А.М. Пєтух, В.В. Войтко, С.В. Бевз, С.А.Яремко

МЕРЕЖІ ЕОМ

ЛАБОРАТОРНИЙ ПРАКТИКУМ

Міністерство освіти і науки України

Вінницький національний технічний університет

А.М. Пєтух, В.В. Войтко, С.В. Бевз, С.А. Яремко

МЕРЕЖІ ЕОМ

ЛАБОРАТОРНИЙ ПРАКТИКУМ

Затверджено Ученою радою Вінницького національного технічного університету як навчальний посібник для студентів спеціальності 7.080403 – “Програмне забезпечення автоматизованих систем”. Протокол № 5 від “27” грудня 2001 року.

Вінниця ВНТУ 2003

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

А.М. Пєтух, В.В. Войтко, С.В. Бевз, С.А.Яремко

МЕРЕЖІ ЕОМ

Лабораторний практикум

Усі цитати, цифровий, фактичний матеріал та бібліографічні відомості перевірені, написання одиниць відповідає стандартам. Зауваження рецензентів враховані.

Вимогам, які використовуються до навчальної літератури, відповідає. До друку і в світ дозволяю на підставі п.2 пункту 15 "Єдиних правил…".

Автори: _____А.М. Пєтух
_____В.В. Войтко
_____С.В. Бевз

Проректор з навчальної та науково-методичної роботи
В.О. Леонтьєв
_____С.А. Яремко
(підпис)

ЗАТВЕРДЖЕНО НА ЗАСІДАННІ КАФЕДРИ ПМОС

Протокол № 6 від 6.11.2001.
Зав. каф. ПМОС _____А.М.Пєтух
(підпис)

Вінниця ВНТУ 2003

Навчальне видання

Пєтух А.М., Войтко В.В., Бевз С.В., Яремко С.А.

МЕРЕЖІ ЕОМ

Лабораторний практикум

ОРИГІНАЛ-МАКЕТ ПІДГОТОВЛЕНО АВТОРАМИ

РЕДАКТОР С.А. МАЛІШЕВСЬКА

Навчально-методичний відділ ВНТУ
Свідоцтво Держкомінформу України
Серія ДК №746 від 25.12.2001 р.
21021, м. Вінниця, Хмельницьке шосе, 95, ВНТУ

Підписано до друку
Формат 29,7*42 $^{1}/_{4}$ Гарнітура Times New Roman
Друк різографічний Папір офсетний
Тираж ___прим. Ум. друк. арк.
Зам. №

Віддруковано в комп'ютерному інформаційно-видавничому центрі
Вінницького національного технічного університету
Свідоцтво Держкомінформу України
Серія ДК №746 від 25.12.2001 р.
21021, м.Вінниця, Хмельницьке шосе, 95

УДК 681.3
П 31

Рецензенти:
С.М. Злепко, доктор технічних наук, професор
В.П. Кожем'яко, доктор технічних наук, професор
В.І. Сачанюк, голова правління ВАТ "Інфракон"

Рекомендовано до видання Ученою радою Вінницького національного технічного університету Міністерства освіти і науки України

Пєтух А.М., Войтко В.В., Бевз С.В., Яремко С.А.
П 31 **Мережі ЕОМ.** Лабораторний практикум.
Навчальний посібник. – Вінниця: ВНТУ, 2003 – 125 с.
У посібнику розглянуті особливості створення програмних мережних засобів та адміністрування мережних операційних систем. Посібник розроблений у відповідності з планом кафедри та програмою до дисципліни "Комп'ютерні мережі".

УДК 681.3

ВСТУП

Дисципліна “Комп’ютерні мережі” передбачена навчальним планом для студентів бакалаврського напрямку “Комп’ютерні науки”; базується на курсах: “Основи програмування та алгоритмічних мов”, “Організація і функціонування ЕОМ та систем”; є теоретичною основою курсів: “Інформаційні мережні технології”, “Методи і засоби комп’ютерних інформаційних технологій”. У результаті вивчення дисципліни студенти повинні вміти створювати мережні програмні засоби; вільно орієнтуватися в мережі INTERNET; створювати програми-сервери та клієнти; мати навички адміністрування сучасних мережних операційних систем.

Навчальний посібник “Мережі ЕОМ. Лабораторний практикум” складається з двох розділів: 1 – “Створення програмних мережних засобів” (який є дуже важливим саме для студентів спеціальності 7.080403 – “Програмне забезпечення автоматизованих систем”) та 2 – “Адміністрування мережних операційних систем”, де розглядаються особливості адміністрування сучасних мережних операційних систем Windows NT, Windows 2000, Novell Netware. При виконанні лабораторних робіт 1-го розділу студент має попередньо ознайомитися з теоретичними відомостями та контрольним прикладом роботи, написати програму відповідно до завдання та перевірити її працездатність. Порядок виконання лабораторних робіт 2-го розділу, залежно від теми заняття, подається окремо у кожній роботі .

У посібнику велика увага приділяється формуванню коротких теоретичних відомостей, які містять базові рекомендації та пояснення щодо виконання лабораторної роботи. Особливої уваги заслуговує подання контрольного прикладу, яким завершуються короткі теоретичні відомості кожної лабораторної роботи. У контрольному прикладі повністю наводиться розв’язування одного варіанту завдання, що дає можливість студентам різного рівня підготовки добре розібратися в основних питаннях роботи. Такий підхід є особливо важливим для студентів заочної форми навчання, оскільки, навіть студент мало знайомий з питаннями дисципліни, уважно прочитавши короткі теоретичні відомості та проглянувши контрольний приклад, зможе самостійно виконати лабораторну роботу.

І. СТВОРЕННЯ ПРОГРАМНИХ МЕРЕЖНИХ ЗАСОБІВ

Лабораторна робота №1

ВИКОРИСТАННЯ ФУНКЦІЙ МОВИ ПРОГРАМУВАННЯ VISUAL C++ ПРИ НАПИСАННІ ПРОГРАМ ПЕРЕДАЧІ ДАНИХ У МЕРЕЖАХ.

Мета роботи: ознайомитись з основними принципами передачі даних в мережах та набути практичні навички з використання функцій мови Visual C++ при написанні програм передачі даних.

Варіанти завдань

1. Написати програму, що пересилає заданий файл за заданою адресою.
2. Написати програму-сервер, яка розсилає інформацію за вказаними адресами клієнтів. А також написати програму-клієнт , що здатна приймати інформацію.
3. Написати програму-сервер, яка в залежності від запиту клієнта відсилає йому той чи інший файл.
4. Написати програму, що пересилає вхідний файл пакетами розміром 1000 байт.
5. Написати програму, що здійснює передачу даних. Адреси користувачів задаються явно.
6. Написати програму, що здійснює передачу даних. Адреси користувачів задаються за замовчуванням.
7. Написати програму, яка використовувала б сокет дейтаграм при передачі даних.
8. Написати програму, яка використовувала б сокет потоків при передачі даних.
9. Написати програму, яка демонструє роботу функцій *accept()* та *connect()*.
10. Написати програму, яка демонструє роботу функцій *send()* та *recv()*.

Контрольний приклад. Напишіть програму, яка пересилає файл data.dat з комп'ютера, що має IP–адресу 10.8.1.15, на комп'ютер з IP–адресою 10.8.1.16, причому так, щоб ім'я вхідного файлу можна було змінити.

Теоретичні відомості

Мережна обробка даних. Використання сокетів

Найбільша віддача від сокетів виявляється при програмуванні мережних додатків для двох учасників діалогу. Наприклад, при розробці додатків для клієнта і сервера сокети використовуються для обміну структурами даних

чи пакетами. Якщо ж розробляється тільки додаток-клієнт, переважно використовують один з інтерфейсів Інтернет високого рівня "Winlnet API", але ми зупинимося на використанні API WinSock.

Що таке сокет?

Сокет — це абстрактний об'єкт для позначення одного з кінців мережного з'єднання. Він призначений для створення механізму обміну даними. Сокети Windows підрозділяються на дві категорії: сокети дейтаграм і сокети потоків. Якщо ви хочете гарантовану, безперебійну доставку даних у потрібному порядку і без дублювання, використовуйте сокети потоків, що є надійною, основаною на з'єднаннях системою обміну даними. Сокети потоків особливо корисні при пересиланні великих обсягів даних без втрат, дублювання і порушення порядку. Крім того, при закритті з'єднання додаток отримає повідомлення про цю подію.

Ініціалізація WinSock. Перший параметр функції WSAStartup() — це значення типу WORD, що визначає максимальний номер версії WinSock, доступний додатку. Перша цифра версії знаходиться в молодшому байті, друга — в старшому. Другий параметр функції містить покажчик на структуру WSADATA. Нижченаведений приклад виклику функції

WSAStartup().

```
#include<winsock.h>
WSADATA wsadata;
Int nRc = WSAStartup(0*0101,&WSAData)
If(nRc)
{
    //Ошибка при инициализации WinSock
    return;
}
if(WSAData.wVersion !=0*0101)
{
    //Поддерживаемая версия не подходит
    //Сообщить об ошибке и завершить работу
    WSACleanup();
    Return;
}
```

Функція WSAStartupt() повертає значення WSASYSNOTREADY, якщо динамічна бібліотека підтримки WinSock або відповідна підсистема мережі неініціалізована, ініціалізована некоректно або не знайдена. Крім того, за допомогою цієї функції додаток повідомляє системі версію WinSock, що повинна використовуватися. Як правило, при викликанні функції WSAStartup() необхідно вказувати максимальний допустимий номер версії. Якщо він менше, ніж версії, підтримувані даною динамічною

бібліотекою, функція WSAStartup поверне значення WSAVERNOTSUPPORTED.

Якщо запитуваний номер версії більше чи дорівнює номеру версії, що підтримується динамічною бібліотекою, поле wVersion структури WSAData буде містити номер версії, що повинна використовуватися, а поле wHighVersion — максимальний номер версії, підтримуваний даною бібліотекою.

Якщо версія в полі wVersion для роботи додатка не підходить, потрібно викликати функцію WSACleanup() і або завершити роботу програми, або спробувати скористатися іншою бібліотекою wsock32.dll — звичайно, для цього її необхідно динамічно завантажити.

Крім номера версії до структури WSADATA входить і інша корисна інформація, у тому числі текстові рядки з описом реалізації WinSock, максимальне число сокетів, доступних процесу, і максимально допустимий розмір дейтаграм.

Створення сокета

Для створення сокета, що буде використовуватися при обміні даними, використовується функція socket (). Вона має три параметри: формат адреси, тип сокета і протокол. Для WinSock версії 1.1 виклик функції socket () , як правило, має вигляд:

```
// Створення сокета дейтаграм.
SOCKET myUDPSOCK = socket (AF_INET, SOCK_DGRAM,
IPPROTO_UDP) ;
// Створення сокета потоку.
SOCKET myUDPSOCK =socket (AF__INET, SOCK_STREAM,
IPPROTOJTCP);
```

Перший параметр означає, що з цим сокетом будуть використовуватися адреси Інтернет; наступні два аргументи конструктора задають тип створюваного сокета і протокол обміну даними через нього. У наведеному прикладі створюються сокет дейтаграм, що використовує протокол UDP, і сокет потоку, що використовує протокол TCP. Це два найпоширеніші типи сокетів у мережах із протоколом TCP/IP, хоча конкретна реалізація WinSock може підтримувати й інші протоколи.

Функція bind() При виклику функції bind() вона отримує дескриптор сокета і покажчик на структуру адреси, а також довжину цієї структури. Дескриптор сокета — це просто значення, що повернув конструктор socket () , а от зі структурою адреси виникають деякі проблеми. Склад цієї структури може залежати (і звичайно залежить) від застосовуваного мережного протоколу, хоча в даному випадку ми розглядаємо тільки адреси TCP/IP.

Для вказання IP-адреси при виклику функції bind() використовується структура sockaddt_in:

```
struct   sockaddr_in   {
short        sin_family;
u_short,   sin_port;
struct     in_addr   sin_addt:;
char si_n_zero[8] ;
};
```

Для IP-адрес поле sin_family завжди має значення AF_INET. Поле sin_port визначає порт, що буде асоційований із сокетом, а поле sin_addr задає фізичну адресу сокета, Поле sin_zero — це просто "заглушка".

У полі sin_addr структури sockaddr_in зберігається фізична IP-адреса комп'ютера у форматі структури in_addr, описаної в файлі заголовку winsock.h (чи winsock2.h для версії 2).

Розмір структури in_addr дорівнює 4 байти, кожний з яких утримує одне з чисел у нотації IP-адреси. Наприклад. IP-адреса 1.2.3.4 у форматі цієї структури подається у вигляді 0x04030201. Подання різних типів даних у вигляді об'єднання (union) дозволяє додатку обробляти дану структуру найбільш зручним для нього способом.

У більшості викликів функції bind() замість поля s_addr можна підставляти INADDR_ANY. Це дозволяє сокету посилати чи приймати дані через будь-яку IP-адресу даного комп'ютера. Звичайно комп'ютер має тільки одну IP-адресу, хоча, в принципі, на ньому може бути встановлено кілька мережних адаптерів, кожний зі своєю IP-адресою. Якщо сокет повинен використовувати тільки одну з них, її необхідно вказати явно. Для цього нерідко використовується функція inet_addr(), що приймає як аргумент ASCII-рядок десяткової нотації IP-адреси з крапкою і повертає змінну типу u_long, що містить цю адресу у форматі поля s_addr. Крім неї, існує функція inet__ntoa (), що виконує обернене перетворення, приймаючи перемінну типу u_long і повертаючи адресу у вигляді ASCII-рядка.

Функції обробки імен і адрес

Для полегшення обробки адрес і портів, використовуваних при виклику функції bind(), в інтерфейсі WinSock є кілька корисних допоміжних функцій.

Функція gothostname() повертає рядок, що містить ім'я локального комп'ютера. У залежності від реалізації WinSock це може бути просто ідентифікатор комп'ютера, наприклад, keymaster, чи повне ім'я в домені, наприклад, keymaster.racotek.com.

Для одержання адреси комп'ютера за його ідентифікатором використовується функція gethostbyname(), що приймає як аргумент рядок

з ім'ям комп'ютера і розшукує адресу, що відповідає цьому імені. Цей пошук може виконуватися локально в списку адрес на даному комп'ютері чи шляхом посилання запиту на сервер імен, у залежності від конфігурації мережі. Функція gethostbyname() повертає структуру hostent, що містить список всіх адрес, які відповідають зазначеному імені, а також інформацію про тип адреси і псевдоніми імені вузла.

Функції асинхронного розпізнавання. У тому випадку, якщо перераховані вище функції посилають запити на сервер по мережі, їхнє виконання може блокуватися на невизначений проміжок часу. Щоб дозволити додатку виконувати під час очікування відповіді на запит інші операції, можна скористатися такими функціями WinSock API: WSAAsyncGetHostByName ( ), WSAAsyncGetHostByAddr ( ), WSAGetServ ByName ( ), WSAAsyncGetServfBy-Port ( ), WSAAsyncGetProtoByName ( ) і WSAAsyncGetPhotoByNumber ( ).

Для цих функцій можна задати системне повідомлення Windows, що буде посилатися зазначеному процесору команд вікна при завершенні викликаної функції. Звичайно це спеціальне повідомлення, що стосується тільки користувача. Воно обробляється за допомогою списку повідомлень; приклад запису з цього списку приведений нижче (у ньому використовується повідомлення WM_USER+1:

```
ON_COMMAND ( WM_USER+1 ,    OnAsyncLookup)
```

Прототип функції процесора команд має такий вигляд:

```
LONG  OnAsynoLockup(WPARAM wParam, LPAPAM   lParam);
```

Ця функція зчитує код помилки і довжину буфера з параметра lParam з використанням макросів WSAGETASYNCERROR ( ) і WSAGETASYNC BUFLEN ( ). Крім того, wParam містить ідентифікатор асинхронного процесора задач, який працює при викликанні функції WSAAsync… .

Якщо необхідно скасувати асинхронний запит до його завершення, використовується функція WSACancelAsyncRequest ( ), у яку передається ідентифікатор запиту, що повертається функцією WSAAsync ( ) .

Створення сокета для сервера

Розглянувши прив'язку адреси до сокета, перейдемо до використання сокета. У наступному прикладі розглядається створення сокета з номером порту 5150:

```
SOCKET  sServSock;
sockaddr_in   addr;
int nSockErr;
```

```
sServSock  =-   socket(AF_INET,
SOCK_DGRAM,IPPROTO_UDP) ;
addr.sin_family=  AF_INET;
addr.sin_port  =  htons ( 5150) ;
addr.sin_addr.s_addr  =  htonl (INADDR_ANY) ;
if(bind(sServSock,(LPSOCKADDR)&addr,sizeof(addr))==
SOCKET_ERROR)
{
nSockErr =  WSAGetLastError();
// Обробка   помилки.
}
```

Прийом даних на сокеті дейтаграм

Після створення сокета в додатку-сервері і прив'язання його до визначеної адреси і порту можна приймати дані від додатка-клієнта. Це здійснюється за допомогою функції recvf rorn(), яка має такий прототип:

```
int   recvfrom (SOCKET   s, char FAR * buf, int  len,
int   flags, struct   sockaddr   FAR *from, int FAR
*fromlen) ;
```

Перший параметр — це дескриптор сокета, що повертається функцією socket(); за ним йдуть покажчик на буфер для прийому нової дейтаграми і довжина буфера. Параметр flags може дорівнювати G_PEEK для заповнення буфера таким чином, що дейтаграма залишається у вхідній черзі. Останні два параметри використовуються для одержання адреси сокета, який відіслав дейтаграму. За цією адресою можна надіслати відповідь.

При успішному зчитуванні дейтаграми функція recvfrom() повертає кількість прочитаних байтів. При помилковому зчитуванні дейтаграми повертається значення SOCKET_ERROR; для одержання коду помилки необхідно викликати функцію WSAGetLastError ().

Пересилання даних із сокета дейтаграм

Відправлення даних виконується функцією sendto():

```
int sendto (SOCKET s, const char FAR *buf,int len,
int flags, const struct sock-addr FAR *to,int to1en);
```

Тут перший параметр, як і раніше, є дескриптором сокета; за ним знаходиться покажчик на буфер, що містить дані для пересилання, і довжина цього буфера. Останні два параметри використовуються для вказування адреси і порту сокета призначення.

Якщо функція sendto() спрацьовує коректно, вона повертає кількість посланих байтів. У випадку помилки функція sendto() повертає SOCKET_ERROR. Для визначення коду помилки необхідно викликати функцію WSAGetLastError ().

При спробі послати дейтаграму більшого розміру, ніж максимально припустимий у даній реалізації WinSock, код помилки буде дорівнювати WSAEMSGSIZE. Максимально припустимий розмір дейтаграми можна визначити шляхом виклику функції WSAStartup ().

Використання сокета потоку. Через те, що сокети потоків виконують дії над протоколами на основі з'єднань, їхнє використання трохи відрізняється від використання сокетів дейтаграм. Так спочатку необхідно встановити з'єднання, а потім уже зчитувати дані з потоку.

Сокети потоків на сервері

Як і сокети дейтаграм, сокети потоків для процесів-серверів створюються шляхом виклику функції socket () і прив'язуються до визначеної адреси функцією bind ():

```
SOCKET   sServSock;
Sockaddr_in   addr;
int.  nSockErr;
sServSock   =   socket(AF_INET,    SOCK_STREAM,
0);
addr.sin_family   =  AF_INET;
addr.sin_port   =   htons ( 5050) ;
addr.sin_addr.s_addr   =   htonl(INADDR_ANY) ;
if(bind("sServSock, (LPSOCKADDR) &addr,    sizeof
(addr ) ) == SOCKET_ERROR)
{
nSockErr  =   WSAGetLastError ( ) ;
//   Обробка   помилки
}
```

Єдина різниця між цим прикладом і прикладом для сокетів дейтаграм полягає в передачі аргументу SOCK_STREAM замість SOCK_DGRAM у функцію socket() . Крім того, як ідентифікатор протоколу передається 0, тобто інтерфейс WinSock автоматично вибирає протокол за сімейством адрес і типом сокета.

Очікування запитів на відкриття з'єднання

Після створення сокета і прив'язування його до адреси необхідно встановити з'єднання з клієнтом. Для цього використовується функція listen() int   listen (SOCKET   s, int backlog);

Виклик цієї функції ініціює очікування запиту клієнта на відкриття з'єднання. Параметр backlog містить кількість запитів, які повинні надійти для того, щоб додаток погодився встановити з'єднання. Наприклад, якщо цей параметр дорівнює 2 і додаток з якихось причин відмовив відкрити з'єднання, третій клієнт, що спробує підключитися до сервера, отримає код помилки WSAECONNREFUSED. Перші два запити будуть відправлені в чергу для наступної обробки сервером.

Відкриття з'єднання

При одержанні запиту клієнта відкриття з'єднання виконується за допомогою функції accept():

```
SOCKET accept (SOCKET s,  struct  sockaddr   FAR
*addr, int FAR *addrlen) ;
```

У якості першого параметру передається сокет, що очікує запит. Другий і третій параметри використовуються для одержання адреси сокета клієнта, що запитує з'єднання. Якщо з'єднання відкривається успішно, функція accept() повертає дескриптор на новий сокет, що буде використовуватися для керування новим з'єднанням. Якщо сталася помилка, функція accept () повертає код INVALID_SOCKET, і для одержання більш докладної інформації про помилку необхідно викликати функцію WSAGetLastError () .

Вихідний сокет продовжить очікувати запити на нові з'єднання, які будуть відкриватися знову ж таки за допомогою функції accept(). Кожне відкрите з'єднання керується окремим сокетом, дескриптор якого повертається з цієї функції:

```
SOCKET sServSock
sockadd r_in  addr:
i nt  nSockErr ;
int. NNumConns=0 ;
SOCKET sConns [5] ;
Sockaddr ConnAddrs[5];
int  nAddrken = sizeof(sockaddr);
// Створення- сокету.
SServSook -  socket(AF_INET, SOCK_ STREAM, 0) ;
// Асоціювання сокета з портом.
Addr.sin_ family = AF_INET;
adr.sin.port = htons (5050) ;
addr.sin_addr.s_addr = htonl(INADDR_ANY);
if (bind (s ServSock, ( LPSOCKADDR) &addr , sizeof
(addr) ) == SOCKET_ERROR)
{
```

```
nSockErr = WSAGetLastError () ;
// Обробка помилки... Завершення.
}
// Очікування запиту клієнта на відкриття з'єднання.
 If(listen(sServSock,2} == SOCKET_ERROR)
{
nSockErr =WSAGetLasrError ( ) ;
 // Обробка  помилки... Завершення.
}
while(nNumConns < 5)
 {
// Прийом запиту і відкриття з'єднання.
 SConns[nNumConns] = accept(sServSock,
ConnAddrs[nNumConns], &nAddrLen);
if(sConns[nNumConns] == INVALID_SOCKET)
 {
nSockEr r == WSAGetLastErrоr() ;
/ / Обробка помилки
}
else {
//Успішне з'єднання з новим сокетом.
 StartNewHandlerThread(sConns[nNumConns]);
 nNumConns++
 } // Кінець циклу while
```

У цьому прикладі для сервера створюється сокет, який прив’язується до адреси і очікує запит клієнта на відкриття з’єднання. Допускається відкриття до п’яти окремих з’єднань; при цьому в масиві зберігається дескриптор сокета і відповідна адреса, а потім для керування з’єднань запускається новий процес-поток.

Запит на відкриття з’єднання

Щоб сервер міг прийняти запит на відкриття з'єднання, цей запит має бути відісланим. Додаток-клієнт здійснює це за допомогою функції connect():

```
int: connect (SOCKET s, const struct sockaddr FAR*
name, int namelen);
```

Параметри sockaddr і namelen використовуються для вказування адреси і порту, до якого необхідно приєднатися. Структура sockaddr, передана у функцію connect(), повинна бути ідентичною структурі, переданій у функцію bind() на сервері. Тут використовуються відомі

заздалегіть адреси портів, які привласнюються сокетам сервера. Розглянемо приклад, у якому створюється сокет клієнта, прив'язуючись до визначеної адреси і надсилає запит на з'єднання.

```
SOCKET   SClntSock;
sockaddr_in   addr;
 sockaddr_in   ServerAddr ;
 int  nSockErr;
//  Створення   сокета,
SClntSock   =   socket (AF_INET,    SOCK_STREAM, 0) ;
//  Присвоєння   адреси   новому   сокету,
Addr.sin_family  =  AF_INET;
addr.sin_port.   =   0;    //   порт   асоціюється
автоматично
addr.sin_addr.s_addr  =   htonl (INADDR _ANY) ;
if (bind (sClntSock, (LPSOCKADDR) &addr, sizeof
(addr)) == SOCKET_ERROR)
{
nSock.Err  =  WSAGetLastError ( ) ;
//  Обробка   помилки   і   завершення.
}
//  Підключення   до  сервера.
 ServerAddr.sin_family   =  AF_INET;
ServerAddr.sin_port.   =   htons ( 5050) ;
ServerAddr.sin_addr.s_addr   =   inet_addr
("192.9.200.93") ;
if (connect (sClntSock,    ServerAddr,    sizeof
(ServerAddr) )  ==  SOCKET_ERROR)
{
nSockErr   =:   WSAGetLas tError ( ) ;
//  Обробка   помилки   і   завершення.
}
```

Відправлення і приймання даних через сокети потоків

Після встановлення з'єднання між сокетом клієнта і сокетом сервера пересилка даних виконується за допомогою функції send():

```
int  send  (SOCKET s, const chaк FAR  *huf, int
len,    int   flags);
```

Функція send() приймає як аргументи покажчик на буфер, який містить дані для пересилання, і його довжину, а також параметр flags. Якщо цей параметр дорівнює MSG_DONTROUTE, у набір даних, який пересилається, не включається інформація про маршрутизацію; якщо його

значення дорівнює MSG_OOB, посилається потік привілейованих (out-of-band) даних.

Обсяг даних, які пересилаються одним викликом функції send() , не повинен перевищувати розміру пакета, максимально допустимого в даній мережі. При спробі пересилання більшого обсягу даних функція send() завершиться аварійно, а функція WSAGetLastError() поверне код помилки WSAEMSGSIZE.

Для прийому даних через сокет потоку використовується функція recv(). Ось її прототип:

```
int recv (SOCKET s, char FAR* buf, int len, int
flags);
```

Параметри buf і len визначають відповідно буфер для прийому даних і його довжину. Параметр flags може приймати значення MSG_OOB для прийому привілейованих даних чи MSG_PEEK для заповнення буфера без видалення даних з вхідної черги.

Якщо у вхідній черзі знаходяться дані для сокета, функція recv() повертає кількість прочитаних байтів, яка дорівнює обсягу доступних даних у вхідній черзі і не перевершує значення len. При коректному закритті з'єднання повертається значення 0, а при аварійному — значення SCCKET_ERROR. Для визначення точного коду помилки необхідно викликати функцію WSAGetLastError() .

Використання функції select ()

У UNIX-реалізації сокетів BSD для модифікації файлових дескрипторів використовувалася функція ioctl(). Оскільки сокети WinSock— це не зовсім файлові дескриптори, необхідно використовувати функцію ioctlsocket() аналогічного призначення. Ця функція може використовуватися для активізації сокета в неблокуючому режимі:

```
ioctlsocket(s, FIONBIO, 1);
```

У неблокуючому режимі виклик функції, яка не може бути виконана негайно, завершується аварійно і повертає код помилки. При цьому функція WSAGetLastError () повертає код WSAEWOULDBLOCK.

За допомогою функції select() додаток може перевіряти стан набору сокетів чи блоку паралельно з очікуванням події. Одним з найбільш розповсюджених застосувань функції select(); є вибір сокетів для очікування прийому даних. Як тільки дані приходять на будь-який з обраних сокетів, функція select () повертає значення, яке містить інформацію про сокети з даними у вхідних чергах. Ці дані можна одержати за допомогою функції recvfrom().

Закриття сокета

Для завершення роботи сокета його необхідно закрити за допомогою функції closesocket():

```
int  closesocket(SOCKET   s);
```

Ця функція має єдиний аргумент — дескриптор сокета, який закривається. Про поточні параметри можна довідатися, викликавши функцію getsockopt(). Результат роботи функції closesocket() визначається параметрами SO_LINGER і SO_DONTLINGER.

Якщо параметр SO_DONTLINGER дорівнює TRUE, функція closesocket() поверне значення негайно, але перед закриттям сокета буде здійснена спроба пересилання усіх даних, які залишилися. Такий прийом називається коректним закриттям.

Якщо параметр SO_LINGER дорівнює TRUE; і встановлена ненульова затримка, то виконується коректне закриття зі спробою пересилання усіх даних, що залишилися в буфері, але функція closesocket() не повертає значення доти, доки не будуть переслані всі дані чи поки не мине термін затримки. Якщо параметр SO_LINGER дорівнює TRUE і затримка дорівнює нулю, сокет закривається негайно, а всі дані, що залишилися в буфері, втрачаються.

Контрольний приклад

Напишіть програму, яка пересилає файл data.dat з комп'ютера, що має IP–адресу 10.8.1.15, на комп'ютер з IP–адресою 10.8.1.16, причому так, щоб ім'я вхідного файлу можна було б змінити.

Client.exe - завантажується з комп'ютера, який передає дані.

Server.exe - завантажується з комп'ютера, який отримує дані.

```
Client.cpp
void CClientDlg::OnButton2()
{
    SOCKET       con_handle;
    sockaddr_in  addr;
    sockaddr_in  con_addr;
    WSADATA      WSAData;
    int          kol;

    UpdateData();

    WSAStartup(0x0101,&WSAData);
    con_handle=socket(AF_INET,SOCK_STREAM,0);
```

```
    addr.sin_addr.s_addr=htonl(INADDR_ANY);
    addr.sin_family=AF_INET;
    addr.sin_port=0;
    if(bind(con_handle,(LPSOCKADDR)&addr,sizeof(socka
ddr_in))==SOCKET_ERROR)
    {
        int error=WSAGetLastError();
        CString st;
        st.Format("%d",error);
        m_Edit2="Error binding"+st;
        UpdateData(FALSE);
        return;
    }
    con_addr.sin_port=htons(5050);
    con_addr.sin_family=AF_INET;
    con_addr.sin_addr.s_addr=inet_addr("10.8.1.16");

    if(connect(con_handle,(const
sockaddr*)&con_addr,sizeof(sockaddr_in))==SOCKET_ERRO
R)
    {
        int error=WSAGetLastError();
        CString st;
        st.Format("%d",error);
        m_Edit2="Error connection"+st;
        UpdateData(FALSE);
        return;
    }
    CString st=ID_PLAY_MUSIC;
    send(con_handle,st,st.GetLength()+1,0);
    CFile
fil("1.wav",CFile::modeRead|CFile::typeBinary);
    char buffer[MAX_FILE_LENGTH];

    ULONG count=fil.GetLength()/MAX_FILE_LENGTH+1;
    send(con_handle,(char*)&count,sizeof(ULONG),0);
    m_Progress.SetRange32(0,count);
    m_Progress.SetPos(0);
    m_Progress.SetStep(1);
    for(ULONG i=0;i<count;i++)
    {
        kol=fil.Read(buffer,MAX_FILE_LENGTH);
        send(con_handle,buffer,kol,0);
        m_Progress.StepIt();
        Pause(NUM_PAUSE);
```

```
    }
    fil.Close();
        Pause(NUM_PAUSE);
    kol=recv(con_handle,buffer,4,0);
    m_Edit2=buffer;

    closesocket(con_handle);

    UpdateData(FALSE);
}
Server.cpp
BOOL CServerDlg::OnInitDialog()
{
    CDialog::OnInitDialog();
    SOCKET       handle;
    SOCKET       con_handle;
    sockaddr_in  addr;
    sockaddr con_addr;
    int          con_len=sizeof(con_addr);
    int          kol;
    char     buffer[MAX_BUFFER];
    WSADATA      WSAData;
    WSAStartup(0x0101,&WSAData);
    handle=socket(AF_INET,SOCK_STREAM,0);
    addr.sin_addr.s_addr=htonl(INADDR_ANY);
    addr.sin_family=AF_INET;
    addr.sin_port=htons(5050);
    if(bind(handle,
(LPSOCKADDR)&addr,sizeof(sockaddr_in))==SOCKET_ERROR)
    {
        int error=WSAGetLastError();
        CString st;
        st.Format("%d",error);
        MessageBox("binding error"+st);
        return FALSE;
    }
    if(listen(handle,2)==SOCKET_ERROR)
    {
        int error=WSAGetLastError();
        CString st;
        st.Format("%d",error);
        MessageBox("listening error"+st);
        return FALSE;
    }
    while(1)
```

```
    {
    if((con_handle=accept(handle,&con_addr,&con_len))
== INVALID_SOCKET)
    {
        int error=WSAGetLastError();
        CString st;
        st.Format("%d",error);
        MessageBox("acepting error"+st);
        return FALSE;
    }

if((kol=recv(con_handle,buffer,MAX_BUFFER,0))==SOCKET
_ERROR)
    {
        int error=WSAGetLastError();
        CString st;
        st.Format("%d",error);
        MessageBox("recving error"+st);
        return FALSE;
    }
    CString st(buffer);
             if(st==ID_PLAY_MUSIC)
             {
                 charbuffer[MAX_FILE_LENGTH];
                 ULONG   count;
                 recv(con_handle,(char*)&count,4,0);
                 CFile
fil("c:\\1.wav",CFile::modeWrite|CFile::typeBinary|CF
ile::modeCreate);
                 for(ULONG i=0;i<count;i++)
                 {

    kol=recv(con_handle,buffer,MAX_FILE_LENGTH,0);
                     fil.Write(buffer,kol);
                     CString st;
                     st.Format("%5d %d\n",i,kol);
                 Pause(NUM_PAUSE);
                 }
                 fil.Close();
                 Pause(NUM_PAUSE*5);
                 BOOL
b=sndPlaySound("c:\\1.wav",SND_NODEFAULT);
                 CString st;
                 if(b)
                     st="yes";
```

```
                else
                    st="no ";

    send(con_handle,st,st.GetLength()+1,0);
            }
    closesocket(con_handle);
    }
    closesocket(handle);

    return TRUE;  // return TRUE unless you set the
focus to a control
                  // EXCEPTION: OCX Property Pages
should return FALSE
}
```

Питання для самоперевірки

1.Що таке сокет та де він використовується?
2.Як створюється сокет?
3.Особливості використання функції bind().
4. Як відбувається приймання даних на сокеті дейтаграм?
5. Як відбувається передача даних на сокеті дейтаграм?
6. Як встановити з'єднання між сокетом, прив'язаним до певної адреси, та клієнтом?
7. Як відбувається відкриття з'єднання при використанні сокетів потоків?
8. Як відбувається посилання запиту на відкриття з'єднання?
9. Розкажіть як працює функція send().
10.Розкажіть як працює функція recv().
11.Розкажіть як працює функція select().
12.Розкажіть як працює функція closesocket().

Література

1.Дэвид Беннет. Руководство разработчика Visual C++ 6. —К., 1998. — 810 с.
2. Кейн Грегори. Использование Visual C++ 6. — М., 2000. — 650 с.

Лабораторна робота №2

ВИКОРИСТАННЯ ФУНКЦІЙ МОВИ ПРОГРАМУВАННЯ DELPHI ДЛЯ НАПИСАННЯ ЧАТУ (ПРОГРАМ “КЛІЄНТ”, “СЕРВЕР”) ПРИ РОБОТІ У МЕРЕЖАХ.

Мета роботи: ознайомитись з основними принципами написання програм “клієнт”, “сервер” на мові програмування Delphi.

Варіанти завдань

Написати програму - чат з такими функціональними можливостями:

- вибору режимів роботи (“активний”, “не турбувати”, “відключений”). У режимі “активний” повідомлення приходять у висхідних вікнах та супроводжуються звуковим сигналом. В режимі “не турбувати” – у мінімізованих вікнах та без звукового супроводу, а в режимі “відключений” – повідомлення не приймаються;
- паралельного ведення окремого діалогу між двома користувачами при загальній кількості користувачів N>2;
- фільтрації повідомлень;
- підключення кількох каналів;
- підключення таймера для повідомлень;
- автоматичного ведення протоколу повідомлень;
- конвертування розкладки клавіатури за допомогою гарячих клавіш;
- визначення IP адреси комп’ютера абонента мережі;
- контролю посилання та приймання повідомлень.
- ознайомлення із списком усіх користувачів мережі, які на даний момент знаходяться в чаті та можливістю поновлення списку через певні інтервали часу.

Теоретичні відомості

Мережна обробка даних. Використання сокетів

Усі мережні програми поділяються на два класи: сервери і клієнти. Відмінність між ними, в основному, полягає у способі з'єднання. Програма-сервер повинна прийняти запит на з'єднання від програми-клієнта і відповісти на нього, а Програма-клієнт надсилає запит на з'єднання програмі-серверу і чекає на відповідь. Після цього розходження між сервером і клієнтом зникають. Між ними відбувається діалог, у якому

обидві сторони рівноправні. Правила організації діалогу визначаються протоколом.

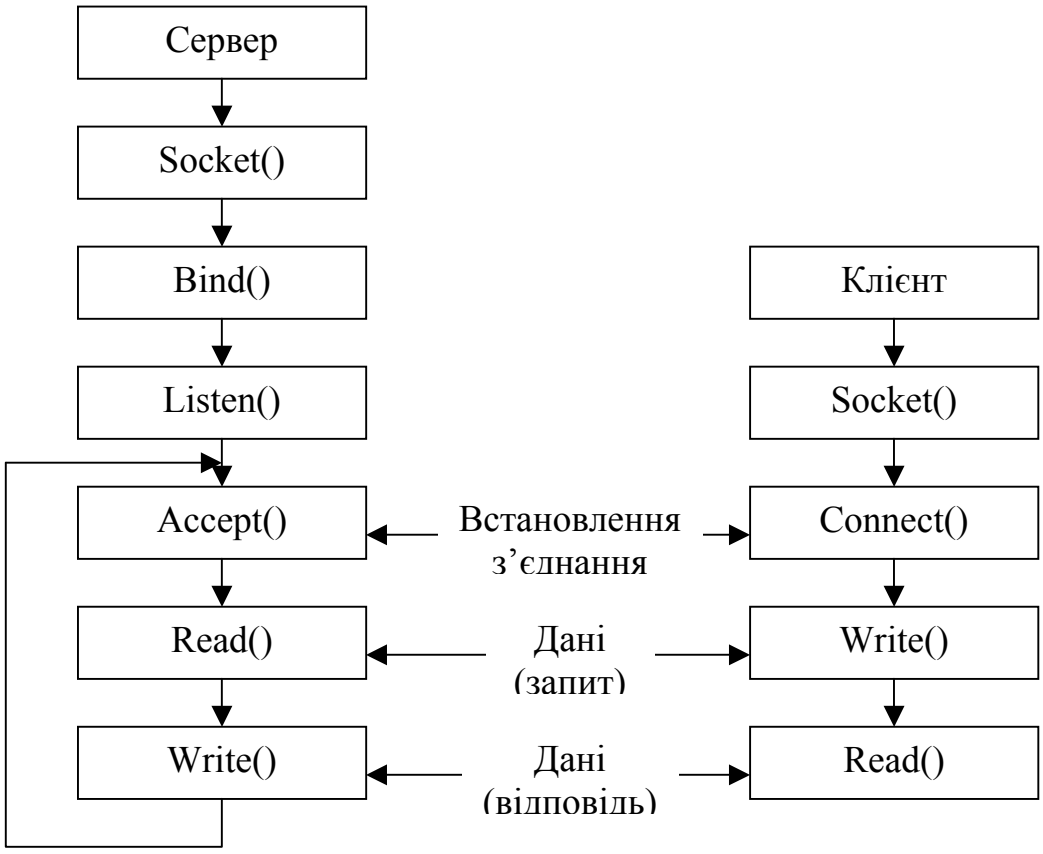


Рисунок 2.1

Між сервером і клієнтом створюється тимчасовий канал зв'язку. Читання і запис відбувається через спеціальні об'єкти — сокети (socket-гніздо). Сокети мають багато загального з описувачами файлів (створюваними, наприклад, функціями *CreateFile* у Windows чи *open* у UNI). Для запису і читання можна використовувати стандартні функції *read* і *write*. Сокеты створюються функцією ***socket()*** і знищуються функцією ***closesocket(). Socket()*** повертає ціле число, що є дескриптором створеного сокета і використовується у всіх випадках для операцій із сокетами. Прямого доступу до вмісту сокета-об'єкта непередбачено, але існують функції ***select()***, ***ioctlsocket()*** у Windows і ***ioctl()*** у UNI, ***getsockopt()***, ***setsockopt()***, що дозволяють маніпулювати поведінкою сокетів і одержувати інформацію чи є в сокеті дані для читання, запису, помилки та ін.

Написати програму клієнта нескладно. Написати програму сервера дещо складніше. Основні проблеми – синтаксичний аналіз запитів, що складає велику частину програми.

Програма-клієнт

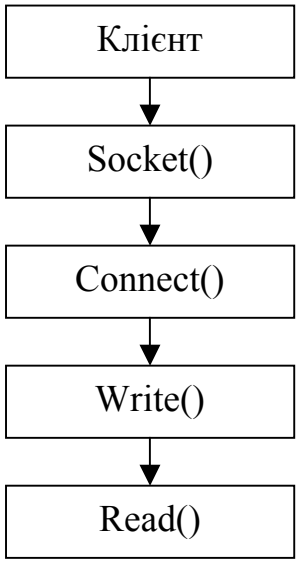


Рисунок 2.2

Розглянемо програму-клієнт. Основний цикл виглядає так:

- створити сокет. Після створення можна установити деякі параметри сокета, наприклад, блокування та неблокування;
- з'єднати сокет із сервером;
- послати серверу запит;
- прийняти відповідь. Можливо, ці два пункти повторяться - у залежності від протоколу, що використовується;
- закрити сокет. Одночасно розривається з'єднання із сервером, якщо сервер не розірвав з'єднання першим.

Основні модулі програми-клієнта подані в табл. 2.1.

Таблиця 2.1 — Основні модулі програми-клієнта

Назва модуля	**Характеристика модуля**
client.dpr	Файл опису проекту Delphi
UMain.dfm	Файл опису вікна Form3
UMain.pas	Модуль з текстом програми

Програма створює вікно, в якому потрібно явно задати адресу сервера, вказати ім’я користувача та при готовності сервера під’єднатись до нього натисканням клавіші “connect”.

Основні модулі програми-сервера подані в табл. 2.2.

Таблиця 2.2 — Основні модулі програми-сервера

Назва модуля	**Характеристика модуля**
Server.dpr	Файл проекту Delphi
Umain.dfm	Файл опис вікна програми
UMain.pas	Текст модуля програми

WWW сервер діє за алгоритмом:

- створення сокета (socket);
- поєднання сокета з локальним портом (bind);
- надсилання повідомлення сокету (listen);
- перехід у режим чекання запиту (accept);
- читає запит (read);
- створення відповіді (write);
- знову переходить у режим чекання.

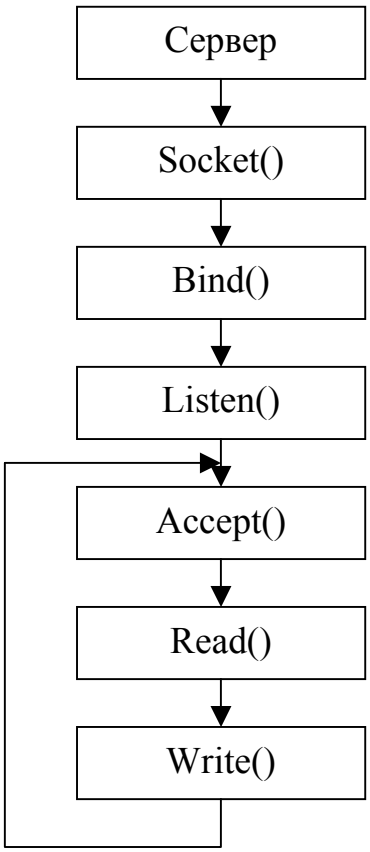


Рисунок 2.3

Контрольний приклад

Напишіть програму–чат (з використанням структури “клієнт” – “сервер”), яка забезпечує з’єднання кількох “клієнтів” із одним “сервером” та між собою, IP–адреса сервера для клієнтів вказується явно (наприклад, 10.8.1.15), вказується ім’я користувача .

Client.exe та **Server.exe** завантажуються на комп’ютерах, які підключені до однієї мережі.Нижче подано текст основних модулів програм **Client.exe** та **Server.exe.**

```
UMain.pas (client)
unit UMain;
interface
uses
```

```
  Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs,
  StdCtrls, ScktComp, Buttons, ExtCtrls, ComCtrls,
Gauges;
type
  TForm3 = class(TForm)
    cs: TClientSocket;
    mMsg: TMemo;
    eMsg: TEdit;
    eAddr: TEdit;
    eUser: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    SpeedButton1: TSpeedButton;
    procedure csRead(Sender: TObject; Socket:
TCustomWinSocket);
    procedure eMsgKeyDown(Sender: TObject; var Key:
Word;
      Shift: TShiftState);
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    pathexe : string;
  end;
var
  Form3: TForm3;
implementation
{$R *.DFM}
procedure TForm3.csRead(Sender: TObject; Socket:
TCustomWinSocket);
begin
 mMsg.Lines.Add(Socket.ReceiveText);
end;
procedure TForm3.eMsgKeyDown(Sender: TObject; var
Key: Word;
  Shift: TShiftState);
begin
if key = 13 then
begin
  cs.Socket.SendText('<' + eUser.text + '>  ' +
eMsg.text);
  eMsg.Clear;
```

```
end; end;
procedure TForm3.Button1Click(Sender: TObject);
begin
 cs.Active := false;
 cs.Address := eAddr.Text;
 cs.Active := true;
end;
procedure TForm3.FormCreate(Sender: TObject);
begin
 cs.Socket.SendText('<' + eUser.text + '>  ' +
'покинул сеть');
  pathexe := ExtractFilePath(Application.ExeName);
end;
end.
UMain.pas (server)
unit UMain;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs,
  StdCtrls, ScktComp;
type
  TForm2 = class(TForm)
    ss: TServerSocket;
    Button1: TButton;
    Button2: TButton;
    Memo1: TMemo;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure ssClientRead(Sender: TObject; Socket:
TCustomWinSocket);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form2: TForm2;
implementation
{$R *.DFM}
procedure TForm2.Button1Click(Sender: TObject);
begin
 ss.Active := true;
end;
```

```
procedure TForm2.Button2Click(Sender: TObject);
begin
  ss.Active := false;
end;
procedure TForm2.ssClientRead(Sender: TObject;
Socket: TCustomWinSocket);
 var
    i : integer;
    s : string;
begin
 s := Socket.ReceiveText;
 memo1.Lines.Add(s);
 try  i := 0;
  repeat
    ss.Socket.Connections[i].SendText(s);
    inc(i);
  until false; except end; end; end.
```

Питання для самоперевірки

1. Особливості та основні принципи побудови структури “клієнт”.
2. Особливості та основні принципи побудови структури “сервер”.
3. Особливості взаємодії та побудови діалогу “клієнт” – ”сервер”.
4. Правила організації діалогу за допомогою протоколів.
5. Що таке сокет, де він використовується та які основні принципи його створення?
6. Розкажіть як працює функція socket().
7. Розкажіть як працює функція closesocket().
8. Використання функцій getsockopt(), setsockopt().
9. Як встановити з’єднання між сокетом, прив’язаним до певної адреси, та клієнтом?
10. Як відбувається посилання запиту на відкриття з’єднання при використанні сокетів потоків?

Література

1. Том Сван. Основы программирования в Delphi для Windows 95. – К.: Диалектика, 1996. – 540 с.
2. Даптеманн Джефф, Мишель Джим, Тейлор Дон. Программирование в среде Delphi. – К.: Диалектика, 1995. – 608 с.

Лабораторна робота №3

ЗАБЕЗПЕЧЕННЯ МОЖЛИВОСТІ ВІДОБРАЖЕННЯ ПРОЦЕСУ ЗАВАНТАЖЕННЯ ІНФОРМАЦІЇ З ВІДДАЛЕНОГО КОМП'ЮТЕРА

Мета роботи: отримати відомості про організацію роботи java-аплетів та написати java-аплет для забезпечення можливості інформування користувача про стан процесу завантаження інформації.

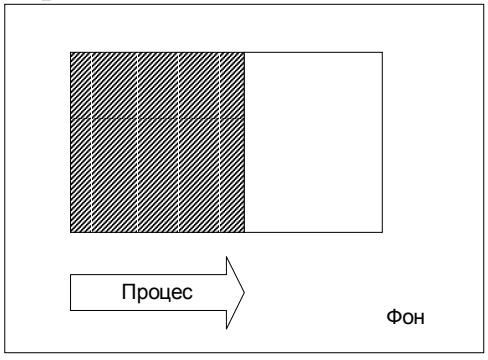
Завдання на лабораторну роботу

1. Ознайомитися з теоретичними відомостями.
2. Відповідно до завдання розробити спосіб ви
3. ведення зображень
4. Написати "каркас" java-аплету.
5. Написати функцію виведення зображення відповідно до варіанта завдання.
6. Враховуючи результати пунктів 4,5 створити повноцінний аплет, який забезпечує виведення зображення у відповідності з варіантом.
7. Внести до звіту результат роботи написаної програми.
8. Зробити необхідні висновки.

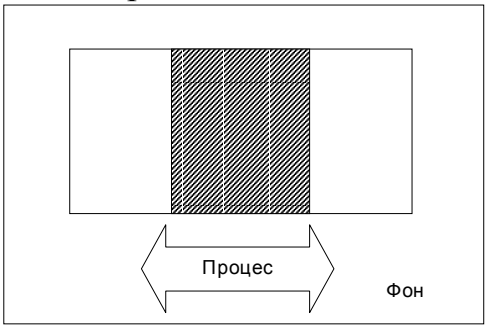
Завдання

Забезпечити виведення зображення згідно з варіантом :

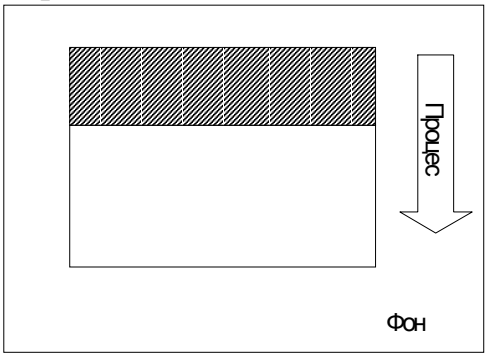
Варіант 1



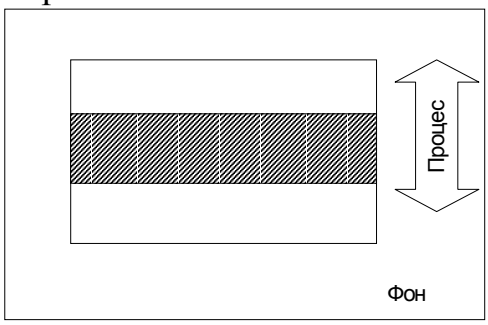
Варіант 2



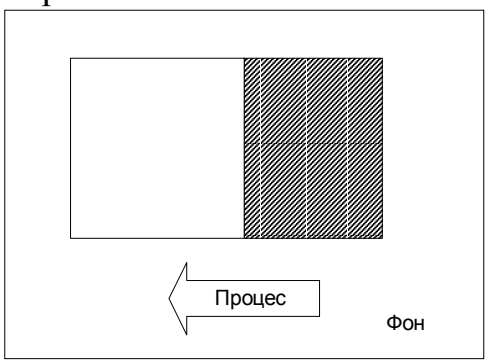
Варіант 3



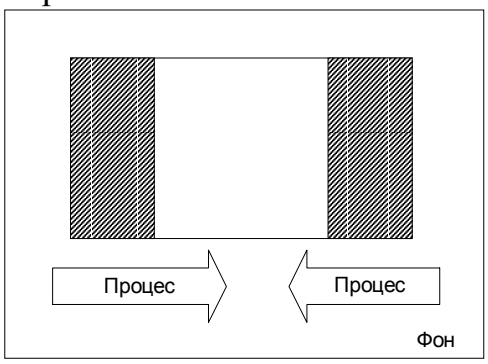
Варіант 4



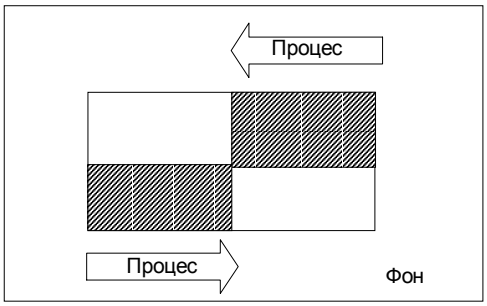
Варіант 5



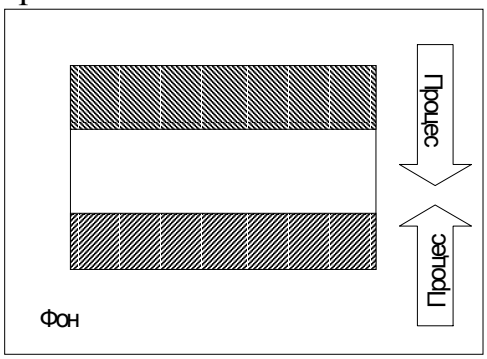
Варіант 6



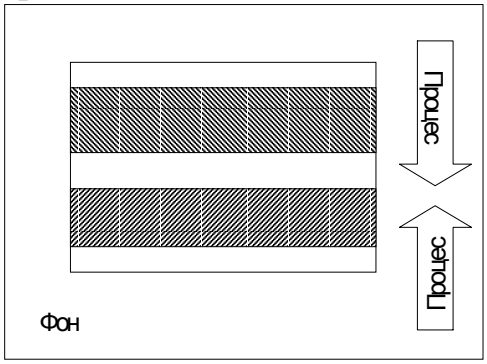
Варіант 7



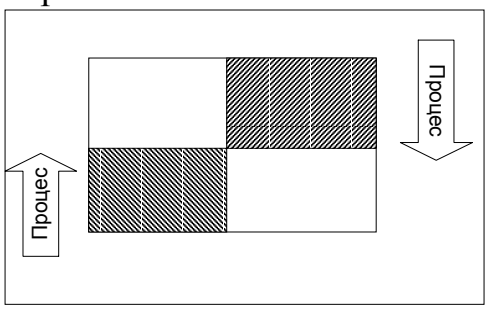
Варіант 8



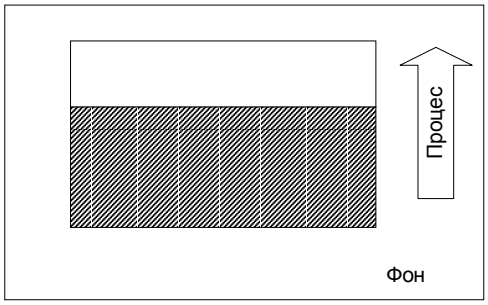
Варіант 9



Варіант 10



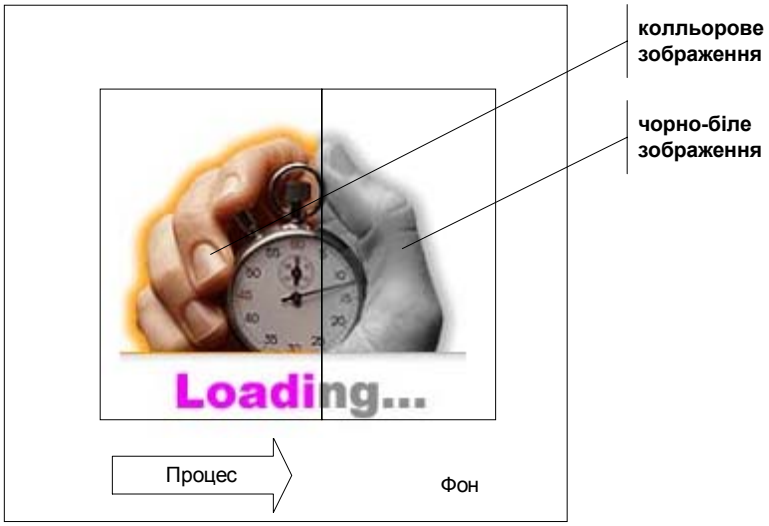
Контрольний приклад


Процес
Фон

Теоретичні відомості

При написанні складних програм-аплетів досить часто постає питання необхідності використання великої кількості графічної інформації і відповідно забезпечення процесу її завантаження. Питання гнучкої і достатньо зрозумілої форми індикації є достатньо складним і вимагає знання декількох технологій за допомогою яких поставлена задача вирішується достатньо швидко.

У найбільш загальному варіанті індикацію можна подати у такому вигляді :



Будь-який java-аплет має набір стандартних методів, які забезпечують його функціонування. Розглянемо кожний зі стандартних методів окремо**.**

Метод init. Метод init визначений у базовому класі Applet, від якого успадковуються всі аплети. Метод init викликається тоді, коли браузер завантажує у своє вікно документ HTML з оператором <APPLET>, який посилається на даний аплет. У цей момент аплет може виконувати ініціалізацію або створювати потоки, якщо він працює у багатопоточному режимі.

Існує протилежність для методу init - метод destroy.

Метод destroy. Перед видаленням аплета з пам'яті викликається метод destroy, який визначений у базовому класі Applet як порожня заглушка. Методу destroy звичайно доручають усі необхідні операції, які варто виконати перед видаленням аплета. Наприклад, якщо в методі init ви створювали які-небудь потоки, у методі destroy їх потрібно завершити.

Метод start. Метод start викликається після методу init у момент, коли користувач починає переглядати документ HTML з вбудованим у нього аплетом.

Метод stop. Доповненням до методу start слугує метод stop. Він одержує керування, коли користувач залишає сторінку з аплетом і завантажує у вікно браузера іншу сторінку. Слід зауважити, що метод stop викликається перед методом destroy.

Метод paint. Найбільш цікавим для нас є метод paint, який виконує рисування у вікні аплета. Виведення зображення забезпечується використанням контекста відображення. Найпростіше уявити собі контекст відображення як полотно, на якому малює художник. Як художник може вибирати для рисування різні інструменти,так і програміст, що створює аплет – Java, може вибирати різні методи класу Graphics і задавати різні атрибути контексту відображення.

Ієрархія класів. З ієрархії класів видно, що клас java.applet.Applet є наступним в ієрархії після класу java.awt.Panel. Цей клас, у свою чергу, визначений у бібліотеці класів java.awt і є наступним після класу java.awt.Container. У класі java.awt.Container немає методу paint, але сам цей клас створений на базі класу java.awt.Component. Але і тут методу paint немає. Цей метод визначений у класі java.awt.Component, який у свою чергу, є наступним після класу java.lang.Object і реалізує інтерфейс java.awt.image.ImageObserver.

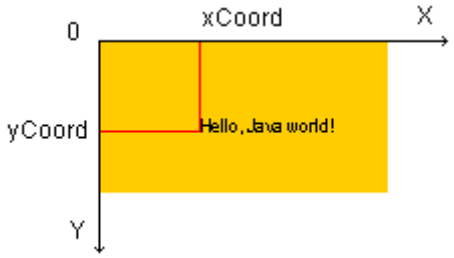
У такий спосіб ми простежили ієрархію класів від класу java.applet.Applet, на базі якого створений наш аплет, до класу java.lang.Object, що є базовим для всіх класів у Java. Метод paint визначений у класі java.awt.Component.

Викликання методу paint. Метод paint викликається коли необхідно перерисувати вікно аплета. Якщо ви створювали програми для операційної системи Windows, то напевно знайомі з повідомленням WM_PAINT, що надходить у функцію вікна при необхідності його перерисовування. Перерисовування вікна додатка Windows і вікна аплета звичайно виконуються асинхронно стосовно роботи аплета. У будь-який момент часу аплет повинен бути готовим перерисувати вміст свого вікна.

Така техніка дещо відрізняється від тої, до якої ви, можливо, звикли, створюючи програми під MS-DOS. Програми MS-DOS самостійно визначають час початку рисування на екрані, причому рисування може виконуватися з різних місць програми. Аплети аналогічно до програм під Windows, виконують рисування у своїх вікнах централізовано. Аплет робить це в методі paint, а програма Windows – при обробці повідомлення WM_PAINT. За своїм змістом цей об'єкт нагадує контекст відображення, з яким добре знайомі творці програм Windows. Контекст відображення можна порівняти з полотном, на якому аплет може рисувати зображення чи писати текст. Численні методи класу Graphics дозволяють задавати різні параметри полотна, такі, наприклад, як колір чи шрифт.

Координатна система. Аплети використовують систему координат, що відповідає режиму відображення MM_TEXT, знайому тим, хто

створював програми під Windows. Початок цієї системи координат розташовано в лівому верхньому куті вікна аплета, вісь X спрямована зліва на право, а вісь Y – зверху вниз.



Методи класу Graphics. У якості базового об'єкта для класу Graphics (повна назва класу java.awt.Graphics) виступає клас java.lang.Object. Наведемо прототипи конструктора цього класу і його методів з короткими поясненнями, розглянемо призначення основних методів, згрупувавши їх за функціями, які виконуються.

Конструктор

- **Graphics**

Методи

- **clearRect**

Стирання вмісту прямокутної області

- **clipRect**

Завдання обмеження області виведення

- **copyArea**

Копіювання вмісту прямокутної області

- **create**

Створення контексту відображення

- **dispose**

Видалення контексту відображення

- **draw3DRect**

Рисування прямокутної області з тривимірним виділенням

- **drawArc**

Рисування сегмента

- **drawBytes**

Рисування тексту з масиву байтів

- **drawChars**

Рисування тексту з масиву символів

- **drawImage**

Рисування растрового зображення

- **drawLine**

Рисування лінії

- **drawOval**

Рисування овалу

- **drawPolygon**

Рисування багатокутника

- **drawRect**

Рисування прямокутника

- **drawRoundRect**

Рисування прямокутника зі скругленими кутами

- **drawString**

Рисування текстового рядка

- **fill3DRect**

Рисування заповненого прямокутника з тривимірним виділенням

- **fillArc**

Рисування заповненого сегмента кола

- **fillOval**

Рисування заповненого овалу

- **fillPolygon**

Рисування заповненого багатокутника

- **fillPolygon**

Рисування заповненого багатокутника

- **fillRoundRect**

Рисування заповненого прямокутника зі скругленими кутами

- **finalize**

Простежування викликання методу dispose

- **getClipRect**

Визначення межі області виведення

- **getColor**

Визначення кольору, обраного в контекст відображення

- **getFont**

Визначення шрифту, обраного в контекст відображення

- **getFontMetrics**

Визначення метрик поточного шрифту

- **getFontMetrics**

Визначення метрик заданого шрифту

- **setColor**

Установка кольору для рисування в контексті відображення

- **setFont**

Установка поточного шрифту в контексті відображення

- **setPaintMode**

Встановлення режиму рисування. Метод **setPaintMode** встановлює в контексті відображення режим рисування, при якому виконується заміщення зображення поточним кольором, встановленим в контексті відображення.

- **setXORMode**

Встановлення маски для рисування. Задаючи маску для рисування за допомогою методу **setXORMode**, ви можете виконати при рисуванні

заміщення поточного кольору на колір, зазначений у параметрі методу, і навпаки, кольору, зазначеного в параметрі методу, на поточний.

Усі інші кольори змінюються непередбачуваним чином, однак така операція є зворотною, якщо ви нарисуєте ту ж саму фігуру два рази на тому самому місці.

- **translate**

Зрушення початку системи координат. Метод **translate** зрушує початок системи координат у контексті відображення таким чином, що вона переміщається в точку з координатами ***(x, y)***, заданими через параметри методу:

- **toString**

Одержання текстового рядка, що подає контекст відображення

Принцип роботи аплета з забезпеченням індикації процесу завантаження графічної інформації

Завантажується два зображення, одне з них є фоном, а інше виводиться поверх першого по одному стовпцю пікселів (в залежності від обраного варіанту). Для забезпечення такої можливості використовується clipping : обмежується область, у якій виводиться задане зображення. Для обмеження області виведення використовується така функція класу Graphics clipRect

```
public abstract void clipRect(int x,  int y,
int width, int height)
```

Розглянемо приклад програми створення аплету .

```
import java.awt.*;
public class ProgresIndicator extends Canvas{
    private Dimension canvasSize = null;
    private Image offScreenImage = null;
    private Graphics offScreenGraphics = null;
// Фон
    private Image blackImage = null;
// Кольорове зображення
    private Image colorImage = null;
// Чорно-біле зображення
    private Image backImage = null;
    private int state = 0;
public ProgresIndicator(Dimension canvasSize, Image
blackImage, Image colorImage, Image backImage){
        this.canvasSize = canvasSize;
        this.blackImage = blackImage;
```

```
        this.colorImage = colorImage;
        this.backImage = backImage;   }
    public void update(Graphics g){
        painBackground(offScreenGraphics);
        paint(g);    }
    public void paint(Graphics g){
        if (offScreenImage == null) {
            offScreenImage =
createImage(canvasSize.width, canvasSize.height);
            offScreenGraphics =
offScreenImage.getGraphics();     }
        if (state!=0){
            paintIncrease(state);     }
        g.drawImage(offScreenImage, 0, 0, this);   }
    private void painBackground(Graphics g){
        g.drawImage(backImage, 0, 0, this);
        int imWidth = blackImage.getWidth(null);
        int imHeight = blackImage.getHeight(null);
        g.drawImage(blackImage, getXOffset(imWidth),
getYOffset(imHeight), this);  }
    private int getXOffset(int imageWidth){
        return (canvasSize.width - imageWidth)/2;  }
    private int getYOffset(int imageHeight){
        return (canvasSize.height - imageHeight)/2;
    }
    public void paintIncrease(final int
currentState){
        int imWidth = colorImage.getWidth(null);
        int imHeight = colorImage.getHeight(null);
        int inc = imWidth*currentState/100;
          // Створюється новий графічний контекст
        Graphics g2 = offScreenGraphics.create();
        // Задається прямокутник відсікання

    g2.clipRect(getXOffset(imWidth),getYOffset(imHeig
ht),inc,imHeight);
        g2.drawImage (colorImage,
getXOffset(imWidth), getYOffset(imHeight), this);  }
    public void increase(final int newState){
        state = newState;
        repaint();   }}
```

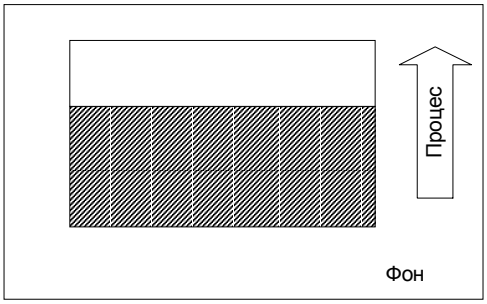
Для демонстрації наведеного прикладу використовується така програма :

```
   import java.awt.*;
   import java.applet.*;

   public class ClientApplet extends Applet
implements Runnable{
    private ProgresIndicator progresIndicator = null;
               private Dimension appletSize = null;
    private Image imageBackGround = null;
    private Image imageBlack = null;
    private Image imageColor = null;

    Thread thread;
    public void init() {
        appletSize = getSize();
        setLayout(new BorderLayout());
        imageBlack = getImage(getDocumentBase(),
getParameter("progImageBlack"));
   imageColor = getImage(getDocumentBase(),
getParameter("progImageColor"));
   imageBackGround = getImage(getDocumentBase(),
getParameter("progImageBack"));
   progresIndicator = new
   ProgresIndicator(appletSize, imageBlack,
   imageColor, imageBackGround);
   add(progresIndicator,BorderLayout.CENTER);
    }

    public void start(){
        if (thread == null) thread = new
Thread(this);
        thread.start();
    }

    public void stop(){
        thread = null;
    }

    public void run(){
        for(int i=0; i<100;i++){
            progresIndicator.increase(i);
```

```
            try{
                thread.sleep(50);
            }catch(InterruptedException e)
            {
            }
    }}}
```

Контрольний приклад

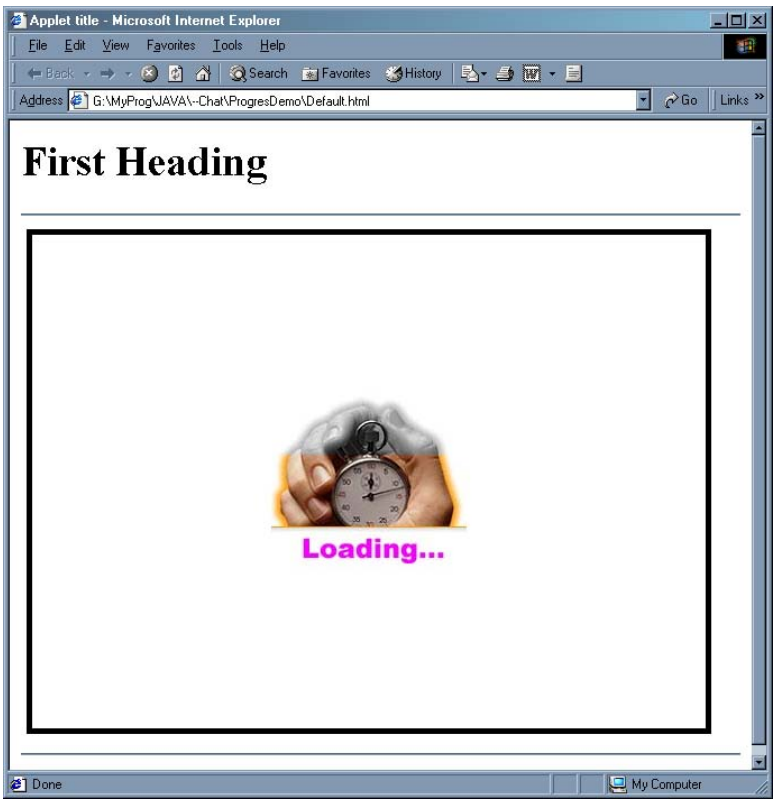
Створити java-аплет, який буде забезпечувати таку форму виведення зображення



Для забезпечення зазначеного способу виведення необхідно змінити лише одну функцію (змінити область відсікання при виведенні зображення) :

```
public void paintIncrease(final int currentState)
{    int imWidth = colorImage.getWidth(null);
     int imHeight = colorImage.getHeight(null);
     int inc = imHeight*currentState/100;
     Graphics g2 = offScreenGraphics.create();
     g2.clipRect(getXOffset(imWidth),getYOffset(imHeig
ht)+(imHeight-inc),imWidth,inc);
     g2.drawImage (colorImage, getXOffset(imWidth),
getYOffset(imHeight), this);}
```

Результат :



Питання для самоперевірки

1. Що таке аплет ?
2. Що таке контекст виведення ?
3. Яке призначення методу paint() ?
4. Коли викликається метод start() ?
5. Які методи аплету і в якій послідовності визиваються браузером?
6. Яка функція дозволяє обмежити область виведення зображення?
7. Як можна вивести зображення у області виведення аплету ?

Література

1. Питер Нортон, Герберт Шилд. Полный справочник по Java. – К.: Диалектика, 1997. – 760 с.

2. Майкл Морган. Java 2. Руководство разработчика. – К.: Диалектика, 1995. – 630 с.

Лабораторна робота №4

ЗАБЕЗПЕЧЕННЯ МОЖЛИВОСТІ НАВІГАЦІЇ ЗА САЙТОМ

Мета роботи : ознайомитися зі способом створення простої навігаційної системи, яка дозволить користувачу швидко і достатньо просто переглядати потрібну сторінку.

Завдання на лабораторну роботу

1. Ознайомитися з теоретичними відомостями.
2. Відповідно до завдання розробити структуру пристроїв навігації.
3. Написати відповідну функцію обробки повідомлень.
4. Написати html-файл, який буде реалізовувати конкретний варіант завдання.
5. Внести до звіту результат роботи написаної програми. Зробити необхідні висновки.

Варіанти завдань

Створити навігацію для сайта :

Варіант 1. Тематичний сайт з програмування (Пункти “Assembler” ,“C++”, “Java”, “JavaScript”, “HTML”, “Інше”).

Варіант 2. Тематичний сайт “електронна бібліотека” (У якості пунктів вкажіть 6-7 письменників).

Варіант 3. Тематичний сайт “Новини” (У якості пунктів вкажіть 6-7 розділів).

Варіант 4. Тематичний сайт “Програмне забезпечення” (У якості пунктів вкажіть 6-7 розділів).

Варіант 5. Тематичний сайт “Музичний каталог” (У якості пунктів вкажіть 6-7 розділів).

Варіант 6. Тематичний сайт “Інформаційний портал” (У якості пунктів вкажіть 6-7 розділів).

Варіант 7. Тематичний сайт “Цікаві історії” (У якості пунктів вкажіть 6-7 розділів).

Варіант 8. Тематичний сайт “Анекдоти” (У якості пунктів вкажіть 6-7 розділів).

Варіант 9. Тематичний сайт “Магазин” (У якості пунктів вкажіть 6-7 розділів).

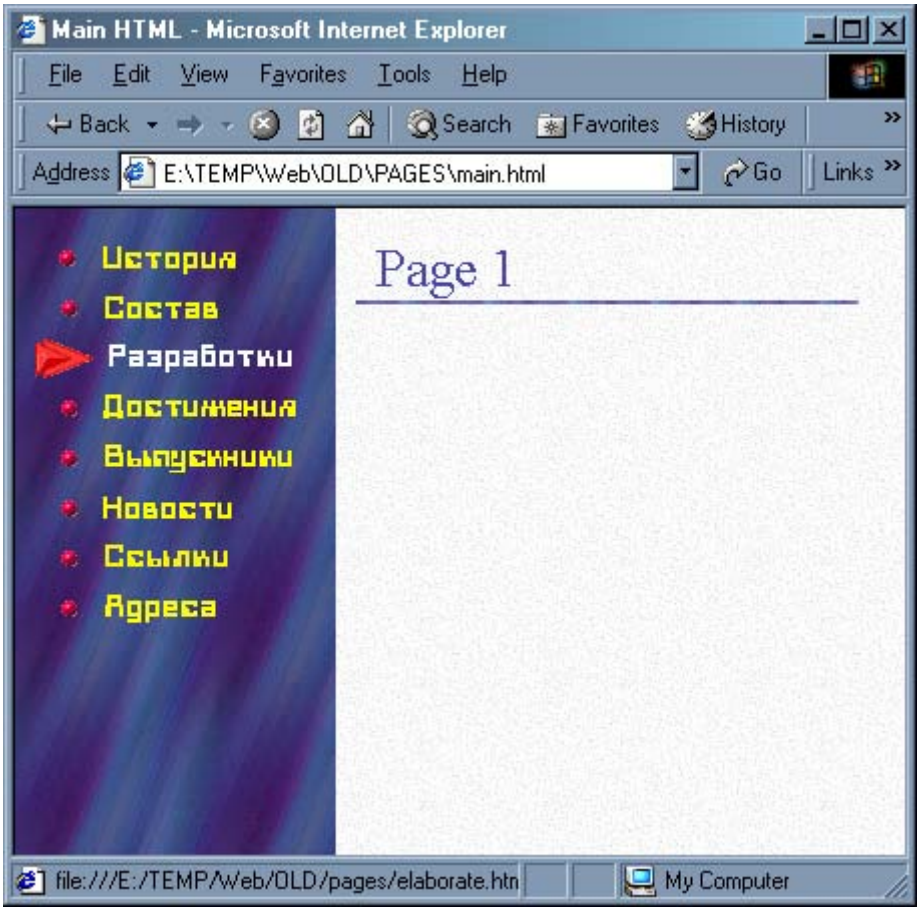
Варіант 10. Тематичний сайт “Комп’ютери” (У якості пунктів вкажіть 6-7 розділів).

Контрольний приклад. Створити навігацію за сайтом, який є сайтом кафедри.

Теоретичні відомості

При створенні веб-сайту досить часто постає питання про створення простої та наочної системи навігації, яка дозволить користувачу швидко отримати доступ до ресурсів, які його цікавлять. Для вирішення задач такого типу можливо використовувати можливості JavaScript.

Навігаційна панель являє собою фрейм з графічними пунктами, які дозволяють перейти на відповідну сторінку. Забезпечується динамічна реакція на дії користувача (при наведенні миші відповідний пункт реагує динамічно):



Після вибору елемента панелі виділення зберігається, що дозволяє допомогти користувачу з ідентифікацією його місця знаходження. Також слід передбачити можливість попереднього завантаження зображень.

Використовуючи JavaScript необхідно мати уявлення про механізм подій, який використовується досить широко.

Існують такі типи подій:

Обробник подій onBlur. Подія blur відбувається, коли поля форми select, text чи textarea втрачають фокус. Обробник подій onBlur виконує програму JavaScript, коли відбувається подія blur.

Обробник подій onChange. Подія change відбувається, коли поля форми select, text чи textarea втрачають фокус і їхні значення змінюються. Обробник подій onChange виконує програму JavaScript, коли відбувається подія change. Обробник подій onChange використовується для підтвердження даних після їхньої зміни користувачем.

Обробник подій onClick. Подія click відбувається при натисненні кнопки миші на об'єкті форми. Обробник подій onClick виконує програму JavaScript, коли відбувається подія click.

Обробник подій onFocus. Подія focus відбувається, коли поле одержує фокус введення з клавіатури чи при натисненні кнопки миші. Вибір результатів усередині поля пов'язано з подією select, а не з подією focus. Обробник подій onFocus виконує програму JavaScript, коли відбувається подія focus.

Обробник подій onLoad. Подія load відбувається, коли Navigator завершує завантаження вікна чи усіх фреймів усередині тега <FRAMESET> . Обробник подій onLoad виконує програму JavaScript, коли відбувається подія load.

Обробник подій onLoad використовується усередині тегу <BODY> чи <FRAMESET>, наприклад, <BODY onLoad="...">.

У відношенні <FRAMESET> і <FRAME> подія onLoad, розміщена усередині фрейму в тегу <BODY>, виконується перед подією onLoad, розміщеною усередині <FRAMESET> у тегу <FRAMESET>.

Обробник подій onMouseOver. Подія mouseOver відбувається щоразу, коли курсор миші попадає на об'єкт. Обробник подій onMouseOver виконує програму JavaScript, коли відбувається подія mouseOver.

Обробник подій onSelect. Подія select відбувається, коли користувач вибирає деякий текст усередині поля text чи textarea. Обробник подій onSelect виконує програму JavaScript, коли відбувається подія select.

Обробник подій onSubmit. Подія submit відбувається, коли користувач відправляє форму на Web-сервер. Обробник подій onSubmit виконує програму JavaScript, коли відбувається подія submit. Ви можете використовувати Обробник подій onSubmit для зупинення передачі даних форми; для цього використовується вираз return, що повертає false в Обробник подій. Будь-яке інше повернене значення відправляє форму.

Обробник подій onUnload. Подія unload відбувається, коли ви виходите з документа. Обробник подій onUnload виконує програму JavaScript тоді, коли відбувається подія unload. Обробник подій onLoad використовується усередині тегів <BODY> чи <FRAMESET>, наприклад, <BODY onUnload="...">.

У відношенні <FRAMESET> і <FRAME> подія onUnload, розміщена усередині фрейму в тегу <BODY>, відбувається перед подією onUnload, розміщеної усередині <FRAMESET> у тегу <FRAMESET>.

Обробник подій. window . *Технічні особливості реалізації*

Для уникнення неприємних моментів при роботі з web-сторінкою попередньо завантажується відповідна графічна інформація. Для попереднього завантаження зображення використовується функція :

```
function preload(name,first,second)
 {
    pics[objCount] = new Array(3);
    pics[objCount][0] = new Image();
    pics[objCount][0].src = first;
    pics[objCount][1] = new Image();
    pics[objCount][1].src = second;
    pics[objCount][2] = name;
    objCount++;
 }
```

Спосіб використання :

```
preload("IM01",”../images/but_01.gif",
"../images/but_02.gif");
```

Іншими словами, вказується назва зображення, яка використовується в подальшому, і відповідні два варіанта зображень, які відповідають різним станам.

Реакція на наведення курсора миші :

```
function on(name){
```

```
     for (i = 0; i < objCount; i++) {
      if (document.images[pics[i][2]] != null)
          if (name != pics[i][2]) {
              if (i != k)
document.images[pics[i][2]].src = pics[i][0].src;
         } else {
             if (i != k)
document.images[pics[i][2]].src = pics[i][1].src;}}}

function off(name){
  for (i = 0; i < objCount; i++) {
      if (document.images[pics[i][2]] != null)
      if (i != k)  document.images[pics[i][2]].src =
pics[i][0].src;  }}
```

Реакція на вибір пункту меню :

```
function on_Click(name){
  for (i=0; i < objCount; i++) {
     if (document.images[pics[i][2]]!= null) {
         if (name == pics[i][2]){
             if(k!=null)
document.images[pics[k][2]].src = pics[k][0].src;
k = i;
document.images[pics[i][2]].src=ics[i][1].src;  }  }  }
}
```

Контрольний приклад

Створити навігацію за сайтом, який є сайтом кафедри.
HTML файл, який реалізує систему навігації, має такий вигляд :

```
<HTML>
<HEAD>
<TITLE> </TITLE>
</HEAD>
<BODY TEXT="#0A0147" LINK="#8888FF" VLINK="#8888FF"
ALINK="#FF0000" BACKGROUND="../images/nav_blue.gif">

<script language="JavaScript">
 var objCount = 0;
 var pics;
 var k;
 pics = new Array();
```

```
 function preload(name,first,second)
 {
    pics[objCount] = new Array(3);
    pics[objCount][0] = new Image();
    pics[objCount][0].src = first;
    pics[objCount][1] = new Image();
    pics[objCount][1].src = second;
    pics[objCount][2] = name;
    objCount++;
 }

 function on(name){
     for (i = 0; i < objCount; i++){
      if (document.images[pics[i][2]] != null)
          if (name != pics[i][2]){
              if (i != k)
document.images[pics[i][2]].src = pics[i][0].src;
          } else {
              if (i != k)
document.images[pics[i][2]].src = pics[i][1].src;
          }
      }
  }

function on_Click(name)
{
  for (i=0; i < objCount; i++) {
     if (document.images[pics[i][2]]!= null){
         if (name == pics[i][2]){
             if(k!=null)
document.images[pics[k][2]].src = pics[k][0].src;
             k = i;
             document.images[pics[i][2]].src =
pics[i][1].src;
         }
     }
  }
}

function off(name){
  for (i = 0; i < objCount; i++){
      if (document.images[pics[i][2]] != null)
```

```
      if (i != k)  document.images[pics[i][2]].src =
pics[i][0].src;
  }
}

preload("IM01", "../images/but_01.gif",
"../images/but_02.gif");
preload("IM11", "../images/but_11.gif",
"../images/but_12.gif");
preload("IM12", "../images/but_21.gif",
"../images/but_22.gif");
preload("IM13", "../images/but_31.gif",
"../images/but_32.gif");
preload("IM14", "../images/but_41.gif",
"../images/but_42.gif");
preload("IM15", "../images/but_51.gif",
"../images/but_52.gif");
preload("IM16", "../images/but_61.gif",
"../images/but_62.gif");
preload("IM17", "../images/but_71.gif",
"../images/but_72.gif");
preload("IM18", "../images/but_81.gif",
"../images/but_82.gif");
preload("UP11", "../images/up11.gif",
"../images/up12.gif");
preload("LEFT11", "../images/left11.gif",
"../images/left12.gif");
preload("RIGHT11", "../images/right11.gif",
"../images/right12.gif");
preload("DOWN11", "../images/down11.gif",
"../images/down12.gif");
</script>
<DIV ALIGN="LEFT">
<A HREF="history.html" onMouseOver="on('IM11')"
onClick="on_Click('IM11')" onMouseOut="off('IM11')"
target="Frame21"><IMG NAME="IM11"
SRC="../images/but_11.gif" TPPABS="" WIDTH=130
HEIGHT=20 BORDER=0 ALT=""></A>
<A HREF="structure.html" onMouseOver="on('IM12')"
onClick="on_Click('IM12')" onMouseOut="off('IM12')"
target="Frame21"><IMG NAME="IM12"
SRC="../images/but_21.gif" TPPABS="" WIDTH=130
HEIGHT=20 BORDER=0 ALT=""></A>
```

```
<A HREF="elaborate.html" onMouseOver="on('IM14')"
onClick="on_Click('IM14')" onMouseOut="off('IM14')"
target="Frame21"><IMG NAME="IM14"
SRC="../images/but_41.gif" TPPABS="" WIDTH=130
HEIGHT=20 BORDER=0 ALT=""></A>
<A HREF="progress.html" onMouseOver="on('IM15')"
onClick="on_Click('IM15')" onMouseOut="off('IM15')"
target="Frame21"><IMG NAME="IM15"
SRC="../images/but_51.gif" TPPABS="" WIDTH=130
HEIGHT=20 BORDER=0 ALT=""></A>
<A HREF="graduates.html" onMouseOver="on('IM17')"
onClick="on_Click('IM17')" onMouseOut="off('IM17')"
target="Frame21"><IMG NAME="IM17"
SRC="../images/but_71.gif" TPPABS="" WIDTH=130
HEIGHT=20 BORDER=0 ALT=""></A>
<A HREF="news.html" onMouseOver="on('IM13')"
onClick="on_Click('IM13')" onMouseOut="off('IM13')"
target="Frame21"><IMG NAME="IM13"
SRC="../images/but_31.gif" TPPABS="" WIDTH=130
HEIGHT=20 BORDER=0 ALT=""></A>
<A HREF="links.html" onMouseOver="on('IM18')"
onClick="on_Click('IM18')" onMouseOut="off('IM18')"
target="Frame21"><IMG NAME="IM18"
SRC="../images/but_81.gif" TPPABS="" WIDTH=130
HEIGHT=20 BORDER=0 ALT=""></A>
<A     HREF="adres.html"     onMouseOver="on('IM16')"
onClick="on_Click('IM16')"    onMouseOut="off('IM16')"
target="Frame21"><IMG                      NAME="IM16"
SRC="../images/but_61.gif"     TPPABS=""     WIDTH=130
HEIGHT=20 BORDER=0 ALT=""></A>
</DIV>
<br>
<BR>
</BODY>
</HTML>
```

Питання для самоперевірки

1. При вирішенні яких проблем доцільно використовувати JavaScript ?
2. Які існують стандартні об’єкти ?
3. Яким чином забезпечується зв’язок HTML та JavaScript ?
4. Які повідомлення підлягають обробленню ?

5. Яким чином реєструються програми для оброблення відповідних переривань?
6. З якою метою використовується попереднє завантаження зображень ?

Література

1. JavaScript Guide Netscape Navigator Version 4.0. – К.: НИПФ “Диа Софт ЛТД”, 2000. – 540 с.

Лабораторна робота №5

ФІЛЬТРАЦІЯ ЗОБРАЖЕНЬ

Мета роботи: ознайомитися зі способом і реалізацією фільтрації зображень.

Завдання на лабораторну роботу

1. Ознайомитися з теоретичними відомостями.
2. Відповідно до завдання розробити спосіб виведення зображень у відповідності з варіантом.
3. Написати функцію виведення зображення відповідно до варіанта.
4. Написати програму виведення зображень.
5. Внести до звіту результат роботи написаної програми. Зробити необхідні висновки.

Варіанти завдань

Варіант завдання подається у формі матриці, елементи якої є ваговими коефіцієнтами, перед матрицею вагових коефіцієнтів вказується нормуючий коефіцієнт.

Варіант 1.	Варіант 2.	Варіант 3.
$\frac{1}{9}\begin{vmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{vmatrix}$	$\frac{1}{16}\begin{vmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{vmatrix}$	$\frac{1}{2}\begin{vmatrix}1 & -1 & 1\\ -1 & 5 & -1\\ 1 & -1 & 1\end{vmatrix}$

Варіант 4. $\frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{vmatrix}$	Варіант 5. $\frac{1}{2}\begin{vmatrix} 2 & -1 & 2 \\ -1 & 4 & -1 \\ 2 & -1 & 2 \end{vmatrix}$	Варіант 6. $\frac{1}{4}\begin{vmatrix} 1 & 2 & 1 \\ 2 & 8 & 2 \\ 1 & 2 & 1 \end{vmatrix}$
Варіант 7. $\frac{1}{4}\begin{vmatrix} -1 & 2 & -1 \\ 2 & 8 & 2 \\ -1 & 2 & -1 \end{vmatrix}$	Варіант 8. $\frac{1}{4}\begin{vmatrix} 1 & -2 & 1 \\ -2 & 8 & -2 \\ 1 & -2 & 1 \end{vmatrix}$	Варіант 9. $\frac{1}{3}\begin{vmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{vmatrix}$
Варіант 10. $\frac{1}{10}\begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix}$		

Теоретичні відомості

При розробці різних графічних програм програміст часто зіштовхується з проблемою поліпшення параметрів зображення. Потрібно, приміром, підвищити його різкість чи усунути перешкоди. Локальна фільтрація надає для цього безліч методів.

Нижче будуть розглянуті лише деякі, найбільш вживані алгоритми, які, однак, дозволяють досягти хороших результатів. Крім поліпшення зображення, фільтри часто використовують для створення різних спеціальних ефектів. Також методи фільтрації дозволяють зменшити навантаження на мережу, оскільки більшу частину об’єму графічної інформації можливо отримати на машині клієнта шляхом застосування методу фільтрації зображень, таким чином об’єм інформації, що передається, зменшується.

У подальшому будемо вважати зображення двомірним масивом, компонент [i,j] якого містить інтенсивність відповідного піксела. Кольорове зображення задається трьома такими масивами. Перший містить значення інтенсивностей червоної складової, другий – зеленої, третій – синьої. Будемо розглядати кожен такий масив окремо.

Обмежимо деяку ділянку зображення прямокутником і поставимо у відповідність кожному пікселю, що попав у цей прямокутник, коефіцієнт. Ділянку зображення назвемо апертурою, а задані на ній коефіцієнти – ваговими. Матриця, елементом якої є вагові коефіцієнти, називається ваговою функцією. Як правило, на практиці використовують апертури розміром у 3x3 чи 5x5 елементів.

Процес фільтрації полягає в переміщенні апертури по зображенню і виконанні якогось алгоритму, що на підставі вагових коефіцієнтів і точки зображення (x,y) видає нове значення інтенсивності цієї точки. Найчастіше використовують лінійні фільтри. При цьому нове значення інтенсивності піксела (x,y) розраховується у такий спосіб :

а) фіксується апертура так, щоб її центр збігався з (x,y);

б) кожне значення вагового коефіцієнта апертури збільшується на значення інтенсивності відповідного йому піксела;

в) складаємо добутки і ділимо на нормуючий коефіцієнт;

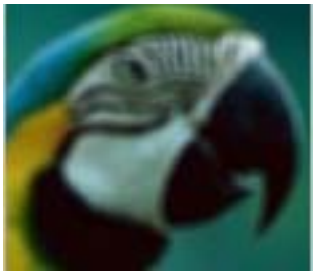
г) отримане значення інтенсивності привласнюється пікселю (x,y) у вихідному масиві.

Як бачимо, для того щоб одержати нове значення інтенсивності, ми порахували значення лінійної функції. Якщо для цього потрібно підрахувати значення нелінійної функції, то такі функції називаються нелінійними. Розрізняють також два типи фільтрів: прості і рекурсивні. У випадку з простими фільтрами існує два масиви зображення : вхідний і вихідний. У випадку з рекурсивним фільтром як вихідний масив використовується вхідний.

Приклади

Згладжування шуму. Такі фільтри також називають низькочастотними, оскільки при частотній інтерпретації процесів фільтрації фільтр є фільтром нижніх частот.

Пом'якшення зображення. Для пом'якшення зображення використовують такі фільтри:



Швидкість обчислень. Очевидно, що у випадку з простими фільтрами вихідний масив не залежить від шляху його обробки. Єдиною необхідною умовою є вимога, щоб кожна точка вихідного масиву була обробленою. Припустимо, ми проходимо зображення фільтром 3x3 по лініях.Тоді при згладжуванні шуму можна використовувати суми, отримані на попередніх кроках.

Крайові ефекти. Перекручування крайніх пікселів відфільтрованого зображення називають крайовими ефектами. Причина перекручування полягає в тому, що при фільтрації крайніх пікселів апертура "виїжджає" за межі зображення. Вплив крайових ефектів є найбільш відчутним при обробці зображень, розміри яких малі. Якщо розміри великі, то крайові пікселі можна не фільтрувати (швидше за все ефект не буде впадати в око).

Можливі ще три варіанти вирішення цієї проблеми. По-перше, можна розширити оброблюване зображення шляхом його обрамлення нульовими елементами. По-друге, можна не розширювати поле зображення, а зменшити розміри апертури при перерахуванні крайових пікселів. Очевидно, що другий метод є складнішим у реалізації, у порівнянні з першим, хоча і дає кращі результати. Третій метод полягає у "зацикленні зображення". Така техніка дає непогані результати при циклічній фільтрації зображення, коли перекручування зображення необхідно виключити.

Перетворення кольору в яскравість. Обчислюємо яскравість пікселя P(R,G,B) (де R,G,B – відповідно червона, зелена, синя складова), використовуючи процедуру перетворення кольору в яскравість K = 0.56*G + 0.33*R + 0.11*B. Одержуємо P(K,K,K) – сірий колір тієї ж яскравості, що і вихідний.

Інвертування зображення. Інвертуємо окремо червону, зелену і синю складові. Потім формуємо колір піксела.

Поліпшення контрастності. Для поліпшення контрастності цей фільтр використовує алгоритм, відповідно до якого окремо взяті значення червоної, зеленої і синьої складових збільшуються в 1.2 разу, якщо вони перевищують 128. Якщо ж ці значення менше 128, то вони діляться на 1.2. Збільшені значення, які не потрапляють у допустимий діапазон, приводяться до максимального значення.

Збільшення різкості зображення. Для кожного елемента вихідного зображення обчислюємо середнє значення сусідніх елементів з квадрата 3x3, але не враховуємо значення центрального піксела. Потім до значення центрального піксела додаємо різницю між цим значенням і середнім значенням сусідніх пікселів, а результат привласнюється відповідним елементом вихідного масиву. Таким чином, переходи між контрастними областями стають ще більш контрастними, а області з плавною зміною кольору залишаються незмінними.

Програма реалізації фільтрів (наведено неоптимізований варіант для покращання розуміння процесу фільтрації) :

```
import java.applet.Applet;
import java.awt.*;
import java.awt.image.MemoryImageSource;
import java.awt.image.PixelGrabber;

public class N_Filter_21 extends Applet
{

    public void init()
    {
        dim = size();
        w = dim.width;
        h = dim.height;
        try
        {
            In_Image = getImage(getDocumentBase(),
getParameter("img"));
            MediaTracker mediatracker = new
MediaTracker(this);
            mediatracker.addImage(In_Image, 0);
            mediatracker.waitForID(0);
            iw = In_Image.getWidth(null);
            ih = In_Image.getHeight(null);
            temp_pixels = new int[iw * ih];
            pix = new int[iw * ih];
            PixelGrabber pixelgrabber = new
PixelGrabber(In_Image, 0, 0, iw, ih, pix, 0, iw);
            pixelgrabber.grabPixels();
        }
        catch(InterruptedException _ex) { }
        Use_F3x3();
```

```
        Out_Image = createImage(new
MemoryImageSource(iw, ih, temp_pixels, 0, iw));
    }

    private int Filt_Buffer_F3x3()
    {
        int k = (c[0][2] + 2 * c[1][2] + c[2][2]) -
(c[0][0] + 2 * c[1][0] + c[2][0]);
        int l = (c[0][0] + 2 * c[0][1] + c[0][2]) -
(c[2][0] + 2 * c[2][1] + c[2][2]);
        new_color = (int)Math.sqrt(k * k + l * l);
        return new_color;
    }

    private void Use_F3x3()
    {
        for(int k = 1; k < ih - 1; k++)
        {
            for(int l = 1; l < iw - 1; l++)
            {
                for(i = 0; i < 3; i++)
                    for(j = 0; j < 3; j++)
                        c[i][j] = pix[((((k - 1) + i)
* iw + l) - 1) + j] >> 16 & 0xff;


                int i1 = Filt_Buffer_F3x3();
                for(i = 0; i < 3; i++)
                    for(j = 0; j < 3; j++)
                        c[i][j] = pix[((((k - 1) + i)
* iw + l) - 1) + j] >> 8 & 0xff;


                int j1 = Filt_Buffer_F3x3();
                for(i = 0; i < 3; i++)
                    for(j = 0; j < 3; j++)
                        c[i][j] = pix[((((k - 1) + i)
* iw + l) - 1) + j] & 0xff;


                int k1 = Filt_Buffer_F3x3();
                temp_pixels[k * iw + l] = 0xff000000
| i1 << 16 | j1 << 8 | k1;  }  }   }
```

```
public void paint(Graphics g)
{
    g.drawImage(Out_Image, 0, 0, null);    }
public N_Filter_21()
{
    c = new int[3][3];
}
Dimension dim;
Math M;
Image In_Image;
Image Out_Image;
int iw;
int ih;
int pix[];
int temp_pixels[];
int w;
int h;
int i;
int j;
int c[][];
int new_color; }
```

Питання для самоперевірки

1. Для чого використовуються фільтри?
2. Які існують типи фільтрів?
3. Яким чином можна прискорити процес фільтрації?
4. Що таке контекст виведення ?
5. Які проблеми виникають при використанні матричних фільтрів?
6. Яке призначення методу paint() ?
7. Які методи аплету і в якій послідовності визиваються браузером?
8. Наведіть приклад будь-якого фільтру, та опишіть його роботу.
9. Як можливо вивести зображення в області виведення аплету ?

Література

1. Питер Нортон, Герберт Шилд. Полный справочник по Java. – К.: НИПФ “Диа Софт ЛТД”, 1997. – 760 с.

2. Майкл Морган. Java 2. Руководство разработчика. – К.: Диалектика, 1995. – 630 с.

Лабораторна робота № 6

ЗНАЙОМСТВО ТА РОБОТА З ІНТЕРНЕТ

Мета роботи: навчитися працювати та знаходити інформацію за допомогою Інтернет, а також ознайомитися з типами підключення до Інтернет та основними програмами перегляду.

Завдання до лабораторної роботи

Завдання до лабораторної роботи для різних варіантів зведені в таблицю:

Перший варіант	
	Завдання для виконання
1	За адресою http://www.ambernet.kiev.ua/~klokol/8dr.html вказати, що є *іменем домену,* де знаходиться маршрут каталогу та ім'я документа
2	За допомогою www.altavista.com здійснити пошук за темою “Ukraine”
3	Виконати пошук інформації про Верховну Раду України за допомогою пошукових каталогів www.yandex.ru, www.rambler.ru. Порівняйте одержані результати
4	Виконати пошук інформації за допомогою www.uaport.net, www.metaukraine.com за словом “інститути”
5	Виконати пошук інформації за ключовими словами: Україна, Вінниця, Вінницький національний технічний університет, з використанням “І”
Другий варіант	
1	За адресою http://www.gradobank.kiev.ua/~visiter/price.html вказати, що є *іменем домену,* де знаходиться маршрут каталогу та ім'я документа
2	За допомогою www.altavista.com здійснити пошук за темою: weather
3	Виконати пошук інформації за допомогою www.uaport.net, www.metaukraine.com за словом “журнали”.
4	Виконати пошук інформації за ключовими словами: Автомобілі, Opel, Omega з використанням “І”
5	За допомогою мережі Інтернет зробити огляд останніх подій
Третій варіант	
1	За адресою http://www.uefa.com./~europa/england.html вказати, що є *іменем домену,* де знаходиться маршрут каталогу та ім'я документа

2	За допомогою www.altavista.com здійснити пошук за ключовими словами: музика, The Beatles
3	Виконати пошук інформації про Державну Думу Росії за допомогою пошукових каталогів www.yandex.ru, www.rambler.ru. Порівняти одержані результати
4	Виконати пошук інформації за ключовими словами: операційні системи, Windows, Windows 2000 з використанням "І"
5	За допомогою мережі Інтернет зробіть огляд існуючих бібліотек
Четвертий варіант	
1	За адресою http://www.microsoft.com.ru./~windoows2000/servisepack.html вказати, що є *іменем домену,* де знаходиться маршрут каталогу та ім'я документа
2	За допомогою www.altavista.com здійснити пошук каталогів mp3 музики
3	Виконати пошук інформації про Києво-Могилянську Академію за допомогою пошукових каталогів www.yandex.ru, www.rambler.ru. Порівняйте одержані результати
4	Виконати пошук інформації за ключовими словами: наука, журнали, лінійні рівняння
5	За допомогою мережі Інтернет переглянути інститути, які знаходяться у вашому місті
П'ятий варіант	
1	За адресою http://www.freesoft.ru./~halava/pack.html вказати, що є *іменем домену,* де знаходиться маршрут каталогу та ім'я документа
2	За допомогою www.altavista.com здійснити пошук за темою "американський долар"
3	Виконати пошук інформації про футбольний клуб "Динамо" за допомогою пошукових каталогів www.yandex.ru, www.rambler.ru. Порівняти одержані результати
4	Виконати пошук інформації за ключовими словами: Україна, літаки, АН-24 з використанням "І"
5	За допомогою мережі Інтернет зробити пошук інформації за темою: реферати, соціологія
Шостий варіант	
1	За адресою http://www.meta.ua/~one/go.html вказати, що є *іменем домену,* де знаходиться маршрут каталогу та ім'я документа
2	Виконати пошук інформації про останні події у світі Формула1 за допомогою пошукових каталогів www.yandex.ru, www.rambler.ru
3	За допомогою www.altavista.com здійснити пошук за темою "криптографія"

4	Виконати пошук інформації за ключовими словами: мистецтво, живопис, віртуальні музеї живопису
5	За допомогою www.uaport.net, www.metaukraine.com зробити пошук інформації за темою: біржові індекси
Сьомий варіант	
1	За адресою http://www.lenta.ru/~news/usa.html вказати, що є *іменем домену,* де знаходиться маршрут каталогу та ім'я документа
2	За допомогою www.altavista.com здійснити пошук за темою ОС Unix
3	Виконати пошук інформації про АР Крим за допомогою пошукових каталогів www.yandex.ru, www.rambler.ru. Порівняти одержані результати
4	Виконати пошук інформації за ключовими словами: Україна, Вінниця, Вінницький державний педагогічний університет, з використанням “І”
5	Виконати пошук інформації про відношення курсу гривні до іноземних валют на сьогоднішній день
Восьмий варіант	
1	За адресою http://www.referats.com./~ekonom/makroekon.html вказати, що є *іменем домену,* де знаходиться маршрут каталогу та ім'я документа
2	Виконати пошук інформації про законодавство України за допомогою пошукових каталогів www.yandex.ru, www.rambler.ru. Порівняйте одержані результати
3	За допомогою www.altavista.com здійснити пошук за темою: виробники відеокамер
4	За допомогою мережі виконати пошук за ключовими словами: спорт, теніс, рейтинг ATP.
5	За допомогою www.uaport.net, www.metaukraine.com зробити пошук інформації за темою: вибори України.
Дев’ятий варіант	
1	За адресою http://www.bbs.com./~hot_news/ban_laden.html вказати, що є *іменем* домену, де знаходиться маршрут каталогу та ім'я документа
2	За допомогою www.altavista.com здійснити пошук за темою: фільми компанії Warner Broses
3	Виконати пошук інформації про українські телеканали за допомогою пошукових каталогів www.uaport.net, www.metaukraine.com . Порівняти одержані результати.
4	Виконати пошук інформації за допомогою www.yandex.ru, www.rambler.ru за темою: карта міста Москва.

5	Виконати пошук інформації за ключовими словами: Україна, аеропорти, аеропорт Бориспіль, з використанням “І”.
Десятий варіант	
1	За адресою http://www.kvazarmikro.ua/~news/price.html вказати, що є *іменем* домену, де знаходиться маршрут каталогу та ім'я документа
2	За допомогою www.altavista.com здійснити пошук за темою: університети України
3	Виконати пошук інформації про роботу за допомогою пошукових каталогів www.yandex.ru, www.rambler.ru. Порівняти одержані результати
4	За допомогою www.uaport.net, www.metaukraine.com зробити пошук інформації за темою: прогноз погоди на сьогодні
5	Виконати пошук інформації за ключовими словами: Україна, Київ, київський зоопарк
Контрольний приклад	
1	За адресою http://www.charm.net/~brooklyn/People/JackKerouac.html вказати, що є *іменем домену,* де знаходиться маршрут каталогу та ім'я документа
2	За допомогою www.uaport.net, www.metaukraine.com здійснити пошук інформації: журнал Chip
3	3. За допомогою www.altavista.com здійснити пошук за темою: business
4	Виконати пошук інформації за допомогою пошукових каталогів www.yandex.ru, www.rambler.ru за темою: реферати, політологія. Порівняти отримані результати
5	Виконати пошук інформації за ключовими словами: Україна, Київ, Інститут Шевченка з використанням “І”

Короткі теоретичні відомості

Доступ у мережу Інтернет, як правило, одержують через "постачальників мережних послуг" (service provider). Постачальники послуг надають різні види телекомунікаційних й інформаційних послуг і кожний з них має свої переваги і недоліки. Природно, що при звертанні до них необхідно вирішити, якими якостями повинні володіти надані послуги, скільки за них платити, і, виходячи з цього, вибирається прийнятний варіант.

Усі постачальники послуг підрозділяються на дві великі категорії: "академічні" (чи некомерційні) і комерційні провайдери. Істотна різниця в них полягає у тому, що для академічних мереж канали зв'язку оплачуються

державою або міжнародними чи національними фондами, і для них вартість підключення визначається, виходячи з витрат на саме підключення й обслуговування. Комерційним провайдерам оплата здійснюється за рахунок клієнта.

Види доступу в Інтернет. В Інтернет є кілька видів доступу. Чим більше можливостей надає вид доступу і чим вища швидкість передачі, тим він більш дорожчий. Розглянемо їх у порядку зменшення вартості.

Безпосередній доступ. Великі організації, такі як інститути, ВУЗи використовують "безпосередній" чи прямий вид доступу. Він дає повний доступ до всіх можливостей мережі. Для організації прямого доступу необхідно пряме з'єднання з найближчим вузлом Інтернет. Для цього можна орендувати виділену телефонну лінію з обраною пропускною здатністю (чим швидшою є передача даних, тим дорожчим є зв’язок; види ліній зв'язку приведені в табл.1), прокласти кабель чи використовувати радіоканал і встановити мережний сервер (вузловий комп'ютер - роутер). Цей комп'ютер відповідає за зв'язок організації з центральним вузлом мережі (чи з іншими вузлами) і пересилання даних в обидва боки. Цей варіант є досить дорогим (особливо, якщо канал орендується в МТС, що є монополістом у встановленні орендної плати). Але, встановивши одноразово таке з'єднання, можна підключати до цього вузла стільки комп'ютерів, скільки необхідно. При цьому необхідно зв'язати їх у локальну обчислювальну мережу разом з вузлом Інтернет (наприклад, через Ethernet).

Таблиця 1 – Види телефонних каналів

Види ліній підключення		
1	**2**	**3**
Вид послуг	**Швидкість**	**Примітки**
Стандартна телефонна лінія тональної частоти	0 -19.2 Kbps 0-28.8 Kbps 0-34.4 Kbps	Доступ по SLIP чи "за викликом" Швидкості залежать від виду модемів і якості каналу зв’язку
Виділена лінія	56 - 64 Kbps 56-115 Kbps до 512 Kbps	Пряме підключення до вузла Інтернет

Продовження таблиці 1

1	2	3
T1	1.544 - 2.0 Mbps	Пряме підключення до вузла по широкополосному каналу, або виділені канали, або канали Frame Relay
T2	6.0 Mbps	Звичайно в мережах не використовується
T3	45.0 Mbps	Основний мережний канал (backbone) для великої корпорації, як правило, оптоволоконі канали зв’язку.

Безпосередній доступ пропонує найбільш гнучке підключення. Кожний з комп'ютерів локальної мережі є повноправним членом Інтернет і може скористатися його можливостями, однак, через високу вартість доступний тільки для організацій, але не для "домашніх користувачів".

Безпосередній Інтернет-доступ звичайно вимагає наявності деякої базової структури локальної мережі. Для її експлуатації, як правило, потрібний освічений персонал, устаткування і документація, на які необхідно передбачити відповідні витрати.

SLIP і PPP. Існують і менш дорогі способи “майже прямого доступу”. Вони називаються SLIP і PPP і є версіями програмного забезпечення Інтернет, що працює на звичайних телефонних лініях, використовуючи стандартні високошвидкісні модеми. Робота з SLIP чи PPP відбувається на звичайній лінії, що звільняється по закінченню сеансу роботи. Цією ж лінією можуть скористатися інші користувачі. Перевага SLIP і PPP полягає в тому, що вони дозволяють працювати в режимі повноправного входу в Інтернет. SLIP і PPP - це протоколи канального рівня.

Що таке SLIP? Serial Line Інтернет Protocol (SLIP) – це Інтернет-протокол, що дозволяє використовувати послідовні лінії так і лінії зв'язку, наприклад, модем і звичайну телефонну лінію. Програмне забезпечення, що реалізує роботу з протоколом SLIP, приймає символи, які приходять з пристрою послідовної передачі даних (модему, послідовного порту і т.д.), розглядає і тлумачить їх, як складову IP-пакета. Потім формує з отриманих даних нормальний IP-пакет і передає цей пакет відповідній програмі, що обробляє IP-пакети, наприклад, модулю TCP. На зворотному шляху SLIP одержує від програми (мережного рівня), яка посилає IP-пакети, IP-пакет, зчитує його зміст, переформатовує, потім поділяє на символи і відправляє його через пристрій послідовної передачі по послідовній лінії в мережу сусідньому вузлу Інтернет.

Що таке PPP? Point to Point Protocol (PPP) – це більш пізній протокол, що виконує ті ж функції, що і SLIP. PPP є досконалішим і кращим від свого попередника, однак, навряд чи він витисне SLIP із використання.

SLIP і PPP дуже зручні для підключення домашнього комп'ютера до локальної мережі, що, у свою чергу, входить у Інтернет. Наприклад, можна скористатися SLIP, щоб підключити домашній комп'ютер до мережі чи організації інституту. І тоді комп'ютер буде мати повний доступ у Інтернет, як будь-який комп'ютер організації, підключений через локальну мережу. Ці протоколи не призначені для підключення середніх чи великих за величиною мереж: через обмежену швидкість вони не можуть обслуговувати одночасно велику кількість користувачів. Тому для мереж середнього чи великого розміру використовують безпосередній доступ.

Існує версія протоколу SLIP, пристосована для роботи на повільних лініях - CSLIP. Це SLIP зі стиснутими заголовками. Цей протокол був створений у Lawrence Berkeley Labs (LBL) Ван Якобсоном, як спосіб підвищити ефективність послідовної передачі і підвищити рівень сервісу прикладних програм, що використовують TCP/IP на повільних лініях.

Доступ "за викликом» (Dial-up Access). Dial-up Access – це спосіб одержання доступу до Інтернет користувачем великої машини, що має прямий доступ у мережу і допускає можливість вилученої роботи. Після одержання логічного імені, доступу до системи і прав користувача на роботу, використовується, наприклад, домашній комп'ютер з модемом для входу в цю машину і роботи в мережі. Доступ за викликом є майже так само зручним, як і постійне підключення, і він є істотно простішим за установкою. Комп'ютер не стає частиною мережі, він просто має доступ до послуг комп'ютера, який приєднано до мережі постійно. Багато організацій надають саме цей вид послуг, до того ж такі послуги є значно дешевшими.

Програми перегляду. Відображення інформації відбувається за допомогою спеціальної програми перегляду - броузера. Броузер може бути графічним чи текстовим. Для того щоб броузери могли інтерпретувати документи вони повинні бути розмічені у відповідності зі специфікацією мови HTML.

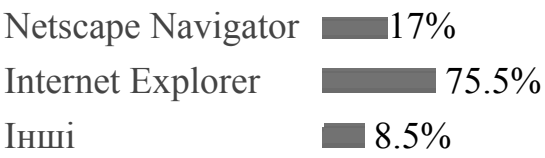
Першим графічним броузером була програма Mosaic, розроблена в національному центрі суперкомп'ютерних додатків США (NCSA). У 1994р. була утворена корпорація Netscape Communications, а її засновниками були творці Mosaic. Незабаром з'явилася комерційна версія графічного броузера Netscape. Netscape Navigator 2.0 умів підтримувати Java-апплети і мову JavaScript. До 1996 р. Netscape мала повне панування у світі веб-броузерів.

На початку 1996 р. на ринку з'являється броузер Microsoft Internet Explorer 2.0, а влітку того ж року Microsoft Internet Explorer 3.0. Версія

IE3.0 підтримувала усі розширення Netscape і незабаром стала на один рівень з ним.

У наш час найбільш поширеними броузерами є Netscape Navigator6.0, та Інтернет Exploer 5.5, хоча поряд з ними мають широке застосування і альтернативні броузери (наприклад, Opera). Нижче наведено графік розподілення частоти використання відповідних броузерів.

Броузери:



Що таке *World Wide Web*? Це спроба подати всю інформацію, доступну в мережі Інтернет, у вигляді сукупності гіпертекстових документів, що дозволяють рухатися від документа до документа через посилання. Посилання чи піктограма – це виділене слово чи словосполучення, що дозволяє розкрити його зміст. Посиланнями також можуть бути графічні картинки. У такий спосіб Web нагадує електронну мультимедійну енциклопедію.

Як вже зазначалося вище, найбільш розповсюдженим броузером є Intrnet Exploer. Після запуску броузера, автоматично завантажується «домашня» (початкова) сторінка або просто “пусте відображення”. Щоб рухатися у Web необхідно ввести в текстовому боксі **Адреса** URL -адреса сторінки Web, наприклад: http://www.vinntsa.com.

У полі „адрес” – буде вказана адреса вашої “домашньої” сторінки. Для заміни на іншу сторінку необхідно виконати таку послідовність дій: “Сервис → Свойства обозревателя → Общие” та в полі “адрес” вказати нову ”домашню” сторінку (ця сторінка буде відображуватися постійно при запуску IE до того моменту, поки в полі „адрес” ви не введете нову). Для пошуку інформації або для роботи з WWW необхідно в полі „адрес”, ввести відповідний адрес ресурсу, який ви бажаєте переглянути. У загальному вигляді це можна записати так: http://www.адреса.com. При виникненні помилок в процесі роботі з WWW для поновлення відповідної сторінки можно скористуватися кнопкою “Обновить”, для завершення роботи: “Файл → Закрыть”.

Пошук інформації у Intrnet Exploer. Існує кілька способів пошуку інформації в Інтернеті. Щоб одержати доступ до систем пошуку, натисніть кнопку **Пошук** на панелі інструментів. Введіть у поле **Пошук** ключове слово чи фразу.

- В адресний рядок введіть команди **go**, **find** чи **?** і далі через відступ введіть ключове слово чи фразу. Internet Explorer розпочне пошук з використанням заздалегідь визначеної системи пошуку.
- Після переходу на веб-сторінку, можна знайти на ній певний текст, вибравши в меню „Правка” пункт „Найти на этой странице”.

Примітка Якщо задана неправильна веб-адреса, з'явиться запит, чи виконувати пошук схожих веб-адрес. Можна налаштувати Internet Explorer так, щоб він виконував пошук автоматично, без запиту.

Netscape Navigator (NN) for Windows 9x

NN може читати і інтерпритувати коди, що вказують комп'ютеру, як виводити текст, а також повідомити, де знаходиться інша інформація, наприклад, відеокліпи й інші документи Web.

Щоб розпочати роботу з програмою Navigator, необхідно переконатися в наявності ще двох програм: це TCP/IP (керує передачею даних, поставляється сервісною компанією) і Dial-Up Networking, яка дозволяє встановлювати з'єднання із сервісною компанією (поставляється з Windows 9x)

NN та IE дозволяють відкривати кілька документів Web одночасно, причому кожен документ з’являється в окремому вікні. Допускається змінювати розміри вікон і переміщати їх, мінімізувати одне вікно при роботі в іншому.

Засоби пошуку інформації в Інтернет

On-line механізми пошуку стають все більш швидкими, інтелектуальними, розвиненими. Деякі вузли пропонують 200 або більше різних утиліт, наприклад засоби пошуку телефонних номерів та адрес, довідкові інструкції та інші види сервісу. Найбільш відомі пошукові системи, такі як AltaVista, HotBot, Infoseek, побудовані на базі високошвидкісних оптоволоконних комунікацій та масштабованих серверних архітектур. При розширенні системи Web-адміністратори просто масштабують сервер, додаючи до нього додаткові процесори, оперативну пам'ять та диски для збільшення навантаження. При наявності високопродуктивного програмного забезпечення такі сервери дозволяють швидко і ефективно здійснювати пошук.

Метою механізму пошуку є ведення постійного індексу, який оновлюється. Таким чином, його інтелектуальні агенти повинні постійно відвідувати Web-вузли, знаходити зміни та модифікувати індекс. Вони повторюють ці візити кожні декілька годин або місяців (в залежності від популярності вузла або числа змін сторінки за певний період). Багато механізмів пошуку керуються популярністю Web-вузла, враховуючи кількість зв'язаних з ним зовнішніх вузлів.

Пошукові системи не обмежують свої подорожі лише системою Word Wide Web. Наприклад Lycos та Alta Vista перевіряють також архіви груп Usenet, вузли ftp та "простір Gopher", а Galaxy шукає інформацію в Hytelnet - каталозі вузлів telnet. Взагалі усі механізми пошуку можна розбити на чотири основні категорії:

а) індекси за ключовими словами;

б) тематичні;

в) спеціалізовані каталоги;

г) інструменти метапошуку.

Багато з відомих служб пошуку, таких як Yahoo, Lycos та Excite, суміщують вказані види пошуку, звертаючись до додаткових механізмів. Наприклад, служба Yahoo для виконання пошуку за ключовими словами використовує сервер AltaVista, а Excite - пропонує повний пошук за вузлами в своїх предметних каталогах одночасно з низкою спеціалізованих індексів за ключовими словами.

В індексах за ключовими словами, прикладом яких є WebCrawler, AltaVista, Open Text та Infoseek, Yandex збираються та індексуються тексти, які виявлені на різних Web-вузлах. Ключовим словом (Keyword) документа називається окреме слово чи словосполучення, яке ілюструє зміст даного документа. У багатьох текстових процесорах ключовим словом є текст, за яким здійснюється пошук потрібної інформації (з допомогою команд “найти” і “заменить”). Тут основний індекс формується з мільйонів документів Web. Механізми пошуку за ключовими словами шукають в індексному вказівнику документи, які відповідають умовам запиту, та повертають URL (а деякі і перші декілька стрічок або короткий опис сторінки). Більшість засобів пошуку за ключовими словами досить швидко повертають результати обробки запиту. Це пов'язано з тим, що вони здійснюють лише самі елементарні операції аналізу вмісту (або обходяться взагалі без нього). Списки документів формуються на базі інформації про зустріч на відповідних Web-сторінках термінів, які задані для пошуку. Щоб викоренити цей недолік, засоби індексування за ключовими словами присвоюють сторінкам коефіцієнти релевантності. В загальному випадку більш високий коефіцієнт буде у сторінки, яка містить пошукові терміни в заголовку, підзаголовку або в мета-дескрипторах HTML-коду. Такий підхід сприяє переміщенню документів з більшим числом входжень даного слова до початку списку.

Якщо потрібно вести пошук засобами Alta Vista, треба набрати текст запиту в поле введення Alta Vista і натиснути кнопку "поиск". Як сформувати запит? В основі запитів на пошук і вибірку інформації лежить апарат алгебри логіки. Однак пошук в Інтернет менш формалізований, ніж у структурованих базах даних.

Як приклад розглянемо деякі запити в Alta Vista. Найпростіший запит - відібрати сторінки в Інтернет, що містять задане слово, наприклад

"Інформатика". Якщо запит складається з декількох слів, то в Alta Vista передбачені такі правила: кілька слів, набраних через відступ, позначають запит, що відповідає логічній операції OR (АБО). Наприклад, за запитом *шкільна інформатика* будуть обрані сторінки, на яких є слово "шкільна" або слово "інформатика" (чи відразу обидва слова). Слід зауважити, що кількість таких документів дуже велика, до них можуть потрапити сторінки, що не мають ніякого відношення до інформатики. Кілька слів, взятих в лапки, сприймаються системою як єдине ціле. Наприклад, за запитом *"Шкільна інформатика"* будуть відібрані документи, у яких є цей символьний рядок.

Слова, з'єднані знаком "+", відповідають логічній операції AND (логічне "І"). Наприклад, за запитом *Шкільна + Інформатика* будуть відібрані документи, у яких містяться обидва ці слова. Зрозуміло, що кількість таких документів буде також великою. Пошук краще проводити, вказуючи слова з маленької літери.

Тематичні каталоги, з іншого боку, передбачають попереднє сканування вмісту документів і віднесення їх до однієї з декількох категорій, список яких можна переглядати. Наприклад, найбільш популярним є тематичний каталог Yahoo, який відображує Web-вузли у вигляді ієрархічної схеми, у вершині якої знаходяться загальні теми, а далі дерево поділяється на спеціалізовані гілки. Сама тематична служба Yahoo не передбачає кількісних оцінок інформаційного змісту, але для цього можна використовувати додаткові інструментальні можливості даного вузла. В інших тематичних каталогах, наприклад Excite або NetGuide Live, для оцінки простоти навігації, якості подання інформації та її змістовності в цілому запрошуються колективи аналітиків.

Не дивлячись на те, що тематичні каталоги не можуть порівнюватися за шириною обхвату з індексом, який складений за ключовими словами, вони здатні зменшити ймовірність звертання до вузлів, які не відносяться до справи. Крім того, в них даються корисні попередження:

- для доступу до того чи іншого вузла необхідний певний тип броузера;
- сторінки вузла містять багато графічної інформації;
- сторінки вузла призначені лише для дорослих.

Між іншим, тематичні каталоги далеко неідеальні, тому необхідно покладатись на міркування аналітика, в результаті чого багато вузлів може опинитись не у тій темі. Багато каталогів призначені для широкої аудиторії, і деякі не зовсім зрозумілі рядовим користувачам. До того ж тематичні каталоги орієнтовані на перегляд, а не на пошук, і практично не пропонують інструментів пошуку даних.

Спеціалізовані індекси призначені для знаходження об'єктів, які відносяться до певної теми чи до певного типу інформації. Наприклад,

індексний вказівник типу Citi.Net - являє собою добірку інформації про міста: від інформації про ресторани до огляду майбутніх подій; служба Deja News виконує пошук в архівах груп новин Usenet; вказівник OKRA обмежується пошуком адрес електронної пошти, а MovieLink – спеціалізується на інформації про кінофільми і за введеним поштовим кодом дає відомості про місцеві театри. Тут, на відміну від популярних пошукових систем, як правило, застосовується менш розвинена і швидкодіюча технологія, але жертвуючи швидкістю досягається більш висока релевантність.

Інструменти мета-пошуку або мета-індекси, дозволяють вдосконалити пошук шляхом передачі запиту одночасно декільком службам. При цьому результат об'єднується в загальний впорядкований за релевантністю список. Наприклад, служба Higway 61 використовує для пошуку інформаційні сервери Yahoo, AltaVista, Lycos, Inforseek, Excite і HotBot. Мета-пошук значно економить час, оскільки зникає необхідність відвідувати і опитувати кожен вузол окремо, і є можливість знайти документи, які інакше не були б знайдені.

Але пошук за мета-індексами нерідко досить повільний, оскільки доводиться очікувати результати від декількох серверів. Перевантажений або зупинений сервер здатен призвести до призупинення усієї процедури. У той же час існують методи впорядкування пошуку, коли результати, які отримуються від більш швидких серверів, виводяться першими, проте багато з існуючих механізмів їх не підтримують.

Ці методи мають ще один недолік: вони не дозволяють скористатися довідковою системою або файлами навчального керівництва кожного механізму пошуку, який застосовується, проте такі документи надають досить цінну допомогу в більш точній постановці запитів. Щоб звернутися до них, необхідно відвідати конкретний вузол. Ще гіршим є те, що в більшості інструментів мета-пошуку при звернені до конкретного механізму застосовуються задані за замовчуванням стилі запиту, а це не дозволяє використовувати їх більш розвинені можливості. Нерідко найбільш оптимальним виходом є застосування нового класу інструментів – персональних програм пошуку.

Універсальна служба пошуку (пошукова система) – це комплекс програм і потужних комп'ютерів, який виконує такі функції:

а) спеціальна програма (пошуковий робот) беззупинно переглядає сторінки "Всесвітньої павутини", вибирає ключові слова й адреси документів, у яких ці слова виявлені. Тут доречно згадати про індексований файл (окремий файл, що містить інформацію про фізичне розташування записів у якому-небудь файлі бази даних). На відміну від цього, щоб переглянути фактичний файл бази даних, програми бази даних користуються індексами, що дозволяють істотно прискорити пошук потрібної інформації;

б) Web–сервер приймає від користувача запит на пошук, перетворює його і передає спеціальній програмі – пошуковій машині;

в) пошукова машина переглядає базу даних індексів, складає список сторінок, що задовольняють умовам запиту (точніше список посилань на ці сторінки) і повертає його Web–серверу;

г) Web–сервер оформляє результати виконання запиту в зручному для користувача вигляді і передає їх на машину клієнта.

Для кращого та прискореного пошуку інформації необхідно: сформувати декілька запитів, що складаються з одного слова. В результаті пошуку буде вказано скільки документів було знайдено. Для зменшення кількості документів, якщо це можливо, спробуйте для запиту взяти поширене слово (автомобіль), більш конкретне визначення „фольксваген" та спеціалізований термін „пассат". Тобто намагайтеся розбити термін, який шукаєте, на більш прості поняття та лише потім їх ускладнювати. Це дозволить краще оцінити "можливості" пошукового сервера, а також скоротити пошук необхідної інформації.

Проведемо порівняльну характеристику найбільш відомих, поширених і потужних пошукових систем :

Таблиця 2

Ресурс	**Опис ресурсу**	**Переваги**	**Недоліки**
Англомовні пошукові системи			
www.google.com	Пошукова машина/каталог. База даних більше 1 млрд. сайтів	Простий у роботі, результати пошуку відповідають запиту, можливість настройки мови інтерфейсу, швидкість пошуку	Можливостей розширеного пошуку небагато
www.altavista.com	Пошукова машина/каталог. Портал. Виконує пошук в новинах, пошук аудіо. Кожен день опрацьовує 50 млн. запитів.	Багато можливостей розширеного пошуку, відповідність запитаупошуку, пошук на 25 мовах	Сервер загромаджений, пошук рос. мовою не ефективний.

Російськомовні системи			
www.rambler.ru	Двомовна (рос., англ.) пошукова машина, повнотекстова, база має більше 350 тис. сторінок. Портал має багато служб	Підтримує всі кодування кирилиці, швидко працює	Уповільнене поновлення бази даних, несвоєчасне видалення застарілих посилань
www.yandex.ru	Двомовна (рос., англ..) пошукова машина/каталог, повнотекстова, здійснює пошук в російськомовному Інтернеті	Система допомоги, простий та зручний інтерфейс	Немає
Українські пошукові системи			
www.uaport.net	Пошукова машина/каталог, портал, пошук ведеться за українськими ресурсами, але є посилання на рос. сайти	Детальний опис результатів пошуку, мало дублюючих посилань	Погана організація каталогу, повільне завантаження першої сторінки
www.metaukraine.com	Пошукова машина/каталог, портал, пошук ведеться тільки в українському Інесі.	Велика база даних, продумана система допомоги	Багато посилань, що повторюються; немає чіткого опису мови запитів

Спеціалізовані довідкові служби – це тематичні каталоги (subject catalogs), у яких зібрані більш-менш структуровані зведення про адреси серверів за тією чи іншою тематикою. На відміну від універсальних баз індексів, тематичні каталоги складаються фахівцями і забезпечують клієнта більш строгою, достовірною систематизованою інформацією про мережу.

Крім того, багато сайтів Інтернет мають у своєму розпорядженні власні механізми пошуку (у межах даного сайта). У першу чергу – це механізм контекстного пошуку, а також спеціалізований пошук за прізвищами (наприклад, персони комп'ютерного бізнесу), товарами

(рекламні сайти), назвами фірм і т.д. Контекстний пошук на поточній сторінці передбачений і в Internet Explorer.

Деякі сторінки в Інтернет (наприклад, сторінки пошукових систем) спеціально призначені для прийому й обробки запитів на пошук. Microsoft у Internet Explorer 5.5 пропонує власну сторінку пошуку.

Internet Explorer не займається пошуком, він приймає від користувача запит, обробляє його і передає відповідній пошуковій системі.

Контрольний приклад

1. За адресою:

http://www.charm.net/~brooklyn/People/JackKerouac.html,

вкажіть, що є *іменем домену,* де знаходиться маршрут каталогу та ім'я документа.

Розглянемо окремі частини адреси URL. Перша частина HTTP (Hyper Text Transfer Protocol) означає *набір правил,* що керують обміном даних у Web. Кожна адреса URL для серверів Web починається з букв http.

Наступна частина URL, а саме **www.charm.net,** називається *іменем домена*. Кожен комп'ютер у Інтернет має унікальне ім'я домена, що відрізняє його від інших комп'ютерів у Інтернет. Імена домена звичайно дають деяке уявлення про організацію, в якій працює сервер.

Далі знаходиться маршрут каталогу (**/~brooklyn/People**), що показує місцезнаходження файлу, а після нього — ім'я документа „**JackKerouac.html”**. Розширення імені файлу означає *Hyper Text Marcup Language* (гіпертекстова мова міток). Ця мова являє собою простий набір команд, що вказує броузеру Web, як виводити документ.

2. За допомогою www.uaport.net, www.metaukraine.com здійснити пошук інформації: журнал Chip.

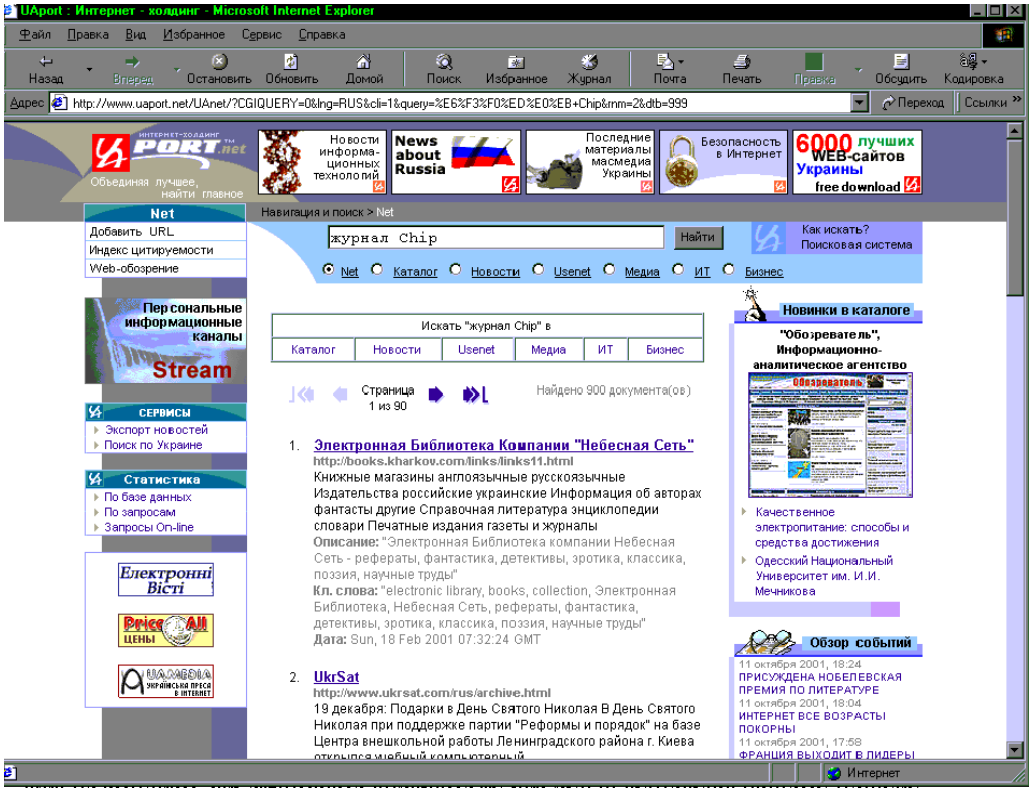
Для цього необхідно за допомогою IE або NN зайти на сайт www.uaport.net, або www.metaukraine.com, та в графі Пошук ввести слово: журнал Chip; після цього пошуковий каталог видасть перелік інформації за запитом, що вас цікавить, серед якої необхідно вибрати ту, що є найбільш відповідною вашому запиту.

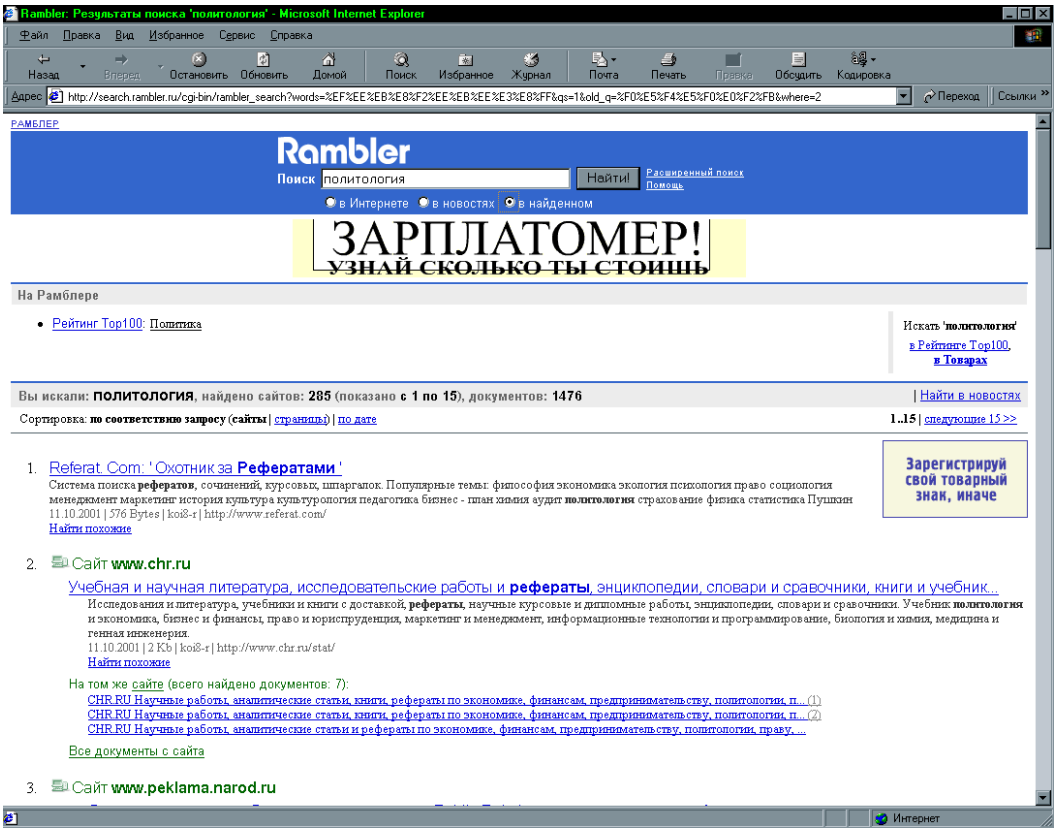
3. За допомогою www.altavista.com здійснити пошук за темою: business . Для цього необхідно за допомогою броузера зайти на сайт www.altavista.com та в графі Search ввести слово: business , після цього пошуковий каталог видасть перелік інформації (гіпертекстових посилань-сайтів) за запитом, що вас цікавить.

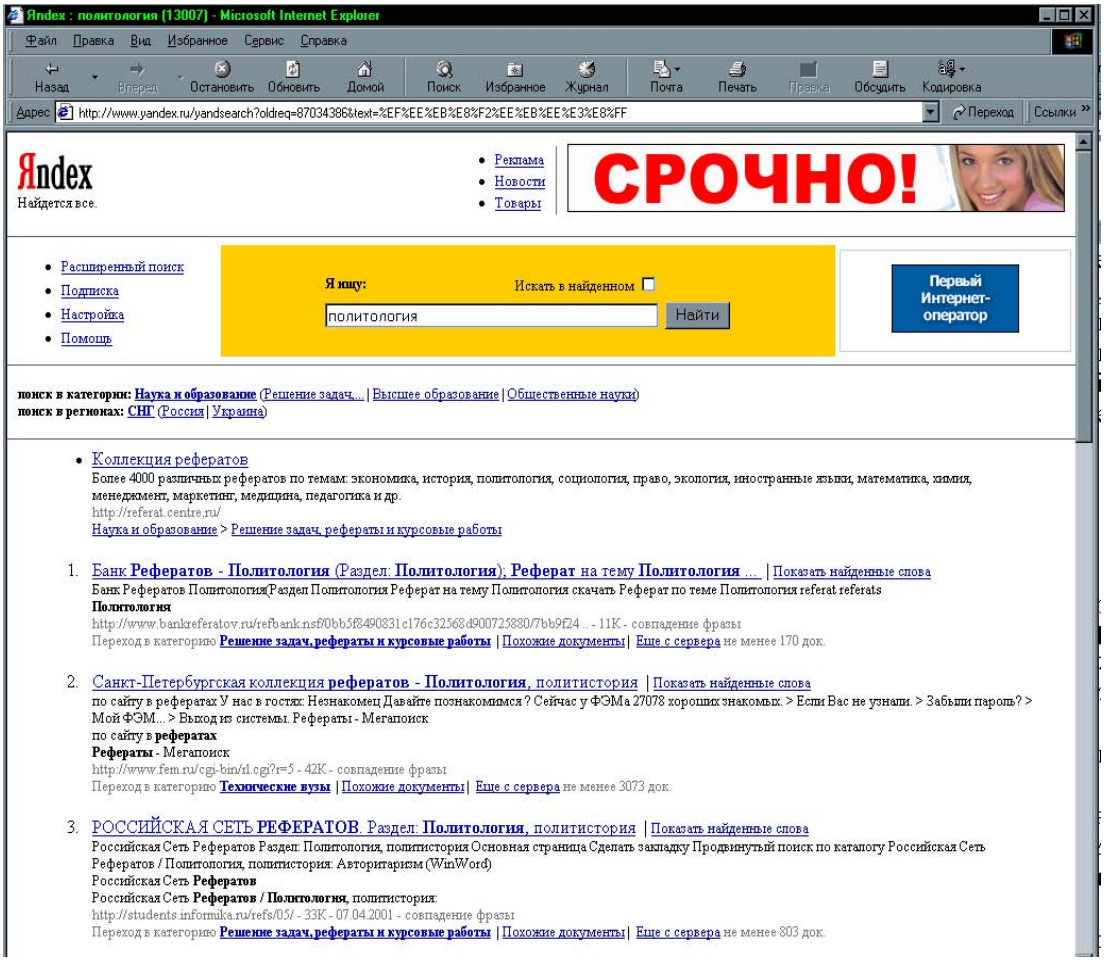
4. Виконайте пошук інформації за допомогою пошукових каталогів www.yandex.ru, www.rambler.ru за темою: реферати, політологія. Порівняти отримані результати

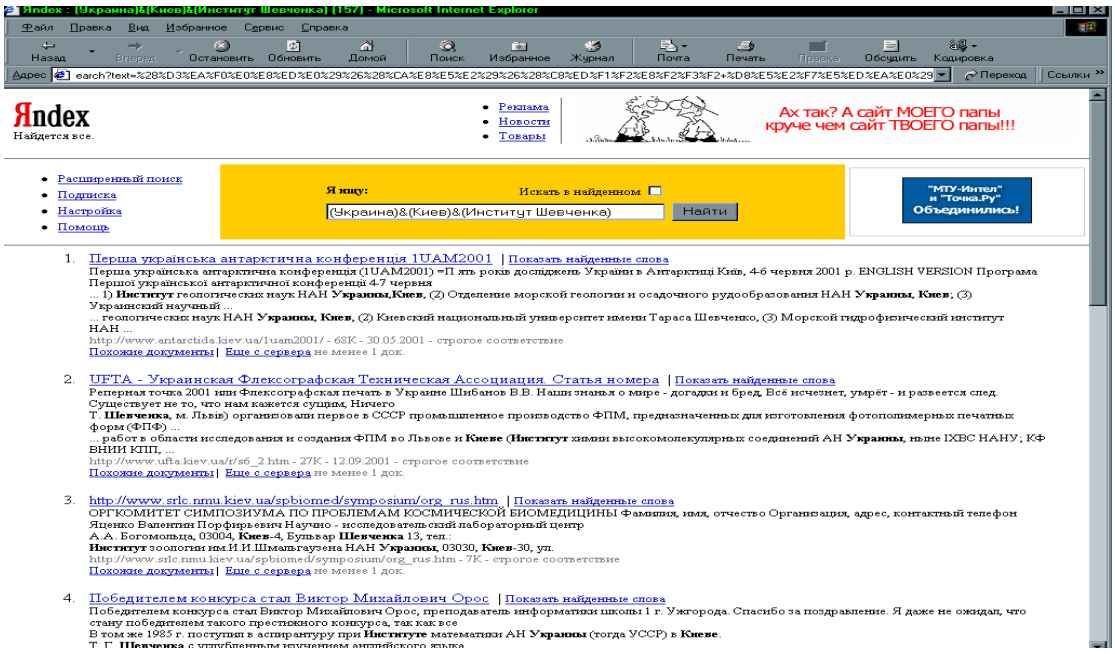
Для цього необхідно за допомогою броузера зайти на сайт www.yandex.ru, www.rambler.ru та в графі “Пошук” ввести слово:

реферати, політологія, після цього пошуковий каталог видасть перелік інформації (гіпертекстових посилань-сайтів) за запитом, що вас цікавить, потім, використовуючи функцію “пошук” введемо слово “політологія”.









Порівнюючи результати пошуку інформації, необхідно зазначити її схожість, але при цьому пошук за допомогою www.yandex.ru виявився більш відповідним запиту.

Виконати пошук інформації за ключовими словами: Україна, Київ, Інститут Шевченка з використанням “І”. Для пошуку необхідної інформації скористаємося пошуковим каталогом www.yandex.ru де в полі “Я ищу” введемо спочатку (Украина)&(Киев)&(Институт Шевченка), після чого отримаємо перелік сайтів з відповідною тематикою, серед яких вибираємо той, що найбільш підходить запиту.

Питання для самоперевірки

1. Перерахуйте можливі види доступу до Інтернет.
2. Що таке броузер? Для чого він використовується?
3. Які Ви знаєте механізми пошуку в Інтернет?
4. Які пошукові машини/каталоги Вам відомі?
5. Що таке тематичні каталоги?
6. Як виконується пошук інформації за ключовими словами?
7. Що означають слова, з'єднані "+" (плюсом)?
8. Назвіть переваги та недоліки тематичних каталогів.

Література

1. Синтия Морган. Средства поиска информации в Інтернет // Компьютеры. - М., 1997. - №6. - С. 21-24, 52-54.
2. Эд Крол. Все об Internet. К.: Диалектика, 1995 – 590 с.

ІІ АДМІНІСТРУВАННЯ МЕРЕЖНИХ ОПЕРАЦІЙНИХ СИСТЕМ

Лабораторна робота №7

ЛОГУВАННЯ В ОПЕРАЦІЙНІЙ СИСТЕМІ WINDOWS NT

Мета роботи: придбати навички логування в операційній системі Windows NT, ознайомитися з наявними відмінностями при роботі з різними рівнями доступу.

Порядок виконання роботи

1. Виконайте логічний вхід у систему Windows NT, ввівши в полі User Name слово Administrator і в полі Password – пароль адміністратора.

2. Створіть необхідні типи користувачів за таким принципом:

- ім'я – Guest + №варіанта (наприклад, для варіанта 3 – Guest3). Група – Guests. Пароль обирається самостійно;

- ім'я – User + №варіанта. Група – Users. Пароль – обирається самостійно;

- ім'я – PUser + №варіанта. Група – Power Users. Пароль обирається самостійно.

3. Створіть новий каталог у розділі C: локального диска з ім'ям LabNt. У каталозі C:\LabNt створіть новий каталог, з ім'ям, що відповідає номеру вашого варіанта (наприклад, для варіанта 5 це буде Guest5). Виконайте копіювання декількох файлів у створений каталог. Серед файлів повинні бути файли таких типів: текстові файли чи документи; файли, що виконуються (програми).

4. Для створеного каталогу встановіть дозволи, додавши права для різних типів користувачів згідно з варіантами, що зазначені в табл. 1.

Таблиця 1 - Варіанти перевірки операцій з файлами

Варіант	Тип користувача 1 типу	Необхідні операції	Тип користувача 2 типу	Необхідні операції
1	**2**	**3**	**4**	**5**
1	G	R, W	U	R, W, D
2	G	X, D	U	R, X, D
3	G	R, X	E, U	R, W, D
4	G	R, D	E, U	X, C

Продовження таблиці 1

1	2	3	4	5
5	G	R, W, C	G	X, D, C
6	U	R, X	E, P	R, W, C
7	U	R, W	P	X, D, C
8	U	X, D	P	R, D
9	U	R, X	E	R, X, D
10	U	R, X, C	U	R, D, C
КП	G	R, D	U	X, D

Зауваження: Якщо серед необхідних типів користувачів немає Everyone (позначається як E), то в списку користувачів необхідно усунути можливість “Full control for everyone”. Якщо такий тип є (наприклад у варіанті 4), потрібно встановити відповідний тип доступу як для зазначеного типу юзера, так і для Everyone (наприклад: варіант 4 – встановлюємо права для Users: X, C та для Everyone - також X, C).

Умовні позначення типів юзерів:

G – Guests;

U – Users;

P – Power Users;

A – Administrators;

E – Everyone.

Умовні позначення операцій :

R (Read) – читання;

W (Write) – запис;

X (eXecute)– виконання програм;

D (Delete) – видалення;

C (Change permissions) – зміна дозволів.

5. Виконайте логічний вхід до системи відповідно до першого типу користувача. Перевірте можливості виконання дій. Якщо є можливості змінювати дозволи (тобто C у завданні), то встановіть зміни відносно дій, позначених у другому типу користувача.

6. Виконайте логічний вхід до системи відповідно до другого типу користувача. Перевірте можливості виконання дій.

7. Виконайте перевірку можливостей згідно з варіантами, що зазначені в табл. 2:

Таблиця 2 - Варіанти перевірки операцій

Варіант	**Тип користувача**	**Необхідні операції**
1	G	зміна системного часу
2	G	зупинка системи
3	G	зміна вигляду робочого столу
4	G	зміна папки Programs
5	U	зміна системного часу
6	U	зупинка системи
7	U	зміна вигляду робочого столу
8	U	зміна папки Programs
9	P	зміна системного часу
10	P	зупинка системи
КП	P	зміна вигляду робочого столу

Теоретичні відомості

Операційна система Windows NT реалізована у двох варіантах: Windows NT Server і Windows NT Workstation. Windows NT Server 4.0 – мережна операційна система з додатками для Інтернет, сервісами файлів та друку, службою віддаленого доступу, вбудованим маршрутизатором, індексуванням файлів та керуванням мережею. Другий варіант Windows NT – Windows NT Workstation 4.0 нагадує NT Server, але вона оптимізована у якості операційної системи для робочої станції.

Домен – це основна одиниця адміністрування і забезпечення безпеки в Windows NT. Для домена існує загальна база даних облікової інформації користувачів домена (user accounts) і ресурсів домена – комп'ютерів (computer accounts) і принтерів (printer accounts). Користувач домена виконує один логічний вхід у домен і одержує доступ відразу до всіх дозволених ресурсів цього домена. Членами домена є як користувачі, так і комп'ютери. При відсутності доменної організації кожен комп'ютер Windows NT Workstation і Windows NT Server зберігає власну базу облікових даних користувачів – SAM (Security Access Manager). У цій базі зберігається вся необхідна системі інформація про користувача – ім'я,

пароль (у зашифрованому вигляді) і так званий SID - Security Identifier. Ідентифікатор SID відіграє ключову роль у процесі надання користувачу доступу до захищених ресурсів системи – файлів, принтерів і т.п. У списку прав доступу ресурсу ACL зберігається інформація про конкретні номери SID, яким дозволений той чи інший вид доступу. Кожен SID є унікальним числом. При зміні імені користувача його SID не змінюється.

Комп'ютер домена також характеризується іменем і ідентифікатором SID між якими є таке ж співвідношення, як між іменем і ідентифікатором SID користувача.

У домені обов'язково є сервер Windows NT Server, що виконує роль первинного контролера домена – Primary Domain Controller, PDC. Цей контролер зберігає первинну копію бази даних облікової інформації користувачів домена – SAM PD. Усі зміни облікової інформації спочатку виробляються саме в цій копії. Основний контролер домена завжди існує в єдиному екземплярі.

Крім основного контролера, у домені можуть існувати кілька резервних контролерів – Backup Domain Controllers, BDC. Ці контролери зберігають репліки бази облікових даних. Усі резервні контролери на додаток до основного можуть обробляти запити користувачів на логічний вхід у домен. База даних SAM BD завжди є копією (з точністю до інтервалу синхронізації) бази SAM PD.

Крім доменної структури, мережа Windows NT може бути організована як *робоча група*. Робоча група – це логічне об'єднання комп'ютерів, звичайно не більше 10, що можуть розділяти свої ресурси. Однак при цьому кожен комп'ютер робочої групи має власну базу облікових даних, що містить інформацію тільки про його локальних користувачів. На комп'ютерах робочої групи можуть бути встановлені і Windows NT Server, і Windows NT Workstation. І користувачі, і ресурси кожного члена робочої групи адмініструються засобами даного комп'ютера. Таким чином, робоча група не дає усіх тих можливостей, що несе в собі централізована доменна довідкова служба.

Якщо користувач виконує операцію логування в комп'ютері, що є членом домена, то забезпечується можливість ввійти в комп'ютер локально або ввійти в домен. Цей вибір ви здійснюєте, коли на панелі *Logon Information* у полі *Domain* вказуєте ім'я комп'ютера або домена.

У мережі Windows NT можуть бути визначені такі типи користувачів і груп користувачів:

- *локальний інтерактивний користувач комп'ютера* (користувач, що заведений у локальній обліковій базі даних комп'ютера, і який працює з ресурсами комп'ютера інтерактивно);
- *локальний мережний користувач комп'ютера* (користувач, що заведений у локальній обліковій базі даних комп'ютера, і який працює з ресурсами комп'ютера через мережу);

- *користувач домена* (користувач, що заведений у глобальній обліковій базі даних домена на PDC);
- *локальна група комп'ютера* (може створюватися на всіх комп'ютерах домена, крім PDC і BDC, у яких вона вироджується в локальну групу домена);
- *локальна група домена* – складається з користувачів домена (заводиться тільки на PDC);
- *глобальна група домена* – складається з користувачів домена (може входити в локальну групу домена).

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

Для однозначної ідентифікації глобальної групи в багатодоменній мережі, використовується її ім'я, наприклад Marketing\Managers, де Marketing – ім'я домена, Managers – ім'я глобальної групи.

Захист виконується для того, щоб розподілити певні можливості між групами користувачів та окремими користувачами. При цьому маємо такі види об’єктів:

- *Каталоги і файли*. Виконуються операції: read, full control, change, add, ...;
- *Принтери;*
- *Операційна система*. Стосовно цього типу об'єктів визначаються права на виконання різних сервісів і утиліт: вхід, архівування файлів, зміна конфігурації панелей Program Manager.

Операції доступу – це дії об'єктів над суб'єктами. Операції можуть бути дозволеними чи забороненими, або взагалі не мати змісту для даної пари об'єкта і суб'єкта.

Уся безліч операцій розділяється на підмножини, що мають особливі назви:

- дозволи (permissions) – це безліч операцій, які можуть бути визначені для суб'єктів усіх типів стосовно об'єктів типу файл, каталог, принтер;
- права (user rights) – визначаються для об'єктів групи 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 відображаються на екрані дисплея, включення нових іконок у Desktop, можливість використання команди Run і т.п.

Контрольний приклад

1. Натисніть Ctrl-Alt-Del. Введіть в полі user name – Administrtor, в полі password – пароль адміністратора.

2. Для створення користувача здійснюємо такі дії: запускаємо програму конфігурування груп користувачів: Start->Programs-> Administrative Tools ->User manager for Domains. Отримуємо вікно, яке зображено на рис. 1.

У меню обираємо команду User->New User. Отримуємо нижченаведене вікно (рис.2), в якому виконуємо введення імені користувача, наприклад, GuestKP. Далі обираємо Groups та додаємо необхідну стандартну групу зі списку.

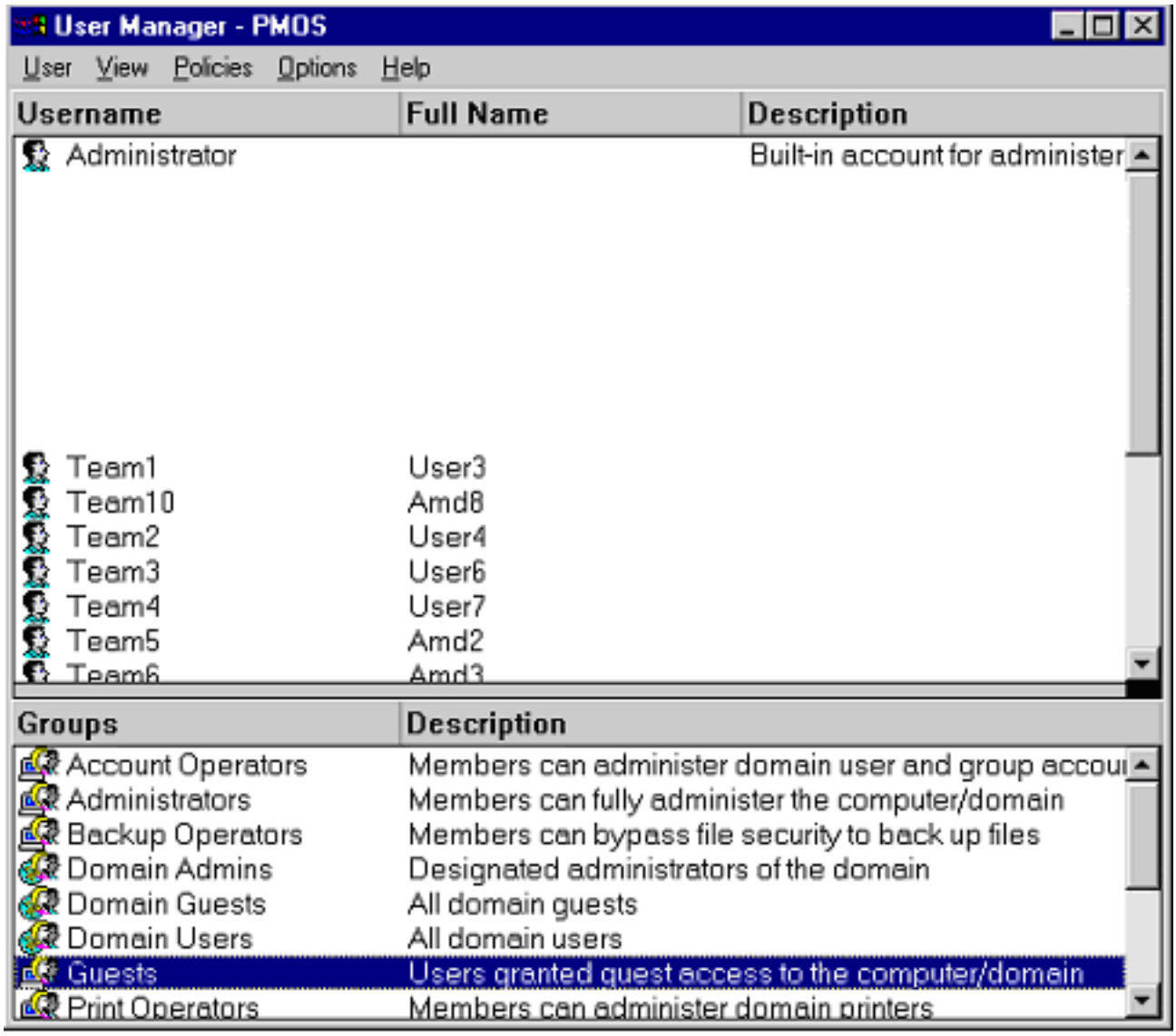


Рисунок 1 - Адміністрування у вікні User manager for Domains

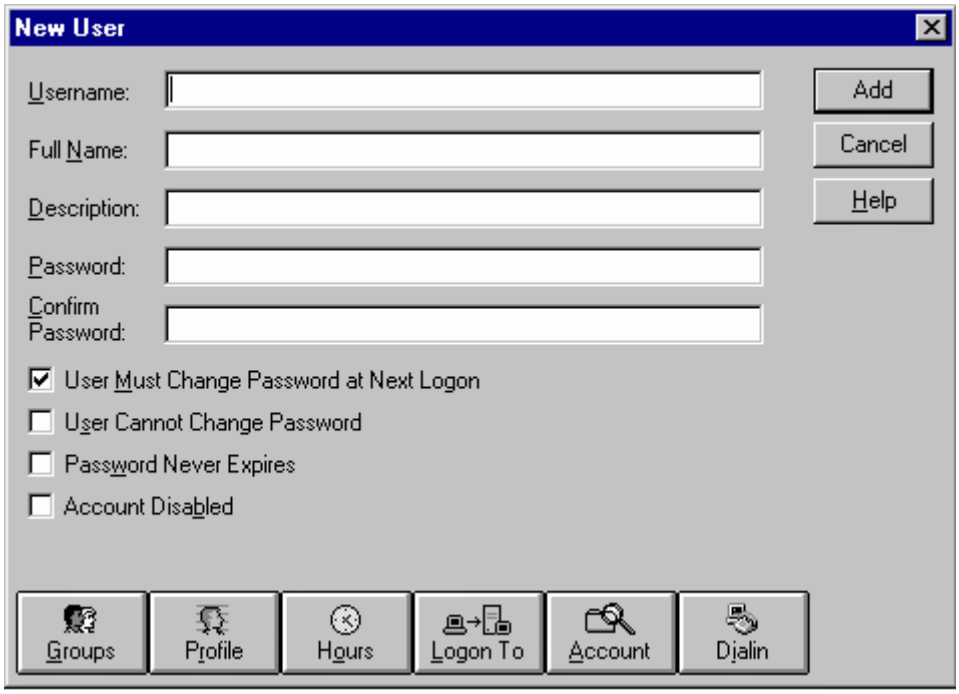


Рисунок 2 - Створення нового користувача

3. Створюємо на диску C каталог LabNt та в ньому папку ForExample. Для перевірки властивостей копіюємо декілька файлів, включаючи й ті, що виконуються. Створюємо документ Microsoft Word.

4. За допомогою explorer знаходимо створену нами папку, натискаючи правою клавішею миші, обираємо пункт properties. Отримуємо вікно (рис. 3). У закладці Security натискаємо кнопку Permissions.

Додаємо нову групу та обираємо для неї Special Directory Access. Обираємо необхідні для прикладу можливості читання та видалення для Guest, виконання файлів і видалення для User. Видаляємо можливість Full Control для Everyone, змінивши її у випадаючому листі на No Access. Після цього проводимо завершення сеансу (log off), виконавши потрібну команду в меню Start.

5. Проводимо логічний вхід до системи як Guest, ввівши відповідний пароль. Заходимо у створену папку та намагаємося виконати певні операції. Для цього запускаємо документ Word та пересвідчуємося, що в результаті маємо доступ тільки в режимі Read-Only. Виконання заборонене. Видалення файлів дозволене.

6. Проводимо логічний вхід до системи як користувач, ввівши відповідний пароль. Заходимо у створену папку і також намагаємося виконати визначені операції. Пересвідчуємося, що на відміну від попереднього логічного входу, тепер нам недоступним є навіть читання файлу, проте дозволеним є виконання файлів, що можуть виконуватися. Також можна вилучати файли.

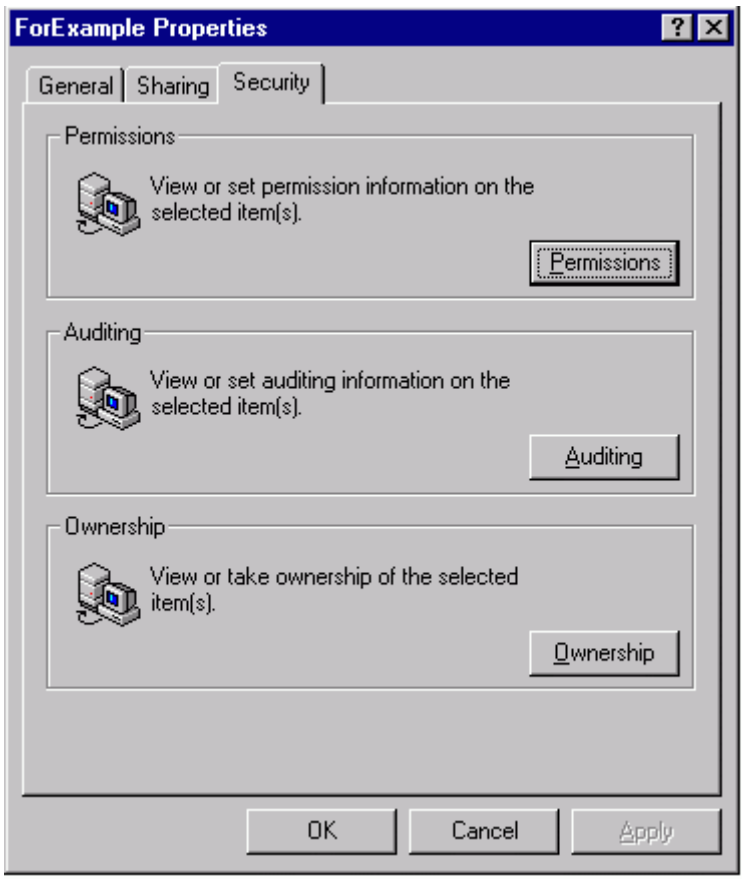


Рисунок 3 - Визначення властивостей

7. Проводимо логічний вхід до системи як адміністратор, ввівши відповідний пароль. Проводимо зміну наявних іконок – міняємо місця, змінюємо назви. Виходимо з системи та входимо знову. Пересвідчуємося, що внесені зміни залишилися.

Питання для самоперевірки

1. Чим відрізняються системи Windows NT Workstation та Windows NT Server?
2. Що таке домен? Що таке робоча група?
3. Які існують типи та групи користувачів?
4. Які розрізняють види об’єктів?
5. Які види операцій доступні для користувачів?

Література

1. Бойс Дж. Расширенное техническое руководство по Windows NT Workstation 4.0. – СК: Пресс, в 2 т. – 1998. – С. 328-480.

2. К. Айвенс. Эксплуатация Windows NT в подлиннике. – СПб.: BHV, 1998. – 552 с.

Лабораторна робота № 8
ОСНОВИ АДМІНІСТРУВАННЯ В WINDOWS NT

Мета роботи: отримати відомості про організацію і функціонування операційної системи Windows NT, отримати навички адміністрування.

Завдання

1. Здійсніть операцію створення і видалення користувачів.

- Створіть нових локальних користувачів (Увага! Щоб уникнути проблем із запам'ятовуванням паролів, призначайте новим користувачам паролі, що збігаються з їх іменами).
- Переіменуйте користувача *Адміністратор* в adm_ws1 (або adm_ws2, adm_ws3, ... - відповідно до імені комп'ютера), для того, щоб можна було в подальших експериментах не плутати адміністраторів різних доменів і робочих станцій.
- Створіть для груп *Адміністратори, Користувачі і Досвідчені користувачі* нових користувачів, дайте їм імена з префіксами типу _ws1, наприклад, Nik_ws2.

Для нових користувачів домена (крім адміністраторів) виконайте такі дії:

- задайте інтервал часу, коли користувачеві дозволений вхід в систему — пункт Hours в меню New User.
- забороніть вхід у систему з деяких комп'ютерів в межах домена. Встановіть дату закінчення терміну дії облікової інформації даного користувача — пункт Account в меню New User.

2. Ознайомтесь з дозволами (permissions), що надаються користувачам і групам користувачів для доступу до файлів і каталогів.

2.1. Створіть новий каталог у розділі локального диска, внесіть у нього декілька файлів з існуючих каталогів шляхом копіювання.

2.2. Перевірте, які індивідуальні дозволи для цього каталогу встановлені операційною системою за замовчуванням.

2.3. Задайте каталогу стандартний дозвіл CHANGE для одного з нових користувачів, наприклад, user_dom1.

2.4. Відмітьте у нижченаведеному списку, які дії ви можете виконати відповідно до свого каталогу:

- переглянути імена файлів у каталозі;
- переглянути атрибути каталогу;
- додати файли і підкаталоги;
- змінити атрибути каталогу;
- перейти в підкаталоги каталогу;
- переглянути власника каталогу і дозволу;
- вилучити каталог;

- змінити дозволи каталогу;
- стати власником каталогу;

3. Виберіть один з файлів даного каталогу. Перевірте, які дії ви можете виконати відповідно до цього файла.

- переглянути дані файла;
- переглянути атрибути файла;
- змінити атрибути файла;
- змінити і додавати дані у файл;
- виконати файл, якщо це програма;
- переглянути власника файла і дозволу;
- вилучити файл;
- змінити дозволи файла;
- стати власником файла.

Таблиця 1 - Варіанти завдань

Варіант	Години	Підключень	Ім’я каталогу
1	18.00-22.00	50	Mydir
2	6.00-12.52	25	Newdir
3	14.00-19.00	19	Dir#1
4	11.25-23.00	20	Labdir
5	14.30-20.00	39	NTlab
6	18.00-4.05	28	AdmNT
7	22.30-6.00	63	LabadmNT
8	8.00-24.18	100	MyDir
9	17.00-21.00	15	New
10	23.45-16.15	30	Мой каталог
кп	11.00-12.00	42	Новый каталог

Теоретичні відомості

Користувачі, ресурси і операції доступу

Адміністрування полягає у створенні облікової інформації користувачів (що визначає ім'я користувача, належність 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 домена. Права доступу до ресурсів задаються різними засобами, у залежності від типу ресурсу.

Можливість використання комп'ютерів Windows NT Workstation як робочих станцій здійснюється з 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 каталогів — з допомогою Server Manager, доступ до принтерів — з панелі Printers.

Типи користувачів і груп користувачів

У мережі Windows NT можуть бути визначені такі типи користувачів і груп користувачів:

- *локальний інтерактивний користувач комп'ютера (*користувач, який заведений у локальну облікову базу даних комп'ютера і який працює з ресурсами комп'ютера інтерактивно);
- *локальний мережний користувач комп'ютера (*користувач, який заведений в локальну облікову базу даних комп'ютера і який працює з ресурсами комп'ютера через мережу);
- *користувач домена (*користувач, який заведений у глобальну облікову базу даних домена на PDC);
- *локальна група комп'ютера (*може створюватися на всіх комп'ютерах домена, крім PDC і BDC, в яких вона вироджується в локальну групу домена);
- *локальна група домена* — складається з користувачів домена (заводиться тільки на PDC);
- *глобальна група домена*. — складається з користувачів домена (може входити до локальної групи домена).

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

Для однозначної ідентифікації глобальної групи в багатодоменній мережі використовується складове її ім'я, наприклад Marketing\Managers, де Marketing — ім'я домена, Managers — ім'я глобальної групи.

Типи об'єкті:

- *каталоги і файли*;
- *принтери;*
- *операційна система*.

Локальні, глобальні і спеціальні групи

Windows NT Server використовує три типи груп: локальні, глобальні і спеціальні. Кожний тип має своє призначення, можливості і обмеження.

Локальна група може визначатися для домена або для комп'ютера. Локальні групи дають користувачам права і дозволи на ресурси того комп'ютера (або домена), де зберігається облікова інформація локальної групи. Доступ до ресурсів комп'ютера — Windows NT Workstation або Windows NT Server може бути визначеним тільки для членів локальної групи цього комп'ютера, навіть якщо ці комп'ютери є членами домена.

Доступ до ресурсів комп'ютера для користувачів домена забезпечується за рахунок механізму включення у локальну групу окремих користувачів домена і глобальних груп домена. Включені користувачі і групи дістають ті ж права доступу, що й інші члени даної групи. Механізм включення глобальних груп у локальні є основним засобом централізованого адміністрування прав доступу в домені Windows NT.

Локальна група не може містити інші локальні групи. Тому в мережі, що використовує модель робочої групи, немає можливості визначити на одному комп'ютері всіх користувачів мережі і надати їм права доступу до ресурсів інших комп'ютерів. У будь-якому випадку локальна група об'єднує деяке число користувачів і глобальних груп, яким привласнюється загальне ім'я — ім'я локальної групи. Локальні групи можуть включати користувачів і глобальні групи не тільки даного домена, але і будь-яких довірених доменів.

Windows NT Workstation і Server підтримують декілька вбудованих локальних груп для виконання системних задач. Адміністратор може створювати додаткові локальні групи для управління доступом до ресурсів. Вбудовані локальні групи поділяють на дві категорії — *адміністратори* (Administrators), які мають усі права і дозволи на даний комп'ютер, і *оператори*, які мають обмежені права на виконання специфічних задач. Для Windows NT Server є такі групи-оператори: оператори архівування (Backup Operator), реплікатори (Replicator), оператори сервера (Serevr Operator), принт-оператори (Print Operator) і оператори облікової інформації (Account Operator). Для Windows NT Workstation є тільки дві групи операторів — Backup Operators і Power Users. Крім того, як на Windows NT Server, так і на Windows NT Workstation, є вбудовані локальні групи Users — для звичайних користувачів, і Guests — для тимчасових користувачів, які не можуть мати профілю і повинні володіти мінімальними правами.

Для спрощення організації надання прав доступу користувачам з іншого домена в Windows NT введене поняття глобальної групи.

Глобальна група користувачів — це група, яка має ім'я і права дійсні для всієї мережі, на відміну від локальних груп користувачів, які мають імена і права, дійсні тільки в межах одного домена. Адміністратор довірного домена може надавати доступ до ресурсів свого домена користувачам з глобальних груп тих доменів, яким даний домен довіряє. Глобальні групи можна включати до складу локальних груп користувачів ресурсного домена. Глобальна група — це деяке число користувачів одного домена, які групуються під одним ім'ям. Глобальним групам можуть даватися права і дозвіл шляхом включення їх в локальні групи, які вже мають необхідні права і дозволи. Глобальна група може містити тільки облікову інформацію користувачів з локальних облікових баз даних, вона не може містити локальні групи або інші глобальні групи.

Існує три типи вбудованих глобальних груп: адміністратор домена (Domain Admins), користувачі домена (Domain Users) і гості домена (Domain Guests). Ці групи спочатку є членами локальних груп адміністраторів, користувачів і гостей відповідно.

Для спрощення адміністрування необхідно використовувати вбудовані групи там, де тільки це можливо. Рекомендується формувати групи в такій послідовності:

1. В обліковому домені необхідно створити користувачів і додати їх до глобальних груп.
2. Включити глобальні групи до складу локальних груп ресурсних доменів.
3. Надати локальним групам необхідні права і дозволи.

Спеціальна група — використовується виключно Windows NT Server для системного доступу. Спеціальні групи не містять облікової інформації користувачів і груп. Адміністратори не можуть приписати користувачів до цих груп. Користувачі або є членами цих груп за замовчуванням (наприклад, кожний користувач є членом спеціальної групи Everyone), або вони стають ними залежно від своєї мережної активності.

Існує 4 типи спеціальних груп:

- Network (Мережна)
- Interactive (Інтерактивна)
- Everyone (Для усіх користувачів)
- Creator Owner (Творець-Власник).

Будь-який користувач, який хоче отримати доступ до ресурсу, розділеного по мережі, автоматично стає членом групи Network. Користувач, який локально ввійшов у комп'ютер, автоматично включається в групу Interactive. Будь-який користувач мережі є членом групи Everyone. Адміністратор може призначити групі Everyone будь-які права. При цьому адміністратор може надати будь-які права користувачеві, не заводячи на нього облікової інформації на своєму комп'ютері. Група Creator Owner містить облікову інформацію користувача, який створив ресурс або володіє ним.

У файловій системі NTFS дозволи групі Creator Owner надаються на рівні каталогу. Власник будь-якого каталога або файла, створеного в даному каталозі, отримує дозволи, притаманні групі Creator Owner. Наприклад, можна призначити якому-небудь каталогу для членів групи Everyone дозвіл Read (Читання), а групі Creator Owner надати доступ Full Control (Повне управління). Будь-який користувач, який створює файли або підкаталоги в цьому каталозі, буде мати до них доступ Full Control.

Вбудовані групи користувачів і їх права

Права визначаються для об'єктів типу „група” на виконання деяких системних операцій: створення резервних списків, вимкнення комп'ютера (shutdown) і т.п. Права призначаються з допомогою User Manager for

Domains. У таблиці 1 наведені права для вбудованих локальних груп домена.

1. Оператори облікової інформації (Accounts operators) не можуть змінювати облікову інформацію адміністраторів або змінювати глобальну групу Domain Admins, або локальні групи Administrators, Server Operators, Account Operators, Print Operators, або Backup Operators.

2. Члени групи Users мають право створювати локальні групи домена, для чого їм необхідно дозволити входити локально в сервер або користуватися утилітою User Manager for Domains.

3. Хоча Everyone має право блокувати сервер, проте тільки користувачі, які можуть також входити локально в цей сервер, можуть дійсно його заблокувати.

Схожі права можна задати і для Windows NT Server, що не виконує роль PDC або BDC, за допомогою утиліти User Manager for Domains, а також для Windows NT Workstation за допомогою утиліти User Manager.

Можливості користувачів визначаються для окремих користувачів і полягають у виконанні дій, що стосуються реорганізації їх операційного середовища:

а) включення нових програмних одиниць (іконок) у групу програм панелі Program Manager;

б) створення програмних груп Program Manager;

в) зміна складу програмних груп;

г) зміна властивостей програмних одиниць (наприклад, включення в стартову групу);

д) запуск програм з меню FILE в Program Manager;

е) встановлення з'єднань з мережним принтером, крім тих (які вже передбачені в профілю користувача).

Можливості користувача є частиною так званого профілю користувача (User Profile), який можна змінювати за допомогою утиліти User Profile Editor. Профіль також включає і установки середовища користувача на його робочому комп'ютері, такі як кольори, шрифти, набір програмних груп і їх зміст.

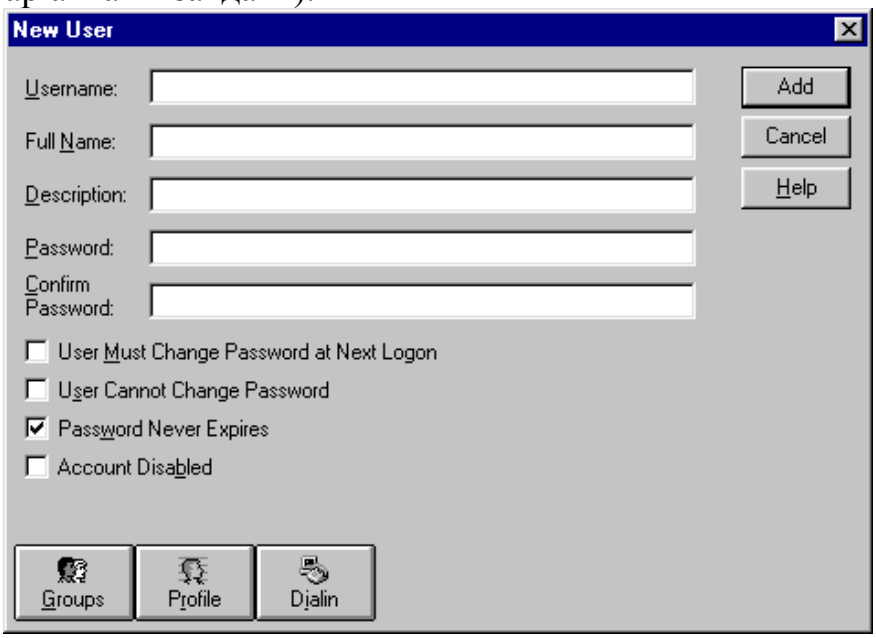
Контрольний приклад

1. Створення і видалення користувачів.

- При створенні облікових записів нових користувачів треба використати утиліту *Диспетчер користувачів.*
- Перейменуємо користувача *Адміністратор* в adm_ws1.
- Створимо для груп *Адміністратори, Користувачі і Досвідчені користувачі* нового користувача з іменем Nik_ws2.

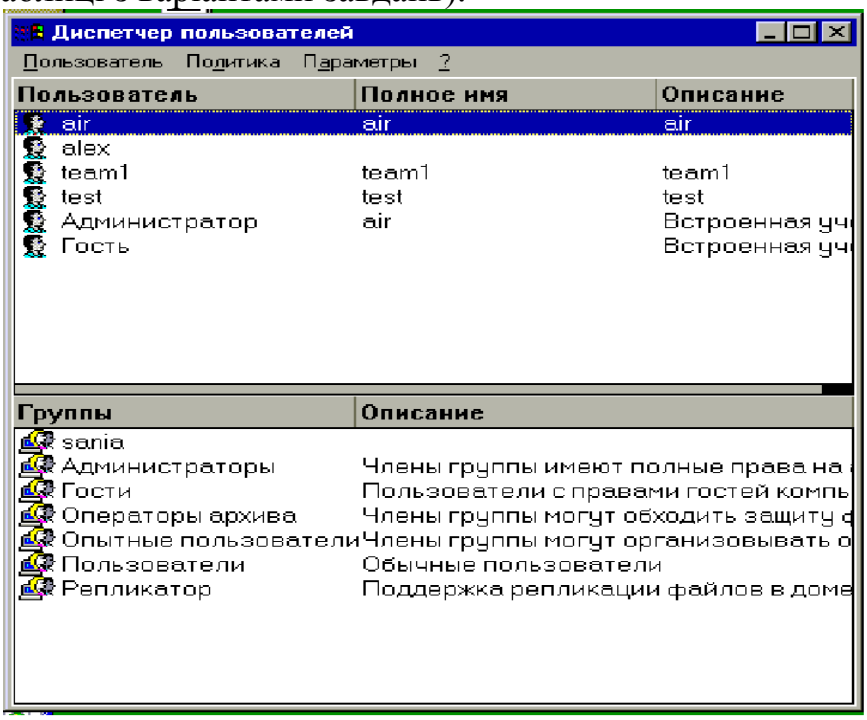
Для нових користувачів домена виконаємо такі дії:

- Задамо інтервал часу, коли користувачеві дозволений вхід в систему — пункт Hours в меню New User,від 11.00 до 12.00 (2-й стовпець таблиці з варіантами завдань).



- Заборонимо вхід з деяких комп'ютерів в межах домена. Для цього в панелі New User треба вибрати Log on To і у вікні діалогу вибрати May Log On To These Workstation. У новому вікні діалогу, що з'явилося, задаємо імена тих комп'ютерів, з яких буде дозволений вхід.

- Встановимо дату закінчення терміну дії облікової інформації даного користувача — пункт Account в меню New User – 42 дні (3-й стовпець таблиці з варіантами завдань).

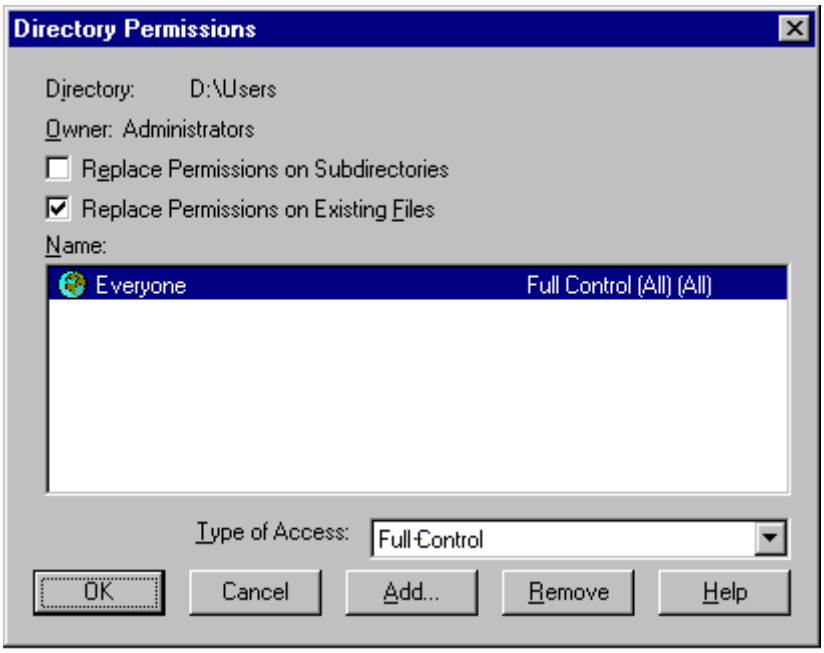


2. Дослідження дозволів (permissions), що надаються користувачам і групам користувачів за доступом до файлів і каталогів.

2.1. Створимо новий каталог “новий каталог” у розділі локального диска, помістимо в нього декілька файлів з існуючих каталогів шляхом копіювання (4-й стовпець таблиці з варіантами завдань).

2.2. Перевіримо, які індивідуальні дозволи стосовно цього каталогу встановлені операційною системою за замовчуванням. Для цього використаємо пункт Properties меню File папки, в яку входить новий каталог. У закладці Security слід натиснути кнопку Permissions.

2.3. Задамо каталогу стандартний дозвіл CHANGE для одного з нових користувачів, наприклад, user_dom1. Для цього в закладці Security натисніть кнопку Permissions, а потім кнопку Add (перед цим корисно видалити дозвіл Full Control для групи Everyone).



Питання для самоперевірки

1. Що таке домен?
2. Які є типи користувачів і групи користувачів в операційній системі Windows NT?
3. Які є типи операцій доступу в Windows NT?
4. Як відрізняються між собою глобальні, локальні та спеціальні групи?
5. Чи рекомендується включати глобальні групи в склад локальних груп ресурсних доменів?

Література

1. Коварт Р. Уотерс Б. Windows NT Server 4. — "Питер", 1999. — 540с.
2. Майнази М и др. Введение в Windows NT Server 4.0. — ЛОРИ, 1998. — 720 с.
3. Ресурсы Microsoft Windows NT Server 4.0. — BHV-Санкт-Петербург, 1999. — 650 с.

Лабораторна робота №9

СИСТЕМА БЕЗПЕКИ WINDOWS NT

Мета роботи: ознайомитися зі структурою системи безпеки Windows NT, навчитися налаштовувати параметри системи безпеки.

Завдання

Зайти у систему Windows NT з правами адміністратора і виконати дії, подані в табл. 1.

Таблиця 1 – Варіанти завдань.

Варіант	Завдання
1	**2**
1	Продивитися журнал. Настроїти журнал так, щоб його очищення відбувалося автоматично. Встановити такі параметри, щоб виконувався аудит таких подій: реєстрація користувача в системі або вихід із неї; створення, зміна і видалення облікових записів користувачів і груп, а також відслідковування процесів.
2	Продивитися журнал. Встановити перегляд тільки помилок і попереджень. Встановити параметри, щоб виконувався аудит таких подій: застосування прав користувачів, зміна політики безпеки, зміни у стратегії аудиту.
3	Обмежити доступ до принтера для звичайних користувачів, встановити аудит усіх подій принтера.
4	Обмежити доступ до файлів \Boot.ini, \Ntdetect.com, \Ntldr, \Autoexec.bat, \Config.sys та директорії \TEMP: для адміністратора — повний контроль, для системи — повний контроль, для інших — тільки читання.

Продовження таблиці 1.

1	2
5	Зайти в систему як Користувач, створити будь-який файл. Зайти в систему як Адміністратор і стати власником попередньо створеного файлу. Обмежити доступ до цього файлу всім користувачам.
6	Створити нову групу користувачів — Студенти. Створити нового користувача, який би належав до групи користувачів Студенти. Встановити обмеження пароля за терміном — 10 днів і за довжиною — не менше 8 символів.
7	Створити нову групу користувачів — Студенти. Створити нового користувача, дозволити йому віддалений доступ до системи та заборонити йому змінювати пароль. Перевірити неможливість зміни паролю.
8	Встановити політику облікових записів. Встановити мінімальну довжину паролю 8 символів, блокування облікових записів після 5 невдалих спроб входу в систему. Встановити політику прав користувачів: дозволити зміну системного часу, володіння файлами та об‘єктами, створення журналів безпеки, керування аудитом лише адміністратору.
9	Створити нову групу користувачів — Студенти. Створити нового користувача. Дозволити йому тільки один раз заходити в систему з даним паролем. Перевірити можливість зміни пароля при кожному вході у систему.
КП	Продивитися журнал. Встановити максимальний розмір архівних журналів—64 Кбайта. Встановити параметри, щоб виконувався аудит таких подій: доступ до файлів, каталогів та принтерів; зміни в привілеях користувачів, стратегії аудита.

Теоретичні відомості

Windows NT проектувалася з урахуванням вимог інформаційної безпеки. Зокрема, Windows NT проектувалася відповідно до вимог Програми Оцінки Довірчих Продуктів (Trusted Product Evaluation Program — TPEP) Національного Центру Комп'ютерної Безпеки (National Computer Security Center — NCSA) США. Ця програма призначена для оцінки рівня безпеки, наданого комп'ютерною системою. Продукту, оціненому за програмою, присвоюється один із рейтингів: A1, B3, B2, B1, C2, C1 або D, за спаданням ступеня безпеки.

Модель безпеки Windows NT була побудована відповідно до вимог класу C2. Найбільш істотними вимогами рівня захисту C2 є:

— власник ресурсу (об'єкта) повинен мати можливість контролю доступу до цього ресурсу;

— операційна система повинна захищати об'єкти таким чином, щоб виключити їх випадкове використання іншими процесами;

— перед тим, як користувачу буде наданий доступ у систему, він повинен ідентифікувати себе шляхом введення унікального імені і пароля. Система повинна мати можливість використання цих унікальних ідентифікаторів для стеження за діями користувачів;

— системний адміністратор повинен мати можливість аудита всіх подій, пов'язаних із безпекою. Доступ до цих даних повинен надаватися тільки авторизованим адміністраторам;

— система повинна захищати себе від зовнішнього впливу, такого як модифікація працюючої підсистеми або системних файлів, що зберігаються на диску.

Базовим у системі безпеки Windows NT є поняття "*облікового запису користувача"* (user account). Для того щоб користувач міг одержати доступ до системи, він, як правило, повинен мати обліковий запис. Обліковий запис містить ім'я користувача і його пароль у зашифрованому вигляді, а також інформацію про групи, до яких належить користувач, інформацію про користувача (наприклад, його дійсне ім'я) і список загальносистемних прав користувача (user rights).

Користувачі в Windows NT для спрощення адміністрування можуть об'єднуватися в групи. Користувач може належати до однієї або декількох груп. Групам також можуть призначатися права доступу до об'єктів.

Кожному *обліковому запису користувача* ставиться у відповідність унікальний ідентифікатор безпеки (Security ID — SID). Цей ідентифікатор є унікальним протягом усього часу життя системи і не може використовуватися повторно, навіть якщо обліковий запис користувача був видалений.

Windows NT дозволяє контролювати локальний доступ до інформації, яка зберігається на диску відповідно до дозволів файлової системи. Windows NT підтримує дві файлові системи: FAT і NTFS. FAT не має засобів захисту на рівні локального користувача, тому вона використовується, тільки якщо комп'ютер має альтернативну операційну систему MS-DOS або Windows 95. NTFS — нова файлова система, підтримувана тільки ОС Windows NT. Основною її відмінністю від FAT є спроможність забезпечувати захист файлів і каталогів відповідно до прав користувачів.

У операційній системі Windows NT керування доступом до файлів і каталогів NTFS покладається не на адміністратора, а на власника ресурсу і контролюється системою безпеки за допомогою *маски доступу* (access mask). Маска доступу включає стандартні, специфічні і родові права доступу. Всі ці права входять у дискредитаційний список контролю доступу (DACL). У списку DACL визначається, яким користувачам і

групам дозволений або заборонений доступ до даного ресурсу. Саме цим списком може управляти власник об'єкта.

Права доступу до об'єкта стануть доступні для перегляду і редагування, якщо вибрати закладку *Безпека* при перегляді властивостей об'єкта.

Створення облікових записів і груп користувачів, керування існуючими записами, а також налагодження політики безпеки, наприклад прав користувачів і політики аудита, проводиться за допомогою *Диспетчера користувачів*. Набір дій, які можна виконати за допомогою диспетчера користувачів, визначається правами поточного користувача. Ці права в основному залежать від того, членом яких груп є даний користувач. Windows NT включає декілька вмонтованих груп, автоматично встановлюваних на комп'ютер.

Найважливішими вмонтованими групами є:

1. *Адміністратори*. Членам групи адміністраторів дозволяється виконувати всі команди диспетчера користувачів.

2. *Досвідчені користувачі*. Членам групи досвідчених користувачів дозволяється створювати облікові записи користувачів і груп, а також змінювати і видаляти ці облікові записи. Крім того, їм дозволяється додавати користувачів у групи досвідчених користувачів, користувачів і гостей, і видаляти користувачів із цих груп.

3. *Користувачі*. Будь-який користувач, що є членом групи користувачів, може створювати групи, змінювати або видаляти створені ним групи, а також включати інших користувачів у ці групи.

Обліковий запис користувача містить набір відомостей про нього, таких як його ім'я і пароль для входу в систему, а також права і дозволи, надані користувачу для роботи із системою і доступу до її ресурсів.

Існує два вмонтовані облікові записи користувачів:

Адміністратор. Обліковий запис користувача з іменем “Адміністратор” визначає відповідального адміністратора робочої станції. Цей користувач повністю керує роботою комп'ютера Windows NT.

Гість. Користувач з іменем *Гість* може створювати файли і видаляти свої файли, а також читати інші файли, якщо системний адміністратор спеціально надав для гостя дозвіл на читання цих файлів.

Вбудований обліковий запис гостя призначений для того, щоб користувач, що працює на комп'ютері дуже рідко або єдиний раз, міг ввійти в систему й одержати до неї обмежений доступ. При установці цей обліковий запис не містить пароля.

Групи використовуються для об'єднання облікових записів користувачів. Включення користувача в групу означає надання йому всіх прав і дозволів, заданих для групи. Ця властивість груп дозволяє швидко надати спільні можливості ряду користувачів.

Диспетчер користувачів дозволяє встановити три типи політики безпеки.

Політика облікових записів визначає режим використання паролів для всіх облікових записів, а також необхідність блокування облікових записів при перевищенні заданого числа невдалих спроб входу в систему за визначений час.

Політика прав користувачів визначає права, що надаються групам і окремим користувачам.

Політика аудита визначає набір подій безпеки, для яких виконується аудит.

Для перегляду і модифікації властивостей облікового запису достатньо відмітити ім'я користувача або групи і на екрані з'явиться діалогове вікно *Властивості користувача***.**

Діалогове вікно містить такі компоненти:

- *основні* — опис користувача, пароль та інші параметри;
- *групи* — обов'язкова інформація про належність користувача до певної групи;
- *профіль користувача* — вибір профілю користувачів;
- *зв'язок* — відомості про віддалений доступ до системи.

Деякі необов'язкові параметри можна і не вводити, але вони є корисними при пошуку того чи іншого користувача за другорядними ознаками.

Властивості груп можна переглянути аналогічно до властивостей користувачів: клацнути мишкою на імені групи, і перед Вами з'явиться діалогове вікно *Властивості локальної групи*.

Аудит — один із засобів захисту Windows NT. З його допомогою можна відслідковувати дії користувачів і ряд системних подій у системі. Фіксуються параметри, які стосуються подій та чиняться користувачами:

- виконана дія;
- ім'я користувача;
- дата і час виконання.

Настройка аудиту дозволяє вибрати типи подій, які підлягають реєстрації, і визначити, які саме параметри будуть реєструватися.

Журнал безпеки (Security Log) містить інформацію про успішні і невдалі спроби виконання дій, що реєструються засобами аудита. Події, які реєструються в цьому журналі, визначаються заданою вами стратегією аудита. Журнал знаходиться у файлі Secevent.evt.

При виборі подій для проведення аудита варто враховувати можливість переповнювання журналу. Для настройки журналу використовуйте діалогове вікно *Настройка журналу подій*.

Для перегляду інформації про помилки і попередження, а також про успішні і невдалі запуски задач використовується програма Перегляд Подій.

За замовчуванням аудит виключений і журнал безпеки не ведеться.

Контрольний приклад

Для того щоб продивитися журнал подій необхідно натиснути кнопку «Пуск» і вибрати в меню Програми команду Адміністрування (Загальне).

Двічі клацнути значок Перегляд подій. На екрані з‘явиться вікно Перегляд подій. В меню Журнал вибрати Журнал безпеки. Вікно Перегляд подій — Журнал безпеки показано на рис. 9.1.

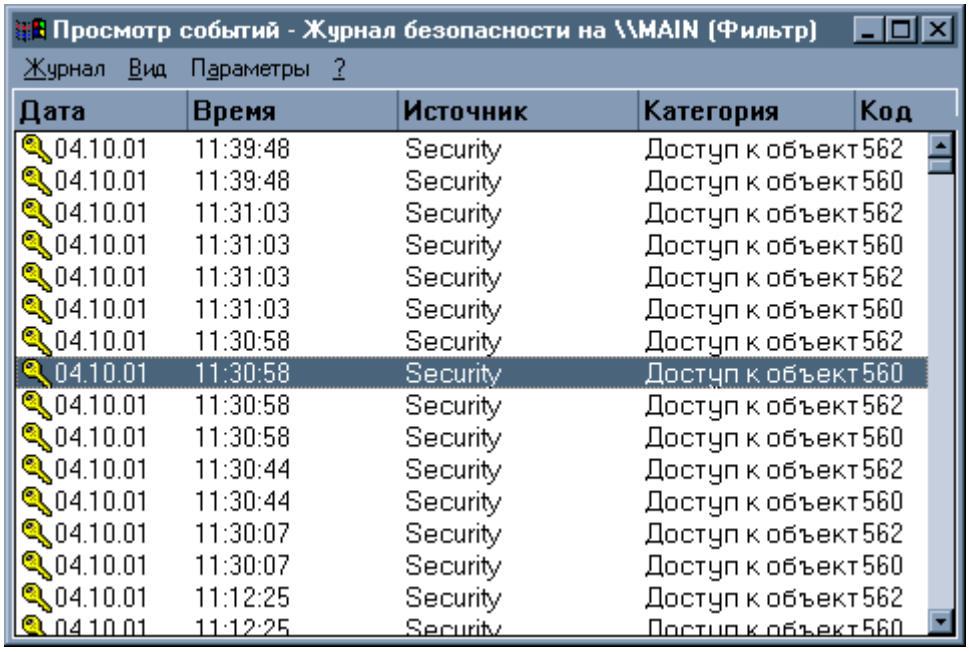


Рисунок 1 - Журнал безпеки

Для того щоб встановити розмір архівних журналів необхідно в меню Журнал вибрати Настройка. З‘явиться вікно (рис. 9.2), де слід встановити максимальний розмір 64 Кбайта.

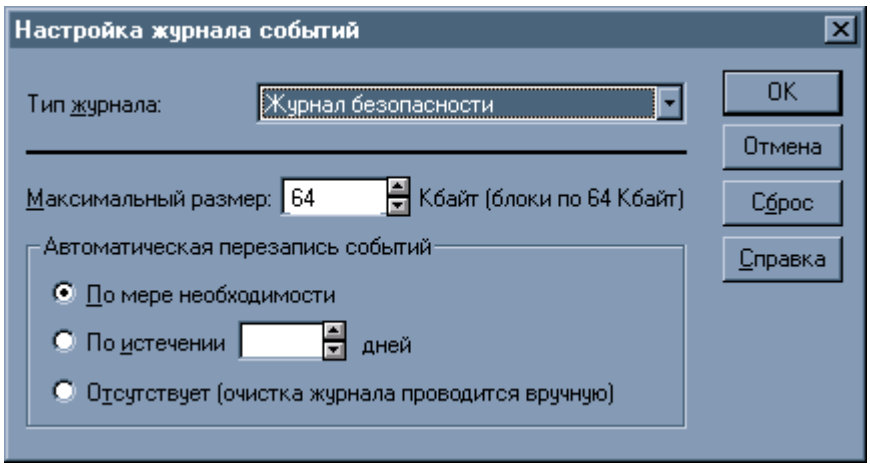


Рисунок 2 - Настроювання журналу подій

Для того, щоб встановити параметри аудиту необхідно натиснути кнопку «Пуск» і вибрати в меню Програми команду Адміністрування (Загальне). Двічі клацнути значок Диспетчер користувачів. В диспетчері в меню Політика необхідно вибрати Політика аудиту. З‘явиться вікно (рис.9.3), де встановлюються необхідні параметри політики аудиту. А саме встановлюємо прапорці навпроти таких подій: доступ до файлів та об‘єктів; керування користувачами та групами; зміни в політиці безпеки. Ставимо прапорці у графах Успіх та Відмова, тоді будуть фіксуватися всі відмічені події, які завершилися як успішно, так і з відмовою.

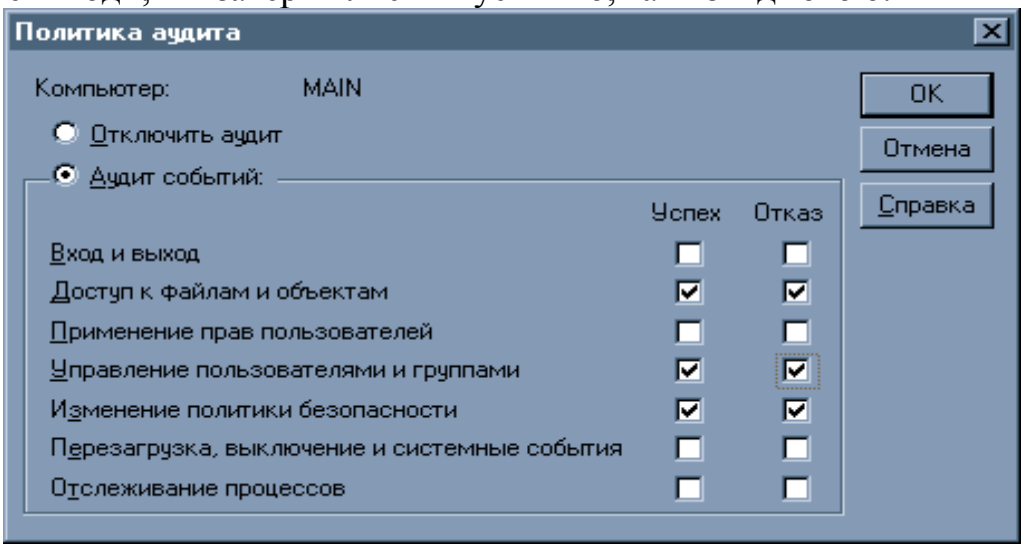


Рисунок 3 - Встановлення параметрів політики аудиту

Питання для самоперевірки

1. Якому рівню відповідає модель безпеки Windows NT? Які найбільш істотні вимоги до цього рівня безпеки?
2. Що таке обліковий запис користувача? Які дані він містить?

3. Чи може ідентифікатор безпеки користувача використовуватися після видалення облікового запису користувача?
4. Які файлові системи підтримує Windows NT?
5. Хто керує доступом до файлів і каталогів NTFS? Як встановити права доступу до об‘єкта?
6. За допомогою якої програми виконується налагодження політики безпеки?
7. Назвіть найважливіші вмонтовані групи користувачів.
8. Які права має користувач з обліковим записом Гість?
9. Які ви знаєте типи політики безпеки?
10. Що таке аудит? Де зберігається інформація про виконання дій у системі?

Література

1. Коварт Р. Уотерс Б. Windows NT Server 4: Учебный курс. — "Питер", 1999. — 540 с.
2. Майнази М и др. Введение в Windows NT Server 4.0. — ЛОРИ, 1998. — 720 с.
3. Ресурсы Microsoft Windows NT Server 4.0. — BHV-Санкт-Петербург, 1999. — 560 с.

Лабораторна робота № 10

АДМІНІСТРУВАННЯ MICROSOFT WINDOWS 2000

Мета роботи**:** ознайомитися з основними принципами адміністрування операційної системи Microsoft Windows 2000 та виконати практичні задачі адміністрування.

Порядок виконання роботи

1. Ознайомитися із можливостями адміністрування операційної системи Microsoft Windows 2000.
2. Зайти до операційної системи Microsoft Windows 2000, зареєструвавшись як користувач, що має права адміністратора.
3. Створити користувача з вказаним у відповідному варіанті іменем.
4. Надати відповідно до свого варіанта права створеному користувачеві.
5. Зайти до системи, реєструючись під іменем створеного користувача.
6. Створити ресурс-папку та надати доступ іншим користувачам відповідно до вашого варіанта.

7. Перевірити наслідки встановлених параметрів.

8. Після закінчення лабораторної роботи привести систему в початковий стан.

Завдання до лабораторної роботи

Таблиця 1 – Варіанти завдань до лабораторної роботи

Варіант	Користувач	Права	Папка-ресурс	Доступ
1	Ваше ім’я	Користувач	Папка	Усім(повний)
2	Назва групи	Досвідчений користувач	Диск	Одному користувачу (читання)
3	Користувач	Користувач	Папка	-
4	Студент	Адміністратор	Папка	Усім (читання)
5	Користувач	Користувач		-
6	Комп’ютер	Досвідчений користувач	Диск	Одному користувачу (повний)
7	Адміністратор	Адміністратор	Папка	Усім (повний)
8	Диспетчер	Досвідчений користувач	Диск	Одному користувачу (повний)
9	Керівник	Досвідчений користувач	Диск	Одному користувачу (читання)
10	Тестувальник	Користувач	Папка	-
КП	Користувач	Адміністратор	Диск	Усім (повний)

Теоретичні відомості

Створення облікових записів і груп займає важливе місце у питанні забезпечення безпеки Windows 2000, оскільки, адміністратор, призначаючи їм права доступу, отримує можливість обмежити користувачів у доступі до конфіденційної інформації комп’ютерної мережі, дозволити або

заборонити їм виконувати в мережі певні дії, наприклад архівацію даних чи завершення роботи комп'ютера. Звичайно, право доступу асоціюється з об'єктом – файлом чи папкою. Це право визначає можливість даного користувача отримати доступ до об'єкта.

В операційній системі Microsoft Windows 2000 існує інструмент "Локальные пользователи и группы" (в англомовному варіанті операційної системи – "Local Users and Groups"), за допомогою якого виконується управління локальними обліковими записами користувачів та груп – як на локальному, так і на віддаленому комп'ютерах. З таким інструментом є можливість працювати як на робочих станціях (варіант операційної системи для загального користування), так і на автономних серверах (варіант системи для організації серверних станцій).

Запускати інструмент "Локальные пользователи и группы" може будь який користувач. Виконувати адміністрування облікових записів можуть тільки адміністратори і члени групи "Досвідчені користувачі" (в англомовному варіанті операційної системи – "Power Users").

Вікно програми "Локальные пользователи и группы" виглядає аналогічно показаному на рис. 1.

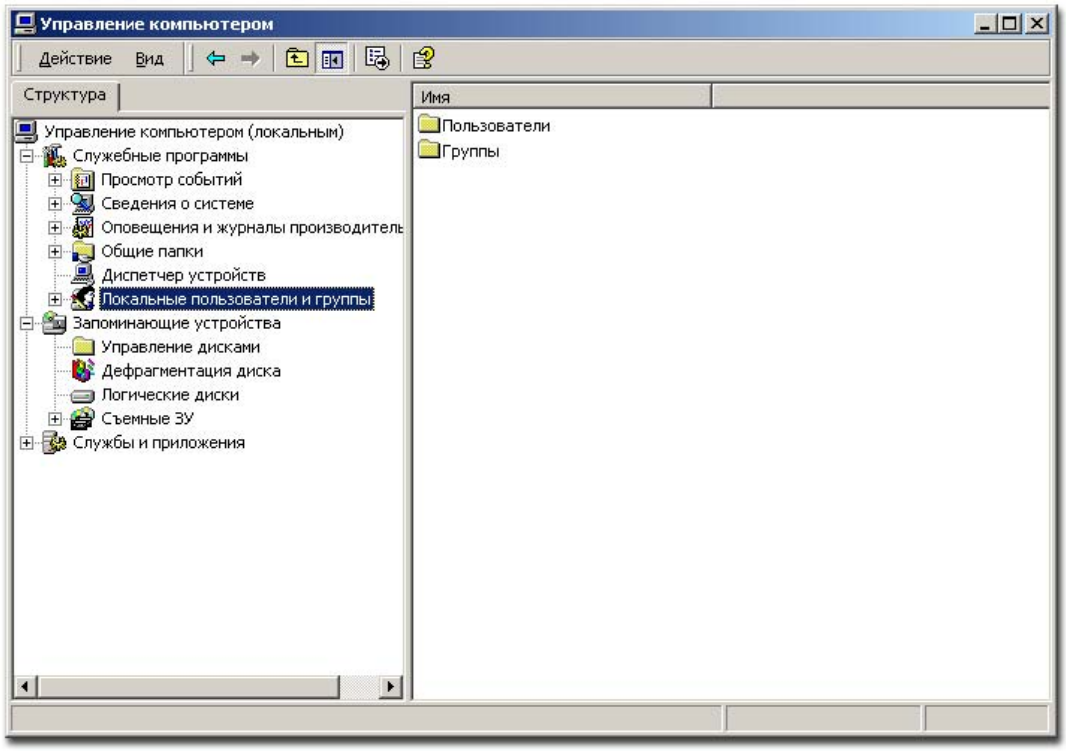


Рисунок 1 - Вікно інструменту "Локальные пользователи и группы"

У вікні можна встановлювати параметри як користувачів, так і груп. Спочатку розглянемо папку "Пользователи" ("Users"), а потім "Групы" ("Groups").

Папка "Пользователи"

Відразу після інсталяції операційної системи Windows 2000, папка “Пользователи” містить два вбудованих облікових записи – “Администратор” та “Гость”; вони створюються автоматично при встановленні Windows 2000.

- *“Администратор”* – цей обліковий запис використовують при встановленні та настроюванні робочої станції чи сервера, які є членами домену. Цей запис не може бути видалений, блокований чи знищений з групи “Администраторы”, йому можна тільки змінити назву.
- *“Гость”* – цей обліковий запис використовується для реєстрації в комп’ютері без використання спеціально створеного облікового запису. Обліковий запис “Гость” не потребує введення паролю і за замовчуванням є блокованим. (Зазвичай, користувач, обліковий запис якого блокований, але не видалений, при реєстрації отримує попередження і входити у систему не може.) Цей запис є членом групи “Гости” (в англ. варіанті – “Guests”). Для цього запису можна надати права доступу для ресурсів системи точно так, як і для будь-якого іншого облікового запису.

Папка “Группы”

Після встановлення операційної системи Microsoft Windows 2000 (робочої станції чи сервера, що є членом домену) папка “Группы” містить шість вмонтованих груп. Вони створюються автоматично при встановленні системи. Нижче описані властивості всіх внутрішніх груп.

- **“Администраторы”** – члени цієї групи мають повний доступ до всіх ресурсів системи. Це єдина група, яка автоматично надає своїм членам увесь набір прав.
- **“Операторы архивации”** – члени цієї групи мають право виконувати архівацію та відновлювати файли у системі, незалежно від того, якими правами ці файли захищені. Крім цього, оператори архівації мають можливість входити до системи і завершувати її роботу, але не мають права змінювати параметри безпеки.
- **“Гости”** – ця група дозволяє виконувати реєстрацію користувача за допомогою облікового запису “Гости” та отримати обмежені права на доступ до мережних ресурсів системи. Члени цієї групи можуть завершувати роботу системи.
- **“Опытные пользователи”** – члени цієї групи мають право створювати облікові записи користувачів, але вони мають права модифікувати параметри безпеки тільки для створених ними облікових записів. Крім цього, вони можуть створювати локальні групи і модифікувати склад членів цих груп. Члени групи “Опытные пользователи” не мають право модифікувати членство в групах “Администраторы” та “Операторы архивации”. Вони не можуть бути власниками файлів, виконувати архівацію чи відновлення каталогів,

завантажувати та припиняти роботу драйверів та модифікувати параметри безпеки та журнал подій.

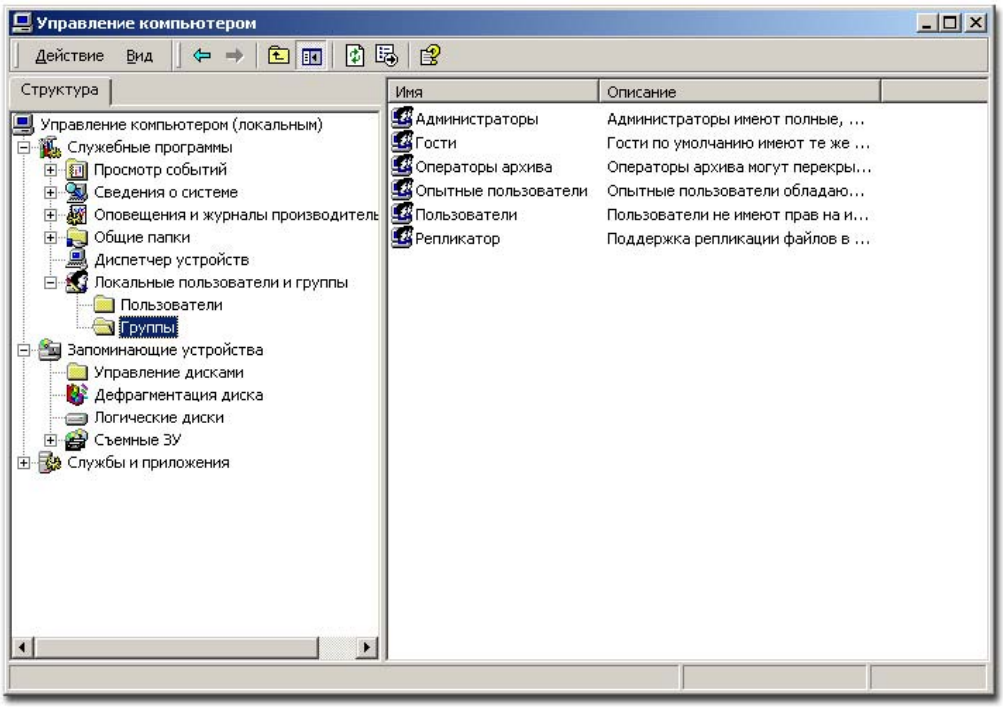


Рисунок 2 - Вікно папки “Группы”

- **“Репликатор”** – членами групи “Репликатор” повинні бути тільки облікові записи, за допомогою яких можна зареєструватися у службі реплікації контролера домену. Ними не варто робити робочі облікові записи.
- **“Пользователи” –** члени цієї групи можуть виконувати більшість функцій користувачів, наприклад, запускати програми, користуватися локальним або мережним принтером, завершати роботу системи чи блокувати робочу станцію. Вони також можуть створювати локальні групи та регулювати склад їх членів. Вони не можуть отримати доступ до загального каталогу чи створити локальний принтер.

Управління обліковими записами

Розглянемо у якості прикладу використання інструментів “Локальные пользователи” та “Группы” для роботи з обліковими записами, розглянемо процедуру створення облікового запису користувача.

Створення облікового запису

Для створення облікового запису:

1. В інструменті “Локальные пользователи и группы” встановіть покажчик миші на папку “Пользователи” та натисніть праву кнопку миші. У контекстному меню, яке з’явилося, виберіть команду “Создать пользователя” (“Create User”).

2. З’явиться вікно “Новый пользователь”. У поле “Пользователь” введіть ім’я користувача, якого ви бажаєте створити. В поле “Полное имя” введіть повне ім’я користувача, якого ви створюєте. В поле “Описание” введіть опис користувача чи його облікового запису. В поле “Пароль” введіть пароль користувача та в полі “Подтверждение” потрібно підтвердити його правильність повторним введенням. Довжина пароля не може перевищувати 14 символів.

3. Встановіть чи зніміть прапорці “Потребовать смену пароля при следующем входе в систему”, “Запретить смену пароля пользователем” (Користувач не може змінювати пароль), “Срок действия пароля неограничен”(Час дії цього пароля не обмежується) та “Отключить учетную запись” (Обліковий запис блоковано).

4. Для того щоб створити ще одного користувача, натисніть кнопку “Создать” та повторіть кроки з 1 по 3. Для завершення роботи натисніть кнопку “Создать” та потім “Закрыть”.

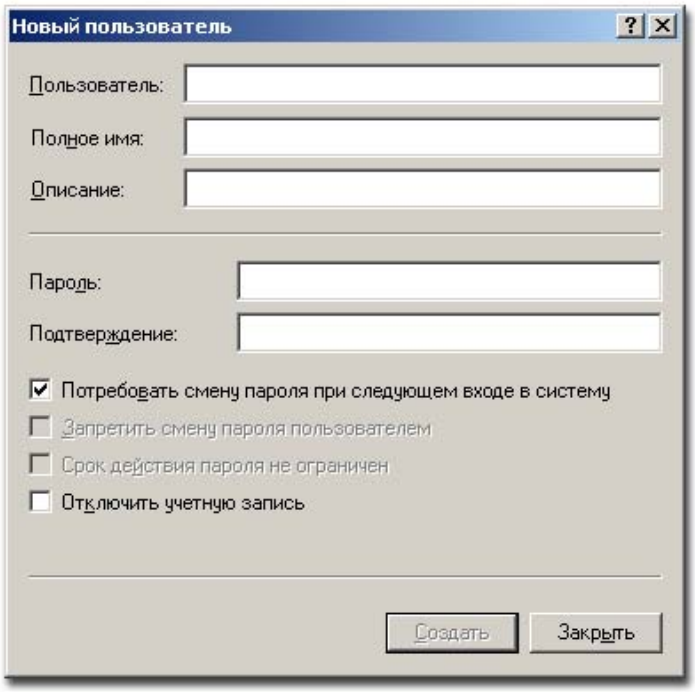


Рисунок 3 - Вікно реєстрації нового користувача

Ім’я користувача повинно бути унікальним у межах одного комп’ютера. Воно може містити до 20 символів верхнього та нижнього регістру. Нижче наведені символи, застосування яких у імені користувача є неможливим: “\”, ”/”, ”[”, ”]”, ”<”, ”>”, ”+”, ”,”, ”:”, ”;”, ”?”, ”*”.

Ім’я користувача не може складатися повністю з крапок чи пропусків.

Зміна та видалення облікових записів

Змінювати, перейменовувати та видаляти облікові записи можна за допомогою контекстного меню, що викликається натисканням на праву

клавішу миші на імені користувача або меню “Действие” на панелі меню інструмента “Локальные пользователи и группы” (при цьому у правому вікні повинен бути вибраним обліковий запис, який змінюється чи видаляється.

Оскільки обліковий запис, що був перейменований, зберігає ідентифікатор безпеки (“идентификатор безопасности”), він зберігає усі свої властивості, наприклад, опис, повне ім’я пароля, членство у групі тощо.

Управління локальними групами

Створення локальної групи

Для створення локальної групи:

1. У вікні інструменту “Локальные пользователи и группы” встановіть курсор миші на папку “Группы” і натисніть праву клавішу миші. У контекстному меню, що з’явилося, виберіть команду “Создать группу”.

2. У поле “Имя” введіть ім’я нової групи.

3. У поле “Описание” введіть опис нової групи.

Ім’я локальної групи повинно бути унікальним у межах одного комп’ютера. Воно може містити до 256 символів верхнього та нижнього регістру. В імені групи заборонено застосування символа зворотного слеша (\).

Зміна членства локальної групи

Щоб добавити чи видалити обліковий запис користувача з групи:

1. У вікні інструменту “Локальные пользователи и группы” виділіть папку “Группы”.

2. У правому підвікні встановіть курсор на групу, що модифікується, і натисніть праву кнопку миші. У контекстному меню, яке з’явилося, виберіть команду “Свойства”.

3. Для того, щоб додати нові облікові записи до групи, натисніть кнопку “Добавить”. Надалі керуйтеся вказівками вікна діалогу “Выбор пользователей или групп”.

4. Для того, щоб видалити з групи певних користувачів, у полі “Члены” вікна властивостей групи виберіть одну чи декілька облікових записів і натисніть кнопку “Удалить”.

До локальної групи можна добавляти як локальних користувачів, створених на комп’ютері, так і користувачів та глобальні групи, які створені в домені, до якого належить цей комп’ютер.

Примітка. Вмонтовані групи не можуть бути видаленими. Видалені групи не можуть бути відновлені. Видалення групи не відображується на користувачах, які до неї входять.

Надання доступу до ресурсів системи

В операційній системі існує можливість надавати права користування тим чи іншим ресурсом. Користувач з відповідними правами

може задавати права доступу до конкретного ресурсу для вказаних користувачів. Наприклад, якщо ви не бажаєте, щоб усі мали доступ до Вашої приватної папки, Ви можете виділити доступ тільки тим користувачам, яким бажаєте. Причому користувачі можуть бути зареєстровані як на локальному комп'ютері, так і в домені, до якого входить комп'ютер. Для встановлення прав доступу потрібно навести курсор миші на бажаний об'єкт та натиснути праву клавішу. У контекстному меню, що з'явиться, потрібно вибрати команду "Доступ". У вікні, що з'явиться, потрібно натиснути на клавішу "Разрешить" і у діалоговому вікні вибрати тих користувачів, яким Ви бажаєте надати доступ до Вашого ресурсу. Доступ може бути наданий у декількох варіантах:

а) на читання;

б) на запис;

в) повний.

Відповідно до наданого доступу користувач має право тільки читати, тільки записувати та читати, записувати і видаляти.

Контрольний приклад

1. Зайдемо у систему як користувач, що має права адміністратора. Для цього при вході в систему потрібно у вікні реєстрації ввести ім'я та пароль користувача з відповідними правами.
2. Для реєстрації нового користувача потрібно визвати інструмент "Локальные пользователи и группы". Для цього потрібно зайти у меню "Пуск", потім вибрати пункт "Настройки". У вікні, що з'явилося, потрібно вибрати ярлик "Администрирование системы". У вікні "Администрирование системы" потрібно вибрати інструмент "Локальные пользователи и группы".
3. Після цього потрібно у правому підвікні вибрати папку "Пользователи".
4. У правому підвікні потрібно натиснути праву клавішу миші та у контекстному меню, яке з'явиться, вибрати пункт "Создать пользователя".
5. У поле "Пользователь" потрібно ввести ім'я – "User", повне ім'я нового користувача, наприклад, "Іванов Олександр Григорович" введіть у поле "Полное имя" та опис – будь-яка інформація, яку ви бажаєте записати про нового користувача, – у поле "Описание".
6. Далі у вікні потрібно ввести пароль користувача, який він буде використовувати при вході у систему. Пароль вводиться два рази: у поле "Пароль" та у поле "Подтверждение" для впевненості, що пароль набраний вірно.

7. Знімемо позначку на умові “Потребовать смену пароля при следующем входе в систему”.

8. Натиснемо клавішу “Создать”. У вікні повинен з’явитися новий користувач з іменем “User”. Для надання йому прав адміністратора, наведемо на нього курсор миші, два рази натиснемо кнопку миші та у вікні, яке з’явиться, з розділу “Группы” виберемо групу “Адмистраторы” і додамо її до прав користувача.

9. Вийдемо з діалогового вікна, натиснувши ОК, та за допомогою меню “Пуск” та пункту “Завершить сеанс” вийдемо у реєстраційну частину системи.

10. У вікні, що з’явилося, введемо ім’я нового користувача – “User” та у наступному полі введемо пароль, який ми призначили новому користувачеві.

11. Після входу у систему ми маємо права адміністратора. У провіднику, вибравши правою кнопкою миші диск, виберемо у контекстному меню пункт „Доступ”.

12. У діалоговому вікні, що з’явиться, надамо доступ для усіх користувачів, вказавши доступ з повними правами.

Питання для самоперевірки

1. Що таке Microsoft Windows 2000?
2. Для чого використовують Windows 2000?
3. Що розуміють під процесом адміністрування?
4. Які основні аспекти адміністрування операційної системи Windows 2000?
5. З якою метою створюються локальні групи?
6. Які права може мати користувач?
7. Що потрібно зробити для реєстрації в операційній системі Windows 2000?

Література

1. Шумов В.С. Администрирование Windows 2000. – bHV, 2000. – 560с.

Лабораторна робота № 11
ЛОГУВАННЯ В ОПЕРАЦІЙНІЙ СИСТЕМІ NOVELL NETWARE

Мета роботи: придбати навички логування в операційній системі Novell Netware.

Порядок виконання роботи

1. Виконайте логічний вхід у систему Novell Netware, використовуючи контекст, відповідно до свого варіанта

Варіант	Контекст
1	user1.0128.vstu
2	user2.0128.vstu
3	user3.0128.vstu
4	user4.0128.vstu
5	user5.0128.vstu
6	user6.0128.vstu
7	user7.0128.vstu
8	user8.0128.vstu
9	user9.0128.vstu
10	user10.0128.vstu

2. Перевірте свій поточний контекст. Використайте параметри команди cx, відповідно до свого варіанта

Варіант	Параметр
1	/cont
2	/T
3	/R
4	/A
5	/C
6	/R

7	/cont
8	/A
9	/T
10	/R

Поясніть результат роботи команди.

3. Перейдіть зразу на два рівня вище і перевірте контекст.

4. Змініть контекст відповідно до варіанта

Варіант	**Контекст**
1	.local.0128.vstu
2	. 0128.vstu
3	.users.user3.0128.vstu
4	.nauka.documents.0128.vstu
5	.programs.0128.vstu
6	.system.0128.vstu
7	.txt.documents.0128.vstu
8	.local.0128.vstu
9	.programs.0128.vstu
10	. 0128.vstu

5. Перевірте інформацію про поточний сеанс. Прокоментуйте інформацію, яку ви бачите на екрані.

6. Змініть свій пароль.

7. Виконайте логічний вихід з системи.

8. Виконайте логічний вхід, використовуючи новий пароль.

Теоретичні відомості

ОС NetWare версій 3.x та 4.x призначена для забезпечення доступу до загальних ресурсів мережі зі сторони декількох користувачів. В якості таких ресурсів виступають файли даних, принтери, модеми, модулі і т.п.

NetWare надає користувачам такі можливості:

- підтримує колективне використання файлів;
- забезпечує доступ до мережних принтерів;
- пропонує засоби для роботи з електронною поштою;
- підтримує роботу СУБД різних типів;
- забезпечує доступ до файлового сервера зі сторони робочих станцій, які функціонують під керуванням різних операційних систем;
- пропонує засоби, що дозволяють об’єднувати віддалені сегменти мережі;
- забезпечує "прозорість" доступу локальних і віддалених користувачів до ресурсів мережі;
- пропонує засоби для надійного збереження даних;
- забезпечує захист ресурсів мережі від несанкціонованого доступу;
- підтримує динамічно розширювані багатосегментні томи на декількох дисках файлового сервера;
- надає засоби керування ресурсами корпоративних мереж: єдиний каталог мережних ресурсів NDS в NetWare 4.1;
- забезпечує передачу і обробку даних з використанням різних протоколів: SPX/IPX, TCP/IP, NetBIOS, AppleTalk;
- підтримує роботу суперсерверів у симетричному режимі функціонування (ОС NetWare 4.1 SMP).

Насамперед NetWare дозволяє користувачам звертатися до загальних файлів, що зберігаються на файловому сервері. Це, з одного боку, дозволяє не дублювати загальні дані на робочих станціях, а, з іншого – забезпечує взаємодію користувачів через файловий сервер.

Якщо всім користувачам мережі необхідно виводити дані на друк, а кількість принтерів менша кількості робочих станцій, то NetWare дозволяє зробити друкувальні пристрої доступними всім клієнтам мережі.

Важливо відмітити, що в NetWare забезпечена можливість доступу до файлового сервера зі сторони робочих станцій, які функціонують під керуванням різних операційних систем: MS-DOS, OS/2, UNIX, Macintosh, Windows NT Workstation, Windows 9x і т.д.

У NetWare використовуються утиліти таких типів:

- утиліти командної стрічки;
- утиліти-меню;
- утиліти, що виконуються під керуванням Windows.

Для виконання утиліти командної стрічки необхідно ввести ім’я програми і параметри. Результати виконання команди відображаються на екрані робочої станції.

При роботі з утилітою-меню використовується текстовий діалоговий інтерфейс. Щоб модифікувати будь-яке поле діалогового вікна утиліти-меню, необхідно підсвітити це поле і натиснути клавішу Enter. Для того, щоб змінити значення поля на альтернативне, використовують клавіші "Left" та "Right". Якщо це поле введення даних, то з’являється курсор, і

можна відредагувати значення поля. Після повторного натискування клавіші Enter (або клавіш "Up", "Down") підсвічується наступне поле діалогового вікна. Якщо поле пов’язане зі списком, то на екрані з’являється цей список. Використовуючи клавіші Del і Ins, можна відредагувати цей список значень.

При роботі з утилітами, які виконуються під керуванням Windows, використовується графічний інтерфейс, що надається цією оболонкою.

Перед реєстрацією в системі користувач повинен запустити утиліту робочої станції, яка встановлює зв’язок з мережею. Команди для такого підключення при встановленні програмного забезпечення робочої станції звичайно записуються в файл AUTOEXEC.BAT. Якщо це зроблено, то користувач може почати реєстрацію. Можна побачити повідомлення про те, що система підключена до сервера. Необхідно пам’ятати, що підключення не означає реєстрацію, а означає лише те, що з сервером встановлений зв’язок.

У відповідь на підказку DOS потрібно переключитися на перший мережний диск. Звичайно він має буквену мітку, що слідує за буквою останнього локального диска. Наприклад, якщо система має локальні жорсткі диски, диски, що моделюються в оперативній пам’яті (псевдодиски), або диски CD-ROM, а останній із цих дисків має буквену мітку H, то перший мережний диск буде мати буквену мітку I. Потрібно перейти на цей диск. Після цього виконати команду реєстрації LOGIN.

Місце знаходження об’єкта в дереві NDS називається його контекстом. Перед тим, як зареєструватися в мережі, користувач повинен знаходитися у вірному контексті. При спробі зареєструватися в неправильному контексті користувач отримає повідомлення про помилку у вигляді "user does not exist in this context" ("у такому контексті користувача не існує"). При цьому для перегляду поточного контексту користувач може дати команду CX (Change Context), а за допомогою іншої команди CX переключитися в потрібний контекст і зареєструватися.

Команда **CX** є ключовою для перегляду поточного контексту дерева і його зміни.

За допомогою команди **CX** можна перейти на рівень вище.

Команда LOGIN буде виконувати пошук вниз по дереву каталогу, але не вверх, тому проблеми з реєстрацією зазвичай усуваються переміщенням вверх по дереву. Якщо потрібно перемістити вверх більш ніж на один рівень, то потрібно дати команду: „CX..”.

Команда **CX** використовується для переміщення по дереву каталогу аналогічно тому, як команда DOS CD застосовується для переміщення в структурі каталогів файлової системи.

По-перше, кожний користувач має об’єкт користувача, який знаходиться в конкретному контексті дерева каталогу. Уявимо, наприклад, що користувач Serg відноситься до відділу Administration, який належить

до Вінницького підрозділу AST Soft. Таким чином, контекстом об'єкта Serg буде:

CN=Serg.OU=Administation.OU=Vinnitsa.O=AST_Soft

Це ім'я формує унікальний маршрут, побудований із складових імен. Складеними іменами являються: назва організації (O), одиниці, яка включає організацію (OU), і загальне ім'я об'єкта (CN). Кожне складене ім'я відділяється за допомогою крапки і явним чином визначається за допомогою буквеного позначення (CN, OU або O). Ці скорочення можна опустити, наприклад:

Serg.Administation.Moscow.AST_Soft

Такий запис називається безтиповим контекстом імені. Однак, у таких випадках операційна система не завжди розпізнає контекст.

Маршрут, сформований послідовністю складених імен, точно визначає розміщення об'єкта відносно кореня дерева каталогу, аналогічно тому, як маршрут чи ім'я файла DOS ідентифікує розміщення файла. Однак, на відміну від маршрутів імен файлів імена NetWare Directory Services мають порядок, зворотний порядку імен файлів. Цей повний маршрут називається повним іменем об'єкта.

Як уже згадувалось, команду CX можна використовувати для перегляду дерева NetWare Directory Services і для визначення свого поточного контексту або контексту інших об'єктів. Наприклад, можна зареєструватися на сервері або томі в дереві каталогу. Команда CX допомагає визначити ім'я цього сервера або тому і його контекст. Дана команда має таку форму:

CX новий_контекст параметри

де "новий_контекст" заміняється тим контекстом, у який потрібно переключитися, а "параметр" – це один із параметрів

Параметр	**Опис**
/CONT	Перераховує контейнери (які включають об'єкти) поточного контексту або заданого контексту
/R	Робить поточний контекст коренем дерева каталогу
/T	Перераховує всі контейнери нижче поточної комірки
/A	Щоб вивести список усіх об'єктів у поточному контексті або нижче, задайте цей параметр з параметром /CONT або /T
/C	Безперервно прокручує виведення на екран
/?	Виводить довідкову інформацію

Щоб побачити свій поточний контекст, потрібно просто набрати команду CX. Для виводу інформації про дерево каталогу і об'єкти в ньому потрібно використовувати параметри. Команда CX /CONT приведе до

виводу на екран списку об’єктів-контейнерів у поточному контексті, наприклад:

```
O=AST_Soft
 ¦
 ¦
 +-OU=Moscow
 ¦
 ¦
 ¦
 ¦
 +-OU=Novgorod
```

Якщо вказати параметр /T, то отримаємо список об’єктів-контейнерів нижче поточного контексту, наприклад:

```
[Root]
 ¦
 ¦
 ¦
 ¦
 +-O=AST_Soft
    ¦
    ¦
    +-OU=Moscow
    ¦   ¦
    ¦   ¦
    ¦   +-OU=Administration
    ¦   ¦   ¦
    ¦   ¦   ¦
    ¦   ¦   L-OU=Servers
    ¦   ¦
    ¦   ¦
    ¦   +-OU=Marketing
    ¦   ¦
    ¦   ¦
    ¦   +-OU=Sales_Moscow
    ¦   ¦
    ¦   ¦
    ¦   +-OU=Service
    ¦
    ¦
    +-OU=Novgorod
        ¦
        ¦
        +-OU=Sales
```

Припустимо, поточним контекстом є [Root], як показано у попередньому прикладі. Щоб зробити своїм контекстом організаційний підрозділ SERVICE, можна дати таку команду:

CX SERVICE.ADMINISTRATION.VINNITSA.AST_SOFT

З цього рівня ви можете переміститися по дереву і зробити поточним контекстом контейнер AST_Soft. Для цього треба дати команду:

CX ...

Кожна крапка надає рівень у дереві NDS. Щоб переключитися з одного повного контексту на другий, необхідно набрати перед контекстом крапку. Наприклад, щоб переключитися з LEVEL1.PATH1 на LEVEL2.PATH2, можна дати команду:

CX. .LEVEL1.PATH2

Реєстрація зводиться до набору в командний рядок команди LOGIN. Якщо користувач не вводить у командний рядок контекст, то виводиться повідомлення з запитом. Потім запитується пароль.
Команда LOGIN має таку форму:

LOGIN сервер/контекст/параметри

де "сервер" – це ім’я сервера, на якому потрібно зареєструватися, "контекст" – ім’я користувача і потрібний для нього контекст, а "параметр" – це один із перерахованих нижче параметрів:

Параметр	Опис
/NoScript або /NS	Запобігає виконанню сценарію реєстрації і не відміняє реєстрацію на тих серверах, де користувач вже зареєстрований
/Script або /S <маршрут>	Замінить "маршрут" іменем файла спеціального сценарію реєстрації, який повинен виконатися при реєстрації
/SWAP	Дозволяє виконати із сценарію реєстрації зовнішні команди
/PR= ім.’я	Замінить "ім’я" на ім’я профільного об’єкта, який повинен виконуватися із сценарію реєстрації. Профільний об’єкт містить сценарій реєстрації, який присвоюється тільки конкретним користувачам, що можуть не належати ні до одного із контейнерів у дереві NDS
/TR	Цей параметр використовується для реєстрації в іншому дереві каталогу. Більшість організацій мають тільки одне дерево каталогу, яке охоплює всю фірму, тому використовується даний параметр рідко

Якщо користувач не може зареєструватися на сервері, то це може вказувати на те, що має місце одне із обмежень реєстрації, або на одну із таких проблем:

- заданий невірний контекст;
- якщо користувач вже зареєструвався на іншій робочій станції і не може зареєструватися на кількох робочих станціях одночасно;
- встановлено обмеження реєстрації, і ви можете зареєструватися тільки протягом певного часу дня;
- користувач можете зареєструватися тільки на конкретній робочій станції.

Якщо користувач зареєструвався, то в будь-який час за допомогою команди SETPASS можна змінити свій пароль. Змінити паролі для інших користувачів можуть супервізори. Команда має вигляд:

SETPASS ім’я_користувача /SE=имя_сервера

де "ім’я_користувача" – це ім’я того користувача, пароль реєстрації якого потрібно змінити, а "ім’я_сервера" – це ім’я сервера. Щоб змінити власний пароль, просто наберіть команду SETPASS. Виводяться повідомлення, які запитують введення старого і нового пароля. Потім для перевірки пароля потрібно ввести його знову.

Команда WHOAMI дає корисну інформацію про поточний сеанс. Для виводу на екран вмісту поточного дерева і контексту імені користувача потрібно набрати:

WHOAMI

Для виводу інформації про сервери потрібно дати команду:

WHOAMI /B

Щоб безперервно прокручувати довгий текст на екрані, потрібно задати параметр /C, а щоб отримати довідку за цією командою, наберіть параметр /?.

Дуже важливо, щоб користувачі правильно завершали сеанс, коли вони покидають робочі станції. Якщо підключена до мережі машина буде залишена без нагляду, зловмисник може отримати доступ до файлів або пошкодити ті області, до яких він звичайно звертатися не може.

Щоб відмінити реєстрацію в мережі, необхідно просто набрати команду LOGOUT. При цьому відміниться реєстрація на всіх серверах мережі. Якщо користувач хоче відмінити реєстрацію на конкретних серверах, то потрібно вказати після команди LOGOUT ім’я сервера.

Контрольний приклад

1. Вводимо login user10.0128.vstu

На запит паролю вводимо відповідно пароль – user10.

2. Для перевірки поточного контексту вводимо cx і натискуємо ENTER. Для варіанта 10 вводимо „cx /r”. Після виконання цієї команди бачимо, що поточним контекстом став корінь дерева каталогів.

3. Переходимо зразу на два рівня вище. Для цього вводимо команду „cx ..”

4. Перевіряємо поточний контекст командою „cx”.

5. Змінюємо контекст. Вводимо „cx .0128.vstu”.

6. Змінюємо свій пароль. Вводимо „setpass”

7. Відповідно до запиту вводимо старий і новий пароль.

8. Виконуємо логічний вихід з системи. Вводимо „logout”

9. Виконуємо логічний вхід у систему, використовуючи новий пароль. Тобто вводимо „LOGIN user10.0128.vstu” і на запит пароля вводимо новий пароль.

Питання для самоперевірки

1. Які можливості надає користувачам Novell Netware?
2. Що розуміють під контекстом об’єкта?
3. Як виконується логічний вхід у систему Novell Netware?
4. Яке призначення команди „cx” ?
5. Які параметри має команда „cx” і з якою метою вони використовуються?
6. Як змінити свій пароль?
7. Як продивитися інформацію про поточний сеанс?
8. Як продивитися інформацію про сервери?

Література

1. Lawrence B. Using NetWare 4.1. Special Edition, QUE, 1999. – 640 p.
2. Гаскин. Netware 5. Полное руководство в 2-х томах. – М.: Век, 2002.
3. Григорьев. NetWare 5. Настольная книга администратора. – М.: ДМК, 2000. – 530 с.

Лабораторна робота № 12
ФАЙЛОВА СИСТЕМА NOVELL NETWARE

Мета роботи: отримати відомості про організацію та функціонування файлової системи Novell NetWare та навчитися використовувати її можливості в роботі.

Завдання на лабораторну роботу

1. Підключитися до сервера з іменем адміністратора та створити два каталогу у кореневому каталозі сервера відповідно до варіанта (див. таб. 1).
2. Надати права доступу до створених каталогів своєму користувачеві.
3. Увійти в систему зі своїм іменем користувача та паролем.
4. Підключити як окремі диски обидва створені каталоги.
5. Створити дерево каталогів на цьому диску згідно з варіантом, та створити у вказаному каталозі будь-який текстовий файл.
6. Скопіювати створений файл на інший раніше підключений диск.
7. Вивести зміст копії на консоль.

8. Відключити диски, створені у пункті 4.

9. Підключитися до сервера з іменем адміністратора та видалити каталоги, що створені при виконанні пункту 1.

Таблиця 1 - Варіанти завдань

№	Ката-лог1	Ката-лог2	Ім'я файлу	Дерево каталогів
1	WORK	BACKUP	REPORT.TXT	__SALES__ |__2000 |__2001 |_AUG |_SEP
2	JAVA	DEMO	LOG.TXT	__JAVA2__ |__SWING |__JDBC |_ORACLE |_MSSQL
3	WORK	MAIN	ERRORS.TXT	__REPORTER__ |__WINUSR |__WINSRG |_SRC |_INCLUDE
4	DECAN	ARHIV	MARKS.TXT	__ITKI__ |__PMOS |__IS |_DIPLOMS |_BAKALAVR
5	DB	DATA	TRANS.TXT	__MAIN__ |__OFFICE |__FILIAL |_DEVELOP |_QA

Продовження таблиці 1

6	MUSIC	2CD	SONGS. TXT	__ROCK__ |__NIRVANA |__RUSSIAN |_ARIA |_BCOFFEE
7	DEMOS	PARTY	READIT. TXT	__INTROS__ |__MUSIC |__THREED |_PHONG |_GOURAUD
8	DOCS	NEEDR EAD	JAPAN. TXT	__LANGS__ |__EUROPE |__EAST |_CHINESE |_JAPANESE
9	PCAD	CUSTO MER	SLIST.TXT	__SCHEMES__ |__ANALOG |__DIGITAL |_LOGIC |_ONPIC
10	DISOP	2BASE	MODLIST.T XT	__KASKOD__ |__TRACES |__MODELS |_MONALISA |_PRMAVERA

Теоретичні відомості

Файлова система NetWare – UNF (Universal File System) забезпечує багато засобів, що покращують продуктивність роботи:

- Алгоритм ліфта для дискових операцій – це засіб дискової системи, що дозволяє визначати приорітети для запитів, які поступають на читання в залежності від встановлення головки диска. Аналогічно тому, як ліфт зупиняється на своєму шляху на визначених поверхах, а не переміщується між ними хаотично, даний алгоритм дозволяє мінімізувати переміщення головки диска та час доступу.

• Кешування файла зменшує число звертань до диска. Файли, які читаються найбільш часто, зберігаються в кеш-буфері диска, звідки їх можна при необхідності зчитати, а не звертатися за потрібною інформацією до диска. Файли в кеш-пам'яті також розбиваються за приорітетами, так що найбільш рідко використовувані файли звільняють у кеш-буфері місце для інших файлів.

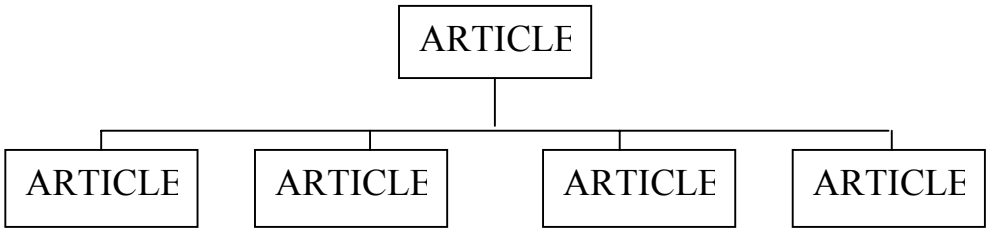
• Фоновий запис дає найвищий приорітет користувачам, яким потрібно виконати читання з диска. В NetWare операції читання з диска обробляються окремо від операцій запису. Такий розподіл дозволяє ОС записувати дані на диск у момент зниження інтенсивності запитів до диска. Фоновий запис підвищує швидкість читання даних з диска.

У NetWare можуть використовуватись файли об'ємом до 2 ГБт, а файлова система підтримує більш ніж 2 млн. каталогів на том і 100000 відкритих файлів. Тома можуть займати декілька дисків, а з додаванням нових дисків об'єм томів може автоматично збільшуватись.

Система відновлення файлів NetWare дозволяє відновлювати видалені файли. Ви можете задати мінімальний об'єм часу, протягом якого файл має бути доступним для відновлення, чи помітити файли для негайного видалення. Можна також зберігати видалені файли, поки є місце на диску. При цьому найбільш старі файли звільнюються та вивільнюють місце для нових. Видалені файли зберігаються навіть при видаленні каталогів.

Файлова система NetWare складається з томів, що є системою дискової пам'яті. Тома є об'єктами в каталозі NDS. Це найвищий (кореневий) рівень у структурі каталогів NetWare.

Сервер NetWare може підтримувати до 64 томів. Перший том створюється при інсталяції, завжди називається SYS і містить структуру каталогів:



System містить команди NetWare та файли, які використовуються системним адміністратором чи супервізором. PUBLIC містить файли NetWare, що доступні всім користувачам NetWare. LOGIN – це каталог, до якого користувачі отримують доступ при реєстрації в системі. Цей каталог відображується на першу доступну літеру мережного диска. MAIL збережено для сумісності з попередніми версіями NetWare.

Томи монтуються при завантаженні сервера NetWare. Том SYS монтується автоматично. Інші томи монтуються за допомогою команд в файлі запуску сервера AUTOEXEC.NCF. Монтувати та демонтувати томи можна за допомогою команд MOUNT та DISMOUNT на консолі сервера NetWare. Томи, що не використовуються чи використовуються рідко, краще не монтувати, оскільки кожен том використовує від 0.5 до 2 Мб пам'яті, залежно від розміру і типу – DOS, Unix чи Macintosh.

При додаванні в систему жорстких дисків до сервера томи можна розширювати (у деяких випадках навіть без зупинки сервера). Том фізично поділяється на сегменти, які можуть бути розташовані на одному чи декількох жорстких дисках. Том може містити до 32 сегментів, але кожен диск може містити лише 8 сегментів тому. Розподілення сегментів по різних дисках покращує продуктивність, оскільки це дозволяє зчитувати та записувати сегменти одночасно. Але це має і свої негативни сторони: при виході з ладу одного з дисків не буде монтуватися весь том. Тому сегментований том рекомендується завжди дублювати.

Каталоги NetWare

Ви можете посилатися на каталоги томів NetWare так само, як на каталоги DOS – з використанням маршрутів. Наприклад, повне им'я каталогу SYSTEM на томі SYS має вигляд SYS:SYSTEM. Якщо цей каталог має підкаталог USR, то його повним іменем буде:
SYS:SYSTEM\USR

Якщо цей каталог знаходиться на сервері SRTF, то на нього потрібно посилатися таким чином:

SRTF\SYS:SYSTEM\USR

Додавати нові каталоги ви можете за допомогою команди DOS MD чи утіліти NetWare Filer. У NetWare можна також використовувати команди DOS CD та RD.

Файли додатків та програм рекомендується зберігати окремо від файлів даних. Це попереджує їх випадкове видалення та полегшує обслуговування, особливо архівування.

Підключення до сервера

Якщо робоча станція налаштована вірно, то при завантаженні з'явиться ще один диск – мережний диск, який розташований на сервері. Ви можете переконатися в цьому, якщо запустите Norton Commander та переглянете список наявних дисків.

Якщо робоча станція має тільки диск **C:** (чи **C:, D:, E:**), новий диск буде позначатися **F:**. Якщо на робочій станції диск **F:** вже існував, новий диск буде позначатись наступною за латинським алфавітом буквою після тої, якою до завантаження мережної оболонки позначався останній локальний диск. Наприклад, якщо на робочій станції були диски **C:, D:, E:, F:, G:** и **H:**, то після підключення до мережі з'явиться диск **I:**.

Отже, у вас з'явився новий диск. Цей диск – результат відображення мережного тому SYS на локальний диск робочої станції. Якщо на сервері існує лише том SYS:, такого відображення цілком досить. Проте, якщо на сервері існує декілька томів, для забезпечення доступу до них необхідно відобразити всі ці томи на локальні диски робочої станції.

Для виконання цієї задачі вам знадобиться програма **map.exe**, яка розташована на сервері в каталозі SYS:PUBLIC.

Нехай нам потрібно відобразити том USERSVOL на диск **Z:**. Це можна зробити, якщо запустити програму **map.exe** з такими параметрами:

map z: = USERSVOL

Після введення такої команди на вашій робочій станції з'явится новий диск Z:, який буде відповідати тому USERSVOL. Зрозуміло, що такий том мусить існувати на сервері.

Ви можете відобразити на локальний диск не лише цілий том сервера, але й окремий каталог. Наприклад, така команда відобразить диск **Y:** на каталог SYS:PUBLIC:

map y: = sys:public

Якщо запустити програму map.exe без параметрів, вона видасть на екран поточне відображення локальних дисків на каталоги і томи сервера.

Якщо створений відображений диск вам більш не потрібен, його можна видалити тією ж програмою **map.exe**. Наприклад, для видалення диска **Z:** введіть команду:

map del z:

Якщо в мережі є декілька серверів, то необхідно перед іменем тому вказати ім'я сервера:

map w: = netlab\sys:

Перейдіть на мережний диск та запустіть існуючу там програму **login.exe**, причому як параметр вкажіть ім'я системного адміністратора – супервізора мережі:

F:

login supervisor

Після цього ви зможете перейти в кореневий каталог мережного диска та переглянути, які каталоги й файли знаходяться там. Якщо ви спробуєте, наприклад, за допомогою програми Norton Commander, вийти в кореневий каталог мережного диска до виконання програми **login.exe**, вам це не вдасться.

Контрольний приклад

1. Підключитися до сервера з іменем адміністратора та створити декілька каталогів у кореневому каталозі сервера згідно з варіантом.

Після завантаження ПЗ робочої станції переходимо на її останній диск (наприклад F:\) командою „CD F:\” запускаємо програму „LOGIN supervisor”. Можливо, що програма вимагатиме введення паролю, – взнайте пароль адміністратора у викладача.

Виходимо у кореневий каталог сервера з допомогою команди „CD..” та створюємо 2 каталоги HELP та MANUAL:

MD HELP

MD MANUAL.

2. Надати права доступу до створених каталогів своєму користувачу. Для цього запускаємо програму „SYSCON”. Далі діємо відповідно до системи меню програми та теоретичних відомостей. Необхідно надати повний доступ до виществорених каталогів користувачу DANNYDIO. Відключаємось від сервера командою LOGOUT

3. Надати права доступа до створених каталогів своєму користувачу. Переходимо на останній диск(наприклад, „F:\”) робочої станції командою „CD F:\” та запускаємо програму „LOGIN DANNYDIO”.

4. Підключити як окремі диски обидва створені каталоги. Використовуємо команди:

MAP Y:=SYS:HELP

MAP Z:=SYS:MANUAL.

5. Створити дерево каталогів на цьому диску згідно з варіантом, та створити у вказаному каталозі будь-який текстовий файл.

Для створення дерева каталогів використовуємо команди MS-DOS md (створити каталог), cd (перейти в каталог), rd(вилучити каталог), оболонку типу Norton Commander чи програму NetWare Filer.

Для створення файлу „EDLIN.TXT” в директорії „Y:\SOFT\DOS” використовуємо команди:

Y:

CD \

CD SOFT\DOS

COPY con EDLIN.TXT.

6. Скопіювати створений файл на інший раніше підключений диск. Для копіювання скористаємось командою NetWare „NCOPY”:

NCOPY Y:\SOFT\DOS\EDLIN.TXT Z:\EDLIN.TXT

7. Вивести зміст копії на консоль. Використовуємо команди:

Z:

CD \

CD SOFT\DOS

COPY EDLIN.TXT con

8. Відключити диски, створені у пункті 4.

Для відключення дисків використаємо команди „MAP DEL Y:” та „MAP DEL Z:”.

Питання для самоперевірки

1. Чи можуть існувати на сервері інші томи, крім SYS?
2. Чому при копіюванні файлів у мережі NetWare потрібно використовувати саме команду NCOPY, а не команду DOS COPY?
3. Поясніть ідею механізму відслідковування транзакцій (TTS) файлової системи NetWare.
4. Що треба зробити, щоб забезпечити можливість використання “нерідних” файлових систем (типу UNIX NFS) сервером NetWare?
5. Якою програмою необхідно користуватися адміністратору мережі для створення користувачів та зміни їх прав доступу?
6. Як можна відображати том окремого диска?
7. Як від’єднати диск від сервера та звільнити його букву?

Література

1. Lawrence B. Using NetWare 4.1. Special Edition, QUE, 1999. – 640 p.

2. Гаскин. Netware 5. Полное руководство в 2-х томах. – М.: Век, 2002.

3. Григорьев. NetWare 5. Настольная книга администратора. – М.: ДМК, 2000. – 530 с.