sys\_exit        equ     1   
sys\_read        equ     3   
sys\_write       equ     4   
sys\_open                equ             5   
sys\_close               equ             6   
sys\_brk                 equ             45   
sys\_newstat             equ             106   
  
O\_RDONLY                equ             0   
O\_WRONLY                equ             1   
O\_RDWR                  equ             2   
  
stdin           equ     0   
stdout          equ     1   
stderr          equ     2   
  
struc STAT           
   .st\_dev:        resd 1          
   .st\_ino:        resd 1       
   .st\_mode:       resw 1       
   .st\_nlink:      resw 1       
   .st\_uid:        resw 1       
   .st\_gid:        resw 1       
   .st\_rdev:       resd 1           
   .st\_size:       resd 1       
   .st\_blksize:    resd 1       
   .st\_blocks:     resd 1       
   .st\_atime:      resd 1       
   .st\_atime\_nsec: resd 1       
   .st\_mtime:      resd 1       
   .st\_mtime\_nsec: resd 1   
   .st\_ctime:      resd 1       
   .st\_ctime\_nsec: resd 1       
   .unused4:       resd 1       
   .unused5:       resd 1       
endstruc   
  
%define sizeof(x) x %+ \_size   
  
%macro write\_string 2   
   mov   eax, sys\_write   
   mov   ebx, stdout   
   mov   ecx, %1   
   mov   edx, %2   
   int   80h   
%endmacro

SECTION     .data   
       szFile          db      "TEST", 0   
       File\_Len    equ     $-szFile   
  
       fSizeMsg        db              "Filesize = "   
       fSizeMsgLen     equ             $-fSizeMsg   
  
       nlMsg           db              0xa   
       nlMsgLen        equ             $-nlMsg   
  
       OBMsg           db              "Original Break = "   
       OBMsgLen        equ             $-OBMsg   
  
       NBMsg           db              "New Break = "   
       NBMsgLen        equ             $-NBMsg   
  
       diffMsg         db              "Difference = "   
       diffMsgLen      equ             $-diffMsg   
  
       fileContentsMsg         db      "File Contents:", 0xa   
       fileContentsMsgLen      equ     $-fileContentsMsg   
  
       hereMsg         db              "Here!", 0xa   
       hereMsgLen      equ             $-hereMsg   
  
  
SECTION     .bss   
       stat            resb    sizeof(STAT)   
       Org\_Break   resd    1   
       TempBuf         resd    1   
       quotient        resb    1   
       remainder       resb    1   
       numbx           resb    1   
       numax           resb    1   
  
SECTION     .text   
       global      \_start   
      
\_start:  
  
      ;~ Get file size   
       mov             ebx, szFile   
       mov             ecx, stat   
       mov             eax, sys\_newstat   
       int             80H   
  
       ;~ Get end of bss section   
       xor             ebx, ebx   
       mov             eax, sys\_brk   
       int             80H   
       mov             [Org\_Break], eax   
       mov             [TempBuf], eax   
       push    eax   
  
       ; extend it by file size   
       pop             ebx   
       add             ebx, dword [stat + STAT.st\_size]   
       mov             eax, sys\_brk   
       int             80H   
  
       ; test buffer size    
       sub eax, [Org\_Break]   
    mov byte [numax], al   
       mov byte [numbx], bl   
       xor eax, eax   
       xor ebx, ebx   
       mov al, [numbx]   
       mov bl, 10   
       idiv bl   
       add al, '0'   
       add ah, '0'   
       mov [quotient], al   
       mov [remainder], ah   
       write\_string fSizeMsg, fSizeMsgLen   
       write\_string quotient, 1   
       write\_string remainder, 1   
       write\_string nlMsg, nlMsgLen   
  
       xor eax, eax   
    xor ebx, ebx   
   mov al, [numax]   
   mov bl, 10   
    idiv bl   
    add al, '0'   
    add ah, '0'   
    mov [quotient], al   
    mov [remainder], ah   
  
       write\_string diffMsg, diffMsgLen   
       write\_string quotient, 1   
    write\_string remainder, 1   
       write\_string nlMsg, nlMsgLen   
  
openFile:   
       ;~ open file   
       mov             ebx, szFile   
       mov             ecx, O\_RDONLY   
       xor             edx, edx   
       mov             eax, sys\_open   
       int             80H   
    xchg    eax, esi   
  
       ; print message   
       write\_string fileContentsMsg, fileContentsMsgLen   
  
       ;~ read in file to buffer   
       mov     ebx, esi   
       mov             ecx, [TempBuf]   
       mov             edx, dword [stat + STAT.st\_size]   
       mov             eax, sys\_read   
       int             80H   
  
       ;~ display to terminal   
    write\_string [TempBuf], edx   
       ;mov            ebx, stdout   
       ;mov            ecx, [TempBuf]   
       ;mov            edx, eax   
       ;mov            eax, sys\_write   
       ;int            80H   
  
       ;~ close file   
       mov             ebx, esi    
       mov             eax, sys\_close   
       int             80H   
  
       ;~ "free" memory   
       mov     ebx, [Org\_Break]   
    mov     eax, sys\_brk   
    int     80H   
     
Exit:     
   mov     eax, sys\_exit   
   xor     ebx, ebx   
   int     80H