MICROPROCESSOR ASSIGNMENTS

NAME : SHRIRANG. R. MHALGI

CLASS : S.E.

DIVISION : B

ROLL NO 222006

PROBLEM STATEMENT :

Write X86 menu driven Assembly Language Program (ALP) to implement OS (DOS) commands TYPE, COPY and DELETE using file operations. User is supposed to provide command line arguments in all cases.

CODE :

;write a ALP menu driven program to implement following dos commands

; 1. Type

; 2. Copy

; 3. Delete

;using file operations

;----------MACROS----------

%macro scall 4

mov rax, %1

mov rdi, %2

mov rsi, %3

mov rdx, %4

syscall

%endmacro

;----------SECTION.BSS----------

section .bss

choice resb 2

filename resb 50

file1 resb 20

file2 resb 20

filehandler resq 1

buffer resb 1024

buffer\_len equ $-buffer

length resq 1

;----------SECTION.DATA----------

section .data

menu\_msg db 10, "----------WELCOME----------"

db 10, "1 : TYPE "

db 10, "2 : COPY "

db 10, "3 : DELETE "

db 10, "0 : EXIT "

db 10, ">> "

menu\_msg\_len equ $-menu\_msg

file\_name\_msg db 10, "ENTER FILE NAME >> "

file\_name\_msg\_len equ $-file\_name\_msg

src\_des\_msg db 10, "ENTER SOURCE AND DESTINATION FILE SEPARATED BY A SPACE >> "

src\_des\_msg\_len equ $-src\_des\_msg

err\_msg db 10, "-----ERROR IN FILE-----", 10

err\_msg\_len equ $-err\_msg

del\_msg db 10, "FILE SUCCESSFULLY DELETED ", 10

del\_msg\_len equ $-del\_msg

cpy\_msg db 10, "FILE SUCCESSFULLY COPIED"

cpy\_msgl equ $-cpy\_msg

;----------SECTION.TXT----------

section .txt

global \_start

\_start :

scall 1, 1, menu\_msg, menu\_msg\_len

scall 0, 0, choice, 2

cmp byte[choice], 31h

je case1

cmp byte[choice], 32h

je case2

cmp byte[choice], 33h

je case3

cmp byte[choice], 30h

je case0

case1 : call type

jmp exit\_proc

case2 : call copy

call exit\_proc

case3 : call delete

call exit\_proc

case0 : call exit\_proc

;----------EXIT\_PROC---------

exit\_proc :

mov rax, 60

mov rdi, 0

syscall

;----------DISPLAY\_PROC----------

;----------TYPE----------

type :

scall 1, 1, file\_name\_msg, file\_name\_msg\_len

scall 0, 0, filename, 50

dec rax

mov byte[filename + rax], 0

scall 2, filename, 2, 0777o

cmp rax, -1

jle error

mov [filehandler], rax

scall 0, [filehandler], buffer, 1024

mov [length], rax

scall 1, 1, buffer, length

scall 3, [filehandler], 0, 0

ret

;----------COPY----------

copy :

scall 1, 1, src\_des\_msg, src\_des\_msg\_len

scall 0, 0, filename, 50

dec rax

mov byte[filename + rax], 0

call file\_n

scall 2, file1, 2, 0777o

cmp rax, -1h

jle error

mov [filehandler], rax

scall 0, [filehandler], buffer, 1024

mov [length], rax

scall 3, filehandler, 0, 0

scall 2, file2, 2, 0777o

cmp rax, -1h

jle error

mov [filehandler], rax

scall 1, [filehandler], buffer, [length]

scall 3, filehandler, 0, 0

scall 1, 1, cpy\_msg, cpy\_msgl

ret

;----------DELETE-----------

delete :

scall 1, 1, file\_name\_msg, file\_name\_msg\_len

scall 0, 0, filename, 50

dec rax

mov byte[filename + rax], 0

scall 2, filename, 2, 0777o

cmp rax, -1

jle error

mov rax, 87

mov rdi, filename

syscall

scall 1, 1, del\_msg, del\_msg\_len

ret

;---------ERROR----------

error : scall 1, 1, err\_msg, err\_msg\_len

jmp exit\_proc

;---------FILE\_N---------

file\_n :

mov rsi, filename

mov rdi, file1

bck : mov al, [rsi]

cmp al, 20h

je nex

mov [rdi], al

inc rsi

inc rdi

jmp bck

nex : mov byte[rdi], 0

mov rdi, file2

inc rsi

bck1 : mov al, [rsi]

cmp al, 0h

je next5

mov [rdi], al

inc rsi

inc rdi

jmp bck1

next5 : mov byte[rdi], 0

ret

OUTPUT :

