Exp No: 11 **Date:** 19/10/2020

Name: Swetha Saseendran

Reg No: 185001183

DISPLAY SYSTEM DATE AND TIME

Aim:

To write assembly language programs to perform the following system operations:

1. Display System Date
2. Display System Time

Programs:

(i) SYSTEM DATE

Algorithm:

- Declare the data segment.
- Initialize data segment with variables to store day, month and year.
- Close the data segment.

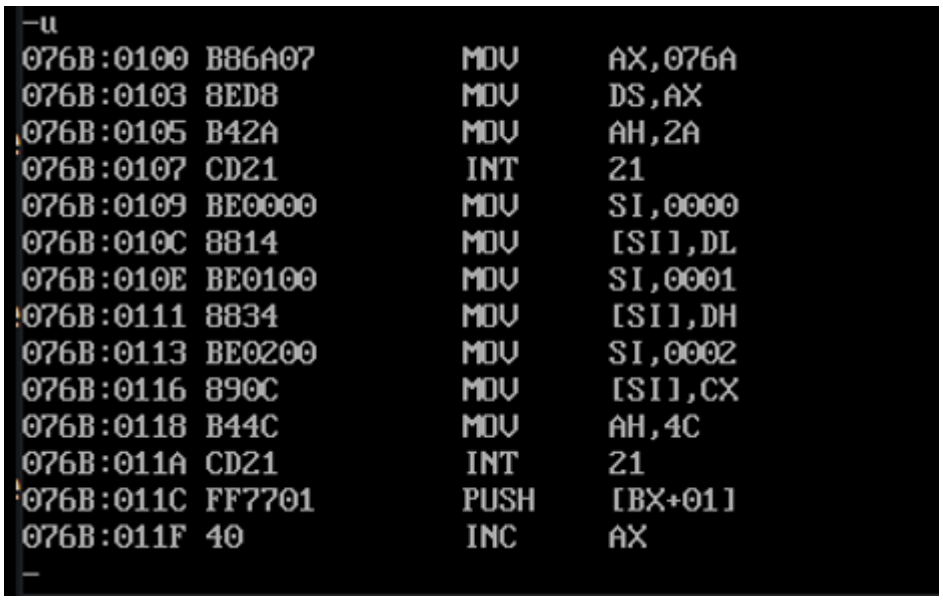
- Declare the code segment.
- Set a preferred offset (preferably 100h)
- Load the data segment content into AX register.
- Transfer the contents of AX register to DS register.
- Load 2Ah to AH register. (DOS function to obtain system date)
- Call interrupt 21h to service the DOS function.
- Load the offset address of variable ‘day’ to SI.
- Transfer contents of DL register through SI to variable ‘day’.
- Load the offset address of variable ‘month’ to SI.

- Transfer contents of DH register through SI to variable ‘month’.
- Load the offset address of variable ‘year’ to SI.
- Transfer contents of CX register through SI to variable ‘year’.
- Introduce an interrupt for safe exit. (INT 21h)
- Close the code segment.

PROGRAM	*COMMENTS*
assume cs:code, ds:data	Declare code and data segment.
data segment	Initialize data segment with values.
day db 01 dup(?)	Variable to store day.
month db 01 dup(?)	Variable to store month.
year db 02 dup(?)	Variable to store year.
data ends	
code segment	Start the code segment.
org 0100h	Initialize an offset address.
start: mov ax, data	Transfer data from “data” to AX.
mov ds, ax	Transfer data from memory location AX to DS.
mov ah, 2Ah	Load 2Ah to AH (DOS code for system date function)
int 21h	Interrupt DOS with 21h to get the system date.
mov si, offset day	Load offset of variable ‘day’ to SI.
mov [si], dl	Copy to ‘day’ the value of DL through SI.
mov si, offset month	Load offset of variable ‘month’ to SI.
mov [si], dh	Copy to ‘month’ the value of DH through SI.
mov si, offset year	Load offset of variable ‘year’ to SI.
mov [si], cx	Copy to ‘year’ the value of CX through SI.
mov ah, 4ch	Moves the hexadecimal value 4c to ah.

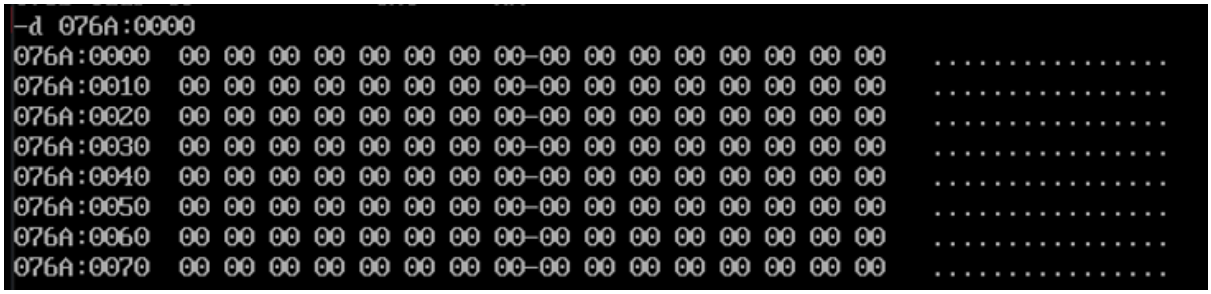
int 21h	When Software interrupt 21 is called with AH=4C, then current process terminates. (i.e., These two instructions are used for the termination of the process).
code ends *end start*	

Unassembled Code:

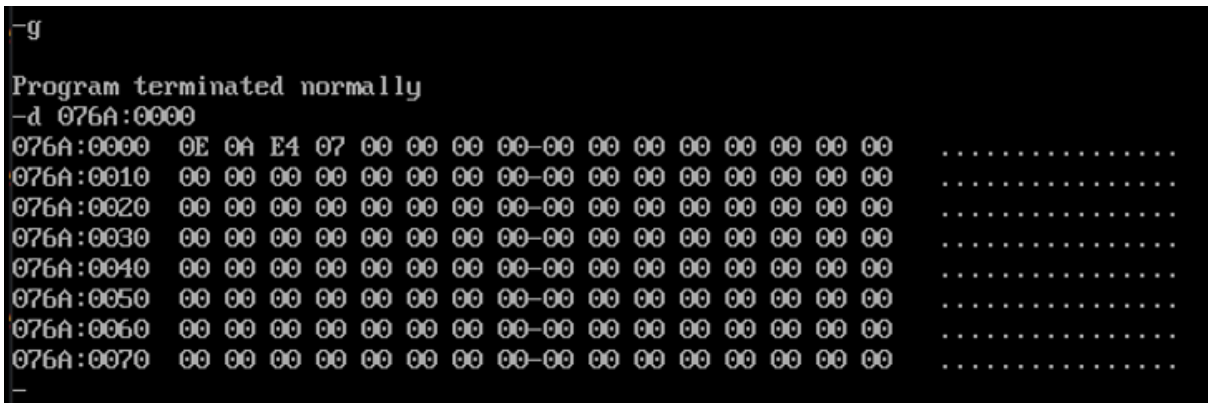


Snapshot of sample input and output:

INPUT:



OUTPUT:



(ii) SYSTEM TIME

Algorithm:

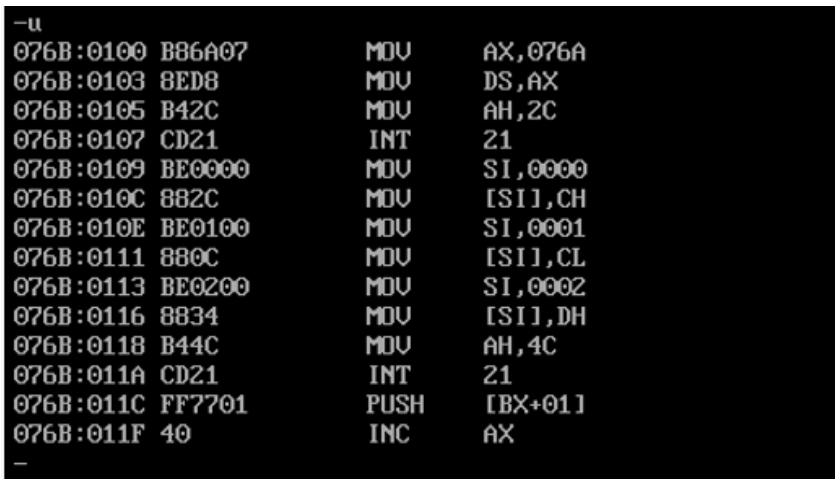
- Declare the data segment.
- Initialize data segment with variables to store hour, minute and second.
- Close the data segment.

- Declare the code segment.
- Set a preferred offset (preferably 100h)
- Load the data segment content into AX register.
- Transfer the contents of AX register to DS register.
- Load 2Ch to AH register. (DOS function to obtain system time)
- Call interrupt 21h to service the DOS function.
- Load the offset address of variable 'hour' to SI.
- Transfer contents of CH register through SI to variable 'hour'.
- Load the offset address of variable 'minute' to SI.
- Transfer contents of CL register through SI to variable 'minute'.
- Load the offset address of variable 'second' to SI.
- Transfer contents of DH register through SI to variable 'second'.
- Introduce an interrupt for safe exit. (INT 21h)
- Close the code segment.

PROGRAM	*COMMENTS*
assume *cs:code, ds:data*	Declare code and data segment.
data segment	Initialize data segment with values.
hour db 01 dup(?)	Variable to store hour.
minute db 01 dup(?)	Variable to store minute.

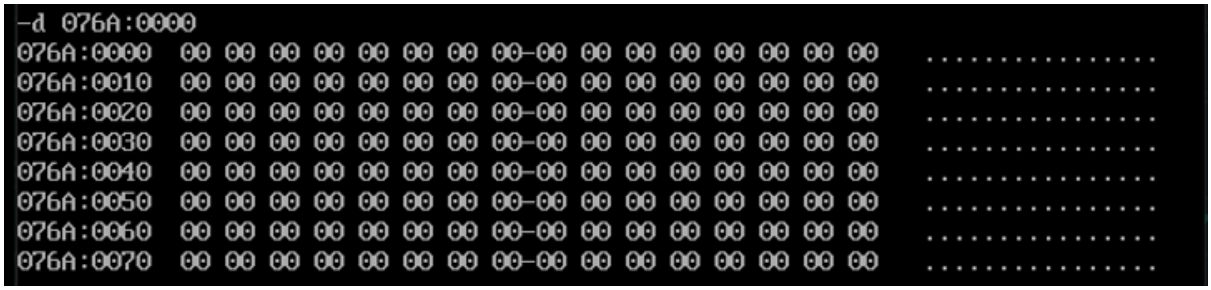
second db 02 dup(?)	Variable to store second.
data ends	
code segment	Start the code segment.
org 0100h	Initialize an offset address.
start: *mov ax, data*	Transfer data from “data” to AX.
mov ds, ax	Transfer data from memory location AX to DS.
mov ah, 2Ch	Load 2Ch to AH (DOS code for system time function)
int 21h	Interrupt DOS with 21h to get the system time.
mov si, offset hour	Load offset of variable ‘hour’ to SI.
mov [si], ch	Copy to ‘hour’ the value of CH through SI.
mov si, offset minute	Load offset of variable ‘minute’ to SI.
mov [si], cl	Copy to ‘minute’ the value of CL through SI.
mov si, offset second	Load offset of variable ‘second’ to SI.
mov [si], dh	Copy to ‘second’ the value of DH through SI.
mov ah, 4ch	Moves the hexadecimal value 4c to ah.
int 21h	When Software interrupt 21 is called with AH=4C, then current process terminates. (i.e., These two instructions are used for the termination of the process).
code ends	
end start	

Unassembled Code:

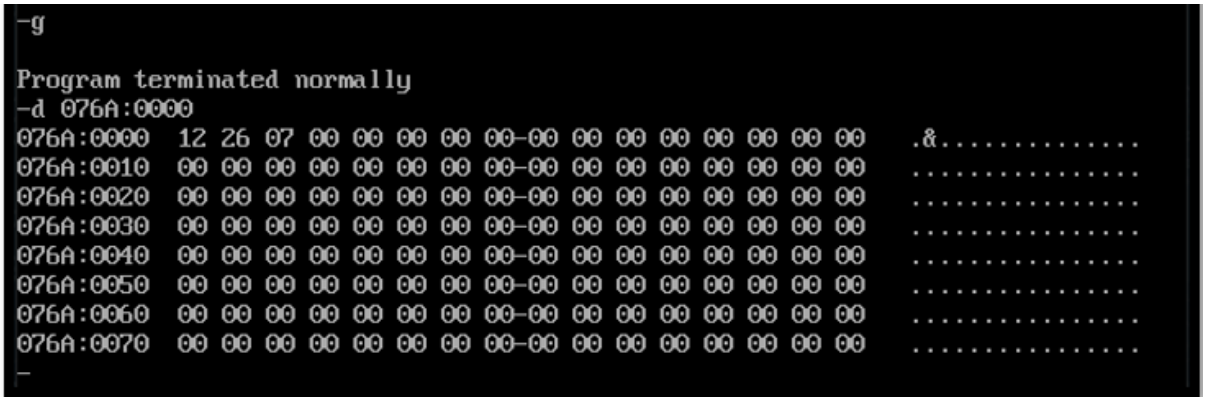


Snapshot of sample input and output:

INPUT:



OUTPUT:



Result:

The assembly level programs were written to perform the above specified system operations, namely, system date and system time and the output was verified.