# Đây là thư viện đã soạn:

%macro sys.print 2

mov edx, %1

mov ecx, %2

mov ebx,1

mov eax,4

int 0x80

%endmacro

sys.writeChar: ;dung thanh ghi al

push rdx

push rcx

push rax

mov edx,1

mov byte[local],al

mov ecx,local

mov ebx,1

mov eax,4

int 0x80

pop rax

pop rcx

pop rdx

ret

%macro sys.writeHex 1 ;dung thanh ghi al

mov al,%1

mov ebx,hexChar

xlat ;tim tren ebx xem co al ko

call sys.writeChar

int 0x80

%endmacro

%macro sys.writeHexDigital 0 ;dung thanh ghi al

mov ebx,hexChar

xlat ;tim tren ebx xem co al ko

call sys.writeChar

int 0x80

%endmacro

sys.writeHexAl: ;lam tren al

push rdx

mov dl,al

shr al,4

sys.writeHexDigital

mov al,dl

and al,15

sys.writeHexDigital

pop rdx

ret

sys.intelHexWord:

push rax

mov eax,edx

shr eax,8

call sys.writeHexAl

mov eax,edx

call sys.writeHexAl

int 0x80

pop rax

ret

section .data

hexChar dw '0123456789abcdef'

hold db 2

# câu 1

## giải thích code:

sys.intelHexWord:

push rax

mov eax,edx

shr eax,8

call sys.writeHexAl

mov eax,edx

call sys.writeHexAl

int 0x80

pop rax

ret

### giải thích:

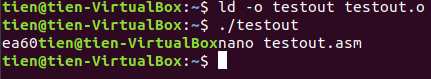
đầu tiên em sẽ nạp số 16-bit là 60000 tương đương với EA60 ở dạng hex. Nhỏ hơn 216



Sys.intelHexWord là hàm đã được code ở thư viện.

1. hàm sẽ cất thanh ghi 32 bit vào thư viện(vì máy em là bản 64 bit nên phải cất rax, nếu máy 32 bit thì chỉ cần push eax).
2. Sau đó đưa giá trị edx vào thanh ghi eax.
3. Sau đó vì phài xử lý 8 bit cao trước nên ta phải shift right để đưa 8bit cao về 8bit thấp. là thanh ghi al
4. Sau đó em gọi hàm writeHexAl(tức là hàm writehexdigital)
5. Là ta có được 8 bit cao (EA)
6. Sau đó em lại đưa trả giá trị ban đầu trước khi shift lại cho eax
7. Sau đó chỉ cần gọi writehexdigital là sẽ dược 8bit thấp. tương đương 60
8. Sau đó kết thúc lệnh int 0x80
9. Pop rax

### Kết quả:



Và đây là kết quả.