PARSHVANATH CHARITABLE TRUST'S
A.P. SHAH INSTITUTE OF TECHNOLOGY
Department of Computer Science and Engineering
Data Science

Department of Computer Science Engineering Data Science

Academic Year: 2022-23 Semester: IV
Class / Branch: S.E.D.S. **Subject: Microprocessor Lab**

Experiment No. 7

1. **Aim: Write an Assembly Language Program to find factorial of a number using procedure.**

2. **Software used:** tasm,tlink ,td,dosemu

3. **Theory :-**

A procedure is a set of code that can be branched to and returned from in such a way that the code is as if it were inserted at the point from which it is branched to. The branch toprocedure is referred to as the *call,* and the corresponding branch back is known as the *return*. The return is always made to the instruction immediately following the call regardlessof where the call is located.

The CALL instruction not only branches to the indicated address, but also pushes the return address onto the stack. The RET instruction simply pops the return address from the stack. The registers used by the procedure need to be stored before their contents are changed, and then restored just before their contents are changed, and then restored just before the procedure is excited.

Procedures are used in the source code by placing a statement of the form at the beginning of the procedure

Procedure name PROC Attribute and by terminating the procedure with a statement

Procedure name ENDP

The attribute that can be used will be either NEAR or FAR. If the attribute is NEAR, the RET instruction will only pop a word into the IP register, but if it is FAR, it will also pop a word into the CS register.

4. **Program:**

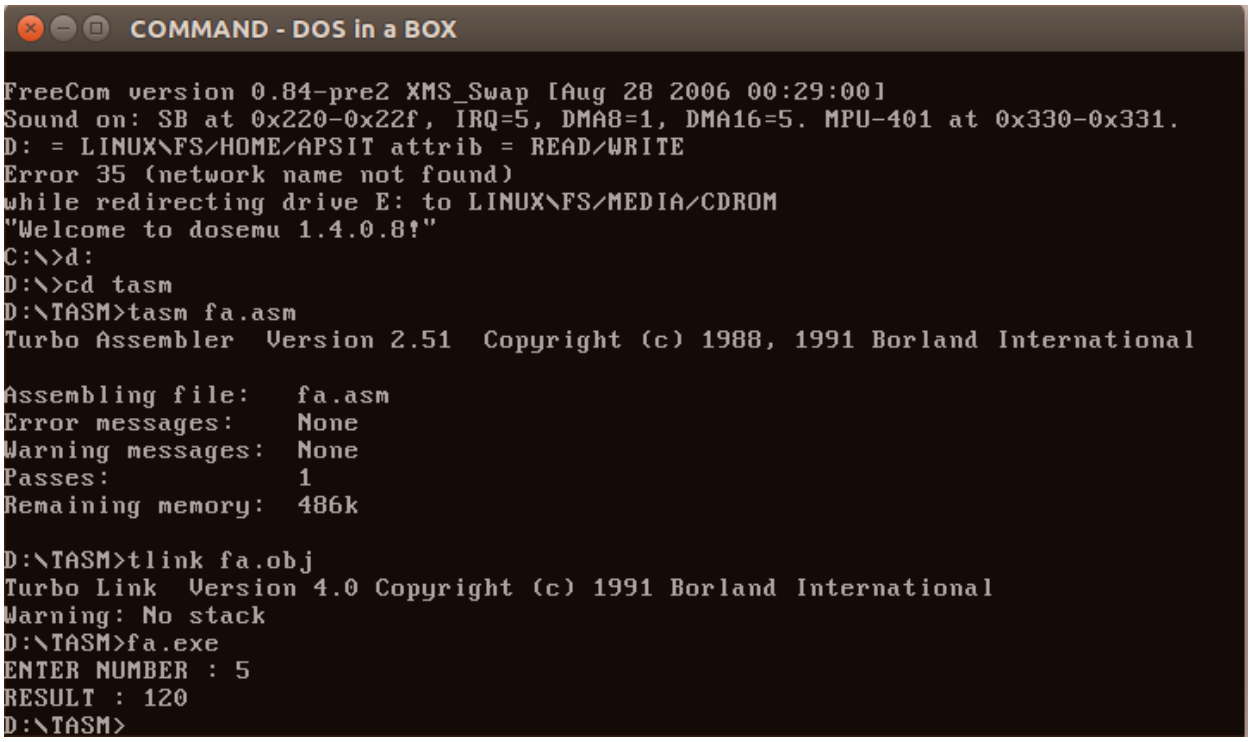
```
.model small
```

.data

```
num db ?
fact db 1h
res db 10 dup ('$')
msg1 db "enter number : $"
msg2 db 10,13,"result : $"
.code
start:
mov ax,@data
mov ds,ax
lea dx,msg1
mov ah,9
int  21h
mov ah,1
int 21h
call factorial
lea si,res
call hex2dec
lea dx,msg2
mov  ah,9
int 21h
lea dx,res
mov ah,9
int 21h
mov ah,4ch
int 21h
hex2dec proc near
mov cx,0
mov bx,10
loop1: mov dx,0
div bx
```

```
add dl,30h
push dx
inc  cx
cmp ax,9
jg loop1
add al,30h
mov [si],al
loop2: pop ax
inc si
mov [si],al
loop loop2
ret
hex2dec endp
factorial proc near
sub al,30h
mov num,al
mov  ah,0
mov al,fact
mov  ch,0
mov cl,num
label1: mul cx
loop  label1
ret
factorial endp
end start
```

Output:



5. Conclusion :-