



1 завдання

1. Особливості архітектури ОС. сімейства Windows XP. Компонентні рішення ядра

Ряди компонентів Windows XP виконуються у привласненому рішенні, інші компоненти — у рішенні ядра. Почнемо розгляд системи з компонентів рішення ядра.

У традиційному розумінні ядро ОС містить усі компоненти привласненого рішення, однак у Windows XP поняття ядра закріплене тільки за одними із цих компонентів.

У Windows XP реалізовано рівень абстрагування від устаткування (у цій системі його звуть HAL, повного абстрактного layer). Для різних апаратних конфігурацій фірма Microsoft або сторонні розробники пишуть постатати різні реалізації HAL.

Хоча код HAL є дуже ефективним, його використання може знижувати продуктивність застосувань низьким рівнем. І такий ризик виокремлюють спеціалізовані пакети DirectX, які дають змогу приєднаним продривам звертатися безпосередньо до апаратного забезпечення, обходячи HAL, на інших рівнях системи.

Ядро Windows XP виконує за базові операції системи. До його основних функцій належать:

- керування контекстом, збереження і відновлення стану потіків.

- планування виконання потоків.
- реалізація засобів підтримки апаратного забезпечення, сформульованих за допомогою HDL (наприклад, передача інформації зберіганням параметрів).

Ядро Windows XP вживає двох видів служб ОС і має набір механізмів для реалізації політики керування ресурсами:

Основним завданням ядра є можлива ефективна завантаження процесорів системи. Ядро постійно перебуває в стані, який дозволяє виконання допоміжних функцій, може порушити тільки перебіг виконання коду ядра багатозадачності не підтримується. Для примусової роботи ядра ніколи не перебіг параметрів, передає під час виконання функцій.

Windows XP не можна віднести до якогось певного класу ОС. Наприклад, хоча за функціональністю ядро системи вживає політику мікроядра, для основної ОС нехарактерна масивна мікроядрова архітектура, основні у привнесеному режимі функціонують в іншій її політиці.

2 завдання

2. Аналіз простого вікна додатку. Описати функцію WinMain.

- Тема 1. Визначення вікна. Класи вікон і параметри вікон. Податок для створення вікна повинен визначити наступні параметри класа вікна (Window class)
- Ім'я вікна (Window name)
 - Стиль вікна (Window style)
 - Батьківське чи вікно батька (Parent window)
 - Розмір (Size)
 - Координати лівого верхнього кута (Location)
 - Місце в порядку розміщення (Position in the Z order)
 - Ідентифікатор ресурсів чи вікна анімації меню (Child window identifier or menu handle)
 - Описувач ресурсів прапорок (resource handle)
 - Податкові дані (Extra data)

2. Ієрархія вікон

3. Стиль вікон

4. Побігливість і черги побігливості.

У типовій додатку Windows головна функція WinMain() повинна виконати ініціалізацію зображення. Варє створювати в системі Windows клас типового вікна, який серія типового вікна буде виводити на екран внутрішні, передній вікно, які не мають такої необхідності зареєструвати. Windows виводить на екран і бачить нову тільки зареєстровані вікна.

2) Створити типові вікно і показати його на екрані.

Горизонтальне вікно наразі не треба створювати.

3) Організувати функції виведення повідомлень, що надходять в додаток, а також додаток повідомлень.

Записати заголовки Windows згідно функційного, коли користувач не подасть команду і дозволити за допомогою Alt+F4. Ці дії призводять до завершення роботи функції і віддалення програми зі списку діючих завдань.

4) Клас вікна і його характеристики

Створення і показ вікна

3) Як вивести повідомлення

1) Для виводу на екран ~~двох~~ вікна, створено вікна програми, треба зареєструвати нові вікна, в якому задаються загальні характеристики всіх вікон даної програми. Це дії виконуються в першій частині функції WinMain()

1) Заповнити структуру типу WNDCLASS.

2) Значення структури функції Wc: hInstance - lpstrClassName.

4) Icon: hIcon, hCursor, ~~hbr~~ IPfh WndProc

h.b Background.

5) Функція RegisterClass() - виводить адресу WNDCLASS

6) Створення і показ вікна → Windows CreateWindow(), ShowWindow()

7) Функція виведення повідомлень - функція Windows GetMessage() Dispatch Message()

3 завдання

3. Написати ПЛДВ програму, яка має експортну функцію що приймає масив та кількість елементів масиву та параметри, які виводяться в порядку збільшення елементів масиву

Рів матриці $matr[5][6]$ знайти суми елементів кожного рядка та записати їх в одновимірний масив. Виведення має бути відсортовані за зростаючим методом шукання.

Рів 12 фр - використання двовимірних масивів

1 - записати суми елементів рядків матричного масиву

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{ int matr[5][6], mas[5]; // matr[5] - масив сум рядків
```

```
int i, j, sum, stk;
```

```
// ввід матриці matr[5][6]
```

```
cout << "Vvod matr[5][6] \n";
```

```
for (i=0; i<5; i++)
```

```
for (j=0; j<6; j++)
```

```
cin >> matr[i][j];
```

```
// формування масиву сум рядків
```

```
for (i=0; sum=0; i<5; i++)
```

```
{ // знаходимо суму елементів рядків
```

```
for (j=0; j<6; j++)
```

```
sum += matr[i][j];
```

```
mas[i] = sum;
```

```
cout << mas[i] << " ";
```

```

for (i=1; i<5; i++)
    for (j=0; j<5-i; j++)
        if (mas[j] > mas[j+1])
            { stk = mas[j];
              mas[j] = mas[j+1];
              mas[j+1] = stk; }
cout << " \ not sortirovanniy massiv \n";
for (i=0; i<5; i++)
    cout << mas[i] << " ";
getch();
}

```