

Display System date and time

Expt No: 11
Date : 16/10/2020

Name: Shivanirudh S G
Reg No: 185001146

Aim:

To display system date and time in 8086.

System date

Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move value 2AH to AH register.
- Move offset of day into SI register, and move value of DL register into SI.
- Move offset of month into SI register, and move value of DH register into SI.
- Move offset of year into SI register, and move value of CX register into SI.

Program:

Program	Comments
assume cs:code, ds:data	Declare code and data segments
data segment	Start of data segment
day db 01 dup(?)	Define byte day for day of month
month db 01 dup(?)	Define byte month for month
year db 02 dup(?)	Define word year for year
data ends	End of data segment
code segment	Start of code segment
start: mov ax, data	Move data to AX register
mov ds, ax	Move contents of AX register to DS register
mov ah, 2ah	Move value 2AH into AH register
int 21h	Request interrupt 21H to capture system date
mov si, offset day	Move offset of day into SI register
mov [si], dl	Move value in DL to address at [SI]
mov si, offset month	Move offset of month into SI register
mov [si], dh	Move value in DH to address at [SI]
mov si, offset year	Move offset of year into SI register
mov [si], cx	Move value in CX to address at [SI]
mov ah, 4ch	To request interrupt
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Unassembled code:

```
D:\>debug Date.exe
-u
0E25:0000 B8240E      MOV     AX,0E24
0E25:0003 8ED8        MOV     DS,AX
0E25:0005 B42A        MOV     AH,2A
0E25:0007 CD21        INT     21
0E25:0009 BE0000      MOV     SI,0000
0E25:000C 8814        MOV     [SI],DL
0E25:000E BE0100      MOV     SI,0001
0E25:0011 8834        MOV     [SI],DH
0E25:0013 BE0200      MOV     SI,0002
0E25:0016 890C        MOV     [SI],CX
0E25:0018 B44C        MOV     AH,4C
0E25:001A CD21        INT     21
0E25:001C 8BEC        MOV     BP,SP
0E25:001E 83EC1A      SUB     SP,+1A
-
```

Input and Output:

```
-d 0e24:0000
0E24:0000 10 0A E4 07 00 00 00 00-00 00 00 00 00 00 00 00 .....
0E24:0010 B8 24 0E 8E D8 B4 2A CD-21 BE 00 00 88 14 BE 01 .$.***.!.
0E24:0020 00 88 34 BE 02 00 89 0C-B4 4C CD 21 8B EC 83 EC ..4.....L.!
0E24:0030 1A 8A 1E B6 2C B7 00 8A-87 B8 2C 3A 46 0A 75 18 .....:F.u.
0E24:0040 D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8 .....:F.u..F..
0E24:0050 73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00 s.....v..v...
0E24:0060 50 EB A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00 P...F...~..u...
0E24:0070 8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C .^.....-F.....
-
```

Figure 1: **Output: 16 10 2020 in hex: 10 0A 07E4**

System time

Algorithm:

- Move the data segment to the AX register and then move it to the DS register.
- Move value 2CH to AH register.
- Move offset of hour into SI register, and move value of CH register into SI.
- Move offset of min into SI register, and move value of CL register into SI.
- Move offset of sec into SI register, and move value of DH register into SI.

Program:

Program	Comments
assume cs:code, ds:data	Declare code and data segments
data segment	Start of data segment
hour db 01 dup(?)	Define byte hour for hour of day
min db 01 dup(?)	Define byte min for minutes
sec db 01 dup(?)	Define word sec for seconds
data ends	End of data segment
code segment	Start of code segment
start: mov ax, data	Move data to AX register
mov ds, ax	Move contents of AX register to DS register
mov ah, 2ch	Move value 2CH into AH register
int 21h	Request interrupt 21H to capture system date
mov si, offset hour	Move offset of hour into SI register
mov [si], ch	Move value in CH to address at [SI]
mov si, offset min	Move offset of min into SI register
mov [si], cl	Move value in CL to address at [SI]
mov si, offset sec	Move offset of sec into SI register
mov [si], dh	Move value in DH to address at [SI]
mov ah, 4ch	To request interrupt
int 21h	Request interrupt routine
code ends	End of code segment
end start	

Unassembled code:

```
D:\>debug Time.exe
-u
0E25:0000 B8240E      MOV     AX,0E24
0E25:0003 8ED8        MOV     DS,AX
0E25:0005 B42C        MOV     AH,2C
0E25:0007 CD21        INT     21
0E25:0009 BE0000     MOV     SI,0000
0E25:000C 882C        MOV     [SI],CH
0E25:000E BE0100     MOV     SI,0001
0E25:0011 880C        MOV     [SI],CL
0E25:0013 BE0200     MOV     SI,0002
0E25:0016 8834        MOV     [SI],DH
0E25:0018 B44C        MOV     AH,4C
0E25:001A CD21        INT     21
0E25:001C 8BEC        MOV     BP,SP
0E25:001E 83EC1A     SUB     SP,+1A
-_-
```

Input and Output:

```
-d 0e24:0000
0E24:0000 0F 02 20 00 00 00 00 00-00 00 00 00 00 00 00 00  ..
0E24:0010 B8 24 0E 8E D8 B4 2C CD-21 BE 00 00 88 2C BE 01  .$.....!.....
0E24:0020 00 88 0C BE 02 00 88 34-B4 4C CD 21 8B EC 83 EC  .....4.L!....
0E24:0030 1A 8A 1E B6 2C B7 00 8A-87 B8 2C 3A 46 0A 75 18  .....:F.u.
0E24:0040 D1 E3 8B 87 FC 13 3B 46-08 75 0D 8A 46 06 D0 D8  .....:F.u..F...
0E24:0050 73 03 E9 B8 02 E9 C0 02-FF 76 0A FF 76 08 B0 00  s.....v.v....
0E24:0060 50 EB A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00  P...F...u....
0E24:0070 8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C  .^.....-F.....
-_-
```

Figure 2: **Output: 15 02 32 in hex: 0F 02 20**

Result:

The 8086 programs were written to display system date and time, and the results observed.