



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

**КАФЕДРА ИУК4 «Программное обеспечение ЭВМ,
информационные технологии»**

ЛАБОРАТОРНАЯ РАБОТА №7

**«Работа с файлами под управлением MS DOS.
Функции файловой системы»**


ДИСЦИПЛИНА: «Системное программирование»

Выполнил: студент гр. ИУК4-31Б


(подпись)

(Суриков Н. С.)
(Ф.И.О.)

Проверил:


(подпись)

(Амеличева К. А.)
(Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:


```

26 .code
27 mPrint macro string
28             mov     ah, 09h
29             lea     dx, string
30             int     21h
31 endm
32
33 start:
34     ; Инициализация сегмента данных
35             mov     ax, @data
36             mov     ds, ax
37             mov     es, ax
38
39     ; Открытие входного файла
40             mov     ah, 3Dh
41             xor     al, al
42             lea     dx, input_file_name
43             int     21h
44             jc      opening_error
45             mov     input_file_ID, ax
46
47     ; Создание выходного файла
48             mov     ah, 3Ch
49             xor     cx, cx
50             lea     dx, output_file_name
51             int     21h
52             jc      creation_error
53             mov     output_file_ID, ax
54
55 file_read_loop:
56     ; Чтение символа из файла
57             mov     ah, 3Fh
58             mov     bx, input_file_ID
59             lea     dx, char_buffer
60             mov     cx, 1                      ; Читаем 1 байт
61             int     21h
62             jc      read_error
63             cmp     ax, 0                      ; Конец файла?
64             je      end_of_file
65
66     ; Проверка на конец строки
67             mov     al, char_buffer            ; Получаем прочитанный символ
68             cmp     al, 0Dh                   ; Возврат каретки?
69             je      end_of_line
70             cmp     al, 0Ah                   ; Перевод строки?
71             je      end_of_line
72
73     ; Сохранение символа в буфер строки
74             lea     di, line_buffer
75             mov     cx, line_length
76             add     di, cx                    ; Указатель на конец строки
77             mov     [di], al                  ; Сохраняем символ
78             inc     word ptr line_length      ; Увеличиваем длину строки
79             jmp     file_read_loop

```

```

80
81 end_of_line:
82 ; Добавляем символы конца строки в конец строки
83         lea    di, line_buffer
84         add    di, line_length
85         mov    al, 0Dh
86         stosb
87         mov    al, 0Ah
88         stosb
89         mov    al, '$'
90         stosb
91
92 ; Выводим строку на экран
93         lea    dx, line_buffer
94         mPrint line_buffer
95
96 ; Строка завершена, проверяем её чётность
97         mov    ax, line_number
98         test   ax, 1
99         jnz    skip_reverse           ; Нечётная строка – пропускаем
100
101 ; Реверсируем чётную строку
102         lea    si, line_buffer
103         lea    di, reverse_buffer
104         add    si, line_length
105         dec    si
106         mov    cx, line_length
107 reverse_loop:
108         std                                ; Устанавливаем направление
109         lodsb                               ; Читаем символ из строки
110         cld                                ; Меняем направление
111         stosb                               ; Записываем в реверсированный буфер
112         loop   reverse_loop
113
114 ; Добавляем символы конца строки (CRLF) в конец строки
115         lea    di, reverse_buffer
116         add    di, line_length
117         mov    al, 0Dh
118         stosb
119         mov    al, 0Ah
120         stosb
121         mov    al, '$'
122         stosb
123
124 ; Выводим перевёрнутую строку на экран
125         lea    dx, reverse_buffer
126         mPrint reverse_buffer
127
128 ; Записываем реверсированную строку в выходной файл
129         lea    dx, reverse_buffer
130         mov    cx, line_length
131         call   write_line
132         jmp    continue_processing
133

```

```

134 skip_reverse:
135 ; Записываем исходную строку в выходной файл
136         lea    dx, line_buffer
137         mov    cx, line_length
138         call   write_line
139
140 continue_processing:
141 ; Увеличение номера строки и сброс длины
142         inc    word ptr line_number
143         mov    word ptr line_length, 0
144 ; Добавляем конец строки в файл вручную
145         lea    dx, newline
146         mov    ah, 40h
147         mov    bx, output_file_ID
148         lea    dx, newline
149         mov    cx, 2
150         int    21h
151         jmp    file_read_loop
152
153 end_of_file:
154 ; Закрытие файлов
155         mov    ah, 3Eh
156         mov    bx, input_file_ID
157         int    21h
158
159         mov    ah, 3Eh
160         mov    bx, output_file_ID
161         int    21h
162
163 ; Успешное завершение
164         mPrint completion_message
165         jmp    end_program
166
167 write_line:
168 ; Записываем строку в файл
169         mov    ah, 40h
170         mov    bx, output_file_ID
171         int    21h
172         jc     write_error
173         ret
174
175 opening_error:
176         mPrint opening_error_message
177         jmp    end_program
178
179 creation_error:
180         mPrint creation_error_message
181         jmp    end_program
182
183 read_error:
184         mPrint read_error_message
185         jmp    end_program
186
187 write_error:
188         mPrint write_error_message

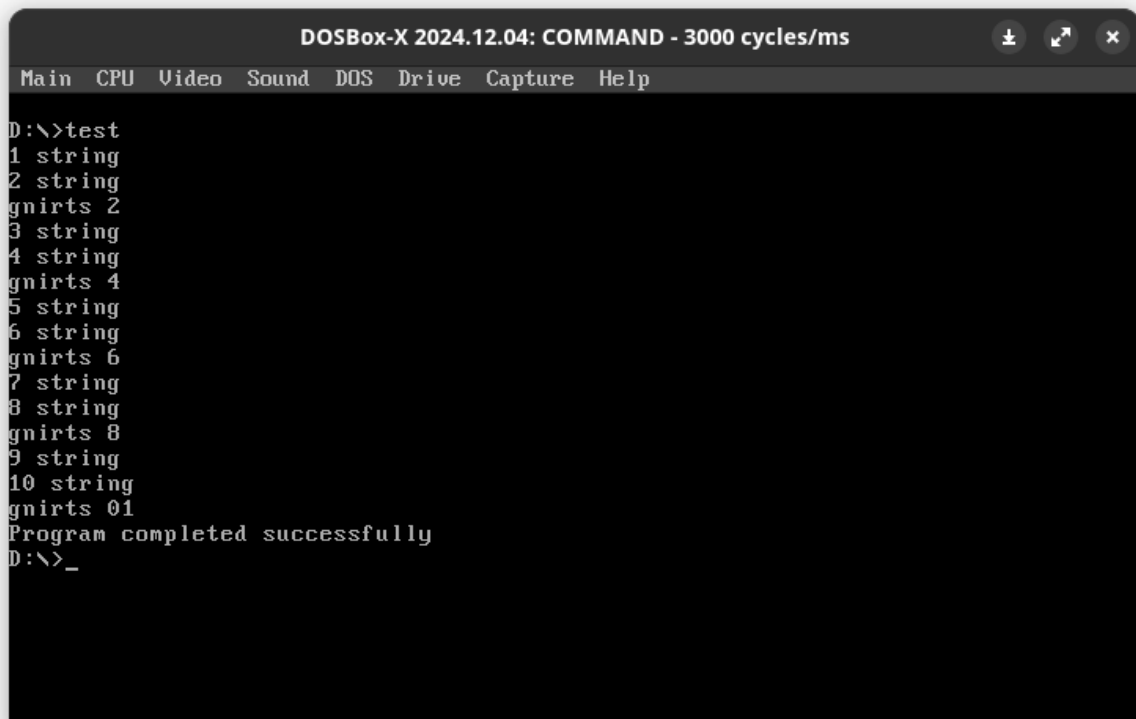
```

```

188             jmp     end_program
189
190     end_program:
191             mov     ax, 4C00h
192             int     21h
193     end start

```

Результат выполнения программы:



```

DOSBox-X 2024.12.04: COMMAND - 3000 cycles/ms
Main CPU Video Sound DOS Drive Capture Help
D:\>test
1 string
2 string
gnirts 2
3 string
4 string
gnirts 4
5 string
6 string
gnirts 6
7 string
8 string
gnirts 8
9 string
10 string
gnirts 01
Program completed successfully
D:\>_

```

infile.txt:	outfile.txt:
1 string 2 string 3 string 4 string 5 string 6 string 7 string 8 string 9 string 10 string	1 string gnirts 2 3 string gnirts 4 5 string gnirts 6 7 string gnirts 8 9 string gnirts 01

Вывод: в ходе выполнения работы были сформированы практические навыки разработки программного кода на языке Ассемблера; изучены способы работы с файлами, цепочечными командами.