Лабораторная работа №12

***Работа с файлами***

Выполнил ст. гр. ІО-83, Гулевский Денис

## Варіанти виконання роботи: 306

|  |  |  |  |
| --- | --- | --- | --- |
| 306 | С длинными именами | Написать программу для слияния 3 текстовых файлов в четвёртый. Т.е. записать в 4-ый файл сначала содержимое первого, потом содержимое 2-ого, потом – 3-его. 4 названия вводить с клавиатуры. | Написать программу, удаляющую из текстового файла все скобки «)». |

.386 ;306

ASSUME CS:CODE, DS:DATA, SS:MYSTACK

DATA SEGMENT USE16

String1 db 'Task 1',0dh,0ah

endString1 equ $

String2 db 'Task 2',0dh,0ah

endString2 equ $

buffer db 64 dup(?)

bx1 dw ?

bx2 dw ?

bx3 dw ?

bx4 dw ?

file1 db 255 dup (?)

file2 db 255 dup (?)

request1 db 'Input file1: '

endRequest1 equ $

request2 db 'Input file2: '

endRequest2 equ $

request3 db 'Input file3: '

endRequest3 equ $

request4 db 'Input file4: '

endRequest4 equ $

err1 db 'Uncorrect number of function',0dh,0ah

err2 db 'File not found',0dh,0ah

err3 db 'Path not found',0dh,0ah

err4 db 'Too many opened files',0dh,0ah

err5 db 'Access denied',0dh,0ah

err6 db 'Uncorrect descriptor',0dh,0ah

err7 db 'Blocks of memory are damaged',0dh,0ah

err8 db 'Not enough memory',0dh,0ah

err9 db 'Uncorrect address of block of memory',0dh,0ah

err10 db 'Unknown mistake',0dh,0ah

endd db ?

errtable dw Err1, Err2, Err3, Err4, Err5, Err6, Err7, Err8, Err9, Err10, Endd

DATA ENDS

MYSTACK SEGMENT USE16 STACK

db 256 DUP(?)

MYSTACK ENDS

CODE SEGMENT USE16

beg:

mov ax, DATA

mov ds, ax

;===========TASK 1=============================

lea dx, String1

mov ah, 40h ;вывод на экран сообщения

mov bx, 1

mov cx, endString1-String1

int 21h

;========file4=================

mov ah, 40h

lea dx, request4

mov cx, endRequest4-request4

int 21h

mov bx, 0

mov cx, 255

lea dx, file1

mov ah, 3Fh ;ввод с клавиатуры

int 21h

mov bx, dx

add bx, ax

mov byte ptr [bx-2], 0

mov ax, 716Ch ;открыть файл с длинным именем

mov bx, 2

mov dx, 1

lea si, file1

int 21h

jc inputErr

mov bx4, ax

;============file1==============

mov ah, 40h

lea dx, request1

mov cx, endRequest1-request1

int 21h

mov bx, 0

mov cx, 255

lea dx, file2

mov ah, 3Fh

int 21h

mov bx, dx

add bx, ax

mov byte ptr [bx-2], 0

mov ax, 716Ch

mov bx, 2

mov dx, 1

lea si, file2

int 21h

jc inputErr

mov bx1, ax

;=============file2===========

mov ah, 40h

lea dx, request2

mov cx, endRequest2-request2

int 21h

mov bx, 0

mov cx, 255

lea dx, file2

mov ah, 3Fh

int 21h

mov bx, dx

add bx, ax

mov byte ptr [bx-2], 0

mov ax, 716Ch

mov bx, 2

mov dx, 1

lea si, file2

int 21h

jc inputErr

mov bx2, ax

;=============file3==========

mov ah, 40h

lea dx, request3

mov cx, endRequest3-request3

int 21h

mov bx, 0

mov cx, 255

lea dx, file2

mov ah, 3Fh

int 21h

mov bx, dx

add bx, ax

mov byte ptr [bx-2], 0

mov ax, 716Ch

mov bx, 2

mov dx, 1

lea si, file2

int 21h

jc inputErr

mov bx3, ax

call read

mov ax, bx2

mov bx1, ax

call read

mov ax, bx3

mov bx1, ax

call read

jmp task2

read PROC NEAR

reading1:

mov bx, bx1

mov cx, 64

lea dx, buffer

mov ah, 3fh

int 21h

mov bx, bx4

mov cx, ax

mov ah, 40h

int 21h

mov bx, 64

cmp bx, ax

je reading1

RET

read ENDP

;========ERROR Handling==========

inputErr:

cmp al, 9

jbe known

mov al,10

known:

xor ah, ah

push ax

shl ax, 1

lea bx, errtable-2

add bx, ax

mov dx, [bx]

pop ax

inc ax

shl ax, 1

lea bx, errtable-2

add bx, ax

mov si, [bx]

sub si, dx

mov cx, si

mov bx, 1

mov ah, 40h

int 21h

jmp end\_prog

;===========TASK 2=============================

task2:

lea dx, String2

mov ah, 40h

mov bx, 1

mov cx, endString2-String2

int 21h

mov ah, 40h

lea dx, request1

mov cx, endRequest1-request1

int 21h

mov bx, 0

mov cx, 255

lea dx, file1

mov ah, 3Fh

int 21h

mov bx, dx

add bx, ax

mov byte ptr [bx-2], 0

mov ax, 716Ch

mov bx, 2

mov dx, 1

lea si, file1

int 21h

jc inputErr

mov bx1, ax

mov ah, 40h

lea dx, request2

mov cx, endRequest2-request2

int 21h

mov bx, 0

mov cx, 255

lea dx, file2

mov ah, 3Fh

int 21h

mov bx, dx

add bx, ax

mov byte ptr [bx-2], 0

mov ax, 716Ch ; создания файла с длинным именнем

mov bx, 2

mov dx, 1

lea si, file2

int 21h

jc inputErr

mov bx2, ax

writing1:

mov bx, bx1

mov cx, 1

lea si, buffer

lea dx, buffer

mov ah, 3fh; чтение с файла с атрибутом бкс1

int 21h

cmp byte ptr ds:[si], 29h

je end\_read

mov bx, bx2

mov cx, ax

mov ah, 40h;запись в файл с атрибутом бкс2

int 21h

mov bx, 1

cmp bx, ax

end\_read:

je writing1

end\_prog:

mov ax, 4c00h

int 21h

CODE ENDS

END beg