Programowanie niskopoziomowe

Materiały uzupełniające

Labolatoria 6

„Operacje na plikach i katalogach”

Cześć materiałów autorstwa prof. A. Timofiejewa

Uwaga: Uwaga od zajęć 06 można wykorzystywać makra i procedury MASM32 obecne w tym asemblerze, zapoznaj się z listą tych procedur w edytorze MASM32 ->Menu Help -> MASM32 Library Reference.

Przykład użycia w lab06przykład.asm

1 Tworzenie katalogu

Funkcja API Win32 **GetCurrentDirectoryA** z pliku user32.inc podaje bieżący katalog.
Argumenty funkcji:

nBufferLength – rozmiar bufora,
lpBuffer – adres bufora do umieszczenia nazwy.

Funkcja zwraca (przez rejestr EAX) długość ciągu znaków (nazwę katalogu ze ścieżką). Ciąg znaków jest zapisany w kodzie ASCII i jest zakończony zerem. Nazwa bieżącego katalogu ze ścieżką nie zawiera ostatniego znaku „\”.

Funkcja API Win32 **CreateDirectoryA** z pliku user32.inc tworzy katalog.
Argumenty funkcji:

lpPathName – adres bufora z nazwą nowego katalogu (ciąg znaków zakończony 0),
lpSecurityAttributes – adres struktury SecurityAttributes opisującej prawa dostępu do katalogu (można ustawić wartość 0, jeśli katalog jest „zwykły”).

Funkcja **CreateDirectoryA** zwraca 0 w przypadku błędu. Opis błędu można otrzymać wywołując funkcję GetLastError.

2 Operacje na łańcuchach znaków

lstrcpyA do kopiowania łańcuchów znaków (ciągu znaków zakończonych zerem), **lstrcatA** kontaktacja , łączenie dwóch łańcuchów, **lstrlenA** do obliczenia długości łańcucha znaków.

Funkcja **lstrcpyA** ma argumenty:

lpString1 – adres bufora – miejsca przeznaczenia,
lpString2 – adres wierszu do kopiowania.

Funkcja **lstrcatA** ma argumenty:

lpString1 – adres pierwszego bufora – (tu także będzie umieszony wynikowy łańcuch znaków)

lpString2 – adres drugiego bufora do połączenia.

Funkcja **lstrlenA** ma argument:

lpString – adres bufora, którego długość jest wyznaczana.

3 Tworzenie lub otwarcie pliku

Funkcja API Win32 **CreateFileA** z pliku user32.inc tworzy lub otwiera plik.

Argumenty funkcji:

lpszName - adres nazwy pliku ze ścieżką,

fdwAccess – tryb dostępu do pliku:

GENERIC_READ – do odczytu,

GENERIC_WRITE – do zapisu, które można połączyć operatorem „OR”,

fdwShareMode – tryb dostępu do pliku ze strony innych aplikacji (można ustawić na 0),

lpsa – adres struktury SECURITY_ATTRIBUTES z informacjami o zabezpieczeniach (można ustawić na 0),

fdwCreate – tryb otwarcia pliku:

CREATE_ALWAYS – kreacja nowego pliku,

OPEN_EXISTING – otwarcie istniejącego pliku,

fdwAttrsAndFlags – dodatkowe atrybuty (można ustawić na 0),

hTemplateFile – deskryptor pliku tymczasowego (można ustawić na 0).

Funkcja zwraca (przez rejestr EAX) uchwyt (HANDLE) pliku, który należy stosować w funkcjach plikowych.

4 Zamknięcie pliku

Do zamknięcia pliku służy funkcja API Win32 **CloseHandle** z pliku user32.inc.

Argumenty funkcji:

hObject – uchwyt pliku.

5 Zapisywanie do pliku

Do zapisywania do pliku służy funkcja API Win32 **WriteFile** z pliku user32.inc.

Argumenty funkcji:

hFile – uchwyt pliku.

lpBuffer – adres bufora z danymi,

nNumberOfBytesToWrite – ilość bajtów do zapisywania,

lpNumberOfBytesWritten – adres zmiennej do przechowywania liczby zapisanych bajtów,

lpOverlapped – adres struktury OVERLAPPED z informacją o nadpisaniu (można ustawić na 0).

Funkcja zwraca (przez rejestr EAX) ilość faktycznie zapisanych bajtów.

6 Odczyt z pliku

Do odczytu z pliku służy funkcja API Win32 **ReadFile** z pliku user32.inc.
Argumenty funkcji:

hFile – uchwyt pliku.
lpBuffer – adres bufora do przyjmowania danych,
nNumberOfBytesToRead – ilość bajtów do odczytu
lpNumberOfBytesRead – adres zmiennej do przechowywania liczby odczytanych bajtów,
lpOverlapped – adres struktury OVERLAPPED z informacją o nadpisaniu (można ustawić na 0).

Funkcja zwraca (przez rejestr EAX) ilość faktyczne odczytanych bajtów.

7 Przemieszczenie w pliku

Do przemieszczenia w pliku służy funkcja API Win32 **SetFilePointer** z biblioteki user32.lib.
Funkcja **SetFilePointer** ma argumenty:

hFile – uchwyt pliku.
lDistanceToMove – odległość (w bajtach)
lpDistanceToMoveHigh – ten argument musi być równy 0, jeśli rozmiar pliku jest mniejszy niż (2^32 – 2), a jeśli rozmiar pliku jest większy niż (2^32 – 2), to ten argument musi być adresem 32-bitowej zmiennej, która razem z argumentem lDistanceToMove tworzy 64-bitową odległość,
dwMoveMethod – opcja wskazująca na regułę liczenia odległości:
FILE_BEGIN – odległość jest liczona od początku pliku,
FILE_CURRENT – odległość jest liczona od aktualnej pozycji,
FILE_END – odległość jest liczona od końca pliku.

Funkcja zwraca (przez rejestr EAX) pozycję wskaźnika pliku. Jeśli rozmiar pliku jest większy niż (2^32 – 2), to argument **lpDistanceToMoveHigh** wskazuje na 32-bitową zmienną, która razem z zawartością rejestru EAX tworzy 64-bitową pozycję.

8 Generowanie liczb pseudolosowych

Do generowania liczb pseudolosowych służy funkcja **nrandom** z biblioteki MASM32.
Funkcja **nrandom** ma argument typu DWORD – zakres liczb (liczba całkowita nieujemna).
Funkcja zwraca (przez rejestr EAX) liczbę pseudolosową.

9 Wyświetlenie numeru błędu

```
.DATA
       formErr DB    "%d=%xh",0Dh,0Ah,0
       nErr    DD    (?)
       bufor   DB    128 dup (0)
       rbuf    DD    (?)
       rout    DD    (?)
       hout    DD    (?)
```

```
.CODE
;--- jeśli blad ----
       call   GetLastError
       mov    nErr,EAX
       INVOKE wsprintfA,OFFSET bufor,OFFSET formErr,nErr,nErr
       mov    rbuf, EAX
       push   0      ;; rezerwa, musi być zero
       push   OFFSET rout  ;; wskaźnik na faktyczną ilość wyprowadzonych znaków
       push   rbuf   ;; ilość znaków
       push   OFFSET bufor  ;; wskaźnik na tekst
       push   hout   ;; deskryptor buforu konsoli
       call   WriteConsoleA ;; wywołanie funkcji WriteConsoleA
```