

Защищено:
Большаков С.А.

Демонстрация ЛР:
Большаков С.А.

"__" _____ 2024 г.

"__" _____ 2024 г.

**Отчет по лабораторной работе № 6 по курсу
Системное программирование**

" Ввод и распечатка параметров командной строки "
(есть ли дополнительные требования - НЕТ)

18
(количество листов)
Вариант № <3>

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-41Б

Бирюкова Е.И.

(подпись)

"__" _____ 2024 г.

СОДЕРЖАНИЕ

1. Цель выполнения лабораторной работы № 6.....	3
2. Порядок и условия проведения работы № 6	3
3. Описание ошибок, возникших при отладке № 6	3
4. Блок-схема программы.....	4
5. Скриншот программы в TD.exe.....	5
6. Текст программы на языке Ассемблера	5
7. Результаты работы программы.....	17
8. Выводы по ЛР № 6.....	18

1. Цель выполнения лабораторной работы № 6

Разработать и отладить программу на языке Ассемблер для ввода, анализа (расшифровки, фактически грамматического разбора) и распечатки параметра командной строки, которые задаются при запуске программы (параметры размещаются в области PSP со смещением 081h, ПРОБЕЛ в DOSBox). Нужно также описать в БНФ синтаксис запуска вашей программы с параметрами в командной строке. Программа должна быть скомпонована в виде *.EXE - исполнимого файла. После запуска нужно проверить правильность первого параметра и наличие второго, после этого выдать соответствующие диагностические сообщения. Изучить структуру PSP и способы получения в программе адреса этого блока. Распечатать заданные параметры.

2. Порядок и условия проведения работы № 6

Предусмотреть ввод и анализ двух позиционных параметров командной строки (параметры читаются из области PSP), адрес PSP получается в программе автоматически. Параметры имеют строгую позицию в командной строке. Первый параметр задает фамилию студента (студентки). Нужно проверить правильность первого параметра – сообщение "Первый параметр верен", параметр распечатать, а наличие второго параметра – сообщение "Второй параметр есть/отсутствует". Написать и оформить в БНФ инструкцию для работы данной программы. Записать параметр в буфер программы. Для проверки параметра использовать команду цепочек CMPSB.

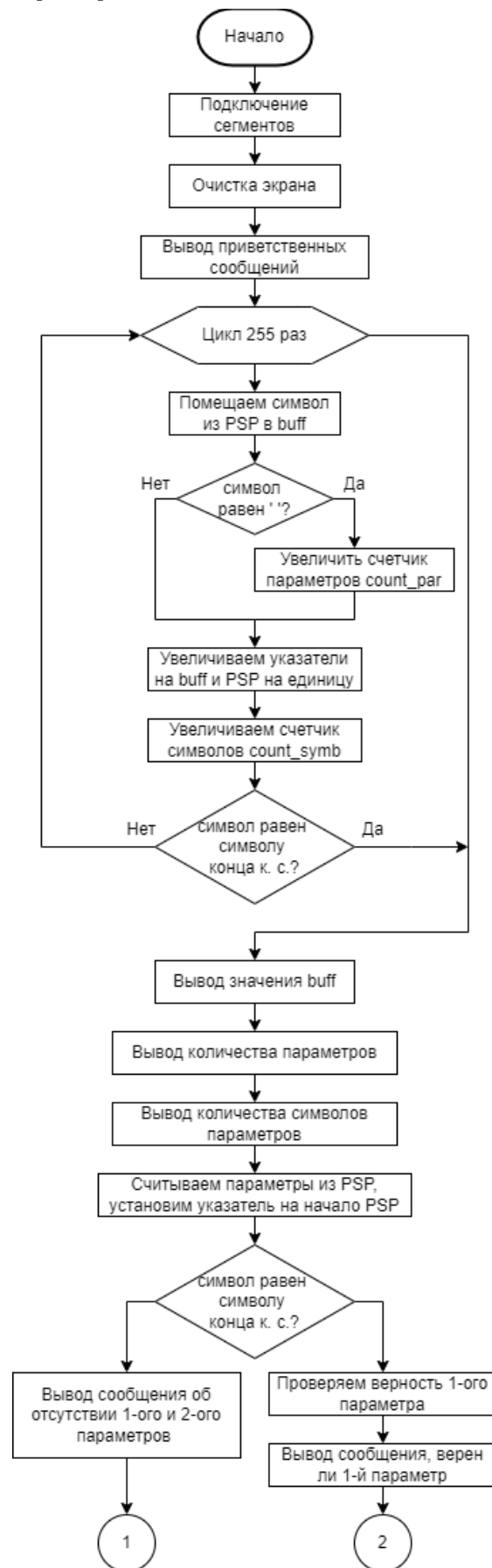
После запуска программы *.EXE список параметров (текст вводимой командной строки сохраняется в PSP программы). Доступ к PSP может быть выполнен с помощью прерывания 21h – 51h или из сегментного регистра ES после первоначального запуска программы. Поле списка параметров начинается в PSP со смещением 081h (См. справочник). В области PSP со смещением 80H содержится число символов введенных параметров (один байт). Примечание. При создании *.COM – это д.т. программы PSP располагается непосредственно в начале программы (ORG 100h – область, в которую загрузчик записывает блок PSP).

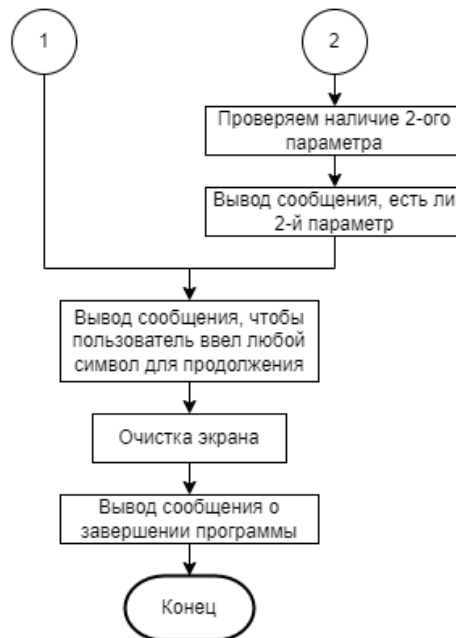
Назначение и формат параметров должен быть следующий: 1. Первый параметр задает фамилию студента в именительном падеже. 2. Второй параметр произвольный – не менее 3-х символов. Все параметры перед завершением программы распечатываются. Данная программа компонуется и выполняется в виде *.EXE модуля.

3. Описание ошибок, возникших при отладке № 6

№ п/п	Проявление ошибки	Причина ошибки	Способ устранения
1.	При значении 1-ого параметра "Бирюковааа" выдается сообщение "1-й параметр верен."	С помощью инструкции cmpsb сравнивается указанное количество символов (8). Первые восемь символов совпадают у параметра и кодового слова.	Добавлена функция check_length_par1, сравнивающая длину 1-ого параметра с длиной кодового слова. Только после этого проводятся остальные проверки.
2.	При попытке вывести количество параметров вместо числа выводится символ.	При попытке вывода числа, как символа, будет выводиться символ, код которого равен этому числу.	Добавление функции hexrg, которая переводит число в шестнадцатеричную систему счисления и выводит данное число.

4. Блок-схема программы





5. Скриншот программы в TD.exe

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File Edit View Run Breakpoints Data Options Window Help

[F5]=Module: lab6 File: 16=1=[F1][F2]

mycode SEGMENT 'CODE'

ASSUME CS: mycode, DS: data_seg, SS:

; Бирюкова Е.И. ИУ5-41Б

; Вариант 3 ЛР6

data_seg SEGMENT

msg_welcome DB 'Добро пожаловать'

msg_exit DB 'Завершение программ'

msg_hint DB 'Нажмите любую кнопку'

msg_par DB 'Введенные параметры:

msg_count_par DB 'Количество пар'

msg_count_symb DB 'Количество сим'

count_par DB 0

count_symb DB 0

HEXTAB DB '0123456789ABCDEF'

CPU 80486

#lab6#start: MOV AX, 5

cs:00CD mov ax, 5

#lab6#190: MOV DS, A

cs:00D0 mov ds, a

#lab6#193: CALL clea

cs:00D2 call #lab

#lab6#196: CALL clrf

cs:00D5 call #lab

#lab6#197: LEA DX, m

cs:00D8 mov dx, 0

#lab6#198: CALL prin

cs:00DB call #lab

#lab6#199: CALL clrf

ds:0000 CD 20 FF 9F

ds:0008 AD DE E0 01

ds:0010 A7 18 89 02

ds:0018 01 01 01 00

ax 0000

bx 0000

cx 0000

dx 0000

si 0000

di 0000

bp 0000

sp 0200

ds 523F

es 523F

ss 5282

cs 524F

ip 00CD

c=0

z=0

s=0

o=0

p=0

a=0

i=1

d=0

ss:0202 0310

ss:0200 52FB

Watches

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

6. Текст программы на языке Ассемблера

Листинг программы:

Turbo Assembler Version 3.1 04/23/24 11:20:45 Page 1

lab6.asm

```

1 0000 mycode SEGMENT 'CODE'
2 ASSUME CS: mycode, DS: data_seg, SS: stack_seg
3
4 ; Бирюкова Е. И. ИУ5-41Б
5 ; Вариант 3 ЛР6
  
```

```

6
7 0000                                data_seg SEGMENT
8 0000 84 AE A1 E0 AE 20      AF+   msg_welcome DB      'Добро пожаловать!$'
9      AE A6 A0 AB AE A2      A0+
10     E2 EC 21 24
11 0012 87 A0 A2 A5 E0 E8      A5+   msg_exit DB 'Завершение программы. Всего доброго!$'
12     AD A8 A5 20 AF E0      AE+
13     A3 E0 A0 AC AC EB      2E+
14     20 82 E1 A5 A3 AE20+
15     A4 AE A1 E0 AE A3      AE+
16     21 24
17 0037 8D A0 A6 AC A8 E2      A5+   msg_hint DB 'Нажмите любую кнопку для продолжения...$'
18     20 AB EE A1 E3 EE      20+
19     AA AD AE AF AA E3      20+
20     A4 AB EF 20 AF E0      AE+
21     A4 AE AB A6 A5 AD      A8+
22     EF 2E 2E 2E 24
23 005F 82 A2 A5 A4 A5 AD      AD+   msg_par DB 'Введенные параметры:$'
24     EB A5 20 AF A0 E0      A0+
25     AC A5 E2 E0 EB 3A      24
26 0074 8A AE AB A8 E7 A5      E1+   msg_count_par DB 'Количество параметров: $'
27     E2 A2 AE 20 AF A0      E0+
28     A0 AC A5 E2 E0 AE      A2+
29     3A 20 24
30 008C 8A AE AB A8 E7 A5      E1+   msg_count_symb      DB 'Количество символов: $'
31     E2 A2 AE 20 E1 A8      AC+
32     A2 AE AB AE A2 3A      20+
33     24
34
35 00A2 00                                count_par DB 0
36 00A3 00                                count_symb DB 0
37
38 00A4 30 31 32 33 34 35      36+   HEXTAB      DB '0123456789ABCDEF'
39     37 38 39 41 42 43  44+
40     45 46
41 00B4 81 A8 E0 EE AA AE      A2+   par1_true DB 'Бирюкова'
42     A0
43 00BC 00                                par1_length DB      0
44 00BD 00                                finish_str DB 0
45 00BE 7F*(24)                      buff DB 127 DUP ('$')
46
47 013D 31 2D A9 20 AF A0      E0+   msg_pr1_v_yes DB '1-й параметр ВЕРЕН.$'
48     A0 AC A5 E2 E0 20      82+
49     85 90 85 8D 2E 24
50 0151 31 2D A9 20 AF A0      E0+   msg_pr1_v_no DB '1-й параметр НЕ ВЕРЕН.$'
51     A0 AC A5 E2 E0 20      8D+
52     85 20 82 85 90 85  8D+
53     2E 24
54
55 0168 32 2D A9 20 AF A0      E0+   msg_pr2_yes DB      '2-й параметр ЕСТЬ.$'
56     A0 AC A5 E2 E0 20      85+
57     91 92 9C 2E 24

```

```

58 017B 32 2D A9 20 AF A0      E0+   msg_pr2_no DB '2-й параметр ОТСУТСТВУЕТ.$'
59     A0 AC A5 E2 E0 20      8E+
60     92 91 93 92 91 92  82+
61     93 85 92 2E 24

```

```

62
63 0195 8F A0 E0 A0 AC A5      E2+   msg_pr1_pr2_no      DB 'Параметры отсутствуют.$'
64      E0 EB 20 AE E2 E1      E3+
65      E2 E1 E2 A2 E3 EEE2+
66      2E 24
67
68 01AC                        data_seg ENDS
69
70 0000                        stack_seg SEGMENT STACK 'STACK'
71 0000 0100*(0000)           DW 256      DUP(0)
72 0200                        stack_seg ENDS
73
74                            ;Вывод одного символа
75 0000                        putch PROC
76 0000 B4 02                  MOV AH, 02h
77 0002 CD 21                  INT 21h
78 0004 C3                    RET
79 0005                        putch ENDP
80
81                            ;Вывод строки
82 0005                        print_str proc
83 0005 B4 09                  MOV AH, 09h
84 0007 CD 21                  INT 21h
85 0009 C3                    RET
86 000A                        print_str endp
87
88                            ;Ввод символа
89 000A                        getch PROC
90 000A B4 01                  MOV AH, 01H
91 000C CD 21                  INT 21H
92 000E C3                    RET
93 000F                        getch ENDP
94
95                            ;Перевод строки и возврат каретки
96 000F                        clrf PROC
97 000F B2 0A                  MOV DL, 0AH
98 0011 E8 FFEC                CALL putch
99 0014 B2 0D                  MOV DL, 0DH
100 0016 E8 FFE7               CALL putch
101 0019 C3                    RET
102 001A                        clrf ENDP
103
104                            ;Очистка экрана при завершении работы
105 001A                        clear_screen PROC
106 001A B4 00                  MOV AH, 00H
107 001C B0 03                  MOV AL,3
108 001E CD 10                  INT 10H
109 0020 B9 0019               MOV CX, 25
110 0023                        labclr:
111 0023 B4 02                  MOV AH , 02H
112 0025 B2 0A                  MOV DL , 10
113 0027 CD 21                  INT 021H
114 0029 E2 F8                  LOOP labclr

```

```

115 002B C3                    RET
116 002C                        clear_screen ENDP
117

```

```

118 002C          check_par1 PROC
119 002C E8 0041    CALL check_length_par1
120 002F 47        INC DI
121
122 0030 A0 00BCr   MOV AL, par1_length
123 0033 3C 08      CMP AL, 8
124 0035 75 14      JNE Check1_false
125
126 0037 B9 0008    MOV CX, 8
127 003A BE 00B4r   MOV SI, offset    par1_true
128 003D F3> A6     REP CMPSB
129 003F 75 0A      JNE Check1_false
130
131              ;1-й параметр верен
132 0041 BA 013Dr   LEA DX, msg_pr1_v_yes
133 0044 E8 FFBE    CALL print_str
134 0047 E8 FFC5    CALL clrf
135 004A C3        RET
136
137              ;1-й параметр НЕ верен-
138 004B          Check1_false:
139 004B BA 0151r   LEA DX, msg_pr1_v_no
140 004E E8 FFB4    CALL print_str
141 0051 E8 FFBB    CALL clrf
142 0054 C3        RET
143 0055          check_par1 ENDP
144
145 0055          check_par2 PROC
146 0055 A0 00BDr   MOV AL, finish_str
147 0058 3C 00      CMP AL, 0
148 005A 74 0A      JE flag_par2_yes
149
150              ;2-ой параметр отсутствует
151 005C BA 017Br   LEA DX, msg_pr2_no
152 005F E8 FFA3    CALL print_str
153 0062 E8 FFAA    CALL clrf
154 0065 C3        RET
155
156              ;2-ой параметр присутствует
157 0066          flag_par2_yes:
158 0066 BA 0168r   LEA DX, msg_pr2_yes
159 0069 E8 FF99    CALL print_str
160 006C E8 FFA0    CALL clrf
161 006F C3        RET
162 0070          check_par2 ENDP
163
164 0070          check_length_par1 PROC
165 0070 57        PUSH DI
166 0071          repeat:
167 0071 FE 06 00BCr INC par1_length
168 0075 47        INC DI
169
170 0076 26: 8A 0D  MOV CL, ES:[DI]
171 0079 80 F9 00  CMP CL, 0

```

```

172 007C 74 09      JE flag_finish
173

```



```

174 007E 26: 80 3D      20          CMP BYTE PTR ES:[DI], ''
175 0082 75 ED          JNE repeat
176 0084 EB 0A 90          JMP contin
177
178 0087          flag_finish:
179 0087 C6 06 00BD r 01    MOV finish_str, 1
180 008C FE 0E 00BC r      DEC par1_length
181
182 0090          contin:
183 0090 FE 0E 00BC r      DEC par1_length
184 0094 5F          POP DI
185 0095 C3          RET
186 0096          check_length_par1 ENDP
187
188 0096          hexpr PROC
189 0096 BB 00A4 r          MOV BX, OFFSET HEXTAB
190 0099 52          PUSH     DX
191 009A 8A C2          MOV AL,DL
192 009C D0 E8 D0 E8 D0 E8 D0+  SHR AL,4
193    E8
194 00A4 D7          XLAT
195 00A5 8A D0          MOV DL , AL
196 00A7 E8 FF56          CALL     putch
197 00AA 5A          POP DX
198 00AB 8A C2          MOV AL,DL
199 00AD 24 0F          AND AL, 0FH
200 00AF D7          XLAT
201 00B0 8A D0          MOV DL,AL
202 00B2 E8 FF4B          CALL     putch
203 00B5 B2 48          MOV DL, 'H'
204 00B7 E8 FF46          CALL     putch
205 00BA C3          RET
206 00BB          hexpr ENDP
207
208          ;Выход из программы
209 00BB          exit_prog PROC
210          ;Очистка экрана -
211 00BB E8 FF5C          CALL clear_screen
212
213          ;Сообщение о выходе из программы
214 00BE BA 0012 r          LEA DX, msg_exit
215 00C1 E8 FF41          CALL print_str
216 00C4 E8 FF48          CALL clrf
217
218 00C7 B0 00          mov al, 00
219 00C9 B4 4C          mov ah, 4ch
220 00CB CD 21          int 021h
221 00CD          exit_prog ENDP
222
223 00CD          START:
224          ;СТАНДАРТНОЕ НАЧАЛО ПРОГРАММЫ
225          ;Подключение сегментов
226 00CD B8 0000 s          MOV AX, data_seg
227 00D0 8E D8          MOV DS, AX
228

```

230	00D2 E8 FF45	CALL clear_screen
231		
232		;Приветствие пользователя
233	00D5 E8 FF37	CALL clrf
234	00D8 BA 0000r	LEA DX, msg_welcome
235	00DB E8 FF27	CALL print_str
236	00DE E8 FF2E	CALL clrf
237		
238		;Сохранение параметров в buff
239	00E1 BB 0081	MOV BX, 81h
240	00E4 BE 0000	MOV SI, 0
241	00E7	save_par:
242	00E7 26: 8A 17	MOV DL, ES:[BX]
243	00EA 88 94 00BEr	MOV buff[SI], DL
244	00EE 80 FA 20	CMP DL, ''
245	00F1 75 04	JNE flag_1
246	00F3 FE 06 00A2r	INC count_par
247	00F7	flag_1:
248	00F7 46	INC SI
249	00F8 43	INC BX
250	00F9 FE 06 00A3r	INC count_symb
251	00FD 26: 8A 0F	MOV CL, ES:[BX]
252	0100 80 F9 00	CMP CL, 0
253	0103 74 02	JE flag_finish_loop
254	0105 E2 E0	LOOP save_par
255	0107	flag_finish_loop:
256	0107 4E	DEC SI
257	0108 FE 0E 00A3r	DEC count_symb
258	010C C6 84 00BEr 24	MOV buff[SI], '\$'
259		
260		;Распечатка параметров
261	0111 BA 005Fr	LEA DX, msg_par
262	0114 E8 FEEE	CALL print_str
263		
264	0117 BE 0000	MOV SI, 0
265	011A	cycle_output:
266	011A 8A 94 00BEr	MOV DL, buff[SI]
267	011E 80 FA 24	CMP DL, '\$'
268	0121 74 06	JE flag2
269	0123 E8 FEDA	CALL PUTCH
270	0126 46	INC SI
271	0127 E2 F1	LOOP cycle_output
272		
273	0129	flag2:
274	0129 E8 FEE3	CALL clrf
275		
276		;Вывод количества параметров
277	012C BA 0074r	LEA DX, msg_count_par
278	012F E8 FED3	CALL print_str
279	0132 8A 16 00A2r	MOV DL, count_par
280	0136 E8 FF5D	CALL hexpr
281	0139 E8 FED3	CALL clrf
282		
283		;Вывод количества символов параметров
284	013C BA 008Cr	LEA DX, msg_count_symb
285	013F E8 FEC3	CALL print_str

```

286 0142 8A 16 00A3r      MOV DL, count_symb
287 0146 E8 FF4D          CALL hexpr
288 0149 E8 FEC3          CALL clrf
289
290                      ;ОСНОВНАЯ ЧАСТЬ ПРОГРАММЫ
291                      ;Считывание параметров и проверка их наличия
292 014C 26: 8A 0E      0080      MOV CL,ES:[80h]
293 0151 80 F9 00        CMP CL,0
294 0154 77 0C          JA flag_check_par1
295
296                      ;Выход из программы, если нет параметров
297 0156 BA 0195r        LEA DX, msg_pr1_pr2_no
298 0159 E8 FEA9          CALL print_str
299 015C E8 FEB0          CALL clrf
300 015F EB 0C 90        JMP flag_exit_prog
301
302                      ;Проверяем 1-й параметр
303 0162                flag_check_par1:
304 0162 B5 00            MOV CH,0
305 0164 BF 0081          MOV DI,81h
306 0167 E8 FEC2          CALL check_par1
307
308                      ;Проверяем наличие 2-ого параметра
309 016A E8 FEE8          CALL check_par2
310
311                      ;Выход из программы
312 016D                flag_exit_prog:
313 016D BA 0037r        LEA DX, msg_hint
314 0170 E8 FE92          CALL print_str
315 0173 E8 FE94          CALL getch
316 0176 E8 FF42          CALL exit_prog
317
318 0179                mycode ENDS
319                END START

```

Turbo Assembler Version 3.1 04/23/24 11:20:45 Page 7
Symbol Table

Symbol Name	Type	Value	Cref (defined at #)
??DATE	Text	"04/23/24"	
??FILENAME	Text	"lab6 "	
??TIME	Text	"11:20:44"	
??VERSION	Number	030A	
@CPU	Text	0101H	
@CURSEG	Text	MYCODE	#1 #7 #68 #70 #72
@FILENAME	Text	LAB6	
@WORDSIZE	Text	2	#1 #7 #68 #70 #72
BUFF	Byte	DATA_SEG:00BE	#45 243 258 266
CHECK1_FALSE	Near	MYCODE:004B	124 129 #138
CHECK_LENGTH_PAR1	Near	MYCODE:0070	119 #164
CHECK_PAR1	Near	MYCODE:002C	#118 306
CHECK_PAR2	Near	MYCODE:0055	#145 309
CLEAR_SCREEN	Near	MYCODE:001A	#105 211 230
CLRF	Near	MYCODE:000F	#96 134 141 153 160 216 233 236 274 281 288
299			
CONTIN	Near	MYCODE:0090	176 #182
COUNT_PAR	Byte	DATA_SEG:00A2	#35 246 279
COUNT_SYMB	Byte	DATA_SEG:00A3	#36 250 257 286
CYCLE_OUTPUT	Near	MYCODE:011A	#265 271

EXIT_PROG	Near	MYCODE:00BB	#209	316	
FINISH_STR	Byte	DATA_SEG:00BD	#44	146	179
FLAG2	Near	MYCODE:0129	268	#273	
FLAG_1	Near	MYCODE:00F7		245	#247
FLAG_CHECK_PAR1	Near	MYCODE:0162		294	#303
FLAG_EXIT_PROG	Near	MYCODE:016D		300	#312
FLAG_FINISH	Near	MYCODE:0087	172	#178	
FLAG_FINISH_LOOP	Near	MYCODE:0107	253	#255	
FLAG_PAR2_YES	Near	MYCODE:0066		148	#157
GETCH	Near	MYCODE:000A		#89	315
HEXPR	Near	MYCODE:0096		#188	280 287
HEXTAB	Byte	DATA_SEG:00A4			#38 189
LABCLR	Near	MYCODE:0023		#110	114
MSG_COUNT_PAR	Byte	DATA_SEG:0074			#26 277
MSG_COUNT_SYMB	Byte	DATA_SEG:008C			#30 284
MSG_EXIT	Byte	DATA_SEG:0012		#11	214
MSG_HINT	Byte	DATA_SEG:0037		#17	313
MSG_PAR	Byte	DATA_SEG:005F			#23 261
MSG_PR1_PR2_NO	Byte	DATA_SEG:0195			#63 297
MSG_PR1_V_NO	Byte	DATA_SEG:0151			#50 139
MSG_PR1_V_YES	Byte	DATA_SEG:013D			#47 132
MSG_PR2_NO	Byte	DATA_SEG:017B		#58	151
MSG_PR2_YES	Byte	DATA_SEG:0168			#55 158
MSG_WELCOME	Byte	DATA_SEG:0000			#8 234
PAR1_LENGTH	Byte	DATA_SEG:00BC			#43 122 167 180 183
PAR1_TRUE	Byte	DATA_SEG:00B4		#41	127
PRINT_STR	Near	MYCODE:0005	#82	133 140 152 159 215 235 262 278 285 298	314
PUTCH	Near	MYCODE:0000		#75	98 100 196 202 204 269
REPEAT	Near	MYCODE:0071		#166	175
SAVE_PAR	Near	MYCODE:00E7	#241	254	
START	Near	MYCODE:00CD		#223	319

Turbo Assembler Version 3.1 04/23/24 11:20:45 Page 8
Symbol Table

Groups & Segments	Bit	Size	Align	Combine	Class	Cref	(defined at #)
DATA_SEG	16	01AC	Para	none	2	#7	226
MYCODE	16	0179	Para	none	CODE	#1	2
STACK_SEG	16	0200	Para	Stack	STACK	2	#70

Текст программы:

```
mycode SEGMENT 'CODE'
ASSUME CS: mycode, DS: data_seg, SS: stack_seg
```

```
; Бирюкова Е. И. ИУ5-41Б
; Вариант 3 ЛР6
```

```
data_seg SEGMENT
    msg_welcome DB 'Добро пожаловать!$'
    msg_exit DB 'Завершение программы. Всего доброго!$'
    msg_hint DB 'Нажмите любую кнопку для продолжения...$'
    msg_par DB 'Введенные параметры:$'
    msg_count_par DB 'Количество параметров: $'
    msg_count_symb DB 'Количество символов: $'

    count_par DB 0
    count_symb DB 0

    HEXTAB DB '0123456789ABCDEF'
```

```

par1_true DB 'Бирюкова'
par1_length DB 0
finish_str DB 0
buff DB 127 DUP ('$')

msg_pr1_v_yes DB '1-й параметр ВЕРЕН.$'
msg_pr1_v_no DB '1-й параметр НЕ ВЕРЕН.$'

msg_pr2_yes DB '2-й параметр ЕСТЬ.$'
msg_pr2_no DB '2-й параметр ОТСУТСТВУЕТ.$'

msg_pr1_pr2_no DB 'Параметры отсутствуют.$'

data_seg ENDS

stack_seg SEGMENT STACK 'STACK'
    DW 256 DUP(0)
stack_seg ENDS

;Вывод одного символа
putch PROC
    MOV AH, 02h
    INT 21h
    RET
putch ENDP

;Вывод строки
print_str proc
    MOV AH, 09h
    INT 21h
    RET
print_str endp

;Ввод символа
getch PROC
    MOV AH, 01H
    INT 21H
    RET
getch ENDP

;Перевод строки и возврат каретки
clrf PROC
    MOV DL, 0AH
    CALL putch
    MOV DL, 0DH
    CALL putch
    RET
clrf ENDP

;Очистка экрана при завершении работы
clear_screen PROC
    MOV AH, 00H
    MOV AL, 3
    INT 10H
    MOV CX, 25
labclr:
    MOV AH, 02H
    MOV DL, 10
    INT 021H
    LOOP labclr
    RET
clear_screen ENDP

check_par1 PROC
    CALL check_length_par1
    INC DI

```

```

MOV AL, par1_length
CMP AL, 8
JNE Check1_false

MOV CX, 8
MOV SI, offset par1_true
REP CMPSB
JNE Check1_false

;1-й параметр верен
LEA DX, msg_pr1_v_yes
CALL print_str
CALL clrf
RET

;1-й параметр НЕ верен
Check1_false:
LEA DX, msg_pr1_v_no
CALL print_str
CALL clrf
RET
check_par1 ENDP

check_par2 PROC
MOV AL, finish_str
CMP AL, 0
JE flag_par2_yes

;2-ой параметр отсутствует
LEA DX, msg_pr2_no
CALL print_str
CALL clrf
RET

;2-ой параметр присутствует
flag_par2_yes:
LEA DX, msg_pr2_yes
CALL print_str
CALL clrf
RET
check_par2 ENDP

check_length_par1 PROC
PUSH DI
repeat:
INC par1_length
INC DI

MOV CL, ES:[DI]
CMP CL, 0
JE flag_finish

CMP BYTE PTR ES:[DI], ' '
JNE repeat
JMP contin

flag_finish:
MOV finish_str, 1
DEC par1_length

contin:
DEC par1_length
POP DI
RET
check_length_par1 ENDP

```

```

hexpr PROC
    MOV BX, OFFSET HEXTAB
    PUSH DX
    MOV AL,DL
    SHR AL,4
    XLAT
    MOV DL , AL
    CALL putch
    POP DX
    MOV AL,DL
    AND AL, 0FH
    XLAT
    MOV DL,AL
    CALL putch
    MOV DL, 'H'
    CALL putch
    RET
hexpr ENDP

```

;Выход из программы

```

exit_prog PROC
    ;Очистка экрана
    CALL clear_screen

    ;Сообщение о выходе из программы
    LEA DX, msg_exit
    CALL print_str
    CALL clrf

    mov al, 00
    mov ah, 4ch
    int 021h
exit_prog ENDP

```

START:

;СТАНДАРТНОЕ НАЧАЛО ПРОГРАММЫ

;Подключение сегментов

```
MOV AX, data_seg
```

```
MOV DS, AX
```

;Очистка экрана

```
CALL clear_screen
```

;Приветствие пользователя

```
CALL clrf
```

```
LEA DX, msg_welcome
```

```
CALL print_str
```

```
CALL clrf
```

;Сохранение параметров в buff

```
MOV BX, 81h
```

```
MOV SI, 0
```

```
MOV CX, 255
```

save_par:

```
    MOV DL, ES:[BX]
```

```
    MOV buff[SI], DL
```

```
    CMP DL, ''
```

```
    JNE flag_1
```

```
    INC count_par
```

```
flag_1:
```

```
    INC SI
```

```
    INC BX
```

```
    INC count_symb
```

```
    MOV CL, ES:[BX]
```

```
    CMP CL, 0
```

```
    JE flag_finish_loop
LOOP save_par
flag_finish_loop:
DEC SI
DEC count_symb
MOV buff[SI], '$'
```

```
;Распечатка параметров
LEA DX, msg_par
CALL print_str
```

```
MOV SI, 0
cycle_output:
    MOV DL, buff[SI]
    CMP DL, '$'
    JE flag2
    CALL PUTCH
    INC SI
LOOP cycle_output
```

```
flag2:
CALL clrf
```

```
;Вывод количества параметров
LEA DX, msg_count_par
CALL print_str
MOV DL, count_par
CALL hexpr
CALL clrf
```

```
;Вывод количества символов параметров
LEA DX, msg_count_symb
CALL print_str
MOV DL, count_symb
CALL hexpr
CALL clrf
```

```
;ОСНОВНАЯ ЧАСТЬ ПРОГРАММЫ
;Считывание параметров и проверка их наличия
MOV CL,ES:[80h]
CMP CL,0
JA flag_check_par1
```

```
;Выход из программы, если нет параметров
LEA DX, msg_pr1_pr2_no
CALL print_str
CALL clrf
JMP flag_exit_prog
```

```
;Проверяем 1-й параметр
flag_check_par1:
MOV CH,0
MOV DI,81h
CALL check_par1
```

```
;Проверяем наличие 2-ого параметра
CALL check_par2
```

```
;Выход из программы
flag_exit_prog:
LEA DX, msg_hint
CALL print_str
CALL getch
CALL exit_prog
```

```
mycode ENDS
```


END START

7. Результаты работы программы

Запустим программу с различными параметрами.

1-й параметр верен, 2-й параметр присутствует.

```
U:\TASM\TASM3>lab6.exe Бирюкова ааааааа
```

```
Добро пожаловать!  
Введенные параметры: Бирюкова ааааааа  
Количество параметров: 02H  
Количество символов: 11H  
1-й параметр ВЕРЕН.  
2-й параметр ЕСТЬ.  
Нажмите любую кнопку для продолжения...
```

1-й параметр верен, 2-й параметр отсутствует.

```
U:\TASM\TASM3>lab6.exe Бирюкова_
```

```
Добро пожаловать!  
Введенные параметры: Бирюкова  
Количество параметров: 01H  
Количество символов: 09H  
1-й параметр ВЕРЕН.  
2-й параметр ОТСУТСТВУЕТ.  
Нажмите любую кнопку для продолжения...
```

1-й параметр не верен, 2-й параметр присутствует.

```
U:\TASM\TASM3>lab6.exe Бирюков ааааааа
```

```
Добро пожаловать!  
Введенные параметры: Бирюков ааааааа  
Количество параметров: 02H  
Количество символов: 10H  
1-й параметр НЕ ВЕРЕН.  
2-й параметр ЕСТЬ.  
Нажмите любую кнопку для продолжения...
```

1-й параметр не верен, 2-й параметр отсутствует.

```
U:\TASM\TASM3>lab6.exe Бирюков_
```

```
Добро пожаловать!  
Введенные параметры: Бирюков  
Количество параметров: 01H  
Количество символов: 08H  
1-й параметр НЕ ВЕРЕН.  
2-й параметр ОТСУТСТВУЕТ.  
Нажмите любую кнопку для продолжения..._
```

Оба параметра отсутствуют.

```
U:\TASM\TASM3>lab6.exe  
Добро пожаловать!  
Введенные параметры:  
Количество параметров: 00H  
Количество символов: 00H  
Параметры отсутствуют.  
Нажмите любую кнопку для продолжения..._
```

8. Выводы по ЛР № 6

В данной лабораторной работе была разработана и отлажена программа на языке Ассемблер, которая выполняет следующие функции:

- Ввод и анализ двух позиционных параметров командной строки (параметры читаются из области PSP).
- Проверка правильности первого параметра и наличие второго параметра.
- Распечатка заданных параметров.

Программа была скомпонована в виде исполнимого файла *.EXE. После запуска программа проверяет правильность первого параметра (фамилия студента) и наличие второго параметра. Затем программа выводит соответствующие диагностические сообщения и распечатывает заданные параметры.

Для получения адреса PSP в программе используется прерывание 21h – 51h или сегментный регистр ES. Поле списка параметров начинается в PSP со смещения 081h. Число символов введенных параметров содержится в PSP со смещением 80H.

Для проверки параметра используется команда цепочек CMPSB.

Программа успешно выполняет поставленные задачи и может быть использована для ввода, анализа и распечатки параметров командной строки.