

Змістовний модуль:
КЕРУВАННЯ ВВЕДЕННЯМ – ВИВЕДЕННЯМ
Розділ 2: ФАЙЛИ

Лекція 9
Файли. Частина 2

ПИТАННЯ ДЛЯ ВИВЧЕННЯ

1. Робота з показниками файлів.
2. Визначення розміру файлів
3. Прямий доступ до даних.

ПОКАЗНИКИ ДЛЯ ФАЙЛІВ

```
DWORD WINAPI SetFilePointer(  
    __in HANDLE hFile,  
    __in LONG IDistanceToMove,  
    __in_out_opt PLONG lpDistanceToMoveHigh,  
    __in DWORD dwMoveMethod  
);
```

HANDLE hFile – дескриптор файлу (файл треба відкрити);

IDistanceToMove – молодша частина зміщення в файлі;

lpDistanceToMoveHigh – адреса для старшої частини зміщення перед викликом та зміщення відносно початку файлу після виконання функції

dwMoveMethod : визначає, відносно чого зміщення

FILE_BEGIN – відносно початку файлу;

FILE_CURRENT – відносно поточної позиції,

FILE_END – відносно кінця файлу.

Якщо треба розмір файлу зменшити, треба встановити показник (**SetFilePointer**) та встановити кінець файлу:

```
BOOL WINAPI SetEndOfFile( __in HANDLE hFile );
```

СТАНДАРТНІ ФУНКЦІЇ ДЛЯ ВИЗНАЧЕННЯ РОЗМІРУ ФАЙЛУ

1. *DWORD WINAPI GetFileSize(
__in HANDLE hFile, // Дескриптор файлу
__out LPDWORD lpFileSizeHigh //старша
частина розміру
);*

Недолік: не можна узнати про наявність помилки.

2. *BOOL WINAPI GetFileSizeEx(
__in HANDLE hFile,
__out PLARGE_INTEGER lpFileSize
// (LowPart, HighPart, QuadPart)
);*

СТАНДАРТНІ ФУНКЦІЇ ДЛЯ ВИЗНАЧЕННЯ РОЗМІРУ ФАЙЛУ. ПРИКЛАД

LARGE_INTEGER liFileSize

BOOL b = GetFileSizeEx (h, & liFileSize);

if (b)

*_tprintf (_T(“%l64d\n”),
 liFileSize.QuadPart);*

else _tprintf (_T(“Error\n”));

Недолік. Для визначення розміру файлу
треба його відкривати.

ВИЗНАЧЕННЯ РОЗМІРУ ФАЙЛУ ТА ВІЛЬНОГО МІСЦЯ НА ДИСКУ

Приклад. Додати до функції копіювання
необхідні оператори для перевірки наявності
вільного місця

```
LARGE_INTEGER NeedSize; // Потрібне місце
ULARGE_INTEGER Free, Total, Available;
b = GetFileSizeEx (hIn, &NeedSize);
if (b) b = GetDiskFreeSpaceEx( 0, &Free, &Total,
    &Available);
b = NeedSize.QuadPart <= Available.QuadPart;
```

РОЗШИРЕНА ФУНКЦІЯ ДЛЯ РОБОТИ З ПОКАЗНИКАМИ

```
BOOL WINAPI SetFilePointerEx(  
__in HANDLE hFile, // дескриптор файлу  
// Зміщення (до виконання операції)(64 біта)  
__in LARGE_INTEGER liDistanceToMove,  
// Зміщення (після виконання операції)(64 біта)  
__out_opt PLARGE_INTEGER lpNewFilePointer,  
// Тип зміщення (FILE_BEGIN, FILE_CURRENT,  
// FILE_END)  
__in DWORD dwMoveMethod  
);
```

РОЗШИРЕНА ФУНКЦІЯ ДЛЯ РОБОТИ З ПОКАЗНИКАМИ. ПРИКЛАД 1. 1

Приклад. Скласти функцію для заміни перших та останніх `len` байтів файлу, якщо розмір повинен бути більше ніж $2 * len$.

`BOOL ChangeBytes (LPCTSTR fName, DWORD len)`

```
{  
// Відкриття файлу  
// Перевірка умови: розмір повинен бути більше ніж  $2 * len$   
// Виділення пам'яті для  $2 * len$  даних  
// Читання перших len байтів  
// Встановлення показника на останні байти  
// Читання останніх len байтів  
// Встановлення показника на початок останніх байтів  
// Запис останніх len байтів  
// Встановлення показника на початок файлу  
// Запис перших len байтів  
// Визволення пам'яті  
// Закриття файлу
```


РОЗШИРЕНІ ФУНКЦІЇ ДЛЯ РОБОТИ З ПОКАЗНИКАМИ. ПРИКЛАД 1. 2

```
// Відкриття файлу
HANDLE h = CreateFile (fName, GENERIC_READ|GENERIC_WRITE,
    FILE_SHARE_READ, 0, OPEN_EXISTING, 0, 0);
// Перевірка умови: розмір повинен бути більше ніж 2 * len
b = h != INVALID_HANDLE_VALUE;
LARGE_INTEGER liSize;
if (b) {
    PBYTE pMem = 0, pMem1, pMem2;
    b = GetFileSizeEx (h, &liSize);
}
b = b && liSize.QuadPart >= 2 * len;

// Виділення пам'яті для 2 * len даних
```

РОЗШИРЕНІ ФУНКЦІЇ ДЛЯ РОБОТИ З ПОКАЗНИКАМИ. ПРИКЛАД 1. 3

```
// Виділення пам'яті для 2 * len даних
using namespace std;
...
PBYTE pMem = 0, pMem1, pMem2;
try{
    pMem = new BYTE [2 * len];
    pMem1 = pMem; pMem2 = pMem + len;
}
catch (std::bad_alloc&)
{
    b = FALSE; pMem = 0;
}
```

РОЗШИРЕНІ ФУНКЦІЇ ДЛЯ РОБОТИ З ПОКАЗНИКАМИ. ПРИКЛАД 1.4

```
// Читання перших len байтів
if (b) b = ReadFile (h, pMem1, len, &dwCount, 0);
// Встановлення показника на останні байти
LARGE_INTEGER liFirst = {0, 0}, liSecond = {-len, -1}, liTemp;
if (b) b = SetFilePointerEx (h, liSecond, &liTemp, FILE_END);
// Читання останніх len байтів
if (b) b = ReadFile (h, pMem2, len, &dwCount, 0);
// Встановлення показника на початок останніх байтів
if (b) b = SetFilePointerEx (h, liSecond, &liTemp, FILE_END);
// Запис останніх len байтів
if (b) b = WriteFile (h, pMem1, len, &dwCount, 0);
// Встановлення показника на початок файлу
if (b) b = SetFilePointerEx (h, liFirst, &liTemp, FILE_BEGIN);
// Запис перших len байтів
if (b) b = WriteFile (h, pMem2, len, &dwCount, 0);
```

РОЗШИРЕНІ ФУНКЦІЇ ДЛЯ РОБОТИ З ПОКАЗНИКАМИ. ПРИКЛАД 1. 5

Використання функції ChangeBytes:

```
TCHAR *fName = _T("x.txt");  
    b = ChangeBytes (fName, 10);  
    _tprintf (_T("ChangeBytes: %s\n"), b? _T("Yes") :  
    _T("No"));  
// Файл x.txt треба розташувати в папці, де знаходиться  
    файл проекту з розширенням vsproj.
```

Відповідь: Yes

Стан файлу x.txt до зміни:

2222222222444444444444441111111111

Стан файлу x.txt після зміни:

1111111111444444444444442222222222

РОЗШИРЕНІ ФУНКЦІЇ ДЛЯ РОБОТИ З ПОКАЗНИКАМИ. ПРИКЛАД 2

Скоротити розмір файлу на задане значення. Якщо розмір файлу менше, то скоротити до нульової довжини

```
BOOL ChangeFileSize (LPCTSTR fName, __int64 size)
```

```
{  
    BOOL b;  
    HANDLE h = CreateFile (fName, GENERIC_READ|GENERIC_WRITE, FILE_SHARE_READ, 0,  
        OPEN_EXISTING, 0, 0);  
    b = h != INVALID_HANDLE_VALUE;  
    if (b) {  
        LARGE_INTEGER liSize, liTemp;  
        b = GetFileSizeEx (h, &liSize);  
        if (b){  
            liSize.QuadPart = liSize.QuadPart < size ? 0 : liSize.QuadPart - size;  
            b = SetFilePointerEx (h, liSize, &liTemp, FILE_BEGIN);  
            if (b) b = SetEndOfFile (h);  
        }  
        CloseHandle (h);  
    }  
    return b;  
}
```

ФУНКЦІЇ ДЛЯ РОБОТИ З ПОКАЗНИКАМИ. ПРИКЛАД 3.1

- Створити “базу даних” з записами різної довжини. Кожний запис має формат: довжина запису, сам запис.
- Створити індексний файл, в якому для кожного запису визначити зміщення в “базі даних”.
- Визначити функцію для отримання запису з заданим номером.

ФУНКЦІЇ ДЛЯ РОБОТИ З ПОКАЗНИКАМИ. ПРИКЛАД 3.2

Створення “бази даних”. Хай записами є рядки з символами.

```
BOOL CreateDB (TCHAR *fName, TCHAR *s[], size_t n){  
    HANDLE h = CreateFile (fName, GW, FSR, 0, OA, 0, 0);  
    DWORD dwCount; BOOL b = h!= IHV;  
    if (b) {  
        for (size_t i = 0; i < n; ++i) {  
            size_t len = (_tcslen (s [i]) + 1) * sizeof (TCHAR);  
            b = WriteFile (h, &len, sizeof (len), &dwCount, 0);  
            if (b) b = WriteFile (h, s[i], len, &dwCount, 0);  
            if (!b) break;  
        }  
        CloseHandle (h);  
    }  
    return b;  
}
```

ФУНКЦІЇ ДЛЯ РОБОТИ З ПОКАЗНИКАМИ. ПРИКЛАД 3.3

Створення індексного файлу (зміщення для початку записів)

```
BOOL CreateIndexBD (TCHAR *DBName, TCHAR *DBIndex) {  
    DWORD dwCount;  
    HANDLE h1 = CreateFile (DBName, GR, FSR, 0, OE, 0, 0);  
    HANDLE h2 = CreateFile (DBIndex, GW, FSR, 0, OA, 0, 0);  
    BOOL b = h1 != IHV && h2 != IHV;  
    if (b) {  
        size_t len; size_t offset = 0;  
        for (size_t i = 0; ; i++) {  
            if (b) b = WriteFile (h2, &offset, sizeof (size_t), &dwCount, 0);  
            if (b){  
                b = ReadFile (h1, &len, sizeof (size_t), &dwCount, 0);  
                if (dwCount == 0) break;  
                SetFilePointer (h1, len, 0, FILE_CURRENT);  
                offset += len + sizeof (size_t);  
            }  
        }  
    }  
    if (h1) CloseHandle (h1);    if (h2) CloseHandle (h2); return b;  
}
```


ФУНКЦІЇ ДЛЯ РОБОТИ З ПОКАЗНИКАМИ. ПРИКЛАД 3.4

Отримання запису по його номеру

```
size_t ReadDBRecord (TCHAR *DBName, TCHAR *DBIndex, size_t i, TCHAR * Rec){  
    size_t res = 0; DWORD dwCount;  
    HANDLE h1 = CreateFile (DBName, GR, FSR, 0, OE, 0, 0),  
        h2 = CreateFile (DBIndex, GR, FSR, 0, OE, 0, 0);  
    BOOL b = h1!= IHV && h2 != IHV;  
    if (b){  
        DWORD dwRecs = GetFileSize (h2, 0) / sizeof (size_t);  
        b = dwRecs > i;  
        if (b){  
            size_t offset; SetFilePointer (h2, i * sizeof (size_t), 0, FILE_BEGIN);  
            b = ReadFile (h2, &offset, sizeof (size_t), &dwCount, 0);  
            if (b){  
                SetFilePointer (h1, offset, 0, FILE_BEGIN);  
                b = ReadFile (h1, &res, sizeof (size_t), &dwCount, 0);  
            }  
            if (b && Rec != 0) b = ReadFile (h1, Rec, res, &dwCount, 0);  
        }  
    }  
    if (h1) CloseHandle (h1); if (h2) CloseHandle (h2);  
    return res;}  
}
```

ФУНКЦІЇ ДЛЯ РОБОТИ З ПОКАЗНИКАМИ. ПРИКЛАД 3.5

Головна програма:

```
...
TCHAR *s[] = {_T("1"), _T("22"), _T("333"), _T("4444"), _T("55555"),
              _T("666666"), _T("7777777")};
// Створення "бази даних"
b = CreateDB (_T("BD.bin"), s, sizeof (s)/ sizeof (TCHAR*));
// Створення індексів для "бази даних"
if (b) b = CreateIndexDB (_T("BD.bin"), _T("BD.ind"));
// Отримання довжини запису, а потім і запису
if (b){
    size_t len = ReadDBRecord (_T("BD.bin"), _T("BD.ind"), 4, 0);
    TCHAR *rec = new TCHAR [len / sizeof (TCHAR)];
    ReadDBRecord (_T("BD.bin"), _T("BD.ind"), 4, rec);
    _tprintf (_T("%s\n"), rec);
    delete []rec;
}
```

ВИСНОВКИ

- Розглянуті операції для роботи з показниками файлів та їх атрибутами.
- При визначенні розміру файлу можна використовувати функції для відкритих файлів (GetFileSize, GetFileSizeEx), та без їх відкриття (GetFileAttributesEx).
- Є можливість зміни атрибутів файлів, що дає змогу запису в файли Тільки для читання, можна встановлювати атрибути пов'язані з компресією файлів та їх шифруванням, що буде розглянуто нижче.

ПИТАННЯ ДЛЯ САМОСТІЙНОГО ВИВЧЕННЯ

1. Перевірити можливість зміни атрибутів файлів в залежності від списку доступу, який задається адміністратором. Зробіть висновки.
2. Зробіть експеримент, в якому значення зміщення для функції `SetFilePointer` перевищує розмір файлу. Зробіть висновок по зміні розміру файлу після такої операції.

МАТЕРІАЛИ ДЛЯ ЕКСПРЕС-КОНТРОЛЮ

- Що таке показник файлу?
- Де знаходиться показник файлу після виконання функції CreateFile?
- Яким чином організувати запис додаткових даних в кінець файлу, вставку додаткових даних на початку файлу без руйнування попередніх даних.
- В яких режимах треба відкривати файл, якщо потрібно змінити його розмір (зменшити, збільшити)
- Які атрибути файлів ви знаєте, як можна змінити атрибути файлу в середовищі, де ви працюєте?
- Наведіть приклад необхідності програмної зміни атрибутів.
- Додайте до класу для роботи з файлом функцію для визначення розміру файлу та поточного положення показника.