

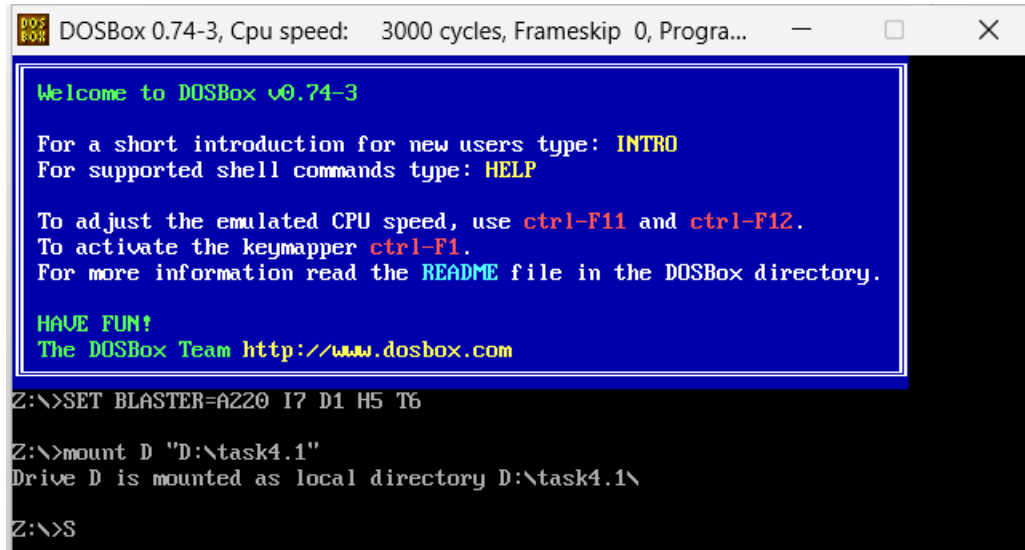
Creating Directories and Files in Assembly using DOS Interrupts

Submitted by:

Al Amin Hossain Nayem

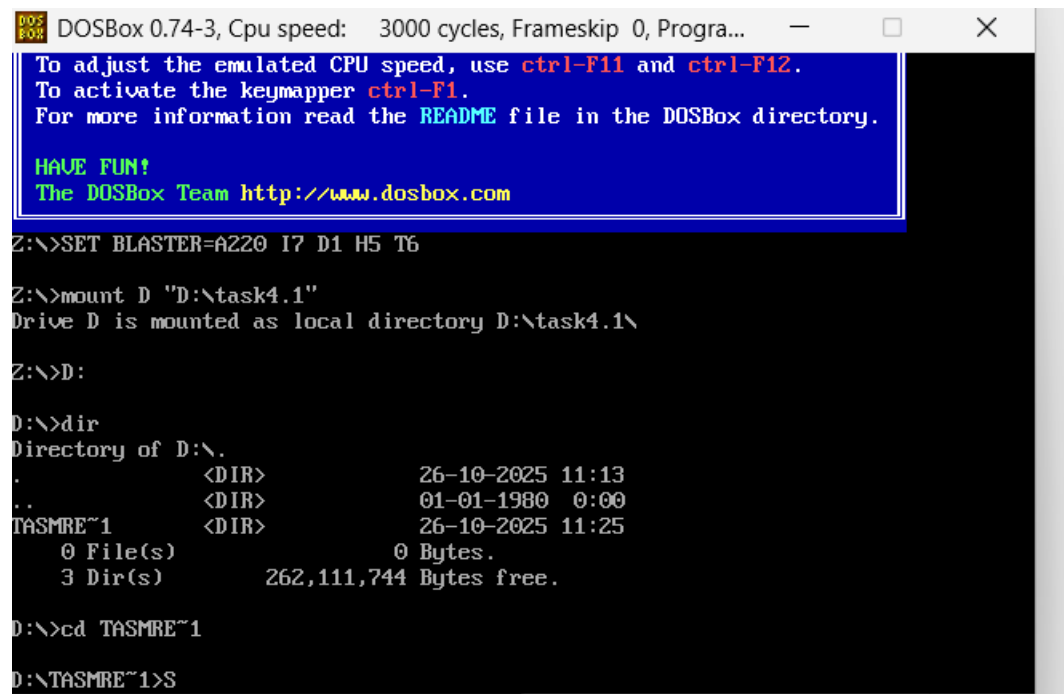
Course: Systemic Programming

1) mount it on dosbox



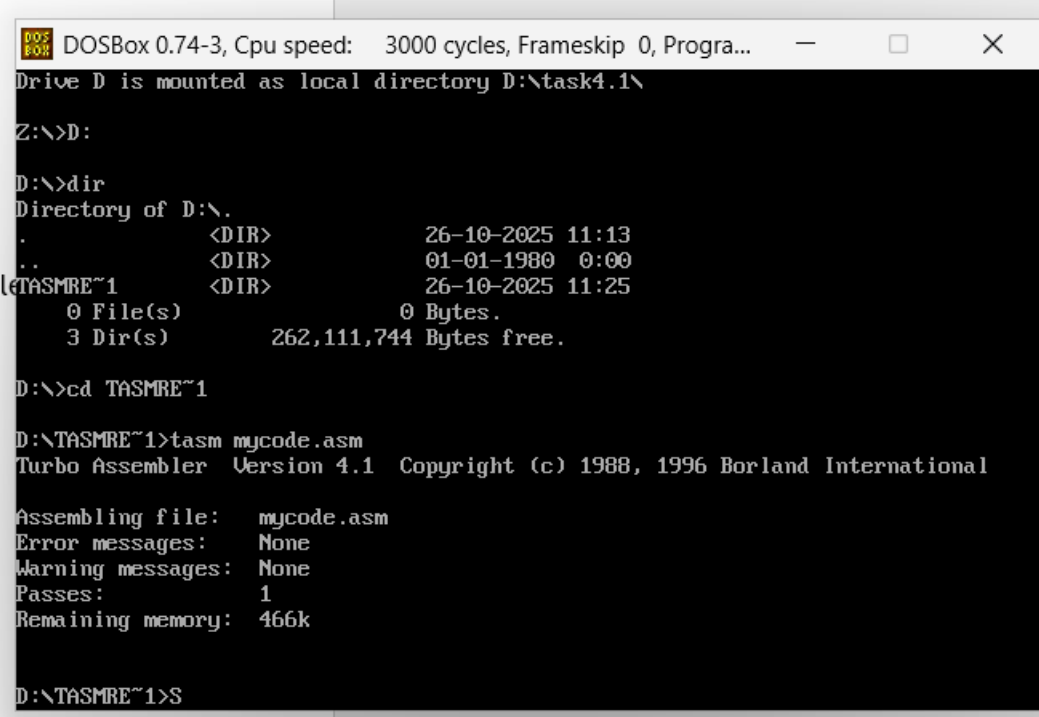
```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
Welcome to DOSBox v0.74-3
For a short introduction for new users type: INTRO
For supported shell commands type: HELP
To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.
HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6
Z:\>mount D "D:\task4.1"
Drive D is mounted as local directory D:\task4.1\
Z:\>S
```

2) go to mounted folder and open tasm file



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.
HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6
Z:\>mount D "D:\task4.1"
Drive D is mounted as local directory D:\task4.1\
Z:\>D:
D:\>dir
Directory of D:\.
.                <DIR>                26-10-2025  11:13
..               <DIR>                01-01-1980   0:00
TASMRE~1         <DIR>                26-10-2025  11:25
0 File(s)        0 Bytes.
3 Dir(s)         262,111,744 Bytes free.
D:\>cd TASMRE~1
D:\TASMRE~1>S
```

3) successfully assembled my assembly source file mycode.asm into an object file called



A screenshot of a DOSBox window titled "DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...". The terminal shows the following commands and output:

```
Z:\>D:
D:\>dir
Directory of D:\.
.                <DIR>                26-10-2025  11:13
..               <DIR>                01-01-1980   0:00
(TASMRE~1       <DIR>                26-10-2025  11:25
0 File(s)        0 Bytes.
3 Dir(s)         262,111,744 Bytes free.

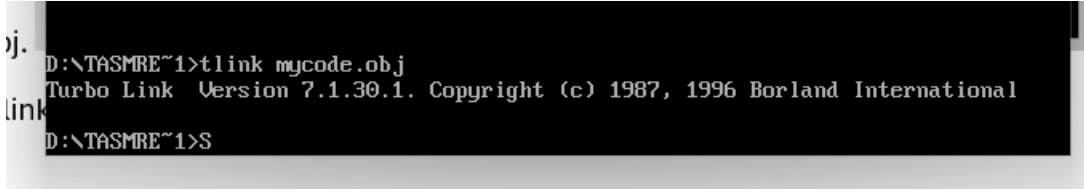
D:\>cd TASMRE~1
D:\TASMRE~1>tasm mycode.asm
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file:   mycode.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  466k

D:\TASMRE~1>S
```

mycode.obj.

4) need to link the object file to create an executable .EXE file




A screenshot of a DOSBox terminal showing the linking process:

```
D:\TASMRE~1>tlink mycode.obj
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International

D:\TASMRE~1>S
```

5)now I will run the executable file



A screenshot of a DOSBox terminal showing the execution of the program:

```
D:\TASMRE~1>tlink mycode.obj
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International

D:\TASMRE~1>mycode
Success! All directories and files created.

D:\TASMRE~1>S_
```

It will create folder name as AL as written in the asm code

1. Introduction

This report explains an Assembly language program written using Turbo Assembler (TASM) to create directories and text files using DOS interrupts. The program demonstrates how to use DOS interrupt 21h functions to perform file management operations such as creating directories, writing files, and displaying messages in DOS.

2. Objective

The objective of this program is to automate the creation of multiple directories and files and to write specific text content into those files using Assembly language.

3. Source Code

```
.model small
.stack 100h

.data
dir1 db 'Al', 0
dir2 db 'Al\Amin', 0
dir3 db 'Al\Amin\Hossain', 0

file1 db 'Al\info.txt', 0
file2 db 'Al\Amin\info.txt', 0
file3 db 'Al\Amin\Hossain\info.txt', 0

content db 'Al Amin Hossain Nayem', 13, 10
content_len equ $ - content

msg_success db 'Success! All directories and files created.', 13, 10, '$'
msg_error db 'Error occurred!', 13, 10, '$'

fileHandle dw ?

.code
main PROC
    mov ax, @data
    mov ds, ax

    lea dx, dir1
    mov ah, 39h
    int 21h

    lea dx, dir2
```

```
mov ah, 39h  
int 21h
```

```
lea dx, dir3  
mov ah, 39h  
int 21h
```

```
lea dx, file1  
call create_write_file
```

```
lea dx, file2  
call create_write_file
```

```
lea dx, file3  
call create_write_file
```

```
lea dx, msg_success  
mov ah, 9  
int 21h
```

```
mov ax, 4C00h  
int 21h  
main ENDP
```

```
create_write_file PROC
```

```
push ax  
push bx  
push cx
```

```
mov si, dx  
mov ah, 3Ch  
mov cx, 0  
int 21h  
jc create_error
```

```
mov fileHandle, ax  
mov bx, ax
```

```

    mov ah, 40h
    mov cx, content_len
    lea dx, content
    int 21h
    jc write_error

    mov ah, 3Eh
    mov bx, fileHandle
    int 21h

    jmp file_done

create_error:
write_error:
file_done:
    pop cx
    pop bx
    pop ax
    ret
create_write_file ENDP

end main

```

4. Explanation of Code

The program starts with a small memory model setup and stack initialization. It defines directory and file paths as data strings and uses DOS interrupt 21h to create directories and files. The content 'Al Amin Hossain Nayem' is written to each file using function 40h. After successful execution, a message is displayed using function 09h, and the program terminates cleanly with function 4Ch.

5. Output

When the program is executed, it creates the following directory structure:

Al\

```

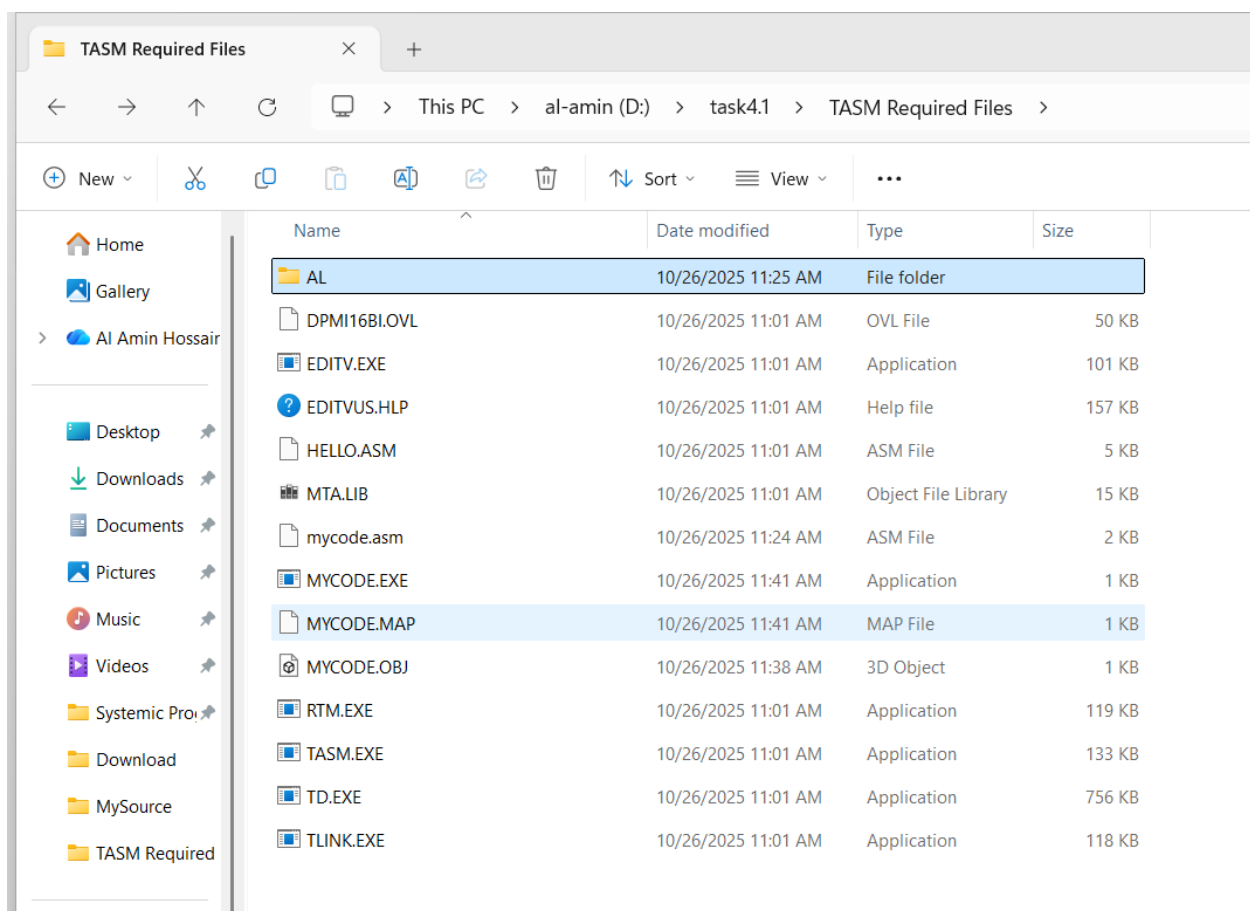
└─ info.txt
└─ Amin\
    └─ info.txt
        └─ Hossain\
            └─ info.txt

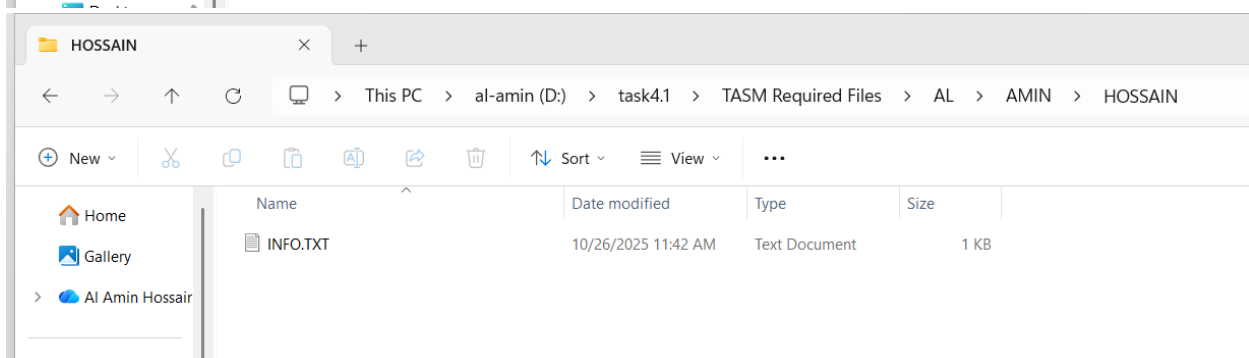
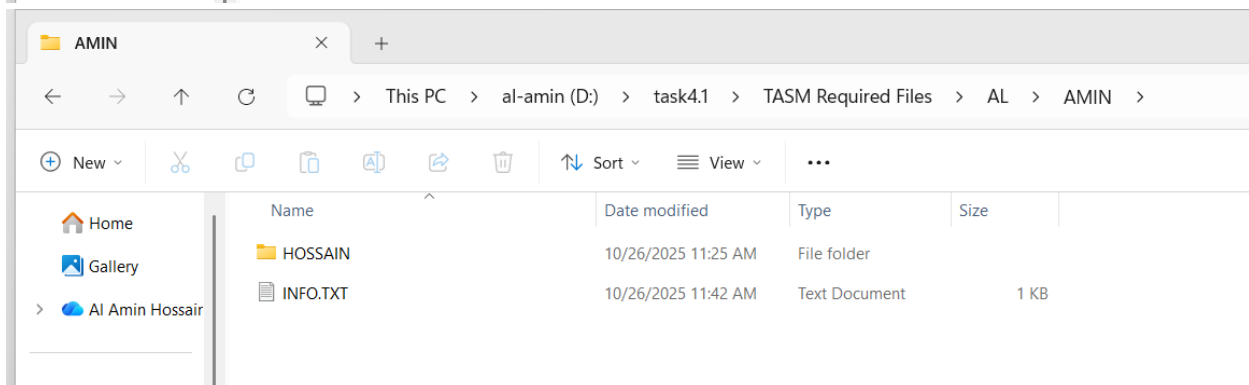
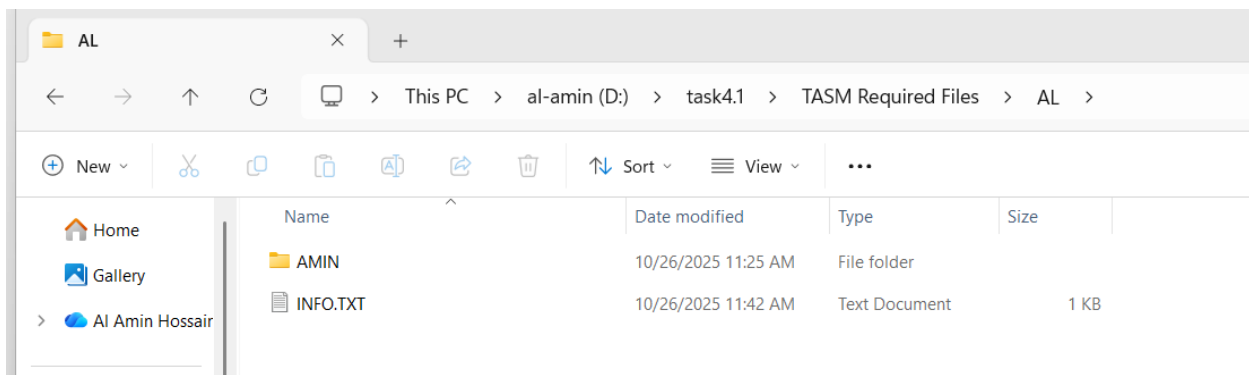
```

Each file contains the text: 'Al Amin Hossain Nayem'

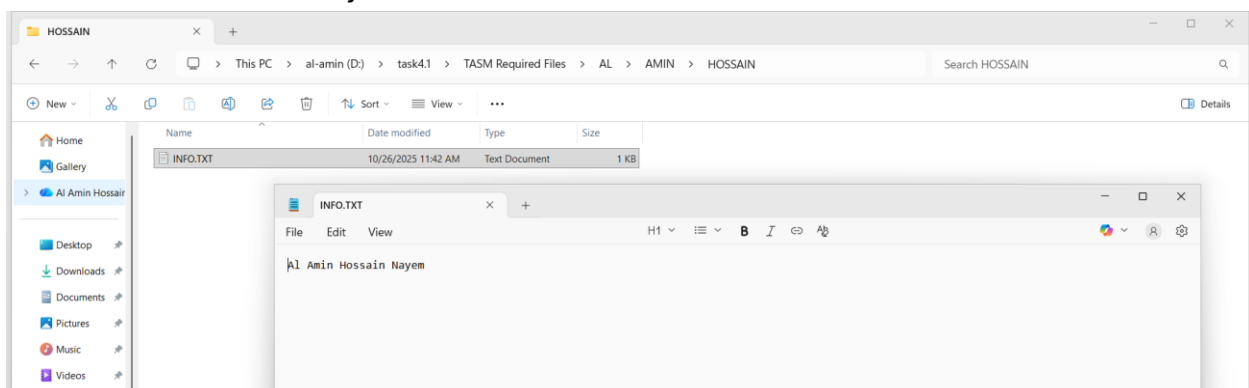
The output displayed on the screen is:

Success! All directories and files created.





and all the txt file have my name on it



6. Conclusion

This program demonstrates the use of DOS interrupts for directory and file management in Assembly language. It provides a clear understanding of how low-level DOS functions can perform high-level file system operations. The successful creation of directories and files confirms that the interrupt calls were used correctly.