будь-який сегмент додаткової пам'яті в будь-яку з виділених сторінок "вікна (ТМ). Хоча мікропроцесор завжди звертається до даних, збереженим у "вікні" (адреса нижче 1 Мбайта), адреси цих даних можуть бути зміщені в додатковій пам'яті щодо "вікна" на кілька мегабайт.

У комп'ютерах на процесорі i8088 для реалізації додаткової пам'яті повинні застосовуватися спеціальні плати з апаратною підтримкою "підкачки" блоків (сторінок) пам'яті і відповідний програмний драйвер. Зрозуміло, плати додаткової пам'яті можуть встановлюватися і в комп'ютер на базі процесорів i80286 і вище.

В області верхньої пам'яті UMA зазвичай розташовуються пристрої з повільною пам'яттю: системна BIOS (System ROM BIOS), розширення BIOS на графічний адаптер (Video ROM BIOS), на контролерах дисків і інтерфейсів (Adapter ROM), ПЗУ початкового завантаження на мережевої карти (Boot ROM), відеопам'ять (Video Memory Buffer). Вони, як правило, реалізовані на 8- або 16-бітних мікросхемах з досить великим часом доступу. Звернення до повільній системній ОЗУ виконується набагато швидше. Для прискорення звернень до пам'яті цих пристроїв застосовується тіньова пам'ять (Shadow Memory) - підміна її системним ОЗУ. Тіньова пам'ять з'явилася на розвинених моделях AT-286, де вона була реалізована апаратно. Процесори класу 386+ дозволяють її реалізувати програмно, за допомогою сторінкової переадресації. Затінення ОЗУ і ПЗУ виконується по-різному. При ініціалізації тіньового ПЗУ (Shadow ROM) вміст затінюють області копіюється в ОЗУ, і при подальшому читанні за цими адресами підставляється ОЗУ, а запис в цю область блокується. При використанні тіньового ОЗУ (Shadow RAM) запис проводиться одночасно в фізичну пам'ять затінюють області і в системне ОЗУ, накладене на цю область. При читанні затіненій області звернення йде тільки до системної пам'яті, що відбувається набагато швидше. Особливо великий ефект від затінення відеопам'яті старих графічних адаптерів, яка з читання буває доступна тільки під час зворотного ходу розгортки, і процесору доводиться довго чекати цього моменту. Однак затінення областей пам'яті, що модифікуються з боку адаптерів, неприпустимо - ці зміни не будуть сприйняті процесором. До розділяється відноситься буферна пам'ять мережевих адаптерів, відеопам'ять адаптерів з графічними сопроцесорами (акселераторами). З цього випливає, що затінення відеопам'яті може бути застосовано тільки до примітивних графічних карт, які встановлюються у слот ISA, і то не у всіх режимах. Зазвичай тіньова пам'ять включається через CMOS Setup окремими областями розміром по 16 Кбайт або більшими, і для кожної області вказують режим затінення (Shadow ROM або Shadow RAM). Можливо її включення і драйверами ОС (наприклад, драйвером EMM386). На сучасних системних платах затінення області системної BIOS виконується завжди, на старих платах затінюванням цієї галузі можна було керувати.

Затінення BIOS відеоадаптера (Video BIOS Shadowing) для роботи в середовищі Windows з «рідними» драйверами графічного адаптера може і не давати приріст продуктивності.

Пристрої комп'ютера, що запам'ятовують, розділяють, як мінімум, на два рівні: основну (головну, оперативну, фізичну) і вторинну (зовнішню) пам'ять.

Основна пам'ять є впорядкованим масивом однобайтових осередків, кожна з яких має свою унікальну адресу (номер). Процесор витягує команду з основної пам'яті, декодує і виконує її. Для виконання команди можуть потрібно звернення ще до декількох елементів основної пам'яті. Зазвичай основна пам'ять виготовляється із застосуванням напівпровідникових технологій і втрачає свій вміст при відключенні живлення.

Багаторівневу схему використовують таким чином. Інформація, яка знаходиться в пам'яті верхнього рівня, зазвичай зберігається також на рівнях з великими номерами. Якщо процесор не виявляє потрібну інформацію на їх рівні, він починає шукати її на наступних рівнях. Коли потрібна інформація знайдена, вона переноситься в швидші рівні.

Виявляється, при такому способі організації у міру зниження швидкості доступу до рівня пам'яті знижується також і частота звернень до нього.

Адреси в основній пам'яті, що характеризують реальне розташування даних у фізичній пам'яті, називаються фізичними адресами. Набір фізичних адрес, з яким працює програма, називають фізичним адресним простором.

У мікросхемах ОЗП динамічного типу елемент пам'яті – це конденсатор р-п переходу МДН-транзистора. Заряджений конденсатор – вважають логічною 1, розряджений – 0. Такі елементи пам'яті не можуть тривалий час зберігати свій стан і тому потребують періодичного відновлення (регенерації) інформації. Час вибірки динамічного ОЗП становить 70-200 нс. Порівняно з ОЗП статичного типу ОЗП динамічного типу характеризується більшою інформаційною ємністю, меншою швидкодією, меншою потужністю споживання, меншою вартістю. Переважно модулі оперативної пам'яті сучасних МПС реалізують на базі ВІС ОЗП динамічного типу.

ОЗП динамічного типу працюють у таких режимах: запису, зчитування, зчитування/модифікації/запису, сторінкового запису, сторінкового зчитування, регенерації. Для забезпечення адресування використовують мультиплексування адресних сигналів у часі. Спочатку відбувається зчитування або запис даних, адреса яких визначається m молодшими розрядами ША, що супроводжується сигналом стробу (Row Address Strobe), який надходить на дешифратор рядків. Після чого відбувається зчитування або запис даних, адреса яких визначається m старшими розрядами ША, що супроводжується сигналом строба (Column Address Strobe) який надходить на дешифратор стовпців. Зчитування інформації відбувається за заднім фронтом сигналу.

У зв'язку з особливостями роботи динамічної пам'яті для скорочення часу доступу до неї застосовуються спеціальні режими роботи оперативної пам'яті: режим розшарування (інтерлівінг) і сторінковий режим.

Використовування режиму інтерлівінгу припускає не зовсім звичайне розбиття пам'яті на банки (частини). Якщо при звичайному розбитті (послідовної адресації) адреси наступного банку починаються після закінчення адрес попереднього, то при інтерлівінгу адреси банків чергують. Тобто, наприклад, після першої адреси першого банку слідує перша адреса другого банку, потім друга адреса першого банку і друга адреса другого банку і т.д. Виходить, що в одному банку парні слова, а в іншому - непарні. Таких чергуючих банків може бути не тільки два, а чотири, вісім, шістнадцять. Об'єми банків при цьому повинні бути однаковими. В результаті такого підходу з'являється можливість починати звернення до наступного слова ще до закінчення процесу доступу до попереднього.

Сторінковий режим припускає посторінкову роботу мікросхем пам'яті, коли вибір сторінки проводиться один раз на всю сторінку, а вибір комірки усередині сторінки може відбуватися набагато швидше. Для підтримки обох режимів застосовуються спеціальні технологічні рішення.

Тут же відзначимо, що при заміні пам'яті комп'ютера або при установці додаткових банків пам'яті треба строго слідувати рекомендаціям виготівників системної плати, оскільки порядок заповнення банків може не бути далеко очевидний.

Типи динамічної пам'яті: асинхронна, синхронна.

1. Асинхронна пам'ять.

FPM DRAM – першою знайшла застосування в ПК модифікацією DRAM, працювала в режимі швидкого сторінкового обміну FPM (Fast Page Mode) або режимі сторінкового обміну (Page Mode). Щоб скоротити час очікування на вибірку даних (на доступ) стандартна пам'ять DRAM розбивається на сторінки. Звичайно для доступу до даних у пам'яті вимагається вказати рядок і стовпець адреси, що займає якийсь час.

EDO DRAM – цей режим іноді називають гіперсторінковим режимом обміну HPM (Hyper Page mode). Це вдосконалений тип пам'яті FPM. Мікросхеми пам'яті EDO враховують перекриття синхронізації між черговими операціями доступу.

BEDO DRAM – у мікросхемах даного типу додатково міститься внутрішній лічильник адреси колонок для пакетного циклу.

2. Синхронна пам'ять.

SDRAM (Synchronous DRAM) – синхронна DRAM. Ефективність SDRAM набагато вища, ніж у її попередників.

DDR SDRAM – (Dual Data Rate – подвоєна швидкість даних) є подальшим розвитком пам'яті, яка за багатьма параметрами та способами виготовлення мало чим відрізняється від звичної пам'яті SDRAM. У мікросхем DDR SDRAM дані усередині пакету передаються з подвоєною швидкістю – вони перемикаються по обох фронтах синхроімпульсів. На частоті 100 МГц DDR SDRAM мають пікову продуктивність 200 Мбит/пін, що у складі 8-байтних модулів DIMM дає продуктивність 1600 Мбайт/с. На високих тактових частотах (100 - 133 МГц) подвійна синхронізація висуває дуже високі вимоги до точності тимчасових діаграм DR DRAM (Rambus).

VC DRAM – ідея архітектури пам'яті з віртуальними каналами (Virtual Channel Memory Architecture), яка полягає в розташуванні між масивом осередків, що запам'ятовують, і зовнішнім інтерфейсом мікросхеми пам'яті набору каналних буферів. При цьому операції обміну даними розділяються на два процеси: “фасадний” обмін даними з каналами і “тиловий” обмін між каналами та масивом осередків, що запам'ятовують. Обидва процеси виконуються по командам з боку зовнішнього інтерфейсу, майже незалежно один від одного.

Динамічна пам'ять найчастіше застосовується у вигляді модулів з розрядністю 1, 2, 4 або 8 байт, які можуть встановлюватися користувачем без яких-небудь додаткових пристроїв. Модулі стандартизовані, тому забезпечується взаємна сумісність.

◆ SIPP і SIMM-30 — найперші модулі з однобайтною організацією, застосовувалися аж до процесорів класу 486.

◆ SIMM-72-pin — 4-байтні модулі, що застосовувалися на системних платах для процесорів класів 486 і Pentium.

◆ DIMM-168 — 8-байтні модулі для Pentium і вище. Існують два покоління, істотно різні по інтерфейсу. Модулі DIMM 168-pin Buffered (1-го покоління), як і слоти для них, зустрічаються рідко і з широко поширеними модулями DIMM 2-го покоління несумісні навіть механічно (по ключах). Найбільш популярне друге покоління з мікросхемами SDRAM. Розрізняють модифікації залежно від наявності буферів або регістрів на управляючих сигналах: Unbuffered, Buffered і Registered.

◆ DIMM-184 — 8-байтні модулі DDR SDRAM для системних плат 6-8-го поколінь процесорів.

◆ DIMM-240 — 8-байтні модулі DDR2 SDRAM для системних плат 7-8-го покоління процесорів.

◆ RIMM — 2-байтні модулі RDRAM для системних плат 6-7-го покоління процесорів.

◆ 50 DIMM (72-, 144- і 200-pin) і SO RIMM — малогабаритні варіанти модулів (для блокнотних ПК).

◆ A1MM (AGP Inline Memory Module), вони ж GPA Card (GPA розшифровується як Graphics Performance Accelerator) — 66-контактні 32- або 16-бітні модулі SDRAM, призначені для розширення пам'яті графічних адаптерів, вбудованих в системну плату.

Не перераховуючи контакти, відрізнити «короткі» модулі SIMM від «довгих» і від модулів DIMM легко за їх розміром: довжина модуля SIMM-30 pin — приблизно 89 мм, SIMM-72 - 108 мм. Модулі DIMM-168, DIMM-184 і DIMM-240 мають однакову довжину близько 134 мм, але у 168-контактних модулів два ключі, у 184- і 240-контактних — один (за рахунок чого більше контактів); крім того, у DIMM-184 і DIMM-240 по два прорізи з боків, а не по одному. У модулів DIMM-240 крок контактів дрібніший (1 мм), ніж у DIMM-184 (1,27 мм), і ключі живлення розташовані вище. Модулі RIMM мають ту ж довжину, але їх легко відрізнити по меншому числу контактів. У модулів RIMM мікросхеми пам'яті закриті пластиною радіатора. Крім того, їх лівий ключ набагато ближчий до центру, ніж у DIMM.

У комп'ютерах, призначених для використання як сервери або могутні станції, нерідко застосовується спеціальні плати пам'яті, що дозволяє встановлювати великі об'єми ОЗП. На таку плату також встановлюються модулі SIMM, DIMM або SO DIMM. Модулі пам'яті застосовуються і в принтерах (лазерних) - DIMM-168, 100-Pin DIMM, A1MM, SO DIMM-144, але іноді для них потрібні і спеціальні (по конструктиву або параметрам) модулі.

Дискові пристрої зовнішньої пам'яті.

Фізична будова жорстких і гнучких дисків. Типи оптичних дисків.

Основним компонентом жорсткого диска є одна або декілька круглих пластинок виготовлених з алюмінію або скла і покриті магнітним шаром, що наноситься з обох сторін пластини. Саме на цих пластинках і зберігається вся інформація. Диски закріплені на загальній осі і обертаються з великою швидкістю (для сучасних – 5400 або 7200 об/хв). Крім того, в корпусі жорсткого диску є блок магнітних головок, які здійснюють читання і запис з поверхонь дисків. Всі головки сполучені разом і не можуть рухатися окремо, тому запис і читання проводяться відразу з усіх поверхонь дисків одночасно.

Диски і головки знаходяться на металічному шасі, що забезпечує жорсткість всієї конструкції, та закриті кришкою, що оберігає поверхню дисків і головку від потрапляння пилу. Дуже важливий надійний захист блоку дисків і головок, адже під час роботи диска, при обертанні пластин, створюється потужний потік повітря і магнітні головки під час читання-запису "парять" над поверхнями дисків не торкаючись їх, а зазор між дисками і головками складає декілька мікрометрів. Блок головок-дисків не містить ніякої електроніки, а підключається до спеціальної плати, на якій знаходиться керуюча електроніка, тобто контролер.

Кожний диск поділяється на треки (Track), концентричні кільця. Крім того, диск розділяється на сектори, області перетину секторів і доріжок називають блоками (block), в блоці зберігається 512 байт корисної інформації.

Зауваження: кожен блок на диску займає 512 байт, з яких під дані відводиться лише 512 байт.

Кількість блоків на одній пластині рівна добутку кількості секторів на кількість доріжок. Кількість сторін, що використовуються, рівна кількості магнітних головок (head) на диску. Загальна формула для розрахунку кількості блоків на диску має вигляд: кількість доріжок * кількість секторів * кількість головок. А об'єм диску рівний кількості блоків помноженому на об'єм одного блоку, тобто на 512 байт. Єдине зауваження: оскільки магнітні головки жорстко скріплені одна з одною, то запис проводиться на всі доріжки всіх сторін одночасно. Отже, замість поняття "доріжка" слід вводити поняття, що описує всі рівновіддалені від центру доріжки на всіх сторонах всіх пластин: сукупність таких доріжок за такими міркуваннями називається циліндром (cylinder).

Разом об'єм жорсткого диска = cylinder * sector * head * 512 byte.

Звичайно дискета являє собою гнучкі пластинку, покриту феромагнітним шаром, звідси англійська назва floppy disk («гнучкий диск»). Ця пластинка містить в пластмасовому корпусі, що захищає магнітний шар від фізичних ушкоджень. Оболонка буває гнучкою або міцною. Запис і зчитування дискет здійснюється за допомогою спеціального пристрою – дисководу (флоп-дисководу).

Дискети звичайно мають функцію захисту від запису, за допомогою якої можна надати доступ до даних тільки в режимі читання.

Дискети зручні у використанні та доступні, але виходять з використання через свою відносну ненадійність – розмагнічування та неможливість відтворення інформації при найменшій руйнації поверхні.

ГМД або дискети широко використовуються в ПК, як для збереження невеликої об'ємів інформації, так і для перенесення даних з одного ПК на інший.

На перших ПК використовувались дискети діаметром магнітного диску 5,25 дюйма (133 мм) з одностороннім магнітним покриттям і об'ємом 160 К. Їх попередником були диски розміром диску, що дозволило збільшити об'єм диску до 1,2 м.

З часом з'явилися дискети з діаметром магнітного диску 3,5 (89 мм), який був розміщений в жорсткому корпусі. Це значно підвищило їх стійкість до механічних пошкоджень. Саме завдяки

цьому такі ГМД є найбільш популярними. Оскільки поверхня ГМД досить чутлива до зовнішніх впливів, то потрібно дотримуватись таких основних правил при роботі з ними:

- не доторкатись відкритої поверхні диску;
- не згинати диск;
- не залишати диск біля джерел електромагнітних і теплових випромінювань;
- зберігати диск від потрапляння його поверхню води, пилу.

Дискета 3,5 має отвір для ініціалізації об'єму диску отвір захисту інформації гнучких диск захисна шорка, яка в закритому положенні захищає поверхню диску у вікні зчитування-запису інформації конверт.

Дискети на 3,5 за об'ємом інформації, класифікують на:

- DD і подвійної щільності на 720 К;
- HD і високої щільності на 1,44 М;
- VHD і надвисокої щільності на 2,8 М.

ГМД типу VHD вперше були розроблені фірмою Toshiba, а надалі її почали випускати фірми Teac, Sony, Chinon. На цих дискетах густина запису була збільшена вдвоє за рахунок використання так званого перпендикулярного методу запису.

Для запису і зчитування інформації з ГМД використовують спеціальні приводи і дисководи. Розглянемо тепер будову гнучкого магнітного диску. Поверхня будь-якого ГМД, умовно розбита на концентричні кола, які називають доріжками або треками, та сектори, що ділять доріжки на сегменти. Сучасні ГМД містять від 40 до 83 доріжки та від 9 до 21 секторів. Один сектор на диску завжди має однаковий об'єм, що дорівнює 512 байт, незалежно від розмірів самого диску. Нумерація доріжок проводиться від зовнішньої, що позначається, як 0-доріжка і зростає до середини диску. Важно заметить, що для секторів нумерація починається з 1, від синхронізаційного отвору і проводиться проти годинникової стрілки. Для двосторонніх дисків вводиться поняття сторін, які нумеруються 0 та 1.

Отже відлік фізичних координат на диску проводиться за принципом:

<сторона> <доріжка> <сектор>.

Запис 0;0;1 і означає, що це 0-ва сторона, 0-ва доріжка та 1-ший сектор.

Крім фізичних координат існують логічні координати, які називають кластерами.

Compact Disc (CD)

Першим оптичним носієм, розробленим ще в 1979 році, став компакт-диск (англ. Compact Disc, CD). Глибина пітів цього носія становить близько 100 нм, ширина - 500 нм. Мінімальна довжина піта CD - від 850 нм. Крок між доріжками спіралі - близько 1,5 мкм. У приводах для читання носіїв цього типу використовується червоний лазер з довжиною хвилі 780 нм, який фокусується на робочій поверхні в точку діаметром близько 1,2 мкм (для кращої уяви: товщина людської волосини становить близько 50 мкм або 50000 нм).

Спочатку CD створювався як носій звукової інформації (Audio CD). Трохи пізніше його почали використовувати для зберігання інших даних. Формат запису Audio CD і CD з даними відрізняється, тому звичайні програвачі не можуть зчитувати інформацію з немюзичних компакт-дисків.

Перші диски були CD-ROM (Read Only Memory - тільки з можливістю читання) і виготовлялися за спеціальною технологією (можливість запису в домашніх умовах відсутня). Пізніше розробили носії, в яких піти являли собою не заглиблення, а ділянки модифікованої поверхні з високим світлопоглинанням, що виникають під спеціальним впливом лазера. Як і у випадку з заглибленими пітами, технологія забезпечувала потрібну зміну інтенсивності відображення лазера і зчитування даних, закодованих у доріжці. Завдяки цьому з'явилися диски типів CD-R і CD-RW (з можливістю запису і неодноразової перезапису користувачами відповідно).

DVD - носій, який з'явився дещо пізніше, має той же фізичний розмір і принципи роботи, що і CD, але використовує дуже щільну робочу поверхню. За рахунок більш "мініатюрної" структури доріжки і тоншого червоного лазера з довжиною хвилі всього 650 нм, DVD здатний зберігати в кілька разів більше даних.

Спочатку DVD розроблявся як альтернатива морально застарілим відеокасетами для зберігання відеоінформації і тому називався Digital Video Disc (цифровий відеодиск). Але носії цього виду чудово підійшли і для запису інших типів даних. Аббревіатуру DVD при цьому почали трактувати як Digital Versatile Disc - цифровий багатоцільовий диск. Але така розшифровка не є офіційною.

В залежності від виду даних, що зберігаються, DVD бувають наступних типів: DVD відео-, DVD-Audio (якість звуку значно вище, ніж на Audio-CD), DVD-Data, DVD змішаного типу.

Сучасні DVD можуть мати одну або дві робочі сторони. При цьому кожна зі сторін може містити один або навіть два робочих шари. Завдяки цій особливості, на DVD максимального обсягу можна записати до 18 ГБ інформації.

DVD і CD, бувають для одноразової запису, або ж можуть перезаписуватися багаторазово (R і RW). При цьому, в залежності від особливостей використовуваного матеріалу і технології запису, розрізняють DVD-R(RW) , DVD+R(RW). У технічні подробиці не вдаватимуся. Скажу тільки, що DVD+R(RW) і приводи з їх підтримкою - кращий і безпроблемніший варіант. При записуванні таких носіїв виникає набагато менше помилок. Крім того, DVD+RW можна записувати поверх вже наявних на ньому даних. У випадку ж з DVD-RW попередньо потрібно провести повне очищення диска.

Blu-ray Disc (BD, англ. blue ray - синій промінь) - оптичний носій нового типу, що відрізняється ще більш високою щільністю робочої поверхні і забезпечує зберігання великого обсягу даних, у тому числі і високоякісного відео високої чіткості. У назві Blu-ray буква "e" була пропущена навмисне, щоб мати можливість зареєструвати торговий знак.

Запис і зчитування BD проводиться за допомогою надтонкого синього лазера з дуже короткою хвилею (405 нм), завдяки чому ширина доріжки на диску звужена до 0,32 мкм, що вдвічі менше, ніж на DVD. Збільшився не тільки обсяг даних, що зберігаються, але і швидкість їх зчитування. У кілька разів тоншим став і захисний шар, але завдяки використанню нових матеріалів, Blu-ray Disc боїться подряпин не більше, ніж оптичні носії інших типів.

Сучасні технології дозволяють записувати на один BD кілька шарів, кожний з яких може містити 23,3 ГБ даних. Поширеними є диски з кількістю шарів до 4. Хоча, вже створено прототипи ємністю до 500 ГБ, що містять 15-20 шарів.

Blu-ray носії продовжують розвиватися і вдосконалюватися. Існують диски для одноразового і багаторазового записування. Розроблена технологія, що дозволяє наносити на один диск шари, призначені для запису DVD і Blu-Ray.

Зовнішні інтерфейси.

Послідовний інтерфейс СОМР-порт. Програмна модель СОМР-порту. Паралельний LPT-порт. Режими роботи. Універсальна послідовна шина USB. Архітектура і взаємодія елементів системи USB. Типи передач і формати інформації, яка передається через шину USB.

Інтерфейс RS-232C призначений для підключення до комп'ютера стандартних зовнішніх пристроїв (принтера, сканера, модему, миші і ін.), а також для зв'язку комп'ютерів між собою. Основними перевагами використання RS-232C в порівнянні з Centronics є можливість передачі на великі відстані (за стандартом довжина сполучного кабелю може доходити до 15 метрів) і набагато більш простий кабель (з меншою кількістю дротів). В той же час працювати з RS-232C дещо складніше. Дані в інтерфейсі RS-232C передаються в послідовному коді (біт за бітом) побайтно. Кожний байт обрамляється стартовим і стоповими бітами. Дані можуть передаватися як в одну, так і в іншу сторону по різних дротах (дуплексний режим). Швидкість передачі - до 14,4 Кбайт/с (115,2 Кбіт/с).

Комп'ютер має 25-контактний роз'єм (типа DB25P) або 9-контактний роз'єм (типа DB9P) для підключення кабелю інтерфейсу RS-232C. Призначення контактів роз'єму приведено в табл. 8.8 (в таблиці застосовані позначення: І - вхідний сигнал комп'ютера, О - вихідний сигнал комп'ютера).

Комп'ютер, сумісний з IBM PC, може мати до чотирьох послідовних портів. Вони маркуються як СОМ1 - СОМ4. Кожен СОМ-порт формується окремим UART 16450, встановленим усередині комп'ютера /UART 8250/16450/. На рис. показана внутрішня структура UART. У ньому маютья вісім восьмибітових регістрів. Адреси введення/виведення цих регістрів обчислюються додаванням зсуву регістра до базової адреси СОМ-порту.

Зсуви і функції регістрів UART такі:

00h - буферний регістр передавача/буферний регістр приймача: використовується для обміну даними;

01h - регістр дозволу переривань: установлює режим запиту переривань;

02h - регістр ідентифікації переривань: перевіряє режим запиту переривань;

03h - регістр формату даних: встановлює формат послідовних даних;

04h - регістр керування модемом: установлює керування модемом (RTS, DTR і т.д.);

05h - регістр стану прийомопередавача: містить інформацію про стан приймача і передавача;

06h - регістр стану модему: містить поточний стан ліній DCD, RI, DSR і CTS;

07h - регістр надоперативної пам'яті: працює як байт пам'яті.

IEEE 1284 (порт принтера, паралельний порт, англ. Line Print Terminal, LPT) — міжнародний стандарт паралельного інтерфейсу для підключення периферійних пристроїв персонального комп'ютера.

В основному використовується для підключення до комп'ютера таких пристроїв як: принтер, сканер та інших зовнішніх пристроїв (часто використовувався для підключення зовнішніх пристроїв зберігання даних), проте може застосовуватися і для інших цілей (організація зв'язку між двома комп'ютерами, підключення будь-яких механізмів телесигналізації і телекерування).

В основі даного стандарту лежить інтерфейс Centronics і його розширені версії (ECP, EPP).

Назва «LPT» походить від назви стандартного пристрою принтера «LPT1» (Line Printer Terminal або Line PrinTer) в операційних системах сімейства MS-DOS.

Базовий інтерфейс Centronics є односпрямованим паралельним інтерфейсом, містить характерні для такого інтерфейсу сигнальні лінії (8 для передачі даних, строб, лінії стану пристрою).

Дані передаються в один бік: від комп'ютера до зовнішнього пристрою. Але повністю односпрямованим його назвати не можна. Так, 4 зворотні лінії використовуються для контролю за станом пристрою. Centronics дозволяє підключати один пристрій, тому для спільного чергового використання декількох пристроїв потрібно додатково застосовувати селектор.

Швидкість передачі даних може варіюватися і досягати 1,2 Мбіт/с.

Стандарт дозволяє використовувати інтерфейс в декількох режимах:

SPP (Standart Paralell Port) — односпрямований порт, повністю сумісний з інтерфейсом Centronics.

Nibble Mode — дозволяє організувати двонаправлений обмін даними в режимі SPP шляхом використання керуючих ліній (4 біт) для передачі даних від периферійного пристрою до контролера. Історично це був єдиний спосіб використовувати Centronics для двостороннього обміну даними.

Byte Mode — рідко використовуваний режим двостороннього обміну даними. Використовувався в деяких старих контролерах до прийняття стандарту IEEE 1284.

EPP (Enhanced Parallel Port) — розроблений компаніями Intel, Xircom та Zenith Data Systems — двонаправлений порт, зі швидкістю передачі даних до 2МБайт/сек. (1991)

ECP (Extended Capabilities Port) — розроблений компаніями Hewlett-Packard та Microsoft — на додаток з'явилися такі можливості, як наявність апаратного стиснення даних, наявність буфера і можливість роботи в режимі DMA.

USB (англ. Universal Serial Bus) — універсальна послідовна шина, призначена для з'єднання периферійних пристроїв обчислювальної техніки. Символом USB є чотири геометричні фігури: квадрат, трикутник, велике коло та мале коло.

Стандарт USB розробили сім компаній: «Compaq», «Digital Equipment», IBM, Intel, «Microsoft», NEC і «Northern Telecom». З листопада 1994 до листопада 1995 року було анонсовано кілька версій протоколу (USB 0.7, 0.8, 0.9, 0.99, 1.0 Release Candidate). Влітку 1996 року на ринку з'явилися перші комп'ютери з портами USB.

Шина USB — послідовний інтерфейс передавання даних для середньо- та низько швидкісних периферійних пристроїв. Для високошвидкісних пристроїв кращою вважалася шина FireWire, хоча з випуском пристроїв на базі USB 3.0 це твердження стало сумнівним.

USB-кабель — це, по суті, дві звиті пари: однією з них передаються дані в кожному напрямку (диференціальне включення), а інша використовується для живлення периферійного пристрою (+5 В, 500 мА). Вбудовані лінії живлення дозволяють використовувати USB-пристрої, що не мають власного блоку живлення, чи заряджати акумулятори переносних пристроїв (фото- та відеокамер, плеєрів тощо), якщо ці пристрої споживають струм силою до 500 мА. Стандарт USB 3.0 допускає навантаження лінії живлення струмом до 900 мА.

З'єднання USB-кабелями формує інтерфейс між USB-пристроями та USB-хостом. Як хост використовується керований з операційної системи USB-контролер, до складу якого входить USB-концентратор, або ж хаб. Цей хаб є відправною точкою в створенні ланцюжка пристроїв, що відповідають вимогами топології «зірка». Він має спеціальну назву — кореневий концентратор. До роз'ємів його портів під'єднується інше USB-приладдя та зовнішні хаби. Загальні їх кількість не може перевищувати 127 пристроїв, увімкнених не більш ніж у п'ять каскадів, не рахуючи рівень кореневого хаба.

Конструкція USB-конекторів розрахована на «гаряче» приєднання та від'єднання пристроїв до хоста (від нього). Це забезпечено більшою довжиною контакту заземлення GND проти інших. Внаслідок цього потенціали корпусів вирівнюються ще до замикання сигнальних контактів, а це забезпечує електроніку приладу від пошкодження статичною електрикою.

Фізичне з'єднання пристроїв здійснюється по топології багатоярусної зірки. Центром кожної зірки є хаб, кожен кабельний сегмент з'єднує дві точки - хаб з іншим хабом або хаб з функцією. В системі USB є тільки один хост-контролер, розташований у вершині піраміди пристроїв і хабів USB. Хост-контролер інтегрується з кореневим хабом (root hub), що забезпечує одну або кілька точок підключення - портів. Контролер USB, що входить до складу чіпсетів багатьох сучасних системних плат зазвичай має двухпортовий хаб.

Логічно пристрій підключеної до будь-якого хабу може розглядатися як підключене безпосередньо до хост-контролера.

"Функції" являють собою пристрої USB, здатний приймати або передавати дані або керуючу інформацію по шині. Фізично в одному корпусі може бути кілька "функцій" з вбудованим хабом забезпечує їх підключення до одного порту

Кожна "Функція" надає конфігураційну інформацію, що описує його можливості та вимоги до ресурсів. Перед використанням функція повинна бути сконфігурована хостом - їй має бути виділена смуга в каналі обрані специфічні опції конфігурації.

Хаб - ключовий елемент системи Plug-and-Play в архітектурі USB. Хаб є кабельним концентратором, точки підключення називаються портами хаба. Кожен хаб перетворює одну точку підключення в їх безліч. Архітектура увазі можливість з'єднання декількох хабів.

У кожного хаба є один вихідний порт (upstream port), призначений для підключення до хосту і Чи до хабу верхнього рівня. Решта порти є спадними (downstream) і призначені для підключення функцій та хабів нижнього рівня. Хаб може розпізнати підключення або відключення пристроїв до цих портів і управляти подачею живлення на їх сегменти. Кожен з цих портів індивідуально може бути дозволений або заборонений і конфігурований на повну або обмежену швидкість обміну. Хаб забезпечує ізоляцію сегментів з низькою швидкістю від високошвидкісних.

Хаби можуть мати можливість управління подачею живлення на низхідні порти, передбачена керування установка обмеження на струм, споживаний кожним портом.

Система USB розділяється на три рівні з певними правилами взаємодії. Пристрій USB ділиться на інтерфейсну частину, частину пристрою і функціональну частину. Хост теж ділиться на три частини - інтерфейсну, системну і ПЗ пристрою. Кожна частина відповідає тільки за певне коло завдань.

Архітектура USB включає в себе чотири базисних типу передач даних:

- Передачі керуючої інформації, яка використовується для конфігурації пристрою під час приєднання і може використовуватися іншим пристроєм для специфічних цілей;
- Передачі даних типу Bulk, генеруються або використовуються при відносно великих обсягах інформації. Виконується контроль правильності прийому з можливістю повторних передач;
- Передачі типу переривання (короткі повідомлення в кілька байт - введений символ або координати покажчиків) з підтвердженням прийому за лімітоване час;
- Ізохронний або потокові передачі даних в реальному часі, які займають заздалегідь обумовлену пропускну здатність USB шини. У даному типі передачі не підтримується механізм підтвердження правильності прийому і повторних передач. Дані передаються з допустимим рівнем помилок. Типовими даними ізохронних передач є передача оцифрованих звукових сигналів.

Клієнтське програмне забезпечення зазвичай запитує передачу даних в каналі за допомогою пакетів запиту вводу / виводу (IRPs) і потім або чекає або повідомляє що вони завершені.

Потоки в каналах поставляють дані як частину пакета даних транзакцій шини не наводячи зміст даних до структури необхідної USB. Дані надійшли в один кінець потоку в каналі, виходять з іншого боку в тому ж самому порядку (FIFO). Потоки в каналах - завжди спрямовані в одну сторону. Потоки в каналі підтримують такі типи передачі bulk, ізохронний, і переривання.

Канали повідомлень взаємодіють з кінцевою точкою не так, як потоки в каналах. Спочатку, хост посилає запит пристрою USB. Цей запит супроводжується передачею (ами) даних у відповідному напрямку. Після цього зчитується стан, що є відповіддю від кінцевої точки.

Щоб розмістити зразок запиту / даних / стану, канали повідомлень роблять потік зв'язку структурованим, що дозволяє командам бути достовірно переданими і ідентифікованими.

Канали повідомлень дозволяють потоку зв'язку мати два напрямки. Створюваний за замовчуванням канал з кінцевої точки 0, є завжди каналом повідомлень.

Основи побудови сайтів. Основи мови HTML.

Основні вимоги процесу побудови сайтів. Основи мови гіпертекстової розмітки. Основні поняття HTML. Основна структура html-документу. Робота з текстом.

Використання електронних носіїв даних, електронних джерел, спілкування в мережі, соціальних мереж та інших сервісів Інтернет є на даний час актуальним та, можна вважати, стало звичною справою. Велика кількість матеріалів та даних представлено на сайтах в глобальній мережі. Створити сайт на сьогодні є справою нескладною та нетривалою. Проте необхідно враховувати певні вимоги до його створення, щоб сайт та представлення даних на ньому було не просто наповненням Web-сторінки, але й мало прийнятну для різних відвідувачів структуру. При розробці сайтів необхідно визначити тему, напрям сайту та його цільову аудиторію. Важливими показниками сайту є його фізична структура, логічна структура, внутрішня та зовнішня оптимізація. Кожний з цих показників забезпечує оптимальні зручність використання, інформативність тексту та наочності, сприятливість кольорового оформлення, неперевантаження комп'ютерної графіки, аудіо та відео об'єктів та інше.

Робота в глобальній мережі Інтернет проходить у веб-сторінках за допомогою спеціальних програм – браузерів. Документи, що містять код розмітки Web-сторінки та зберігаються з розширенням .html або .htm, називають html-документами. Такі документи опрацьовуються браузерами і виводяться на екран як Web-сторінки. Вказані документи створюються в звичайних текстових редакторах або в спеціальних програмних засобах. Набір кодів, призначених для розміщення об'єктів на Web-сторінці задають мовою HTML (Hyper Text Markup Language – мова гіпертекстової розмітки). Сайти – це сукупність Web-сторінок, що об'єднанні загальною темою і розміщені на одному вузловому комп'ютері чи в певному каталозі Web-сервера. На Web-сторінках розміщують текст, таблиці, списки, зображення, відео-фрагменти тощо. Все, що можна розмістити на Web-сторінці і описати мовою HTML, будемо називати об'єктами. Опис правил розміщення окремого об'єкта на Web-сторінці здійснюється за допомогою тегу. Тег – це код мови HTML, що впливає на результати відображення об'єкту. За правилами синтаксису мови тег поміщується у кутові дужки. Теги бувають парні та одинарні. Парні описують спосіб відображення об'єкту і містять між собою його елементи.

Приклад. Теги <I> Ласкаво просимо до міста Суми </I> - означають, що текст, який міститься між ними, буде написаний курсивом. Весь інший текст до та після цих тегів курсивом відобразатися не буде.

Пара тегів та об'єкт між ними називають контейнерами, елементами, дескрипторами. Якщо необхідно додати певні елементи оформлення окремих об'єктів, то використовують атрибути. Атрибут – додаткова вказівка до тегу, що може бути використана для різних об'єктів. Атрибут розміщується у початковому тегу та містить певне значення.

Приклад. Атрибут align визначає розташування об'єкту. Значенням цього атрибуту може бути center, left, right, top та інші. Такий атрибут може застосовуватись при опису тексту, малюнку, горизонтальної лінії, таблиці тощо.

Кожний html-документ містить стандартну структуру:

```
<HTML>
  <HEAD>
    <TITLE> Назва сторінки </TITLE>
  </HEAD>
<BODY>
Об'єкти, що розташовуються на сторінці
```

</BODY>

</HTML>

Теги:

<HTML>, </HTML> - визначають початок та кінець документа з гіпертекстовою розміткою.

<HEAD>, </HEAD> - визначають окрему частину html-документа, де описуються окремі компоненти Web-сторінки (її заголовки, дані про час створення, автора, загальні вимоги до стильового оформлення об'єктів).

<TITLE> Назва сторінки </TITLE> - виводить у відповідне місце Web-сторінки її назву

<BODY>, </BODY> - описують оформлення тексту, таблиць, зображень та інших об'єктів на сторінці, тобто те, що безпосередньо на ній розміщується.

В переважній більшості на сайтах розташовано текст, що супроводжується наочністю. Для розташування тексту спеціальних атрибутів вказувати не потрібно. Але для опису окремих його елементів (заголовків, розмір шрифту, його тип, відображення напівжирним шрифтом, курсивом, розташування по ширині, по центру, лівому чи правому краю та інше) необхідно вказувати спеціальні теги.

Тег	Його значення
 Текст 	Написання тексту напівжирним шрифтом
<I> Текст </I>	Написання тексту курсивом
<U> Текст </U>	Написання тексту підкресленим
<STRIKE> Текст </STRIKE>	Написання тексту перекресленим
^{Текст}	Написання тексту як верхнього індексу
_{Текст}	Написання тексту як нижнього індексу
<P>	Створення абзацу. Може бути парним або одинарним тегом
<H1> Текст </H1>	Задає заголовок тегу

Основи мови HTML.

Розміщення об'єктів на Web-сторінці: списки, таблиці, зображення, відео та аудіо-об'єкти. Форматування об'єктів.

Робота зі списками.

Списки на Web-сторінці використовують переважно трьох типів:

Невпорядкований (маркований)

Впорядкований (нумерований)

Список визначень

Кожен зі списків заключається в пару тегів і має власні атрибути.

Список	Теги	Атрибути	Приклад
впорядкований	<pre> </pre>	<p>type – тип лічильника</p> <p>type=A – нумерація великими латинськими літерами</p> <p>type=a – нумерація маленькими латинськими літерами</p> <p>type=I – нумерація великими римськими цифрами</p> <p>type= i – нумерація маленькими римськими цифрами</p> <p>type=1– нумерація арабськими цифрами</p> <p>start= число – початок нумерації</p>	<pre><OL type=1 start=5> мандарин апельсин помело </pre> <p><i>Результат:</i></p> <p>5 мандарин 6 апельсин 7 помело</p>
невпорядкований	<pre> </pre>	<p>type – тип лічильника</p> <p>type=disc – маркером є диск (●)</p> <p>type=circle– маркером є коло (○)</p> <p>type=square – маркером є квадрат (□)</p>	<pre><UL type=circle> мандарин апельсин помело </pre> <p><i>Результат:</i></p> <p>○ мандарин ○ апельсин ○ помело</p>
список визначень	<pre><DL> <DT> термін <DD> значення <DD>значення </DL></pre>		<pre><DL> <DT> кома <DD> знак пунктуації <DD> стан непритомності <DT> замок <DD> пристрій на дверях для обмеження доступу <DD> споруда, укріплена будівля </DL></pre> <p><i>Результат:</i></p>

			кома знак пунктуації стан неприготовності замок пристрій на дверях для обмеження доступу споруда, укріплена будівля
--	--	--	--

Робота з таблицями.

Тег	Атрибут	Приклад
<p><TABLE> ... </TABLE> - окреслює таблицю</p>	<p>width="n" - ширина в пікселях або % border="n" - товщина рамки bordercolor="#FFFFFF" - колір рамки bgcolor = "#FFFFFF" – колір фону cellspacing = "n" - відступ між рамками комірок cellpadding = "n" - відступ між рамкою і текстом align= "значення" – визначає розташування таблиці (left, center, right)</p>	<pre><TABLE border="4" bordercolor="#000000" cellpadding="0" cellspacing="0"> <TR> <TD width="10" height="10" bgcolor="#000000"></TD> <TD width="10" height="10" bgcolor="#ffffff"></TD> <TD width="10" height="10" bgcolor="#000000"></TD> </TR> <TR> <TD width="10" height="10" bgcolor="#ffffff"></TD> <TD width="10" height="10" bgcolor="#000000"></TD> <TD width="10" height="10" bgcolor="#ffffff"></TD> </TR> </TABLE></pre>
<p><TR>...</TR> - визначення рядка <TD>...</TD> - визначення комірки <TH>...</TH> - визначення заголовку таблиці</p>	<p>width="n" –ширина комірки в пікселях height="n" – висота комірки в пікселях bgcolor="#000000" –фоновий колір комірки background="image.gif" – заповнює фон комірки зображенням &nbsp; (non-breaking space – пробіл, що не розривається) Теги, що встановлюють шрифт (, <I>, <FONT SIZE="n", FONT COLOR="#FFFFFF"), необхідно повторювати для кожної комірки або групи комірок</p>	<pre><TR> <TD width="10" height="10" bgcolor="#ffffff"></TD> <TD width="10" height="10" bgcolor="#000000"></TD> <TD width="10" height="10" bgcolor="#ffffff"></TD> </TR> </TABLE></pre>
<p>COLSPAN="n" – розтягування комірки по горизонталі</p>	<p>Об'єднання комірок відбувається шляхом</p>	<pre><TD COLSPAN="2"> <TD ROWSPAN="7"></pre>

ROWSPAN="n" – розтягування комірки по вертикалі	розтягування їх по горизонталі та вертикалі	
<CAPTION> назва таблиці </CAPTION> - задає назву таблиці і розглядається з таблицею як нерозривний елемент назва таблиці задається після тегу <TABLE>.	align="значення" - визначає положення щодо таблиці значення можуть бути: top – значення за замовчуванням, заголовок над таблицею по центру; left, right – зліва та справа над таблицею bottom – заголовок під таблицею по центру	<CAPTION align=top> Приклади кольорів та їх кодування </CAPTION>

Вставка зображень, відео та аудіо фрагментів

Тег	Атрибут	Приклад
Створення фонового зображення прописується як атрибут тегу BODY	<BODY background="images.gif">	<BODY background="рис1.gif">
 - вставка зображення в певному місці сторінки	SRC (source) – адреса місця розташування зображення (обов'язковий атрибут) width – ширина зображення в пікселях або у відсотках від ширини вікна; height – висота зображення в пікселях hspace – горизонтальний відступ у пікселях vspace – вертикальний відступ у пікселях align= значення (left, right, top, middle, інші) – розташування зображення	
<video> <source src="URL"> </video>	src – шлях до відео autoplay – автоматичне завантаження відео після завантаження сторінки controls – відображає панель керування height – висота області для відтворення loop – повторення відео після закінчення	<video loop=2 autoplay > <source src="відеофрагмент1"> </video>

	poster – адреса зображення, що буде відображатись замість відео	
<pre><audio> <source src="URL"> </audio></pre>	src – шлях до аудіо autoplay – автоматичне завантаження після завантаження сторінки controls – відображає панель керування loop – повторення аудіо після закінчення preload – завантаження аудіо разом з завантаженням сторінки	<pre><audio preload > <source src="аудіофрагмент2"> </audio></pre>

Основи мови HTML.

Розбиття сторінки на частини (блоки, фрейми). Створення та робота з гіперпосиланнями. Створення закладки (якоря).

Створення блоків здійснюється за допомогою тегів `<div>...</div>`. Такий елемент призначений для групування об'єктів сторінки. Розбиття на блоки відбувається для оздоблення сторінки та оформлення окремих елементів іншим чином. Блок складається з контенту, заданих полів, визначених меж та їх оформлення. Контент визначає розмір блока. Для цього задаються ширина та висота тієї частини блоку, де розміщується контент (`width`, `height`). Поля (`padding`) задають для того, щоб окреслити порожній простір навколо контенту. Межі (`border`) визначаються типом оформлення.

Розбиття сторінки на окремі елементи можна проводити за допомогою фреймів. Але використання фреймів є дещо застарілим, оскільки оформлення блоків є більш зручним та мобільним. Окрім того для оздоблення стилем (`css`) використовують саме блоки як окремий елемент. Проте оскільки фреймами визначали розбиття досить тривалий час, то вони і досі опрацьовуються браузерами та можуть використовуватись.

Фрейми – це елементи HTML, що дозволяють розділити сторінку на декілька незалежних вікон. Задається тегами `<FRAMESET> ... </FRAMESET>`. Обов'язковим атрибутом є `cols` або `rows`, що задають спосіб розбиття сторінки – горизонтальний та вертикальний. Кожний елемент керується тегом `<FRAME>`, для якого є також обов'язковий атрибут `src`=«ім'я документу». Даний атрибут визначає той файл, що буде відображатись в середині фрейму.

Приклад.

```
<HTML>
<HEAD>...</HEAD>
<FRAMESET cols="60%,40%">
<FRAME src="tabl_tegov.html">
<FRAME src="Table_color.html">
</FRAMESET>
</HTML>
```

Даний опис розбиває сторінку на 2 частини вертикально. В першій відображається документ `tabl_tegov.html`, а в другій - `Table_color.html`. Документи повинні бути описані окремо і знаходитись в тій же папці, де міститься вказаний опис розбиття на фрейми.

Деякі атрибути:

`SRC` — задає ім'я файлу, що відобразатиметься у фреймі;

`NAME` — задає ім'я фрейму;

`SCROLLING` — визначає наявність (значення `yes`) або відсутність (значення `no`) смуг прокручування у вікні фрейму (за замовчуванням — `yes`);

`NORESIZE` — забороняє користувачу змінювати розміри фрейму;

`BORDER` — визначає ширину розділювальної смуги між фреймами в пікселях;

`BORDERCOLOR` — визначає колір розділювальної смуги між фреймами;

`MARGINHEIGHT` — додає порожнє поле, висота якого визначена в пікселях, між верхньою межею фрейму і початком тексту

або графіки;

`MARGINWIDTH` — додає порожнє поле, ширина якого визначена в пікселях, між боковими межами фрейму і початком тексту

або графіки.

Гіперпосилання задаються тегами <A>... . Між тегами вводиться текст, що буде відображатись як гіпертекст (при натисканні на нього відбудеться переміщення на вказаний файл).

Гіперпосилання може бути зовнішнє, тобто на інший файл чи сторінку у глобальній мережі, внутрішні (закладка, якір), тобто на місце в самому документі.

Атрибути для даного тегу такі:

href = «URL» (або назва файлу зі вказанням шляху до нього) – той файл або Web-сторінка, що буде відкриватися при натисканні на гіперпосилання.

target = значення – відображення результату гіперпосилання відбувається:

- top - на всьому просторі вікна браузера;
- blank - у новому вікні браузера;
- self - у вікні, що містить це посилання;
- parent - у вікні, що є безпосереднім власником набору фреймів;

title – текст підказки.

Приклад:

<A href="prim_2.html" target=self title=«Додаткові дані» переглянути тут
гіперпосилання встановлено на текст.

 - гіперпосилання встановлюється на зображення

Створення внутрішнього посилання, тобто закладки відбувається так:

Назва розділу

Основи гіпертекстової розмітки веб-сторінок.

Таблиці каскадних стилів. Особливості приєднання до html-документу. Переваги використання css-файлів. Вбудовані стилі, групування селекторів.

Таблиці каскадних стилів (CSS- Cascading Style Sheets) – це набір правил оформлення й форматування, що може бути застосований до різних елементів сторінки.

Стильове оформлення об'єктів задається в середині тегів <STYLE> </STYLE> і оформлюється таким чином:

Позначка тегу (називається селектор) {властивість: значення вла-ті (називається опис стилю)}

Приклад.

```
<STYLE> P {font-size: 40pt; color: green; font-family: "Comic Sans MS"}
</STYLE>
```

Ці рядки означають, що для будь-якого тексту, який задається після тегу <P>, розмір шрифту відповідатиме 40 пунктів, колір тексту буде зелений, до тексту застосовуватиметься шрифт Comic Sans MS.

В CSS можна використовувати коментарі, які починаються з символів /* і закінчуються символами */

Стили всередині html-файлу описуються у голові документа.

Приклад.

Лістинг html-файлу, де всі заголовки 1 рівня відобразатимуться червоним кольором, заголовки другого рівня – блакитним, третього – помаранчевим.

```
<HTML>
<HEAD>
<TITLE> CSS </TITLE>
<STYLE TYPE="text/css">
h1 {color: red;}
h2 {color: blue;}
h3 {color: orange;}
</STYLE>
</HEAD>
<BODY>
<h1> ПРИКЛАД 1 </h1>
<h2>ПРИКЛАД 2</h2>
<h3>ПРИКЛАД 3</h3>
</BODY>
</HTML>
```

Застосовують 2 типи використання стилів:

1. Вставка CSS-кода всередину документа.
2. Запис таблиці стилів у зовнішньому файлі та встановлення посилання на цей файл

Приклад внутрішнього опису:

```
<html>
<head> <title>Символ для маркера</title>
<style>
li { list-style-type: none; /* Встановлений маркер зі списку спочатку треба прибрати */ }
li:before { content: "☀"; /* Додати можна будь-який інший символ */ }
</style> </head>
```

```
<body bgcolor=green >
<h2> <font color=silver> Цікаві свята </h2>
<ul>
  <li> Венеціанський карнавал </li>
  <li> Сніговий фестиваль в Саппоро</li>
  <li>Фестиваль фарб у Індії</li>
</li> Фестиваль фарб в Індії</li>
</ul> </body> </html>
```

Зовнішнє приєднання.

```
...
<LINK rel="stylesheet"
href="prim_1.css"
type="text/css">
```

Окремим файлом з назвою prim_1.css (розширення вказує на тип файлу, що відноситься до таблиць каскадних стилів) у файлі описується типи заголовків:

```
...
h1{color:red;}
h2{color:blue;}
h3{color:orange;}
...
```

Можна використовувати вбудовані стилі (Inline Styles), які вставляють у теги заголовків <H1>..., абзацу <P>, тіла <BODY>, а також у теги <DIV>, ...

Також можна групувати селектори для різних тегів з однаковою властивістю, наприклад колір (H1, H2, H3 {color:blue}), розташування тощо.

Основи побудови сайтів.

Поняття CMS, функції, особливості використання, приклади. Використання шаблонів Web-сторінок, вимоги до існування сайту, створеного за допомогою шаблонів.

Система керування змістом або контентом (Content management system, CMS) – це інформаційна система, комп'ютерна програма, сервіс, що використовується для ефективного керування змістом сайту людиною без спеціальної освіти або навичок роботи у сайтобудуванні. Найчастіше сюди входять шаблони для заповнення пересічним користувачем для створення сайту, інструменти для його оздоблення та додаткові функції, корисні при наповненні контенту сайту. Найчастіше CMS використовують модулі, які користувач може обирати, компонувати та налаштовувати за власними потребами. Зустрічається такий набір модулів, як: динамічне меню, новини, пошук по сайту, розташування фотографій, інформаційні сторінки, блог, зв'язок з адміністратором та інші. Сайти, створені на основі CMS використовують переважно веб-сервери, СУБД такі, як MySQL, веб-додатки для роботи тощо.

Існують як платні, так і безкоштовні сервіси. Тка, наприклад, сервіси Ucoz, Joomla та інші пропонують як безоплатні шаблони, так і платні. Безоплатні шаблони виводяться в простір Інтернет за рахунок реклами, що прикріплюється до безкоштовного шаблону. Оформлення платних сайтів супроводжується договором та вимогами покупця щодо рекламних оголошень та підтримки сервісів.

Функціями платформи є формувати зміст сайтів безпосередньо на основі шаблону та підтримувати сайт без спеціальних навичок. Користувач використовує власні зображення, текст та інші дані для наповнення сторінки. При цьому відбувається відокремлення контенту сайту від його дизайну. Сам шаблон (оформлення сайту) залишається незмінним після його вибору, а контент адміністратор сайту може змінювати в будь-який час. Всі дані зберігаються в базах даних і звідти завантажуються в мережу.

При використанні будь-якої з існуючих систем існують певні вимоги до оформлення сайтів. По-перше, це при використанні безоплатного хостингу розташування реклами на сайті, по-друге – це сайт повинен бути «популярним», тобто якщо сайт не відвідується протягом певного часу він вилучається з простору мережі.

Рекомендована література

1. LibreOffice // Матеріал з Вікіпедії – вільної енциклопедії [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/LibreOffice>
2. Microsoft – офіційна домашня сторінка [Сайт]. – Режим доступу: <https://www.microsoft.com/uk-ua/>
3. Microsoft Office // Матеріал з Вікіпедії – вільної енциклопедії [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Microsoft_Office
4. Web-дизайн. Навчально-методичний посібник. – Суми: СумДПУ імені А.С. Макаренка, 2008. – 72 с.
5. Windows 7 // Матеріал з Вікіпедії – вільної енциклопедії [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Windows_7
6. Бекаревич Ю.Б., Пушкина Н.В. Самоучитель Microsoft Access 2013 / Ю.Б. Бекаревич, Н.В. Пушкина. – СПб.: БХВ-Петербург, 2014. – 464 с.
7. Берман Н.Д. MS PowerPoint 2010 : учебное пособие / Н.Д. Берман, Т.А. Бочарова, Н.И. Шадрина. – Хабаровск : Изд-во Тихоокеан. гос. ун-та, 2013. – 88 с.
8. Берман Н.Д. Визуализация данных в MS Excel 2010 : учеб. пособие / Н.Д. Берман. – Хабаровск : Изд-во Тихоокеан. гос. ун-та, 2014. – 72 с.
9. Бураков П.В., Петров В.Ю. Введение в системы баз данных : Учебное пособие. – Санкт-Петербург, 2010. – 130 с.
10. Васильева М.О. Пай А.В. Работа в текстовом редакторе MS Word 2007/2010 : учебно-методическое пособие. – Нижний Тагил, 2012. – 74 с.
11. Глущенко Л.О., Тиркусова Н.В. Работа з електронними таблицями та базами даних. Навчальний посібник. – Суми: СумДУ, 2006. – 101 с.
12. Голицына О.Л., Максимов Н.В., Попов И.И. Базы данных: Учебное пособие. – М.: ФОРУМ: ИНФРА-М, 2006. – 352 с.
13. Гук М. Аппаратные средства IBM PC : [фундаментальное руководство] / Михаил Гук. – Санкт-Петербург : Питер, 2001. – 815 с.
14. Гуржій А.М., Коряк С.Ф., Самсонов В.В., Соляров О.Я. Контроль та керування корпоративними комп'ютерними мережами: інструментальні засоби та технології : Навчальний посібник для студентів технічних спеціальностей вищих навчальних закладів. – Харків : Компанія СМІТ, 2004. – 544 с.
15. Добро пожаловать в справочное руководство [Электронный ресурс]. – Режим доступа: LibreOffice!https://help.libreoffice.org/Main_Page/ru
16. Дуг Лоу. Microsoft PowerPoint 2010 для чайников. – С.-Пб.: Компьютерное изд-во «Диалектика», 2011. – 320 с.
17. Информатика в схемах і таблицях : навчальний посібник / О. В. Семеніхіна, В. Г. Шамоля, О. М. Удовиченко, А. О. Юрченко. – Суми : Видавництво «МакДен», 2013. – 76 с.
18. Карчевский Е.М., Филиппов И.Е. Access 2010 в примерах : Учебно-методическое пособие / Е.М. Карчевский, И.Е. Филиппов. – Казань: Казанский федеральный университет, 2011. – 118 с.
19. Квинт И. HTML и CSS (на 100%). – СПб.: Питер, 2008. – 352 с.
20. Колomoец Г. П. Организация компьютерных сетей: учебное пособие/ Г. П. Колomoец. – Запорожье: КПУ, 2012. – 156 с.
21. Коннолли Т. Базы данных: проектирование, реализация и сопровождение: теория и практика / Т. Коннолли, К. Бегг, А. Строчан : пер. с англ. – 3-е изд. – М.: Вильямс, 2003. – 1440 с.
22. Корнелл П. Анализ данных в Excel. – Москва: Эксмо, 2007. – 215 с.

23. Кравчук С.О., Шонін В.О. Основи комп'ютерної техніки: Компоненти, системи, мережі: навч. посіб. для студ. вищ. навч. закл. – К.: ІВЦ, 2005. – 344 с.
24. Крол Э. Все об Internet. – К.: Торгово-издат. бюро ВНУ, 1995. – 590 с.
25. Леонов В. PowerPoint 2010 с нуля / Леонов Василий. – М.: Эксмо, 2010. – 320 с.
26. Лиман Ф.М. Математична логіка і теорія алгоритмів : Навч. посібник для студ. фіз.-мат. спец. вищ. пед. навч. закл. – Суми : Слобожанщина, 1998. – 151 с.
27. Матросов А, Сергеев А., Чаунин М. HTML-4.0. – Спб: «БХВ-Петербург», 2003. – 672 с.
28. Мержевич В. Основы верстки [Электронный ресурс] / В. Мержевич – Режим доступа: – <http://htmlbook.ru>
29. Методичні рекомендації та завдання до самостійних робіт з дисципліни «Інформатика». Тема: Текстовий процесор MS Word 2010 (для студентів економічних спеціальностей) /укладачі: Д.В. Бельков, Є.М. Єдемська. – Донецьк: ДонНТУ, 2012. – 122 с.
30. Молочков В. Компьютерная графика для Интернета. Самоучитель. – СПб.: Питер, 2004. – 368 с.
31. Морзе Н.В. Основы інформаційно-комунікаційних технологій. – К.: Видавнича група ВНУ, 2008. – 352с.
32. О LibreOffice по-русски [Электронный ресурс]. – Режим доступа: <http://ru.libreoffice.org/home/>
33. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы : учеб. пособ. / В. Олифер, Н. Олифер. – 4-е изд. – Санкт-Петербург: Питер, 2014. – 943 с.
34. Операционная система [Электронный ресурс]. – Режим доступа: http://www.wikiznanie.ru/wikipedia/index.php/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0
35. Операційна система // Матеріал з Вікіпедії – вільної енциклопедії [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0
36. Пасічник О.Г Основни веб-дизайну / О.Г.Пасічник, О.В. Пасічник, І.В. Стеценко: [Навч. посібник].– Вид.група ВНУ. – 2009. 336с.
37. Пикуза В., Гаращенко А. Економічні і фінансові розрахунки в Excel. – Санкт-Петербург: Питер, 2007. – 396 с.
38. Романченко В. Microsoft Office 2010 – огляд «технічної» версії [Електронний ресурс]. – Режим доступу: <http://easy-code.com.ua/2012/08/microsoft-office-2010-oglyad-technichno%D1%97-versi%D1%97-ms-office-programni-kerivnictva-statti/>
39. Русаков М.Ю. HTML. Основы [Электронный ресурс]. – Режим доступа: <http://myrusakov.ru/html-osnovy.html>
40. Семикопенко А.А. Учебник html. Основы html 2009 / А.А. Семикопенко [Электронный ресурс] – Режим доступа: <http://www.webremeslo.ru/html/glava0.html>
41. Система керування вмістом // Матеріал з Вікіпедії – вільної енциклопедії [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B2%D0%BC%D1%96%D1%81%D1%82%D0%BE%D0%BC
42. Скотт Мюллер. Модернизация и ремонт ПК. – Москва, С-Пб., Киев, 2004. – 1342 с.
43. Столлингс В. Операционные системы. 4-е изд. – М.: Вильямс, 2002. – 848 с.
44. Таненбаум Э. Архитектура комп'ютера: 5-е издание. – С-Пб.: Питер, 2007. – 843 с.

45. Тарнавський Ю.А. Internet-технології : Конспект лекцій / Ю. А. Тарнавський; МАУП. – Київ: МАУП, 2004. – 120 с.
46. Технология работы в LibreOffice: текстовый процессор Writer, табличный процессор Calc : практикум / авт.-сост. В.А. Павлушина ; Ряз. гос. ун-т им. С.А. Есенина. – Рязань, 2012. – 80 с.
47. Холленд Р.Ч. Микропроцессоры и операционные системы / Р.Ч. Холленд. – Москва, 1991. – 190 с.
48. Хоумер А., Улмен К. Dynamic HTML: Справочник. – СПб.: Питер, 2000. – 512 с.
49. Что такое CMS [Электронный ресурс]. – Режим доступа: <http://blogwork.ru/chto-takoe-cms-ili-sistema-upravleniya-kontentom/>
50. Шамшина Н.В. Використання табличного процесора MS Excel. Практикум. – Суми: Вид-во СумДПУ ім. А.С. Макаренка, 2015. – 65 с.
51. Шамшина Н.В. Інформатика. Система управління базами даних Microsoft Access. Навчальний посібник. – Суми: Вид-во СумДПУ ім. А.С. Макаренка, 2015. – 72 с.
52. Шаньгин В. Ф. Защита информации в компьютерных системах и сетях. / В. Ф. Шаньгин, Москва: ДМК Пресс, 2012. – 592с.

Навчальне видання

ФАХОВЕ ВСТУПНЕ ВИПРОБУВАННЯ З ІНФОРМАТИКИ

Частина I

Дегтярєва Неля Валентинівна
Петренко Сергій Іванович
Удовиченко Ольга Миколаївна
Безуглий Дмитро Сергійович

Комп'ютерна верстка
О.М. Удовиченко

Фізико-математичний факультет
СумДПУ імені А.С. Макаренка
вул. Роменська, 87
м. Суми, 40002
тел. (0542) 68 59 10

<http://fizmatsspu.sumy.ua/>