

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 к. с."

(есть ли дополнительные требования - НЕТ)

9

(количество листов)

Вариант № 11

ИСПОЛНИТЕЛЬ:

студент группы **ИУ5-41Б**

Ларкин Б. В.

(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 ЛР № 6	3
4. Инструкция в БНФ для запуска программы.....	3
5. Блок-схема программы.....	4
6. Текст программы на языке Ассемблера (.LST).....	5
7. Скриншот программы в TD.exe.....	8
8. Результаты работы программы.....	8
9. Выводы по ЛР № 6.....	9

1. Цель выполнения лабораторной работы № 6

Лабораторная работа №6 выполняется для получения навыков разработки программ и процедур на Ассемблере, использующих параметры командной строки и соответствующей их получению алгоритм.

2. Порядок и условия проведения 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. Для этого надо изучить раздел 4 методических указаний к ЛР. 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.

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 командой MOVSB. Для проверки параметра использовать команду цепочек CMPSB.

Пример возможного вывода результата работы ЛР № 6:

Первый параметр верен= Иванов
Второй параметр отсутствует!

Или

Первый параметр неправильный
Второй параметр есть!

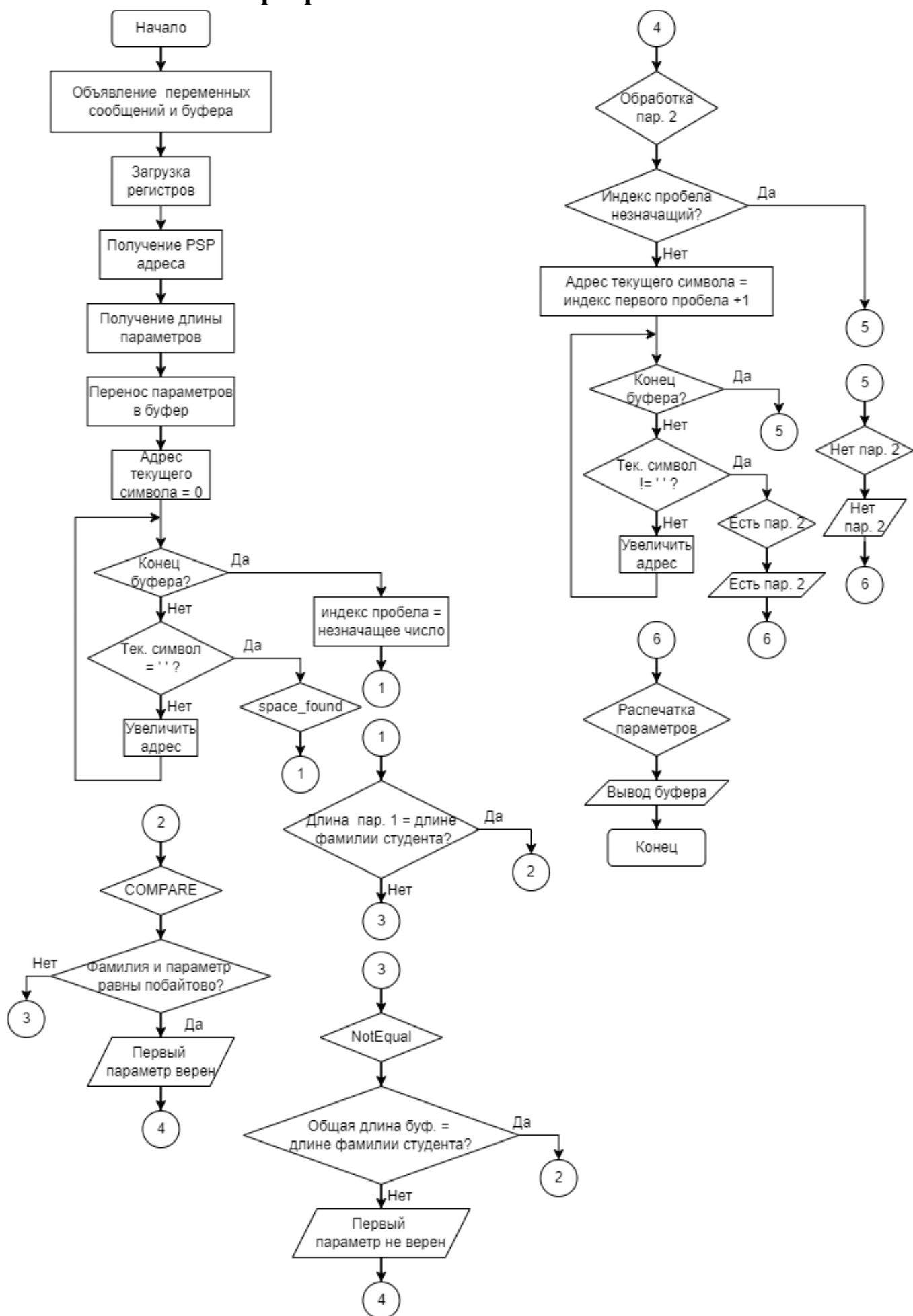
3. Описание ошибок, возникших при отладке ЛР № 6

№ п/п	Проявление ошибки	Причина ошибки	Способ устранения
1.	Неверная форма адресации данных	Попытка сложения переменной и регистра в правой части MOV	MOV переменной в левую часть выражения, затем INC на регистр
2.	Некорректные результаты поиска пробела в буфере	Наличие в DOSBOX пробела в начале строки параметров	Неучет первого символа в буфере

4. Инструкция в БНФ для запуска программы

<запуск программы> := <имя программы> _ <список параметров программы>
<имя программы> := laba6.exe | laba6
<список параметров программы> := <Параметр 1>[_<Параметр 2>]
<Параметр 1> := <Фамилия пользователя>
<Параметр 2> := <любая последовательность символов до пробела>

5. Блок-схема программы



6. Текст программы на языке Ассемблера (.LST)

Turbo Assembler Version 3.1
laba6.asm

05/02/24 00:05:42

Page 1

```
1          ;JP #6 2024, Ларкин Борис ИУ5-41
2 0000      STSEG SEGMENT STACK 'STACK'
3 0000 0100*(0000)    DW 256 DUP(0)
4 0200      STSEG ENDS
5
6 0000      MYCODE SEGMENT 'CODE'
7          ASSUME CS:MYCODE, DS:MYCODE, SS:STSEG
8 0000      START:
9          ; Загрузка регистров
10 0000 0E        PUSH CS
11 0001 1F        POP DS
12 0002 1E        PUSH DS
13 0003 07        POP ES
14 0004 B4 51      MOV AH , 51H ;Получение ID текущего процесса (PSP адрес)
15 0006 CD 21      int 21H
16 0008 8E C3      MOV ES , BX
17 000A 8C 06 00E0r  mov save_psp , ES
18          ; получение счетчика ES
19 000E 26: A0 0080  MOV AL , ES:128
20 0012 48        DEC AX ;не учитываем незначащий пробел
21 0013 A2 017Br   MOV COUNT , AL
22          ; перерезапись
23 0016 8A C8      MOV CL, AL
24 0018 B5 00      MOV CH , 0
25 001A 0E        PUSH CS
26
27 001B 07        POP ES
28 001C 8E 1E 00E0r  MOV DS , save_psp
29 0020 BE 0082      MOV SI , 130 ;Не копируем пробел
30 0023 BF 00F8r    LEA DI , BUF
31 0026 F3> A4      REP MOVSB
32 0028 0E        PUSH CS
33 0029 1F        POP DS
34          ;;;;Заполнение буфера
35 002A B7 00      MOV BH , 0
36 002C 8A 1E 017Br  MOV BL , COUNT
37 0030 C6 87 00F8r 24 90  MOV BUF+[BX], '$'
38 0036 B4 09      MOV AH, 09H
39 0038 BA 00F8r    LEA DX, BUF
40 003B CD 21      INT 21h
41 003D E8 0095     CALL LFCR
42
43          ; СРАВНЕНИЕ
44 0040 FC        CLD ;очистка флагов
45 0041 B0 20      MOV AL, ''
46 0043 8B CB      MOV CX, BX ;в BX лежит COUNT
47 0045 BB 0000     MOV BX, 0
48 0048          find_space:
49 0048 38 87 00F8r  CMP [BUF+BX], AL ;Если пробел => конец ПАП1
50 004C 74 0C      JE space_found
51 004E 43        INC BX
52 004F E2 F7      loop find_space
53 0051 C6 06 00E2r FF 90  MOV space_index, 255 ;Значение, не имеющее смысла
54 0057 EB 05 90    JMP COMPARELEN
55
56 005A          space_found:
57 005A 88 1E 00E2r  MOV space_index, BL
58
59 005E          COMPARELEN:
60          ;Сравнение длины
```

```

61 005E B1 06          MOV CL,      6 ;Ларкин
62 0060 3A 0E 00E2r    CMP CL,      space_index ;if 6: len=6
63 0064 75 15          JNE NotEqual
64
65 0066                COMPARE:
66                    ;Длина верна, сравниваем побайтово
67 0066 FC              CLD
68 0067 BE 00F8r        LEA SI, BUF
69 006A BF 018Ar        LEA DI, PAR1
70 006D F3> A6          REPE CMPSB
71 006F 75 0A          JNZ NotEqual
72
73 0071 B4 09           MOV AH,      09H
74 0073 BA 0193r        MOV DX , OFFSET MSG1CORRECT ; правильно
75 0076 CD 21           INT 21h
76 0078 EB 0E 90        JMP Par2Processing
77
78 007B                NotEqual:
79 007B 38 0E 017Br     CMP COUNT, CL
80 007F 74 E5           JE COMPARE
81 0081 B4 09           MOV AH,      09H
82 0083 BA 01B4r        MOV DX , OFFSET MSG1WRONG ; Ошибка
83 0086 CD 21           INT 21h
84
85 0088                Par2Processing:
86 0088 80 3E 00E2r FF 90 CMP space_index, 255
87 008E 74 14           JE NPAR2
88 0090 B0 20           MOV AL,      ''
89 0092 B9 0099         MOV CX,      COUNT-space_index
90 0095 B7 00           MOV BH,      0
91 0097 8A 1E 00E2r     MOV BL,      space_index
92 009B                find_par2:
93 009B 38 87 00F8r     CMP [BUF+BX], AL ;Если не пробел => есть ПАР2
94 009F 75 0D           JNZ YPAR2
95 00A1 43              INC BX
96 00A2 E2 F7           loop find_par2
97 00A4                NPAR2:
98 00A4 B4 09           MOV AH,      09H
99 00A6 BA 01EEr        LEA DX,      MSGNPAR2 ; Параметра 2 нет
100 00A9 CD 21          INT 21h
101 00AB EB 08 90        JMP PARPRINT
102
103 00AE                YPAR2:
104 00AE B4 09           MOV AH,      09H
105 00B0 BA 01D4r        LEA DX,      MSGYPAR2 ; Параметр 2 есть
106 00B3 CD 21          INT 21h
107
108 00B5                PARPRINT: ;ПЕЧАТЬ АРГУМЕНТОВ
109 00B5 E8 001D         CALL LFCR
110 00B8 B4 09           MOV AH,      09H
111 00BA BA 017Cr        MOV DX , OFFSET MSGPRINT ; ПРОПЕЧАТКА слово
112 00BD CD 21          INT 21h
113                ;;;;;;;;;;;;;;;;;
114 00BF B4 09           MOV AH,09
115 00C1 BA 00F8r        MOV DX , OFFSET BUF
116 00C4 CD 21          INT 21h
117
118                ; Ожидание завершения программы
119 00C6 B4 01           MOV AH,      01H
120 00C8 CD 21          INT 021H
121                ; Выход из прораммы
122 00CA B4 4C           MOV AH,      4Ch
123 00CC B0 00           MOV AL,      0
124 00CE CD 21          INT 21H
125
126

```

```

127 ; Процедуры программы
128 ;;;;;;;;;; cmpsb
129 ; Процедура вывода символа DL
130 00D0 PUTCH PROC
131 00D0 B4 02 MOV AH, 2
132 00D2 CD 21 INT 21H
133 00D4 C3 RET
134 00D5 PUTCH ENDP
135 ; Процедура перевода строки
136 00D5 LFCR PROC
137 00D5 B2 0A MOV DL, 10
138 00D7 E8 FFF6 CALL PUTCH
139 00DA B2 0D MOV DL, 13
140 00DC E8 FFF1 CALL PUTCH
141 00DF C3 RET
142 00E0 LFCR ENDP
143
144 00E0 0000 save_psp dw 0
145 00E2 00 space_index db 0
146 00E3 8E E8 A8 A1 AA A0 20+ MSG1 DB 'Ошибка параметров!', 10,13,$'
147 AF A0 E0 A0 AC A5 E2+
148 E0 AE A2 21 0A 0D 24
149 00F8 7F*(00) BUF DB 127 DUP (0)
150 0177 24 0A 0D 24 DB '$', 10,13,$'
151 017B 00 COUNT DB 0
152 017C 8F 90 8E 8F 85 97 80+ MSGPRINT db 'ПРОПЕЧАТКА!',10,13,$'
153 92 8A 80 21 0A 0D 24
154 018A 8B A0 E0 AA A8 AD 0A+ PAR1 DB 'Ларкин',10,13,$'
155 0D 24
156 0193 8F A5 E0 A2 EB A9 20+ MSG1CORRECT db 'Первый параметр верен = Ларкин',10,13,$'
157 AF A0 E0 A0 AC A5 E2+
158 E0 20 A2 A5 E0 A5 AD+
159 20 3D 20 8B A0 E0 AA+
160 A8 AD 0A 0D 24
161 01B4 8F A5 E0 A2 EB A9 20+ MSG1WRONG DB 'Первый параметр неправильный!',10,13,$'
162 AF A0 E0 A0 AC A5 E2+
163 E0 20 AD A5 AF E0 A0+
164 A2 A8 AB EC AD EBA9+
165 21 0A 0D 24
166 01D4 82 E2 AE E0 AE A9 20+ MSGYPAR2 DB 'Второй параметр ЕСТЬ !',10,13,$'
167 AF A0 E0 A0 AC A5 E2+
168 E0 20 85 91 92 9C 20+
169 21 20 0A 0D 24
170 01EE 82 E2 AE E0 AE A3 AE+ MSGNPAR2 DB 'Второго параметра НЕТ !',10,13,$'
171 20 AF A0 E0 A0 AC A5+
172 E2 E0 A0 20 8D 85 92+
173 20 21 20 0A 0D 24
174 0209 MYCODE ENDS
175 END START

```

Symbol Table

Symbol Name	Type	Value	Cref	(defined at #)
??DATE	Text	"05/02/24"		
??FILENAME	Text	"laba6 "		
??TIME	Text	"00:05:42"		
??VERSION	Number	030A		
@CPU	Text	0101H		
@CURSEG	Text	MYCODE	#2	#6
@FILENAME	Text	LABA6		
@WORDSIZE	Text	2	#2	#6
BUF	Byte	MYCODE:00F8	30 37 39 49 68 93 115	#149
COMPARE	Near	MYCODE:0066	#65	80
COMPARELEN	Near	MYCODE:005E	54	#59
COUNT	Byte	MYCODE:017B	21 36 79 89	#151

FIND_PAR2	Near	MYCODE:009B	#92 96
FIND_SPACE	Near	MYCODE:0048	#48 52
LFCR	Near	MYCODE:00D5	41 109 #136
MSG1	Byte	MYCODE:00E3	#146
MSG1CORRECT	Byte	MYCODE:0193	74 #156
MSG1WRONG	Byte	MYCODE:01B4	82 #161
MSGNPAR2	Byte	MYCODE:01EE	99 #170
MSGPRINT	Byte	MYCODE:017C	111 #152
MSGYPAR2	Byte	MYCODE:01D4	105 #166
NOTEQUAL	Near	MYCODE:007B	63 71 #78
NPAR2	Near	MYCODE:00A4	87 #97
PAR1	Byte	MYCODE:018A	69 #154
PAR2PROCESSING	Near	MYCODE:0088	76 #85
PARPRINT	Near	MYCODE:00B5	101 #108
PATCH	Near	MYCODE:00D0	#130 138 140
SAVE_PSP	Word	MYCODE:00E0	17 28 #144
SPACE_FOUND	Near	MYCODE:005A	50 #56
SPACE_INDEX	Byte	MYCODE:00E2	53 57 62 86 89 91 #145
START	Near	MYCODE:0000	#8 175
YPAR2	Near	MYCODE:00AE	94 #103

Groups & Segments	Bit	Size	Align	Combine	Class	Cref	(defined at #)
MYCODE	16	0209	Para	none	CODE	#6 7 7	
STSEG	16	0200	Para	Stack	STACK	#2 7	

7. Скриншот программы в TD.exe

The screenshot displays the TD.exe debugger interface. The left pane shows the assembly code for the 'MYCODE' segment, starting with 'START: ; Загрузка регистров'. The right pane shows the CPU 80486 registers (ax, bx, cx, dx, si, di, bp, sp, ip) and segment registers (cs, ds, ss, ip). The bottom status bar shows the current instruction: 'buf byte [127]'.

Assembly Code (Left Pane):

```

; IP #6 2024, Ларкин Борис ИУ5-41
STSEG SEGMENT STACK 'STACK'
    DW 256 DUP(0)
STSEG ENDS

MYCODE SEGMENT 'CODE'
    ASSUME CS:MYCODE, DS:MYCODE, SS:S
START:
; Загрузка регистров
    PUSH CS
    POP DS
    PUSH DS
    POP ES
    MOV AH, 51H ; Получение ID текуще
    int 21H
    MOV ES, BX
    mov save_psp, ES
; получение счетчика ES

```

Registers (Right Pane):

Register	Value
ax	0000
bx	0000
cx	0000
dx	0000
si	0000
di	0000
bp	0000
sp	0200
ip	0000
cs	4EE9
ds	4EF9
ss	4F19
ip	0000

Status Bar: buf byte [127]

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

8. Результаты работы программы

Ларкин sth

Первый параметр верен = Ларкин

Второй параметр ЕСТЬ !

ПРОПЕЧАТКА!

Ларкин sth

9. Выводы по ЛР № 6

Разработан файл .ASM и соответствующие файлы приложения и листинга на языке Ассемблер. Программа выполняется один раз: получает на вход параметры из командной строки, обрабатывает их наличие и совпадение с предварительно заданными, выводит сами параметры и результаты их обработки на экран.