64bit ALP to convert Hex to Bcd: Anom Devgun

Gr:21810017

|  |
| --- |
| %macro scall 4 |
|  |

|  |
| --- |
| mov rax,%1 |
|  |

|  |
| --- |
| mov rdi,%2 |
|  |

|  |
| --- |
| mov rsi,%3 |
|  |

|  |
| --- |
| mov rdx,%4 |
|  |

|  |
| --- |
| syscall |
|  |

|  |
| --- |
| %endmacro |
|  |

|  |
| --- |
| section .data |
|  |

|  |
| --- |
| m1 db "enter 4 digit hex number=",10 |
|  |

|  |
| --- |
| l1 equ $-m1 |
|  |

|  |
| --- |
| m2 db "the 64 bit no. is =",10 |
|  |

|  |
| --- |
| l2 equ $-m2 |
|  |

|  |
| --- |
| m3 db "equivalent BCD number is =",10 |
|  |

|  |
| --- |
| l3 equ $-m3 |
|  |

|  |
| --- |
| section .bss |
|  |

|  |
| --- |
| num resq 10 |
|  |

|  |
| --- |
| char\_ans resq 1 |
|  |

|  |
| --- |
| count resq 1 |
|  |

|  |
| --- |
| arr resq 1 |
|  |

|  |
| --- |
| section .text |
|  |

|  |
| --- |
| global \_start |
|  |

|  |
| --- |
| \_start: |
|  |

|  |
| --- |
| scall 1,1,m1,l1 |
|  |

|  |
| --- |
| scall 0,0,num,10 |
|  |

|  |
| --- |
| call accept\_proc |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| scall 1,1,m3,l3 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| call h2b |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| ;/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* EXIT \*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |

|  |
| --- |
| mov rax,60 |
|  |

|  |
| --- |
| mov rdi,0 |
|  |

|  |
| --- |
| syscall |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| accept\_proc: |
|  |

|  |
| --- |
| mov rsi,num |
|  |

|  |
| --- |
| mov rbx,0 |
|  |

|  |
| --- |
| mov rax,0 |
|  |

|  |
| --- |
| mov rcx,4 |
|  |

|  |
| --- |
| back: |
|  |

|  |
| --- |
| rol rbx,04 |
|  |

|  |
| --- |
| mov al,[rsi] |
|  |

|  |
| --- |
| cmp al,39h |
|  |

|  |
| --- |
| jbe next |
|  |

|  |
| --- |
| sub al,07h |
|  |

|  |
| --- |
| next: |
|  |

|  |
| --- |
| sub al,30h |
|  |

|  |
| --- |
| add rbx,rax |
|  |

|  |
| --- |
| inc rsi |
|  |

|  |
| --- |
| dec rcx |
|  |

|  |
| --- |
| jnz back |
|  |

|  |
| --- |
| ret |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| h2b: |
|  |

|  |
| --- |
| mov rbp,arr |
|  |

|  |
| --- |
| mov rax,rbx |
|  |

|  |
| --- |
| mov rbx,0Ah |
|  |

|  |
| --- |
| back2: |
|  |

|  |
| --- |
| xor rdx,rdx |
|  |

|  |
| --- |
| div rbx |
|  |

|  |
| --- |
| mov [rbp],rdx |
|  |

|  |
| --- |
| inc rbp |
|  |

|  |
| --- |
| inc byte [count] |
|  |

|  |
| --- |
| cmp rax,00h |
|  |

|  |
| --- |
| jne back2 |
|  |

|  |
| --- |
| dec rbp |
|  |

|  |
| --- |
| display\_bcd: |
|  |

|  |
| --- |
| xor rdx,rdx |
|  |

|  |
| --- |
| mov rdx,[rbp] |
|  |

|  |
| --- |
| add dl,30h |
|  |

|  |
| --- |
| mov [char\_ans],dl |
|  |

|  |
| --- |
| scall 1,1,char\_ans,1 |
|  |

|  |
| --- |
| dec rbp |
|  |

|  |
| --- |
| dec byte [count] |
|  |

|  |
| --- |
| jnz display\_bcd |
|  |

Ret

