

MICROPROCESSOR LABORATORY

ASSIGNMENT NO. 6

Name :- Ojus Pravin Jaiswal

Roll No. :- SACO19108

Division :- A

Assignment No. 6

Program :

```
%macro disp 2
```

```
    mov rax,1
```

```
    mov rdi,1
```

```
    mov rsi,%1
```

```
    mov rdx,%2
```

```
    syscall
```

```
%endmacro
```

```
section .data
```

```
    rmodemsg db 10,"Processor is in Real Mode"
```

```
    rmsg_len:equ $-rmodemsg
```

```
    pmodemsg db 10,"Processor is in Protected Mode"
```

```
    pmsg_len:equ $-pmodemsg
```

```
    gdtmsg db 10,"GDT Contents are :: "
```

```
    gmsg_len:equ $-gdtmsg
```

```
    ldtmsg db 10,"LDT Contents are :: "
```

```
    lmsg_len:equ $-ldtmsg
```

```
    idtmsg db 10,"IDT Contents are :: "
```

```
    imsg_len:equ $-idtmsg
```

```
trmsg db 10,"Task Register Contents are :: "
```

```
tmsg_len:equ $-trmsg
```

```
mmsg db 10,"Machine Status Word :: "
```

```
mmsg_len:equ $-mmsg
```

```
promsg db 10,"Processor Information :: "
```

```
promsg_len:equ $-promsg
```

```
colmsg db ':'
```

```
newline db 10
```

```
section .bss
```

```
gdt resd 1
```

```
resw 1
```

```
ldt resw 1
```

```
idt resd 1
```

```
resw 1
```

```
tr resw 1
```

```
cr0_data resd 1
```

```
dnum_buff resb 04
```

```
section .text
```

```
global _start:
```

```
_start:
```

```
smsw eax
mov [cr0_data],eax
bt eax,0
jc prmode
disp rmodemsg,rmsg_len
jmp nxt1
```

```
prmode:disp pmodemsg,pmsg_len
```

```
nxt1:sgdt[gdt]
```

```
    sldt[ldt]
```

```
    sidt[idt]
```

```
    str[tr]
```

```
disp gdtmsg,gmsg_len
```

```
mov bx,[gdt+4]
```

```
call disp_num
```

```
mov bx,[gdt+2]
```

```
call disp_num
```

```
disp colmsg,1
```

```
mov bx,[gdt]
```

```
call disp_num
```

```
disp ldtmsg,lmsg_len
```

```
mov bx,[ldt]
call disp_num
```

```
disp idtmsg,imsg_len
```

```
mov bx,[idt+4]
call disp_num
```

```
mov bx,[idt+2]
call disp_num
```

```
disp colmsg,1
```

```
mov bx,[idt]
call disp_num
```

```
disp trmsg,tmsg_len
```

```
mov bx,[tr]
call disp_num
```

```
disp mswmsg,mmsg_len
```

```
mov bx,[cr0_data+2]
call disp_num
```

```
mov bx,[cr0_data]
call disp_num
```

disp newline,1

disp promsg,promsg_len

mov eax,00h

call disp_num

cpuid

call disp_num

exit: mov eax,01

mov ebx,00

int 80h

disp_num:

mov esi,dnum_buff

mov ecx,04

up1:rol bx,4

mov dl,bl

and dl,0fh

add dl,30h

cmp dl,39h

jbe skip1

add dl,07h

skip1:mov [esi],dl

inc esi

loop up1

disp dnum_buff,4

ret

</> Code

≡ Input

>_ Output

▶ Run

📄 Save

```
1 %macro disp 2
2     mov rax,1
3     mov rdi,1
4     mov rsi,%1
5     mov rdx,%2
6     syscall
7 %endmacro
8
9 section .data
10
11     rmodemsg db 10,"Processor is in Real Mode"
12     rmsg_len:equ $-rmodemsg
13
14     pmodemsg db 10,"Processor is in Protected Mode"
15     pmsg_len:equ $-pmodemsg
16
17     gdtmsg db 10,"GDT Contents are :: "
18     gmsg_len:equ $-gdtmsg
19
20     ldtmsg db 10,"LDT Contents are :: "
```

</> Code

≡ Input

>_ Output

▶ Run

📄 Save

```
21     lmsg_len:equ $-ldtmsg
22
23     idtmsg db 10,"IDT Contents are :: "
24     imsg_len:equ $-idtmsg
25
26     trmsg db 10,"Task Register Contents are :: "
27     tmsg_len:equ $-trmsg
28
29     mswmsg db 10,"Machine Status Word :: "
30     mmsg_len:equ $-mswmsg
31
32     promsg db 10,"Processor Information :: "
33     promsg_len:equ $-promsg
34
35     colmsg db ':'
36
37     newline db 10
38
39 section .bss
40     gdt resd 1
```

</> Code ≡ Input >_ Output

▶ Run

📄 Save

```
41     resw 1
42     ldt resw 1
43     idt resd 1
44     resw 1
45     tr resw 1
46     cr0_data resd 1
47     dnum_buff resb 04
48
49 section .text
50 global _start:
51 _start:
52
53     smsw eax
54     mov [cr0_data],eax
55     bt eax,0
56     jc prmode
57     disp rmodemsg,rmsg_len
58     jmp nxt1
59
60 prmode:disp pmodemsg,pmsg_len
```

</> Code ≡ Input >_ Output

▶ Run

📄 Save

```
61
62 nxt1:sgdt[gdt]
63     sldt[ldt]
64     sidt[idt]
65     str[tr]
66
67     disp gdtmsg,gmsg_len
68
69     mov bx,[gdt+4]
70     call disp_num
71
72     mov bx,[gdt+2]
73     call disp_num
74
75     disp colmsg,1
76
77     mov bx,[gdt]
78     call disp_num
79
80     disp ldtmsg,lmsg_len
```

</> Code ≡ Input >_ Output

▶ Run

📄 Save

```
81     mov bx,[ldt]
82     call disp_num
83
84     disp idtmsg,imsg_len
85
86     mov bx,[idt+4]
87     call disp_num
88
89     mov bx,[idt+2]
90     call disp_num
91
92     disp colmsg,1
93
94     mov bx,[idt]
95     call disp_num
96
97     disp trmsg,tmsg_len
98
99     mov bx,[tr]
100    call disp_num
```


Code Input Output

Run

Save

```
101
102     disp mswmsg,mmsg_len
103
104     mov bx,[cr0_data+2]
105     call disp_num
106
107     mov bx,[cr0_data]
108     call disp_num
109
110     disp newline,1
111
112     disp promsg,promsg_len
113
114     mov eax,00h
115     call disp_num
116     cpuid
117     call disp_num
118
119 exit: mov eax,01
120       mov ebx,00
```

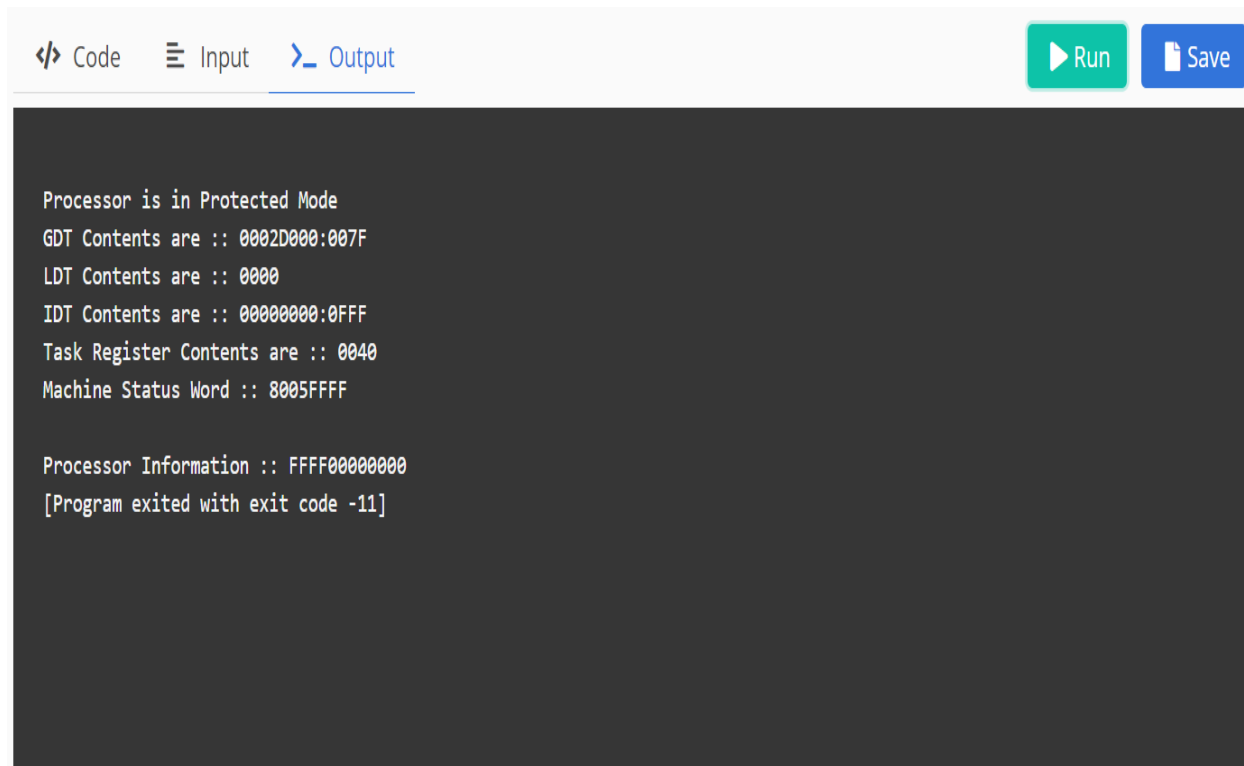
Code Input Output

Run

Save

```
118
119 exit: mov eax,01
120       mov ebx,00
121       int 80h
122
123 disp_num:
124     mov esi,dnum_buff
125     mov ecx,04
126 up1:rol bx,4
127     mov dl,bl
128     and dl,0fh
129     add dl,30h
130     cmp dl,39h
131     jbe skip1
132     add dl,07h
133 skip1:mov [esi],dl
134         inc esi
135         loop up1
136 disp dnum_buff,4
137 ret
```

Output :

A screenshot of a code editor interface. At the top, there is a header bar with three tabs: 'Code' (with a code icon), 'Input' (with a list icon), and 'Output' (with a document icon). The 'Output' tab is currently selected. To the right of the tabs are two buttons: a green 'Run' button with a play icon and a blue 'Save' button with a floppy disk icon. Below the header bar is a large dark gray area representing the output console. It contains several lines of white text showing system information and a final exit message.

```
Processor is in Protected Mode
GDT Contents are :: 0002D000:007F
LDT Contents are :: 0000
IDT Contents are :: 00000000:0FFF
Task Register Contents are :: 0040
Machine Status Word :: 8005FFFF

Processor Information :: FFFF00000000
[Program exited with exit code -11]
```