extern printf

segment .data

;;sign in/sign out

;;make a menu

;;sign in = type first name, type last name, time 12:32, export to file

;;sign out = type first name type last name, time 12:42, edit file

;;calculate = type name, locate them sign in/out, subtraction, multiply by hourly rate

section .data

prompt1: db "Please [FirstName] [LastName] [Hour:Minute] ",10

len1: equ $-prompt1

welcome: db "Welcome to Project Timesheet",10

db " [0] Sign-in ",10

db " [1] Sign-Out ",10

db " [2] Calculate Pay ",10

db " [3] Quit ",10

lenMenu: equ $ - welcome

option: times 4 db 0

fname: time 256 db 0

pathname: dd "/afs/umbc.edu/users/h/j/hj69411/home/cmpe310/proj4/fix.txt";output file

result\_format db "Payout: %d", 10, 0

section .bss

num1: resb 4

num2: resb 4

section .text

global main

main:

push ebp ; set up stack frame

mov ebp,esp

mov eax,4 ; use the write function

mov ebx,1 ; use the standard output

mov edx,lenMenu ; set the number of characters to print

mov ecx,welcome ; string to print

int 0x80 ; call kernel to print the message

mov eax,3 ; use the read function

mov ebx,0 ; use the standard input

mov ecx,option ; load in string1

mov edx,4 ; read at most 255 characters

int 0x80 ; call kernel to read the string

mov BYTE [option + eax -1], 0 ; insert string terminator

mov eax,4 ; use the write function

mov ebx,1 ; use the standard output

mov edx,len1 ; set the number of characters to print

mov ecx,prompt1 ; string to print

int 0x80 ; call kernel to print the message

mov eax,3 ; use the read function

mov ebx,0 ; use the standard input

mov ecx,fname ; load in string1

mov edx,256 ; read at most 255 characters

int 0x80 ; call kernel to read the string

mov BYTE [fname + eax -1], 0 ; insert string terminator

;This is part is all checking part which switches the block according to our <operand> + - / \*

cmp byte[option], 0x30 ;0

je Sign

cmp byte[option], 0x31 ;1

je Sign

cmp byte[option], 0x32 ;2

je Calc

cmp byte[option], 0x33 ;3

je exit

jmp exit ;;if none of the options picked exit

Sign:

mov eax,5

mov ebx, pathname

mov ecx, 101o

mov edx, 700o

int 80h

mov ebx, eax

mov eax,4

mov ecx, fname

mov edx,15

int 80h

Calc:

mov dword [num1], 90;; time one

mov dword [num2], \_\_float32\_\_(10.00); time two

sub esp, 8

fld dword [num2]

fisub dword [num1]

fstp qword [esp]

; Multiply num1 by num2

mov eax, dword [num1]

imul dword [num2]

mov dword [result], eax ; Store the result

mov ecx, [result]

; Print the result

push ecx; put hamming distance in stack

push result\_format; put format string in stack

call printf; call printf to print result

add esp, 12

exit:

mov esp,ebp ; takedown stack frame

pop ebp

mov eax,0 ; normal, no error, return value

ret ; return