Alexander Tracy

3/8/12

Lab 4 Prelab

1)    Read file  
file        DB    ‘Lab4.bin’, ‘$’        ; in data\_seg  
input        DB    FFh, 255    dup(‘$’)  
  
; open file, get file handle  
lea dx, file  
mov ah, 3dh  
mov al, 0  
int 21h  
jc error  
mov bx, ax    ; file handle now in bx  
  
; now that we have file handle, read file into input  
lea dx, input  
mov cx, FFh  
int 21h  
jc error  
  
2)    Pseudocode  
  
read proc near  
    ; opens the file, reads it to input  
  
    ; open file, get file handle

lea dx, file

mov ah, 3dh

mov al, 0

int 21h

jc error

mov bx, ax    ; file handle now in bx

; now that we have file handle, read file into input

lea dx, input

mov cx, FFh

int 21h

jc error

closeFile

ret

read endp  
display proc near  
    loopa:  
        call nextItem  
        printS item  
        cmp ax, 0    ;  ax will contain the address of the next item in the list  
                ;     if 0, then the list is finished  
        je loopa  
        jmp done  
    done:  
        ret  
display endp  
  
error proc near  
    ; returns error message based on error type  
    ; al will contain error code  
      
    cmp al, 1  
    je error1  
    cmp al, 2  
    je error2  
    jmp errorUnknown  
      
      
    error1:  
        printS errorOpeningFile  
        ret  
    error2:  
        printS errorReadingFile  
        ret  
    error:  
        printS errorUnknown  
        ret  
error endp  
  
addScore proc near  
    ; adds variable to end of array  
    loopa:  
        call nextItem  
        cmp ax, 0  
        jne loopa  
    ; now the last item of array is loaded  
    mov cx, [bx+2]     ; movs the length of last item to cx  
    add cx, bx        ; adds the length of the last item to the address of last item  
                ; to get address of where new last item will be stored  
    mov [bx], cx        ; make first byte of current item point to where new  
                ; last item will be stored  
    mov bx, cx  
    loopb:  
        ; loops through string to be added  
        mov [bx], ax  
        inc bx  
        loop loopb  
addScore endp  
  
sort proc near  
    ; sorts the list  
      
    loopa:  
        mov swapped, 0  
        loopb:  
            cmp ax, bx  
            jb swap  
            inc ax  
            inc bx  
            loop loopb  
        swap:  
            mov swapped, 1  
            call swapItems  
            jmp loopb  
        cmp swapped 0  
        jne loopa  
      
sort endp  
  
save proc near  
    ; save the list to file  
    openFile  
    loopa:  
        call nextItem  
        writetoFile  
        cmp cl, 0  
        je loopa  
    closeFile  
save endp