

## ЗМІСТ

Вступ.....	3
1. Дослідження об'єкту практики.....	4
2. Дослідження процесу складання, монтажу, налагодження і тестування створюваного апаратного забезпечення комп'ютерних систем і мереж.....	7
3. Універсальне та спеціалізоване апаратне забезпечення комп'ютерних систем і мереж, які використовуються на об'єкті практики.....	14
4. Універсальне та спеціалізоване програмне забезпечення комп'ютерних систем і мереж, які використовуються на об'єкті практики.....	15
5. Рекомендації щодо поліпшення існуючих на об'єкті практики технологічних процесів створення та використання комп'ютерних систем і мереж та їх програмного забезпечення.....	17
6. Індивідуальне завдання по виробничій практиці.....	18
Висновки.....	20
Література.....	21

## ВСТУП

У кожному вузі України передбачено проходження студентами виробничої практики, яка є обов'язковим складовим елементом підготовки будь-якого спеціаліста. Під час проходження практики студент закріплює здобуті знання і навички, отримуючи практичний досвід роботи в реальних економічних умовах.

Студент може мати досконалі теоретичні знання, але може розгубитися в ситуації, коли виникне необхідність прийняти рішення в реальних економічних умовах. Тому необхідно забезпечувати студентам змогу пройти виробничу практику ще під час навчання у вузі, для того, щоб спробувати себе не тільки в ролі теоретика, але й практика.

Виробнича практика для студентів спеціальності 7.091501 "Комп'ютерні системи та мережі" є невід'ємною складовою частиною процесу їх професійної підготовки і проводиться на сучасних підприємствах та -організаціях, оснащених відповідним апаратним та програмним забезпеченням комп'ютерних систем та мереж.

Виробнича практика передбачає отримання необхідного обсягу практичних знань та умінь відповідно до кваліфікаційного рівня інженера-системотехніка.

Метою виробничої практики для студентів спеціальності 7.091501 є оволодіння студентами сучасними методами ефективного застосування комп'ютерних та мережних технологій для вирішення задач комплексної автоматизації підприємств і установ, вивчення практичних рішень в галузі організації та побудови регіональних, корпоративних та локальних комп'ютерних мереж, розподілених комп'ютерних систем, дослідження характеристик, виконуваних функцій, етапів розробки, тестування та впровадження використовуваних на об'єктах практики універсального та спеціалізованого апаратного та програмного забезпечення комп'ютерних систем та мереж.

## 1. ДОСЛІДЖЕННЯ ОБ'ЄКТУ ПРАКТИКИ

Українське державне підприємство поштового зв'язку "Укрпошта" (УДППЗ "Укрпошта") є національним оператором поштового зв'язку України згідно з розпорядженням КМУ від 10 січня 2002 року.

Діяльність УДППЗ "Укрпошта" регламентується Законом України "Про поштовий зв'язок", іншими Законами України, а також нормативними актами ВПС.

Україна з 1947 року є членом Всесвітнього поштового союзу (ВПС), двічі обиралась і нині входить до складу Адміністративної ради, вищого виконавчого органу ВПС. У 1991 році Адміністрація зв'язку України увійшла до Регіональної співдружності в галузі зв'язку (РСЗ), з 1994 року УДППЗ "Укрпошта" є членом Асоціації європейських державних поштових операторів "PostEurop".

Укрпошта функціонує як самостійна господарська одиниця з 1994 року, коли галузь зв'язку в результаті реформування була розподілена на поштовий та електров'язок. Тоді було утворено два об'єднання, у тому числі Українське об'єднання поштового зв'язку "Укрпошта", яке в липні 1998 року реорганізовано в нині діюче УДППЗ "Укрпошта" згідно з Програмою реструктуризації Укрпошти, затвердженою постановою Кабінету Міністрів України від 04.01.98 за №1 .

До складу підприємства входять 31 філіалів (27 регіональних дирекцій, у тому числі дві міських – Київська та Севастопольська, Дирекція з оброблення та перевезення пошти, дирекція "Автотранспошта", дирекція "Видавництво "Марка України", дирекція "Головний навчальний центр "Зелена Буча").

Мережа поштового зв'язку охоплює всі регіони України і налічує 24 поштами, 400 вузлів поштового зв'язку та 26 центрів поштового зв'язку, 3517 міських, 9880 сільських, 125 пересувних, 22 сезонних відділення та 1824 пункти поштового зв'язку.

У системі Укрпошти працюють понад 115 тис. чол., з них понад 48 тис. листонош.

Доставка поштових відправлень в Україні забезпечується власним автомобільним транспортом. Для перевезення пошти та друкованих видань діють 33 магістральні автомобільні маршрути довжиною понад 29 тисяч кілометрів. Парк транспортних засобів, задіяних на перевезенні, становить понад 4,3 тис. автомобілів.

Українська пошта здійснює прямий поштовий обмін з 58 країнами світу; двосторонній обмін поштовими переказами з 17 країнами, односторонній – з 5, двосторонній обмін електронними переказами між Україною і Москвою, обмін відправленнями з післяплатою – з 10 країнами. УДППЗ “Укрпошта” як національний поштовий оператор має забезпечувати надання універсальних послуг поштового зв’язку, перелік яких і відповідні тарифи затверджуються Кабінетом Міністрів. В Україні до цих послуг належить пересилання простих та рекомендованих поштових карток, листів, бандеролей, а також посилок без оголошеної цінності масою до 10кг. Загалом, Укрпошта пропонує споживачам понад 70 видів послуг, у тому числі з використанням найсучасніших інформаційних технологій, насамперед електронної і гібридної пошти, електронного переказу, послуг Інтернет. У перспективі - здійснення передплати через Інтернет, електронної торгівлі, надання інформаційних послуг, послуг IP-телефонії з передачею відеозображення тощо.

У підприємстві діють корпоративні комунікаційна мережа та автоматизовані системи для оптимізації управлінських рішень, автоматизації обробки технологічної і фінансової документації. Швидкими темпами впроваджується комп’ютерна техніка на робочих місцях обслуговування клієнтів, що дає змогу розширювати номенклатуру послуг та підвищувати їх якість.

УДППЗ “Укрпошта” бере участь у реалізації Національної програми інформатизації “Електронна Україна”. Йде цілеспрямована робота зі створення

в об'єктах поштового зв'язку Інтернет-пунктів з метою забезпечення доступу широких верств населення до всесвітньої інформаційної мережі. Нині у філіалах підприємства діють 78 таких пунктів, планується працювати в цьому напрямку і надалі.

Важливим напрямом діяльності Укрпошти є вдосконалення організаційно-технологічної структури підприємства. Нині розпочато реалізацію Програми оптимізації мережі поштового зв'язку, що включає проведення широкомасштабного комплексу робіт, спрямованих на підвищення ефективності виробництва, поліпшення якості надання послуг та скорочення непродуктивних витрат. Суть оптимізації полягає в реорганізації районних вузлів зв'язку шляхом їх укрупнення і створення міжрайонних вузлових центрів, побудованих за принципом функціональної доцільності, а не за адміністративно-територіальним поділом регіонів України.

Підприємство є стабільною структурою, що має давні традиції, користується довірою населення, забезпечує доступність послуг на всій території держави.

## **2. ДОСЛІДЖЕННЯ ПРОЦЕСУ СКЛАДАННЯ, МОНТАЖУ, НАЛАГОДЖЕННЯ І ТЕСТУВАННЯ СТВОРЮВАНОВОГО АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ**

Під архітектурою будемо розуміти сукупність принципів і правил у відповідності з якими будуються обчислювальні системи різних видів.

Розглядаючи питання архітектури комп'ютерних мереж (КМ), насамперед, необхідно визначити їх призначення та область застосування. Так основним призначенням комп'ютерної мережі є надання великому числу користувачів одночасного доступу до її обчислювальних ресурсів. Виходячи з цього, комп'ютерна мережа може бути визначена як система розподіленої обробки інформації, що складається з комп'ютерів, територіально-розосереджених і взаємодіючих між собою за допомогою засобів зв'язку. Комп'ютери, що входять до складу мережі, виконують досить широке коло функцій, основними серед яких є:

- організація доступу до мережі;
- управління передачею інформації;
- надання обчислювальних ресурсів і послуг абонентам мережі.

Відповідно до цього по функціональній ознаці всю безліч систем КМ можна розділити на абонентські, комутаційні і головні (Host) підсистеми.

Абонентська підсистема являє собою комп'ютер, орієнтований на роботу в складі КМ і забезпечує користувачам доступ до її обчислювальних ресурсів.

Комутаційні підсистеми є вузлами комутації мережі передачі даних і забезпечують організацію складових каналів передачі даних між абонентським підсистемами. Як керуючі елементи вузлів комутації використовуються процесори телеобробки або спеціальні комутаційні (мережеві) процесори.

Великою різноманітністю характеризуються Host підсистеми або мережеві сервери. Сервером прийнято називати спеціальний комп'ютер, що виконує основні сервісні функції, такі як: управління мережею, збір, обробку,

зберігання і надання інформації абонентам КМ. У зв'язку з великим числом сервісних функцій доцільне розділення серверів за їх функціональним призначенням. Наприклад, файл-сервер визначається як мережевий комп'ютер, що здійснює операції по зберіганню, обробці і наданню файлів даних абонентам КМ. У свою чергу, комп'ютер, що забезпечує абонентським системам ефективний доступ до КМ, отримав назву сервер доступу і т.д.

### Класифікація КМ

В залежності від основної класифікаційної ознаки покриття території всі КМ поділяють на групи:

- Локальні (Local Area Network);
- Глобальні (Wide Area Network);
- Регіональні (Metropolitan Area Network);
- Корпоративні КМ.

Головною вимогою, яка ставиться до мереж, є виконання мережею її основної функції – забезпечення користувачам потенційної можливості доступу до розділюваних ресурсів усіх комп'ютерів, об'єднаних у мережу. Всі інші вимоги – продуктивність, надійність, сумісність, керованість, захищеність, розширюваність і масштабованість – зв'язані з якістю виконання цієї основної задачі.

Хоча всі ці вимоги дуже важливі, часто поняття «якість обслуговування» (Quality of Service, QoS) комп'ютерної мережі трактується більш вузько – у нього включаються тільки дві найважливіші характеристики мережі – продуктивність і надійність.

Незалежно від обраного показника якості обслуговування мережі існують два підходи до його забезпечення. Перший підхід, мабуть, покажеться найбільш природним з погляду користувача мережі. Він полягає в тому, що мережа (точніше, персонал який її обслуговує) гарантує користувачу дотримання деякої числової величини показника якості обслуговування. Наприклад, мережа може

гарантувати користувачу А, що кожен з його пакетів, посланих користувачеві В, буде затриманий мережею не більш, ніж на 150 мс. Або, що середня пропускна здатність каналу між користувачами А і В не буде нижче 5 Мбіт/с, при цьому канал буде допускати пульсацію трафіка в 10 Мбіт на інтервалах часу не більш 2 секунд. Технології frame relay і АТМ дозволяють будувати мережі, що гарантують якість обслуговування по продуктивності.

Другий підхід полягає в тому, що мережа обслуговує користувачів відповідно до їхніх пріоритетів. Тобто якість обслуговування залежить від ступеня привілейованості користувача чи групи користувачів, до якої він належить. Якість обслуговування в цьому випадку не гарантується, а гарантується тільки рівень привілеїв користувача. Таке обслуговування називається обслуговуванням best effort – з найбільшим старанням. Мережа намагається по можливості більш якісно обслужити користувача, але нічого при цьому не гарантує. За таким принципом працюють, наприклад, локальні мережі, побудовані на комутаторах із пріоритезацією кадрів.

### Продуктивність

Потенційно висока продуктивність – це одне з основних властивостей розподілених систем, до яких відносяться комп'ютерні мережі. Ця властивість забезпечується можливістю розпаралелення робіт між декількома комп'ютерами мережі. На жаль, цю можливість не завжди вдається реалізувати. Існує кілька основних характеристик продуктивності мережі:

- час реакції;
- пропускна здатність;
- затримка передачі і варіація затримки передачі.

Час реакції мережі є інтегральною характеристикою продуктивності мережі з погляду користувача. Саме цю характеристику має на увазі користувач, коли говорить: «Сьогодні мережа працює повільно».

У загальному випадку час реакції визначається як інтервал часу між



виникненням запиту користувача до якої-небудь мережної служби й одержанням відповіді на цей запит.

Очевидно, що значення цього показника залежить від типу служби, до якої звертається користувач, від того, який користувач і до якого сервера звертається, а також від поточного стану елементів мережі – завантаженості сегментів, комутаторів і маршрутизаторів, через які проходить запит, завантаженості сервера і т.п.

Тому є сенс використовувати також і середньозважену оцінку часу реакції мережі, усереднюючи цей показник по користувачам, серверам і за часом дня (від якого в значній мірі залежить завантаження мережі).

Час реакції мережі звичайно складається з декількох складових. У загальному випадку в нього входить час підготовки запитів на клієнтському комп'ютері, час передачі запитів між клієнтом і сервером через сегменти мережі і проміжне комунікаційне устаткування, час обробки запитів на сервері, час передачі відповідей від сервера клієнту і час обробки одержуваних від сервера відповідей на клієнтському комп'ютері.

Зрозуміло, що користувача розкладання часу реакції на складові не цікавить – йому важливий кінцевий результат; однак для мережного фахівця дуже важливо виділити з загального часу реакції складові, відповідним етапам мережної обробки даних, – передачу даних від клієнта до сервера через сегменти мережі і комунікаційне устаткування.

Пропускна здатність відображає обсяг даних, переданих мережею чи її частиною за одиницю часу. Пропускна здатність вже не є користувацькою характеристикою, тому що вона говорить про швидкість виконання внутрішніх операцій мережі – передачі пакетів даних між вузлами мережі через різні комунікаційні пристрої. Зате вона безпосередньо характеризує якість виконання основної функції мережі – транспортування повідомлень – і тому частіше використовується при аналізі продуктивності мережі.

Найчастіше при проектуванні, налаштуванні й оптимізації мережі

використовуються такі показники, як середня і максимальна пропускні здатності. Середня пропускна здатність окремого елемента чи всієї мережі дозволяє оцінити роботу мережі на великому проміжку часу, протягом якого в силу закону великих чисел піки і спади інтенсивності трафіку компенсують один одного. Максимальна пропускна здатність дозволяє оцінити можливості мережі справлятися з піковими навантаженнями, характерними для особливих періодів роботи мережі, наприклад ранкових годин, коли співробітники підприємства майже одночасно реєструються в мережі і звертаються до розділюваних файлів і баз даних.

Пропускна здатність можна вимірювати між будь-якими двома вузлами мережі, наприклад між клієнтським комп'ютером і сервером, між вхідним і вихідним портами маршрутизатора. Для аналізу і налаштування мережі дуже корисно знати дані про пропускна здатність окремих елементів мережі.

Важливо відзначити, що через послідовний характер передачі пакетів різними елементами мережі загальна пропускна здатність мережі будь-якого складеного шляху в мережі буде дорівнювати мінімальній з пропускних здатностей складових елементів маршруту. Для підвищення пропускної здатності складеного шляху необхідно в першу чергу звернути увагу на самі повільні елементи – у даному випадку таким елементом, швидше за все, буде маршрутизатор. Варто підкреслити, що коли переданий по складеному шляху трафік буде мати середню інтенсивність, яка перевершує середню пропускна здатність самого повільного елемента шляху, то черга пакетів до цього елемента буде рости теоретично до нескінченності, а практично – доти, поки не заповниться його буферна пам'ять, а потім пакети просто почнуть відкидатися і губитися.

Звичайно при визначенні пропускної здатності сегмента чи пристрою в переданих даних не виділяються пакети якогось визначеного користувача, чи додатка комп'ютера – підраховується загальний обсяг переданої інформації. Проте для більш точної оцінки якості обслуговування така деталізація бажана, і

останнім часом системи керування мережами всі частіше дозволяють її виконувати.

### Надійність і безпека

Однією з перших цілей створення розподілених систем, до яких відносяться й КМ, було досягнення більшої надійності в порівнянні з автономними ПК.

Для оцінки надійності складних систем застосовується такий набір характеристик, зокрема: готовність, цілісність даних, узгодженість, імовірність доставки пакета, безпека, відмовостійкість.

Готовність, коефіцієнт готовності (availability) означає частку часу, протягом якого система може бути використана. Готовність може бути поліпшена шляхом введення надмірності в структуру системи: ключові елементи системи повинні існувати в декількох екземплярах, щоб при відмові одного з них функціонування системи забезпечували інші.

Щоб систему можна було віднести до високонадійних, вона повинна як мінімум мати високу готовність, але цього недостатньо. Необхідно забезпечити цілісність даних і захист їх від спотворень. Крім цього, повинна підтримуватися узгодженість (несуперечність) даних, наприклад, якщо для підвищення надійності на декількох файлових серверах зберігається кілька копій даних, то потрібно постійно забезпечувати їхню ідентичність.

Оскільки мережа працює на основі механізму передачі пакетів між кінцевими вузлами, то однією з характеристик надійності є імовірність доставки пакета вузлу призначення без спотворень. Поряд з цією характеристикою можуть використовуватися й інші показники: імовірність втрати пакету, імовірність спотворення окремого біта переданих даних, співвідношення загублених пакетів до доставленого.

Іншим аспектом загальної надійності є безпека (security), тобто здатність системи захистити дані від несанкціонованого доступу. У розподіленій системі

це зробити набагато складніше, ніж у централізованій. Іншим вразливим місцем можуть бути залишені без догляду персональні комп'ютери. Крім того, завжди є потенційна погроза взлому захисту мережі від неавторизованих користувачів, якщо мережа має виходи в глобальні мережі загального користування.

Ще однією характеристикою надійності є відказостійкість (fault tolerance). У мережах під відказостійкістю розуміється здатність системи приховати від користувача відмову окремих її елементів. Наприклад, якщо копії таблиці бази даних зберігаються одночасно на декількох файлових серверах, то користувачі можуть просто не помітити відмову одного з них.

Розширюваність (extensibility) означає можливість порівняно легкого додавання окремих елементів мережі (користувачів, комп'ютерів, додатків, служб), нарощування довжини сегментів мережі і заміни існуючої апаратури більш потужною (продуктивною). При цьому принципово важливо, що легкість розширення системи іноді може забезпечуватися в деяких дуже обмежених межах.

Масштабованість (scalability) означає, що мережа дозволяє нарощувати кількість вузлів і довжину зв'язків у дуже широких межах, при цьому продуктивність мережі не погіршується. Для забезпечення масштабованості мережі доводиться застосовувати додаткове комунікаційне устаткування і спеціальним чином розподіляти структуру мережі.

Прозорість (transparency) мережі досягається в тому випадку, коли мережа представляється користувачам не як безліч окремих комп'ютерів, зв'язаних між собою складною системою кабелів, а як єдина традиційна робоча станція із системою поділу часу

Сумісність або інтегрованість означає, що мережа здатна містити в собі найрізноманітніше програмне й апаратне забезпечення, тобто в ній можуть співіснувати різні операційні системи, що підтримують різні стеки комунікаційних протоколів, і працювати апаратні засоби і додатки різних виробників. Мережа, яка складена з різнотипних елементів, називається

неоднорідною чи гетерогенною, а якщо гетерогенна мережа працює без проблем, то вона є інтегрованою. Основний шлях побудови інтегрованих мереж – використання модулів, виконаних відповідно до відкритих стандартів і специфікацій.

### **3.УНІВЕРСАЛЬНЕ ТА СПЕЦІАЛІЗОВАНЕ АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ, ЯКІ ВИКОРИСТОВУЮТЬСЯ НА ОБ'ЄКТІ ПРАКТИКИ**

У своїй роботі Золочівський вузол поштового зв'язку використовує наступний апаратний комплекс:

- сервер-роутер на базі Intel Pentium III 550 Mhz, 128 Mb RAM, 40 Gb ATA 100, 2 x LAN-CARD (вихід в мережу Інтернет та підтримка корпоративної мережі);

- сервер для бухгалтерії на базі AMD Duron 1.4 Ghz, 128Mb RAM, 20 Gb ATA 100;

- локальна мережа - Ethernet. Побудована з використанням двох 8-ми портових switch фірми Surecom. Як фізичне середовище передачі даних використовується вита пара UTP-5;

- 3 модеми IDC-2814 , що забезпечують доступ до корпоративної мережі через міжміський телефонний зв'язок;

- блоки безперебійного живлення фірми APC(в залежності від потреб використовуються різні моделі);

- персональні комп'ютери різноманітних конфігурацій (від Pentium 100 Mhz до Pentium IV 1,7 Ghz). Використовуються як офісні машини (набір текстів, електронних таблиць, формування звітності та ін.);

- принтери Epson LX-300, Epson LX-300+, HP 1100A, OKI B4200L.

На підприємстві використовується резервування електромережі: від короткотермінових зникнень електроенергії комп'ютерна техніка захищена блоками безперебійного живлення; якщо електроживлення не відновлюється через 3 хвилини – вмикається автономна дизельна-електростанція.

#### **4.УНІВЕРСАЛЬНЕ ТА СПЕЦІАЛІЗОВАНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ, ЯКІ ВИКОРИСТОВУЮТЬСЯ НА ОБ'ЄКТІ ПРАКТИКИ**

У підрозділах адміністрації, технічної підтримки і бухгалтерії використовується операційна система Windows 95, 98.

На сервері-роутері встановлено операційну систему FreeBSDRELEASE 5.2.1.

Сервером для бухгалтерії керує SCO Open Server 4.5.

Захист від зовнішніх атак здійснюється відповідним налаштуванням граничного маршрутизатора під керуванням FreeBSD (налаштуванню підлягає служба IPWF2), який перешкоджає проникненню помилкового пакету даних у мережу.

Для доступу до ресурсів Unix-машин на підприємстві використовується стандартна система захисту цієї операційної системи. Захист від доступу, що не санкціонується, здійснюється за допомогою процедури введення логіну-пароля.

Доступ до конкретного ресурсу здійснюється за допомогою різних серверів.

Авторизація - це право на доступ до деякого об'єкта або на виконання якої-небудь системної дії. У більшості систем UNIX всі рішення з приводу доступу приймаються на основі простих дискреційних критеріїв, або виходячи з того, чи є root власником процесу, що здійснює доступ. Кореневий бюджет має повноваження на виконання таких системних дій, які не може виконувати ніякий інший процес. Операційна Система визначає два типи авторизації: авторизація ядра і авторизація підсистеми. Авторизація ядра пов'язана з процесами. (Процес - це програма, що виконується в системі в даний момент.) Вони дозволяють процесу виконувати певні дії, якщо процес наділений необхідними привілеями. Авторизація підсистеми пов'язана з користувачами. Вони дозволяють користувачеві виконувати спеціальні дії за допомогою команд

підсистеми (надійні утиліти і програми). Підсистема - це зв'язний набір файлів, пристроїв і команд, що служать для виконання деякої функції.

На серверах відключені невживані сервіси, що знижує можливість проникнення стороннього, але не виключає його зі стовідсотковою імовірністю. Кожний день на спеціальних інформаційних серверах публікується інформація про виявлення «дір» в захисті конкретної версії програмного забезпечення. У обов'язок системного адміністратора мережі входить постійно стежити за такого роду інформацією і своєчасно оновлювати версії програмних продуктів, що використовуються в роботі фірми.

Фізичний доступ до серверів обмежений тільки працівниками технічного персоналу.

До відповідних ресурсів серверів мають доступ тільки авторизовані користувачі, ведеться сувора політика регулярної зміни паролів, що дозволяє отримати додаткові гарантії безпеки.

Також, на підприємстві використовуються автоматизовані робочі місця (АРМ):

- „Електронний переказ” Система призначена для прийому-передачі грошових електронних переказів, внутрішньо корпоративних телеграм, службових повідомлень.

- „Автоматизована система реєстрованої кореспонденції”. Призначена для полегшення роботи з реєстрованою кореспонденцією: покликана підняти надійність та швидкість проходження реєстрованої кореспонденції. На даний момент відбувається її впровадження в роботу.

Нажаль детальніша інформація по цих системах знаходиться під грифом ДСК.



## **5. РЕКОМЕНДАЦІЇ ЩОДО ПОЛІПШЕННЯ ІСНУЮЧИХ НА ОБ'ЄКТИ ПРАКТИКИ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ СТВОРЕННЯ ТА ВИКОРИСТАННЯ КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ ТА ЇХ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Розглянувши існуючий комплекс апаратного і програмного забезпечення можна виділити такі недоліки:

- відсутність системи автоматичного резервного копіювання важливої інформації на серверах та робочих станціях;
- моральна зношеність частини апаратного забезпечення;
- використання на робочих станціях застарілих операційних систем Windows 95 та 98;
- недостатня документованість програмних продуктів власного виробництва (АРМ);
- часткова недостача комп'ютеризованих робочих місць в період створення різноманітних форм звітності.

Для вирішення даних задач необхідно таке апаратне забезпечення:

- RAID-система створення гарантованої копії інформації з можливістю гарячої заміни носіїв (з підтримкою системи запису та корекції помилок RAID5) для використання на серверах;
- заміна частини персональних комп'ютерів на нові, створення нових комп'ютеризованих робочих місць.

Програмне забезпечення:

- Провести модернізацію операційних систем до Windows 2000 або Windows XP з останніми виправленнями (Service Pack 4 для Windows 2000 та Service Pack 1 для Windows XP);
- Norton Ghost Enterprise Edition або аналогічне, яке дозволяє проводити резервне копіювання важливої інформації з робочих станцій на виділений

сервер або робочу станцію.

## 6. ІНДИВІДУАЛЬНЕ ЗАВДАННЯ.

Моїм індивідуальним завданням було написати програму, яка автоматизувала процес створення сортувальної таблиці передплатних видань.

Завдання наступне: в районі є 28 відділень зв'язку. Нехай 1-ше відділення виписало 30 примірників, 2-ге – 40, 3-тє – 25, 4-те – 5, інші не виписали жодного; в загальному всі відділення виписали 100 примірників якогось видання. В такому вигляді до Львівської дирекції і йде замовлення на передплачене видання („нам потрібно 100 примірників певного видання”). При цьому інформація про конкретного замовника заноситься до спеціальної картотеки форм СП-5 („1-ше відділення замовило 30 примірників певного видання” і т.д.).

На наступному етапі з заданою періодичністю видання починає надходити з обласної дирекції. Приходить воно у вигляді стосу з ста примірників. Для того, щоб знати, скільки примірників перенаправити у певні районні відділення нам потрібна сортувальна таблиця наступного вигляду:

	Відділення1	Відділення2	Відділення3	...	Відділення 28
Видання	30	25	4		-
Видання	2	5	5		5
Видання	3	3	3		25

Створюється вона на основі даних карток СП-5 з картотеки.

Заповнення такої таблиці трудомісткий процес: одна людина витрачає на нього в середньому 25-30 годин, тобто це приблизно 3-4 дні безперервної клопіткої роботи. За умови автоматизації цього процесу за допомогою комп'ютерної техніки, створення такої таблиці займе приблизно 4-6 годин за умови введеної картотеки форм СП-5.

Реалізація.

Дане завдання я розділив на дві частини: створення картотеки

періодичних видань і підтримання її в актуальному стані; формування сортувальної таблиці.

Мовою реалізації була вибрана мова Web-програмування PHP.

Перша частина програми складається з кількох модулів. Спочатку користувач вводить в Web-формі індекс потрібного видання; дані з форми попадають в скрипт, який перевіряє коректність введеного індексу (введений індекс порівнюється з індексами всіх передплатних видань України, для цього існує спеціальний електронний каталог видань в вигляді файлу бази даних), також, користувач заповнює виведену форму СП-5 для заданого видання. Після виконання цього скрипта керування передається іншому модулю: цей модуль вносить введені дані з Web-форми, створеної попереднім скриптом в файл бази даних (Data Base File).

Друга частина програми складається з двох скриптів: перший створює на основі файлів картотеки форм СП-5 сортувальну таблицю у вигляді файлу баз даних, інший виводить дані з цього файлу в вигляді придатному для друку.

## **ВИСНОВКИ**

Виробнича практика є джерелом накопичення практичного досвіду роботи по вибраній спеціальності та закріплення теоретичних знань. Вона відіграє важливу роль у підготовці спеціалістів високого рівня, які б володіли сучасними технологіями, вміло використовували методологію та методики проектування інформаційних систем

В ході проходження виробничої практики було проведено дослідження об'єкта, його структури та функцій, наявного апаратного та програмного забезпечення і комплексу задач, що вирішується ним. Також було проведено ознайомлення з технологічними процесами складання, монтажу, налагодження і тестування комп'ютерних мереж.

На основі аналізу існуючих апаратних та програмних засобів були виявленні недоліки побудови існуючого комплексу апаратно-програмних засобів та намічені шляхи виправлення даної ситуації.

### **Список використаної літератури:**

1. Немет Э., Снайдер Г., Сибасс С., Хейн Т. Р., // H50 UNIX: Руководство системного администратора. Для профессионалов / Пер. с англ. – СПб.: Питер; К.:Издательская группа BHV, 2002. – 928 с.: ил.
2. Армстронг (мл.), Джеймс // A83 Секреты UNIX: 2-е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2001. – 1072 с.: ил. – Парал. тит. англ.
3. Олифер В.Г., Олифер Н.А. "Компьютерные сети: технологии, протоколы. Учебник." - Питер, 2000 - 672 с.
4. Николайчук Я.М., Яцків В.В., Гринчишин Т.М. Перспективи створення телефонних ліній на основі відкритого оптичного каналу зв'язку // Вісник Технологічного університету Поділля. – 2002. Т2– №3. – С. 47 – 50.

Додаток 1. Скрипт перевірки правильності вводу індексу; вивід для заповнення форми СП-5.

```
<?php
echo "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 3.2 Final//RU\">
<HTML>
<HEAD>
<TITLE>Заголовок документа</TITLE>
</HEAD>
<BODY>";

//Перевірка на правильність вводу індексу
$type = $_GET['index'] + 10;
if ($type == 10) {
echo "Введено неправильний індекс.";
return;
}

$finding_counter = 0;

//відриваємо базу
$db = dbase_open ("C:\Apache2\htdocs\BD\Kpit.dbf", 0);

//кількість записів в базі
$records_counter = dbase_numrecords($db);

//Пошук по базі
for ($c=0; $c<$records_counter; $c++) {
$temp_record = dbase_get_record ($db,$c);
if ($temp_record[0] == $_GET['index']) {
```

```

$temp_record[1] = convert_cyr_string ($temp_record[1], d, w);
//перекодування назви видання з cp866 в win-1251

$temp_record[1] = str_replace ("", "\", $temp_record[1]);

echo "<BR>Індекс: "; echo $_GET['index'];

echo "<BR>Видання: $temp_record[1]";

$finding_counter = 1;

}

if ($finding_counter == 1)

break;

}

if ($finding_counter == 0) {

echo "Видання з індексом "; echo $_GET['index']; echo " не знайдено.";

return;

}

$today = getdate();

echo "<FORM ACTION=\"/Scripts/SP1_Zl_in2.php\">

Кількість комплектів:

<INPUT TYPE=\"text\" SIZE=\"1\" NAME=\"amount\">

<br>

&#160 01&#160&#160&#160 02&#160&#160&#160 03&#160&#160&#160

04&#160&#160&#160 05&#160&#160&#160 06&#160&#160&#160&#160

07&#160&#160&#160 08&#160&#160&#160 09&#160&#160&#160

10&#160&#160&#160 11&#160&#160&#160 12<BR>

<INPUT TYPE=\"text\" SIZE=\"1\" NAME=\"01\">

<INPUT TYPE=\"text\" SIZE=\"1\" NAME=\"02\">

```



```
<INPUT TYPE="text" SIZE="1" NAME="03">
```

```
<INPUT TYPE="text" SIZE="1" NAME="04">
```

```
<INPUT TYPE="text" SIZE="1" NAME="05">
```

```
<INPUT TYPE="text" SIZE="1" NAME="06">
```

```
<INPUT TYPE="text" SIZE="1" NAME="07">
```

```
<INPUT TYPE="text" SIZE="1" NAME="08">
```

```
<INPUT TYPE="text" SIZE="1" NAME="09">
```

```
<INPUT TYPE="text" SIZE="1" NAME="10">
```

```
<INPUT TYPE="text" SIZE="1" NAME="11">
```

```
<INPUT TYPE="text" SIZE="1" NAME="12">
```

```
<br>
```

```
<INPUT TYPE="hidden" NAME="index" VALUE=""; echo  
$_GET['index']; echo " \">
```

```
<INPUT TYPE="hidden" NAME="name" VALUE=""; echo  
$temp_record[1]; echo " \">
```

```
<INPUT TYPE="hidden" NAME="smsp" VALUE=""; echo  
$temp_record[5]; echo " \">
```

```
<INPUT TYPE="hidden" NAME="month" VALUE=""; echo  
$today['mon']; echo " \">
```

```
<INPUT TYPE="hidden" NAME="year" VALUE=""; echo $today['year'];  
echo " \">
```

```
<BR> <BR>
```

```
<INPUT TYPE="submit" SIZE="5" VALUE="Прийняти">
```

```
</FORM>";
```

```
echo "</BODY>
```

```
</HTML>";
```

```
?>
```

Додаток 2. Скрипт переносу даних з Web-форми в файл бази даних.

```
<?php
```

```
$bd = dbase_open ("C:\\Apache2\\htdocs\\BD\\2004_ZI_SP1.DBF", 2);
```

```
$arr_to_bd[0] = $_GET['index'];
```

```
$arr_to_bd[1] = convert_cyr_string ($_GET['name'], w, a);
```

```
$arr_to_bd[2] = $_GET['amount'];
```

```
$arr_to_bd[3] = $_GET['01'];
```

```
$arr_to_bd[4] = $_GET['02'];
```

```
$arr_to_bd[5] = $_GET['03'];
```

```
$arr_to_bd[6] = $_GET['04'];
```

```
$arr_to_bd[7] = $_GET['05'];
```

```
$arr_to_bd[8] = $_GET['06'];
```

```
$arr_to_bd[9] = $_GET['07'];
```

```
$arr_to_bd[10] = $_GET['08'];
```

```
$arr_to_bd[11] = $_GET['09'];
```

```
$arr_to_bd[12] = $_GET['10'];
```

```
$arr_to_bd[13] = $_GET['11'];
```

```
$arr_to_bd[14] = $_GET['12'];
```

```
$arr_to_bd[15] = $_GET['smsp'];
```

```
$arr_to_bd[16] = $_GET['month'];
```

```
$arr_to_bd[17] = $_GET['year'];
```

```
//$records_counter = dbase_numrecords( $bd);
```

```
dbase_add_record ($bd, $arr_to_bd);  
  
//dbase_replace_record ($bd, $arr_to_bd, $records_counter);  
  
echo "<br> $temp";  
  
echo "Запис добавлено. <br>";  
  
dbase_close($bd);  
  
echo "<A HREF='\"\\SP1_Zl_in.html\"'> Ввести нову квитанцію. </A>";  
  
?>
```

Додаток 3. Скрипт створення файлу бази даних сортувальної таблиці.

```
<?php  
  
$bd_1 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_1_SP5.DBF", 0);  
$bd_2 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_2_SP5.DBF", 0);  
$bd_3 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_3_SP5.DBF", 0);  
$bd_4 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_4_SP5.DBF", 0);  
$bd_5 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_5_SP5.DBF", 0);  
$bd_6 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_6_SP5.DBF", 0);  
$bd_7 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_7_SP5.DBF", 0);  
$bd_8 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_8_SP5.DBF", 0);  
$bd_9 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_9_SP5.DBF", 0);  
$bd_10 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_10_SP5.DBF", 0);  
$bd_11 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_11_SP5.DBF", 0);  
$bd_12 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_12_SP5.DBF", 0);  
$bd_13 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_13_SP5.DBF", 0);  
$bd_14 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_14_SP5.DBF", 0);  
$bd_15 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_15_SP5.DBF", 0);
```

```
$bd_16 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_16_SP5.DBF", 0);
$bd_17 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_17_SP5.DBF", 0);
$bd_18 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_18_SP5.DBF", 0);
$bd_19 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_19_SP5.DBF", 0);
$bd_20 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_20_SP5.DBF", 0);
$bd_21 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_21_SP5.DBF", 0);
$bd_22 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_22_SP5.DBF", 0);
$bd_23 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_23_SP5.DBF", 0);
$bd_24 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_24_SP5.DBF", 0);
$bd_25 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_25_SP5.DBF", 0);
$bd_26 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_26_SP5.DBF", 0);
$bd_27 = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_27_SP5.DBF", 0);
$bd_ZI = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_ZI_SP5.DBF", 0);
$bd_ST = dbase_open("C:\\Apache2\\htdocs\\BD\\2004_04_ST.DBF", 2);

$records_counter = dbase_numrecords($bd_1);

//Внесения данных по ВЗ Б.К
for ($i=1; $i<=$records_counter; $i++) {
    $temp_record = dbase_get_record ($bd_1, $i);
    $records_counter_ST = dbase_numrecords($bd_ST);
    for ($e=1; $e<=$records_counter_ST; $e++) {
        $temp_record_ST = dbase_get_record ($bd_ST, $e);
        if ($temp_record[0] == $temp_record_ST[0]) {
            for ($q=0;$q<=29;$q++)
                $temp_rec[$q] = $temp_record_ST[$q];
        }
    }
}
```

```
$temp_rec[2] = $temp_record_ST[2] + $temp_record[6];  
dbase_replace_record ($bd_ST, $temp_rec, $e);  
break;  
}  
}  
}
```

//Внесення даних по ВЗ Бібщани

```
for ($i=1; $i<=$records_counter; $i++) {  
$temp_record = dbase_get_record ($bd_2, $i);  
$records_counter_ST = dbase_numrecords($bd_ST);  
for ($e=1; $e<=$records_counter_ST; $e++) {  
$temp_record_ST = dbase_get_record ($bd_ST, $e);  
if ($temp_record[0] == $temp_record_ST[0]) {  
for ($q=0; $q<=29; $q++)  
$temp_rec[$q] = $temp_record_ST[$q];  
$temp_rec[3] = $temp_record_ST[3] + $temp_record[6];  
dbase_replace_record ($bd_ST, $temp_rec, $e);  
break;  
}  
}  
}
```

//Внесення даних по ВЗ Бортків

```
for ($i=1; $i<=$records_counter; $i++) {  
$temp_record = dbase_get_record ($bd_3, $i);
```

```
$records_counter_ST = dbase_numrecords($bd_ST);  
for ($e=1; $e<=$records_counter_ST; $e++) {  
    $temp_record_ST = dbase_get_record ($bd_ST, $e);  
    if ($temp_record[0] == $temp_record_ST[0]) {  
        for ($q=0;$q<=29;$q++)  
            $temp_rec[$q] = $temp_record_ST[$q];  
        $temp_rec[4] = $temp_record_ST[4] + $temp_record[6];  
        dbase_replace_record ($bd_ST, $temp_rec, $e);  
        break;  
    }  
}  
}
```

*//Внесення даних по ВЗ В.Вільшаниця*

```
for ($i=1; $i<=$records_counter; $i++) {  
    $temp_record = dbase_get_record ($bd_4, $i);  
    $records_counter_ST = dbase_numrecords($bd_ST);  
    for ($e=1; $e<=$records_counter_ST; $e++) {  
        $temp_record_ST = dbase_get_record ($bd_ST, $e);  
        if ($temp_record[0] == $temp_record_ST[0]) {  
            for ($q=0;$q<=29;$q++)  
                $temp_rec[$q] = $temp_record_ST[$q];  
            $temp_rec[5] = $temp_record_ST[5] + $temp_record[6];  
            dbase_replace_record ($bd_ST, $temp_rec, $e);  
            break;  
        }  
    }  
}
```

```
}  
}  
}
```

```
//Внесення даних по ВЗ Вороняки
```

```
for ($i=1; $i<=$records_counter; $i++) {  
    $temp_record = dbase_get_record ($bd_5, $i);  
    $records_counter_ST = dbase_numrecords($bd_ST);  
    for ($e=1; $e<=$records_counter_ST; $e++) {  
        $temp_record_ST = dbase_get_record ($bd_ST, $e);  
        if ($temp_record[0] == $temp_record_ST[0]) {  
            for ($q=0;$q<=29;$q++)  
                $temp_rec[$q] = $temp_record_ST[$q];  
            $temp_rec[6] = $temp_record_ST[6] + $temp_record[6];  
            dbase_replace_record ($bd_ST, $temp_rec, $e);  
            break;  
        }  
    }  
}
```

```
//Внесення даних по ВЗ Глиняни
```

```
for ($i=1; $i<=$records_counter; $i++) {  
    $temp_record = dbase_get_record ($bd_6, $i);  
    $records_counter_ST = dbase_numrecords($bd_ST);  
    for ($e=1; $e<=$records_counter_ST; $e++) {  
        $temp_record_ST = dbase_get_record ($bd_ST, $e);
```

```
if ($temp_record[0] == $temp_record_ST[0]) {  
  for ($q=0;$q<=29;$q++)  
    $temp_rec[$q] = $temp_record_ST[$q];  
  $temp_rec[7] = $temp_record_ST[7] + $temp_record[6];  
  dbase_replace_record ($bd_ST, $temp_rec, $e);  
  break;  
}  
}  
}
```

//Внесения данных по ВЗ Гологори

```
for ($i=1; $i<=$records_counter; $i++) {  
  $temp_record = dbase_get_record ($bd_7, $i);  
  $records_counter_ST = dbase_numrecords($bd_ST);  
  for ($e=1; $e<=$records_counter_ST; $e++) {  
    $temp_record_ST = dbase_get_record ($bd_ST, $e);  
    if ($temp_record[0] == $temp_record_ST[0]) {  
      for ($q=0;$q<=29;$q++)  
        $temp_rec[$q] = $temp_record_ST[$q];  
      $temp_rec[8] = $temp_record_ST[8] + $temp_record[6];  
      dbase_replace_record ($bd_ST, $temp_rec, $e);  
      break;  
    }  
  }  
}
```



```
//Внесения даних по В3
```

```
for ($i=1; $i<=$records_counter; $i++) {  
    $temp_record = dbase_get_record ($bd_8, $i);  
    $records_counter_ST = dbase_numrecords($bd_ST);  
    for ($e=1; $e<=$records_counter_ST; $e++) {  
        $temp_record_ST = dbase_get_record ($bd_ST, $e);  
        if ($temp_record[0] == $temp_record_ST[0]) {  
            for ($q=0;$q<=29;$q++)  
                $temp_rec[$q] = $temp_record_ST[$q];  
            $temp_rec[9] = $temp_record_ST[9] + $temp_record[6];  
            dbase_replace_record ($bd_ST, $temp_rec, $e);  
            break;  
        }  
    }  
}
```

```
//Внесения даних по В3
```

```
for ($i=1; $i<=$records_counter; $i++) {  
    $temp_record = dbase_get_record ($bd_9, $i);  
    $records_counter_ST = dbase_numrecords($bd_ST);  
    for ($e=1; $e<=$records_counter_ST; $e++) {  
        $temp_record_ST = dbase_get_record ($bd_ST, $e);  
        if ($temp_record[0] == $temp_record_ST[0]) {  
            for ($q=0;$q<=29;$q++)  
                $temp_rec[$q] = $temp_record_ST[$q];  
        }  
    }  
}
```

```

$temp_rec[10] = $temp_record_ST[10] + $temp_record[6];
dbase_replace_record ($bd_ST, $temp_rec, $e);
break;
}
}
}

//Внесения даних по В3
for ($i=1; $i<=$records_counter; $i++) {
    $temp_record = dbase_get_record ($bd_10, $i);
    $records_counter_ST = dbase_numrecords($bd_ST);
    for ($e=1; $e<=$records_counter_ST; $e++) {
        $temp_record_ST = dbase_get_record ($bd_ST, $e);
        if ($temp_record[0] == $temp_record_ST[0]) {
            for ($q=0; $q<=29; $q++)
                $temp_rec[$q] = $temp_record_ST[$q];
            $temp_rec[11] = $temp_record_ST[11] + $temp_record[6];
            dbase_replace_record ($bd_ST, $temp_rec, $e);
            break;
        }
    }
}

//Внесения даних по В3
for ($i=1; $i<=$records_counter; $i++) {
    $temp_record = dbase_get_record ($bd_11, $i);

```

```

$records_counter_ST = dbase_numrecords($bd_ST);
for ($e=1; $e<=$records_counter_ST; $e++) {
$temp_record_ST = dbase_get_record ($bd_ST, $e);
if ($temp_record[0] == $temp_record_ST[0]) {
for ($q=0;$q<=29;$q++)
$temp_rec[$q] = $temp_record_ST[$q];
$temp_rec[12] = $temp_record_ST[12] + $temp_record[6];
dbase_replace_record ($bd_ST, $temp_rec, $e);
break;
}
}
}

```

*//Внесения данных по ВЗ*

```

for ($i=1; $i<=$records_counter; $i++) {
$temp_record = dbase_get_record ($bd_12, $i);
$records_counter_ST = dbase_numrecords($bd_ST);
for ($e=1; $e<=$records_counter_ST; $e++) {
$temp_record_ST = dbase_get_record ($bd_ST, $e);
if ($temp_record[0] == $temp_record_ST[0]) {
for ($q=0;$q<=29;$q++)
$temp_rec[$q] = $temp_record_ST[$q];
$temp_rec[13] = $temp_record_ST[13] + $temp_record[6];
dbase_replace_record ($bd_ST, $temp_rec, $e);
break;
}
}
}

```

```
}  
}  
}
```

```
//Внесения даних по В3
```

```
for ($i=1; $i<=$records_counter; $i++) {  
    $temp_record = dbase_get_record ($bd_13, $i);  
    $records_counter_ST = dbase_numrecords($bd_ST);  
    for ($e=1; $e<=$records_counter_ST; $e++) {  
        $temp_record_ST = dbase_get_record ($bd_ST, $e);  
        if ($temp_record[0] == $temp_record_ST[0]) {  
            for ($q=0;$q<=29;$q++)  
                $temp_rec[$q] = $temp_record_ST[$q];  
            $temp_rec[14] = $temp_record_ST[14] + $temp_record[6];  
            dbase_replace_record ($bd_ST, $temp_rec, $e);  
            break;  
        }  
    }  
}
```

```
//Внесения даних по В3
```

```
for ($i=1; $i<=$records_counter; $i++) {  
    $temp_record = dbase_get_record ($bd_14, $i);  
    $records_counter_ST = dbase_numrecords($bd_ST);  
    for ($e=1; $e<=$records_counter_ST; $e++) {  
        $temp_record_ST = dbase_get_record ($bd_ST, $e);
```

```

if ($temp_record[0] == $temp_record_ST[0]) {
for ($q=0;$q<=29;$q++)
$temp_rec[$q] = $temp_record_ST[$q];
$temp_rec[15] = $temp_record_ST[15] + $temp_record[6];
dbase_replace_record ($bd_ST, $temp_rec, $e);
break;
}
}
}

//Внесения данных по ВЗ
for ($i=1; $i<=$records_counter; $i++) {
$temp_record = dbase_get_record ($bd_15, $i);
$records_counter_ST = dbase_numrecords($bd_ST);
for ($e=1; $e<=$records_counter_ST; $e++) {
$temp_record_ST = dbase_get_record ($bd_ST, $e);
if ($temp_record[0] == $temp_record_ST[0]) {
for ($q=0;$q<=29;$q++)
$temp_rec[$q] = $temp_record_ST[$q];
$temp_rec[16] = $temp_record_ST[16] + $temp_record[6];
dbase_replace_record ($bd_ST, $temp_rec, $e);
break;
}
}
}
}

```

```
//Внесения даних по В3
for ($i=1; $i<=$records_counter; $i++) {
$temp_record = dbase_get_record ($bd_16, $i);
$records_counter_ST = dbase_numrecords($bd_ST);
for ($e=1; $e<=$records_counter_ST; $e++) {
$temp_record_ST = dbase_get_record ($bd_ST, $e);
if ($temp_record[0] == $temp_record_ST[0]) {
for ($q=0;$q<=29;$q++)
$temp_rec[$q] = $temp_record_ST[$q];
$temp_rec[17] = $temp_record_ST[17] + $temp_record[6];
dbase_replace_record ($bd_ST, $temp_rec, $e);
break;
}
}
}
```

```
//Внесения даних по В3
for ($i=1; $i<=$records_counter; $i++) {
$temp_record = dbase_get_record ($bd_17, $i);
$records_counter_ST = dbase_numrecords($bd_ST);
for ($e=1; $e<=$records_counter_ST; $e++) {
$temp_record_ST = dbase_get_record ($bd_ST, $e);
if ($temp_record[0] == $temp_record_ST[0]) {
for ($q=0;$q<=29;$q++)
$temp_rec[$q] = $temp_record_ST[$q];
```

```

$temp_rec[18] = $temp_record_ST[18] + $temp_record[6];
dbase_replace_record ($bd_ST, $temp_rec, $e);
break;
}
}
}

//Внесения данных по В3
for ($i=1; $i<=$records_counter; $i++) {
    $temp_record = dbase_get_record ($bd_18, $i);
    $records_counter_ST = dbase_numrecords($bd_ST);
    for ($e=1; $e<=$records_counter_ST; $e++) {
        $temp_record_ST = dbase_get_record ($bd_ST, $e);
        if ($temp_record[0] == $temp_record_ST[0]) {
            for ($q=0; $q<=29; $q++)
                $temp_rec[$q] = $temp_record_ST[$q];
            $temp_rec[19] = $temp_record_ST[19] + $temp_record[6];
            dbase_replace_record ($bd_ST, $temp_rec, $e);
            break;
        }
    }
}

//Внесения данных по В3
for ($i=1; $i<=$records_counter; $i++) {
    $temp_record = dbase_get_record ($bd_19, $i);

```

```
$records_counter_ST = dbase_numrecords($bd_ST);  
for ($e=1; $e<=$records_counter_ST; $e++) {  
    $temp_record_ST = dbase_get_record ($bd_ST, $e);  
    if ($temp_record[0] == $temp_record_ST[0]) {  
        for ($q=0;$q<=29;$q++)  
            $temp_rec[$q] = $temp_record_ST[$q];  
        $temp_rec[20] = $temp_record_ST[20] + $temp_record[6];  
        dbase_replace_record ($bd_ST, $temp_rec, $e);  
        break;  
    }  
}  
}
```

*//Внесения данных по ВЗ*

```
for ($i=1; $i<=$records_counter; $i++) {  
    $temp_record = dbase_get_record ($bd_20, $i);  
    $records_counter_ST = dbase_numrecords($bd_ST);  
    for ($e=1; $e<=$records_counter_ST; $e++) {  
        $temp_record_ST = dbase_get_record ($bd_ST, $e);  
        if ($temp_record[0] == $temp_record_ST[0]) {  
            for ($q=0;$q<=29;$q++)  
                $temp_rec[$q] = $temp_record_ST[$q];  
            $temp_rec[21] = $temp_record_ST[21] + $temp_record[6];  
            dbase_replace_record ($bd_ST, $temp_rec, $e);  
            break;  
        }  
    }  
}
```



```
}  
}  
}
```

```
//Внесения даних по В3
```

```
for ($i=1; $i<=$records_counter; $i++) {  
    $temp_record = dbase_get_record ($bd_21, $i);  
    $records_counter_ST = dbase_numrecords($bd_ST);  
    for ($e=1; $e<=$records_counter_ST; $e++) {  
        $temp_record_ST = dbase_get_record ($bd_ST, $e);  
        if ($temp_record[0] == $temp_record_ST[0]) {  
            for ($q=0;$q<=29;$q++)  
                $temp_rec[$q] = $temp_record_ST[$q];  
            $temp_rec[22] = $temp_record_ST[22] + $temp_record[6];  
            dbase_replace_record ($bd_ST, $temp_rec, $e);  
            break;  
        }  
    }  
}
```

```
//Внесения даних по В3
```

```
for ($i=1; $i<=$records_counter; $i++) {  
    $temp_record = dbase_get_record ($bd_22, $i);  
    $records_counter_ST = dbase_numrecords($bd_ST);  
    for ($e=1; $e<=$records_counter_ST; $e++) {  
        $temp_record_ST = dbase_get_record ($bd_ST, $e);
```

```

if ($temp_record[0] == $temp_record_ST[0]) {
for ($q=0;$q<=29;$q++)
$temp_rec[$q] = $temp_record_ST[$q];
$temp_rec[23] = $temp_record_ST[23] + $temp_record[6];
dbase_replace_record ($bd_ST, $temp_rec, $e);
break;
}
}
}

//Внесения даних по ВЗ
for ($i=1; $i<=$records_counter; $i++) {
$temp_record = dbase_get_record ($bd_23, $i);
$records_counter_ST = dbase_numrecords($bd_ST);
for ($e=1; $e<=$records_counter_ST; $e++) {
$temp_record_ST = dbase_get_record ($bd_ST, $e);
if ($temp_record[0] == $temp_record_ST[0]) {
for ($q=0;$q<=29;$q++)
$temp_rec[$q] = $temp_record_ST[$q];
$temp_rec[24] = $temp_record_ST[24] + $temp_record[6];
dbase_replace_record ($bd_ST, $temp_rec, $e);
break;
}
}
}
}

```

```
//Внесения даних по В3
for ($i=1; $i<=$records_counter; $i++) {
$temp_record = dbase_get_record ($bd_24, $i);
$records_counter_ST = dbase_numrecords($bd_ST);
for ($e=1; $e<=$records_counter_ST; $e++) {
$temp_record_ST = dbase_get_record ($bd_ST, $e);
if ($temp_record[0] == $temp_record_ST[0]) {
for ($q=0;$q<=29;$q++)
$temp_rec[$q] = $temp_record_ST[$q];
$temp_rec[25] = $temp_record_ST[25] + $temp_record[6];
dbase_replace_record ($bd_ST, $temp_rec, $e);
break;
}
}
}
```

```
//Внесения даних по В3
for ($i=1; $i<=$records_counter; $i++) {
$temp_record = dbase_get_record ($bd_25, $i);
$records_counter_ST = dbase_numrecords($bd_ST);
for ($e=1; $e<=$records_counter_ST; $e++) {
$temp_record_ST = dbase_get_record ($bd_ST, $e);
if ($temp_record[0] == $temp_record_ST[0]) {
for ($q=0;$q<=29;$q++)
$temp_rec[$q] = $temp_record_ST[$q];
```

```

$temp_rec[26] = $temp_record_ST[26] + $temp_record[6];
dbase_replace_record ($bd_ST, $temp_rec, $e);
break;
}
}
}

//Внесения даних по В3
for ($i=1; $i<=$records_counter; $i++) {
$temp_record = dbase_get_record ($bd_26, $i);
$records_counter_ST = dbase_numrecords($bd_ST);
for ($e=1; $e<=$records_counter_ST; $e++) {
$temp_record_ST = dbase_get_record ($bd_ST, $e);
if ($temp_record[0] == $temp_record_ST[0]) {
for ($q=0; $q<=29; $q++)
$temp_rec[$q] = $temp_record_ST[$q];
$temp_rec[27] = $temp_record_ST[27] + $temp_record[6];
dbase_replace_record ($bd_ST, $temp_rec, $e);
break;
}
}
}

//Внесения даних по В3
for ($i=1; $i<=$records_counter; $i++) {
$temp_record = dbase_get_record ($bd_27, $i);

```

```
$records_counter_ST = dbase_numrecords($bd_ST);  
for ($e=1; $e<=$records_counter_ST; $e++) {  
    $temp_record_ST = dbase_get_record ($bd_ST, $e);  
    if ($temp_record[0] == $temp_record_ST[0]) {  
        for ($q=0;$q<=29;$q++)  
            $temp_rec[$q] = $temp_record_ST[$q];  
        $temp_rec[28] = $temp_record_ST[28] + $temp_record[6];  
        dbase_replace_record ($bd_ST, $temp_rec, $e);  
        break;  
    }  
}  
}
```

*//Внесення даних по ВЗ Золочів*

```
$records_counter = dbase_numrecords($bd_ZI);  
for ($i=1; $i<=$records_counter; $i++) {  
    $temp_record = dbase_get_record ($bd_ZI, $i);  
    $records_counter_ST = dbase_numrecords($bd_ST);  
    for ($e=1; $e<=$records_counter_ST; $e++) {  
        $temp_record_ST = dbase_get_record ($bd_ST, $e);  
        if ($temp_record[0] == $temp_record_ST[0]) {  
            $temp_rec[0] = $temp_record_ST[0];  
            $temp_rec[1] = $temp_record_ST[1];  
            for ($q=2;$q<29;$q++)  
                $temp_rec[$q] = $temp_record_ST[$q];
```

```
$temp_rec[29] = $temp_record_ST[29] + $temp_record[6];  
dbase_replace_record ($bd_ST, $temp_rec, $e);  
break;  
}  
}  
}  
dbase_close ($bd_1);  
dbase_close ($bd_2);  
dbase_close ($bd_3);  
dbase_close ($bd_4);  
dbase_close ($bd_5);  
dbase_close ($bd_6);  
dbase_close ($bd_7);  
dbase_close ($bd_8);  
dbase_close ($bd_9);  
dbase_close ($bd_10);  
dbase_close ($bd_11);  
dbase_close ($bd_12);  
dbase_close ($bd_13);  
dbase_close ($bd_14);  
dbase_close ($bd_15);  
dbase_close ($bd_16);  
dbase_close ($bd_17);  
dbase_close ($bd_18);
```

```

dbase_close ($bd_19);
dbase_close ($bd_20);
dbase_close ($bd_21);
dbase_close ($bd_22);
dbase_close ($bd_23);
dbase_close ($bd_24);
dbase_close ($bd_25);
dbase_close ($bd_26);
dbase_close ($bd_27);
dbase_close ($bd_Z1);
dbase_close ($bd_ST);
?>

```

Додаток 4. Скрипт виводу сортувальної таблиці для друку.

```
<?php
```

```
$bd = dbase_open ("C:\\Apache2\\htdocs\\BD\\2004_04_ST.DBF", 0);
```

```
$rec_counter = dbase_numrecords ($bd);
```

```
echo "<TABLE BORDER>
```

```
";
```

```
echo "<TR><TD>Індекс</TD><TD>Назва
```

```

видання</TD><TD>Б.К</TD><TD>Біб</TD><TD>Бор</TD><TD>В.В</TD><T
D>Вор</TD><TD>ГЛН</TD><TD>Гол</TD><TD>Гонч</TD><TD>Єл</TD><T
D>Жук</TD><TD>Заст</TD><TD>Кур</TD><TD>Кол</TD><TD>Нов</TD><
TD>Підг</TD><TD>Плг</TD><TD>Пом</TD><TD>Пол</TD><TD>Поч</TD>
<TD>Ремз</TD><TD>Сас</TD><TD>Скв</TD><TD>Слв</TD><TD>Снв</TD
><TD>Стр</TD><TD>Чер</TD><TD>Ясн</TD><TD>Зол</TD></TR>";

```

```

for ($i=1; $i<=$rec_counter; $i++) {
// if ($i==100) break;

$temp_record2 = dbase_get_record ($bd, $i);

if ($temp_record2[2] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

echo "<TD>$temp_record2[1]</TD>";

for ($t=2; $t<29; $t++)

if ($temp_record2[$t] == 0)

echo "<TD><CENTER>&#160</CENTER></TD>";

else

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[3] != 0){

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[4] != 0){

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

```



```

else

if ($temp_record2[5] != 0){

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[6] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[7] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[28]</CENTER></TD></TR>";}

else

if ($temp_record2[8] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[28]</CENTER></TD></TR>";}

```

```
else

if ($temp_record2[9] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[10] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[11] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[12] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}
```

```
else

if ($temp_record2[13] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[14] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[15] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[16] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}
```

```

else

if ($temp_record2[17] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[18] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[19] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[20] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

```

```
else

if ($temp_record2[21] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[22] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[23] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[24] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}
```

```

else

if ($temp_record2[25] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[26] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[27] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

else

if ($temp_record2[28] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

```

```
else

if ($temp_record2[29] != 0) {

echo "<TR><TD><CENTER>$temp_record2[0]</CENTER></TD>";

for ($t=1; $t<29; $t++)

echo "<TD><CENTER>$temp_record2[$t]</CENTER></TD>";

echo "<TD><CENTER>$temp_record2[29]</CENTER></TD></TR>";}

}

echo "

</TABLE>

";

dbase_close ($bd);

?>
```