.586

.model flat, stdcall

include D:\masm32\include\kernel32.inc

include D:\masm32\include\user32.inc

includelib D:\masm32\lib\kernel32.lib

includelib D:\masm32\lib\user32.lib

.data

CaptionN db "Я прога на асемблері. ",0

TextN db "Здоровенькі були!", 10, 13, "Даніленко Наталія",0

res dd 256 dup(0)

Text db 'EAX=xxxxxxxx',13,10,

'EBX=xxxxxxxx',13,10,

'ECX=xxxxxxxx',13,10,

'EDX=xxxxxxxx',0

Caption0 db "CPUID 0",0

Caption1 db "CPUID 1",0

Caption2 db "CPUID 2",0

Caption00 db "CPUID ..00h",0

Caption01 db "CPUID ..01h",0

Caption02 db "CPUID ..02h",0

Caption03 db "CPUID ..03h",0

Caption04 db "CPUID ..04h",0

Caption05 db "CPUID ..05h",0

Caption08 db "CPUID ..08h",0

Model db 32 dup(0)

CaptionModel db "CPUID 0 Model",0

.code

;ця процедура записує 8 символів HEX коду числа

;перший параметр - 32-бітове число

;другий параметр - адреса буфера тексту

dwordToStrHex proc

push ebp

mov ebp,esp

mov ebx,[ebp+8] ;другий параметр

mov edx,[ebp+12] ;перший параметр

xor eax,eax

mov edi,7

@next:

mov al,dl

and al,0Fh ;виділяємо одну шістнадцяткову цифру

add ax,48 ;так можна тільки для цифр 0-9

cmp ax,58

jl @store

add ax,7 ;для цифр A,B,C,D,E,F

@store:

mov [ebx+edi],al

shr edx,4

dec edi

cmp edi,0

jge @next

pop ebp

ret 8

dwordToStrHex endp;

main:

invoke MessageBoxA, 0, ADDR TextN, ADDR CaptionN, 0

mov eax, 0

cpuid

mov dword ptr[Model], ebx

mov dword ptr[Model+4], edx

mov dword ptr[Model+8], ecx

invoke MessageBoxA, 0, ADDR Model, ADDR CaptionModel, 0

mov eax, 0

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call DwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call dwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call dwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call dwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption0, 0

mov eax, 1

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call dwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call dwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call dwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call dwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption1, 0

mov eax, 2

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call dwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call dwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call dwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call dwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption2, 0

mov eax, 80000000h

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call dwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call dwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call dwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call dwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption00, 0

mov eax, 80000001h

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call dwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call dwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call dwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call dwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption01, 0

mov eax, 80000002h

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call dwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call dwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call dwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call dwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption02, 0

mov eax, 80000003h

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call dwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call dwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call dwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call dwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption03, 0

mov eax, 80000004h

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call dwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call dwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call dwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call dwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption04, 0

mov eax, 80000005h

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call dwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call dwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call dwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call dwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption05, 0

mov eax, 80000008h

cpuid

mov dword ptr[res], eax

mov dword ptr[res+4], ebx

mov dword ptr[res+8], ecx

mov dword ptr[res+12], edx

push [res] ;значення регістру EAX з масиву res

push offset [Text+4] ;адреса, куди записуються 8 символів

call dwordToStrHex

push [res+4] ;значення регістру EBX з масиву res

push offset [Text+18]

call dwordToStrHex

push [res+8] ;значення регістру ECX з масиву res

push offset [Text+32]

call dwordToStrHex

push [res+12] ;значення регістру EDX з масиву res

push offset [Text+46]

call dwordToStrHex

invoke MessageBoxA, 0, ADDR Text, ADDR Caption08, 0

invoke ExitProcess, 0

end main;