

Защищено:  
Большаков С.А.

Демонстрация ЛР:  
Большаков С.А.

"\_\_" \_\_\_\_\_ 2020 г.

"\_\_" \_\_\_\_\_ 2020 г.

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

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

9  
(количество листов)  
Вариант № 1

ИСПОЛНИТЕЛЬ:

студент группы **ИУ5Ц-62**

**Гусев С.Р.**

\_\_\_\_\_  
(подпись)

"\_\_" \_\_\_\_\_ 2020 г.

## СОДЕРЖАНИЕ

|  |   |
|--|---|
| 1. Цель выполнения лабораторной работы №6.....     | 3 |
| 2. Порядок и условия проведения работы №6 .....    | 3 |
| 3. Описание ошибок, возникших при отладке №6 ..... | 4 |
| 4. Блок-схема программы .....                      | 4 |
| 5. Текст программы на языке Ассемблера .....       | 5 |
| 6. Результаты работы программы.....                | 9 |
| 7. Выводы по ЛР №6.....                            | 9 |

## 1. Цель выполнения лабораторной работы №6

Разработать и отладить программу на языке Ассемблер для ввода, анализа (расшифровки, фактически грамматического разбора) и распечатки параметров командной строки, которые задаются при запуске программы (параметры размещаются в области PSP со смещением 081h). Нужно также описать в БНФ синтаксис запуска вашей программы с параметрами в командной строке для разработанной программы. Для этого надо изучить раздел 4 методических указаний к ЛР. Программа должна быть скомпонована в виде \*.EXE - исполнимого файла.

## 2. Порядок и условия проведения работы №6

Студенты разрабатывают работоспособную программу на языке Ассемблер по заданию ЛР, выполняют следующие действия (порядок выполнения работы):

- Знакомятся и осмысливают задание на ЛР.
- Разрабатывают алгоритм реализации задачи (блок-схема программы можно оформить в MS VISIO, MS WORD или на листе бумаги).
- Выполняют написание текста программы на языке Ассемблер и вводят его в отдельном текстовом редакторе (можно использовать текстовый редактор ASM\_ED.EXE – есть на сайте) или в интегрированной оболочке (например, в QC).
- Выполняют отладку программы в отладчике (TD.EXE), демонстрируют преподавателю умение работать в отладчике, выполняя различные действия (выполнение по шагам, просмотр данных и т.д.).
- Формируют исполнимый модуль программы заданного типа (COM или EXE см. задание).
- Демонстрируют преподавателю работоспособную программу.
- По требованию преподавателя, если нужно, вносят изменения в программу и демонстрируют знание действий необходимых для создания исполнимого модуля (это предварительная сдача ЛР).
- Оформляют отчет по данной лабораторной работе в соответствии с требованиями приведенными ниже и на основе шаблона отчетов по ЛР.
- На основе отчета по ЛР (распечатанного) выполняют защиту ЛР у преподавателя (ответы на контрольные вопросы), после чего в журнале отмечается: срок сдачи ЛР, срок защиты ЛР, оценка за защиту данной ЛР и выполнение дополнительных требований к ЛР. На защите задаются вопросы, перечисленные в разделе “Контрольные вопросы по каждой ЛР и общие вопросы”, а также вопросы по листингу программы (отметьте себе, не по тексту программы, по листингу).

Работа считается выполненной полностью и в срок, если студент полностью сдал и защитил отчет ЛР в срок. Если студент сделал работу с дополнительными требованиями, то это обязательно отмечается в журнале ЛР и учитывается в оценке при подведении итогов семестра по данной дисциплине и на экзамене. Если студент выполнил все ЛР с дополнительными требованиями и получил отметки не ниже “хорошо”, то на зачете он освобождается от решения задачи (задачи на зачете заключаются в написании процедуры на языке Ассемблер или командного файла) и может претендовать на получение автоматической оценки по курсовой работе - ОТЛИЧНО, при своевременной ее сдаче.

Если преподаватель обнаруживает (поверьте, сделать очень просто), что программа и отчет по ней сделаны самостоятельно (проще - списаны), то он отмечает данный факт в журнале, а на зачете в этом случае задаются дополнительные вопросы по лабораторным

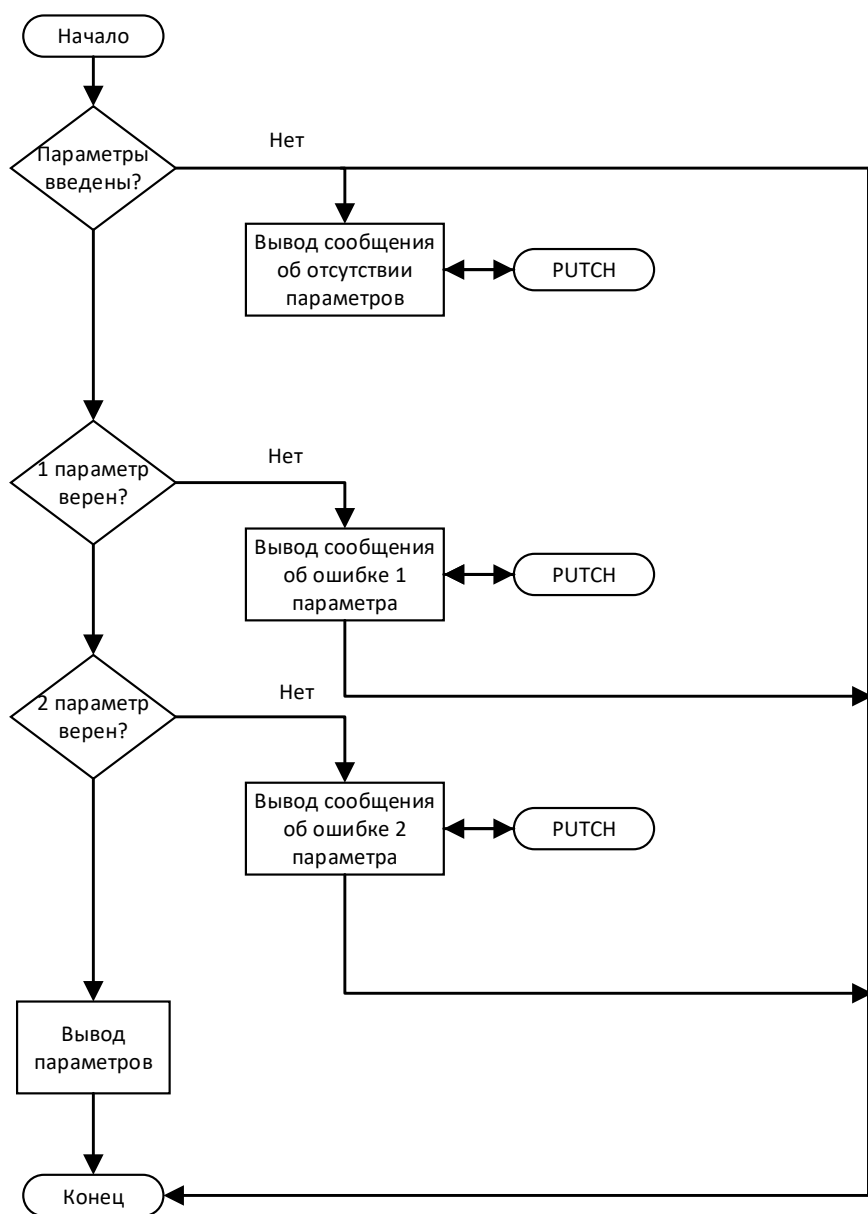
работам, методическому пособию и материалам лекций. Кроме того, студент в этом случае не вправе рассчитывать на оценку по курсовой работе выше чем – удовлетворительно.

Для выполнения цикла лабораторных работ по курсу полезно познакомиться с указанными выше разделами методических указаний к ЛР, подготовленных преподавателем (отдельный документ – есть на сайте – оранжевая кнопка).

### 3. Описание ошибок, возникших при отладке №6

| № п/п | Проявление ошибки                                | Причина ошибки                                | Способ устранения                                     |
|-------|--|---|---|
| 1.    | Ошибка компиляции                                | Неправильное использование оператора 'je'     | Использовать оператор правильно                       |
| 2.    | Зависание программы, бесконечный вывод в консоль | Неправильно указано количество итераций цикла | Занести верное количество итераций цикла в регистр cx |

### 4. Блок-схема программы



## 5. Текст программы на языке Ассемблера

Turbo Assembler Version 3.1  
V:\LR\LR6\LR6.asm

05/04/20 10:20:53

Page 1

```
1 0000 MYCODE SEGMENT 'CODE'
2 ASSUME CS:MYCODE, DS:MYCODE
3
4 0000 4E 6F 20 73 65 63 6F+ NoStr DB 'No second param$'
5 6E 64 20 70 61 72 61+
6 6D 24
7 0010 59 65 73 20 73 65 63+ YesStr DB 'Yes second param$'
8 6F 6E 64 20 70 61 72+
9 61 6D 24
10 0021 47 75 73 65 76 ME DB 'Gusev'
11 0026 20 20 20 20 20 20 20+ mystr DB ' $'
12 20 20 20 20 20 20 20+
13 20 20 20 20 20 20 24
14 003B 45 72 72 6F 72 3A 46+ NCoolStr Db 'Error:First param not cool $'
15 69 72 73 74 20 70 61+
16 72 61 6D 20 6E 6F 74+
17 20 63 6F 6F 6C 20 24
18 0057 46 69 72 73 74 20 70+ CoolStr Db 'First param cool $'
19 61 72 61 6D 20 63 6F+
20 6F 6C 20 24
21 0069 0D 0A 24 CLR DB 0dh,0ah,'$'
22
23 006C CLRF PROC
24 006C BA 0069r MOV DX, OFFSET CLR
25 006F B4 09 MOV AH, 09H
26 0071 CD 21 INT 021H
27 0073 C3 RET
28 0074 CLRF ENDP
29
30 0074 GETCH PROC
31 0074 B4 08 MOV AH, 08H
32 0076 CD 21 INT 21H
33 0078 C3 RET
34 0079 GETCH ENDP
35
36 0079 PUTCH PROC
37 0079 B4 02 MOV AH, 02H
38 007B CD 21 INT 21H
39 007D C3 RET
40 007E PUTCH ENDP
41
42 007E CLRSCR PROC
43 007E B8 0003 MOV AX, 03H
44 0081 CD 10 INT 10H
45 0083 C3 RET
46 0084 CLRSCR ENDP
47
48 0084 OUTSTR PROC
49 0084 B4 09 MOV AH, 09H
50 0086 CD 21 INT 021H
51 0088 C3 RET
52 0089 OUTSTR ENDP
53
54
55 0089 START:
56 0089 0E PUSH CS
57 008A 1F POP DS
```

```

58 008B E8 FFF0          CALL CLRSCR
59
60 008E 26: 8A 0E 0080    MOV CL,ES:[80h]
61 0093 80 F9 00          CMP CL, 0 ;нев пааамеваов - влеодим
62 0096 75 03             JNE good ;ебли ебвм пааамевал
63 0098 E9 008F          JMP FIN
64 009B                   good:
65 009B B6 00             MOV DH,0
66 009D 51               PUSH CX
67 009E 80 E9 01          SUB CL, 1
68 00A1 B0 01             MOV AL,1
69 00A3 BE 0082           MOV SI,82h ;облабвм пааамеваов
70 00A6 BB 0026r          mov BX, offset mystr
71 00A9                   circle:
72 00A9 26: 8A 14          MOV DL, ES:[SI]
73 00AC 88 17             MOV [BX], DL
74 00AE 80 FA 20          CMP DL,' '
75 00B1 75 02             JNE loop1 ;ебли не аавно
76 00B3 B6 01             MOV DH,1
77 00B5                   loop1:
78 00B5 83 C3 01          ADD BX,1
79 00B8 83 C6 01          ADD SI,1
80 00BB 80 FE 00          CMP DH,0
81 00BE 75 02             JNE next ;ебли не аавно
82 00C0 04 01             ADD AL,1
83 00C2                   next:
84 00C2 E2 E5             LOOP circle
85
86 00C4 1E               PUSH DS
87 00C5 07               POP ES
88 00C6 8A C8            MOV CL, AL
89 00C8 50               PUSH AX;аазмеа 1 пааам б паобелом
90 00C9 80 E9 01          SUB CL, 1
91 00CC BE 0021r          MOV SI,OFFSET ME
92 00CF BF 0026r          MOV DI,OFFSET mystr
93 00D2 80 F9 05          CMP CL, 5;колво бгкв в дамилии
94 00D5 75 18             JNE err0r
95 00D7                   srav: ;баавнение ме с mystr побимволмно
96 00D7 A6               CMPSB
97 00D8 75 15             JNE err0r
98 00DA E2 FB            LOOP srav
99 00DC 80 F9 00          CMP CL,0
100 00DF 75 0E            JNE err0r
101
102 00E1 BA 0057r          MOV DX, OFFSET CoolStr
103 00E4 E8 FF9D          CALL OUTSTR
104 00E7 B2 3D            MOV DL,'='
105 00E9 E8 FF8D          CALL PUTCH
106 00EC EB 0C 90          JMP yes
107 00EF                   err0r:
108 00EF BA 003Br          MOV DX, OFFSET NCoolStr
109 00F2 E8 FF8F          CALL OUTSTR
110 00F5 B2 3D            MOV DL,'='
111 00F7 E8 FF7F          CALL PUTCH
112 00FA                   yes:
113 00FA BB 0026r          MOV BX, OFFSET mystr
114 00FD                   loop2:

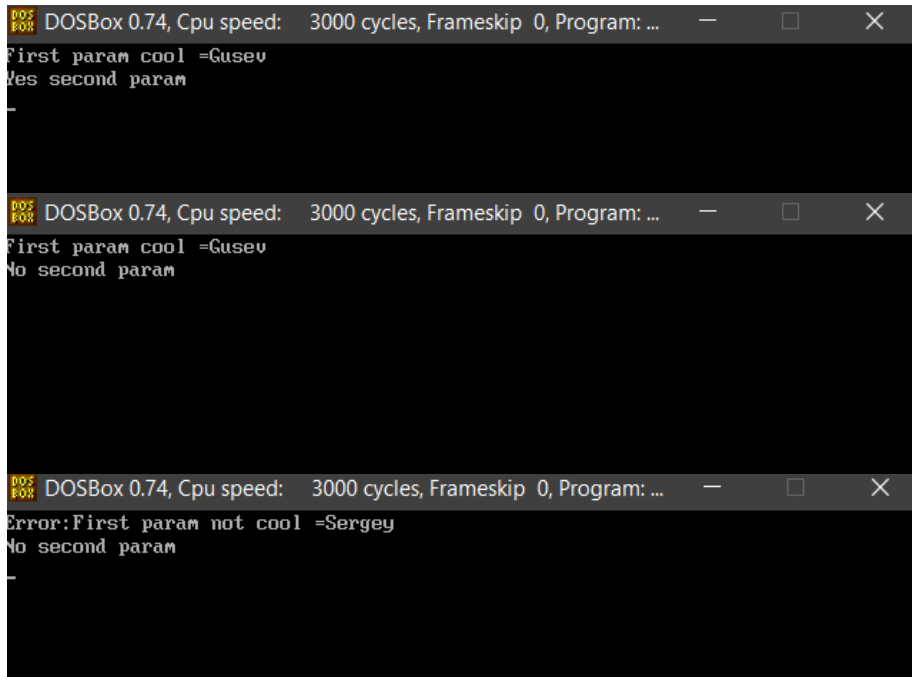
```

```
115 00FD 8A 17          MOV DL,[BX]
116 00FF 80 FA 20      CMP DL,''
117 0102 74 08          JE goaway
118 0104 E8 FF72        CALL PUTCH
119 0107 83 C3 01      ADD BX,1
120 010A EB F1          JMP loop2
121 010C                goaway:
122 010C E8 FF5D        CALL CLRF
123 010F 58             POP AX
124 0110 59             POP CX
125 0111 3A C8          CMP CL, AL ;cl-длина ком бваоки al-1 пааам
126 0113 75 09          JNE ok
127 0115 BA 0000r       MOV DX, OFFSET NoStr
128 0118 E8 FF69        CALL OUTSTR
129 011B EB 07 90       JMP nok
130 011E                ok:
131 011E BA 0010r       MOV DX, OFFSET YesStr
132 0121 E8 FF60        CALL OUTSTR
133 0124                nok:
134 0124 E8 FF45        CALL CLRF
135 0127 E8 FF4A        CALL GETCH
136
137 012A                FIN;;влеод
138 012A B0 00          MOV AL, 0
139 012C B4 4C          MOV AH, 4CH
140 012E CD 21          INT 21H
141
142 0130                MYCODE ENDS
143                    END START
```

| Symbol Name       | Type     | Value            | Cref      | (defined at #) |                 |                |  |
|-------------------|----------|------------------|-----------|----------------|-----------------|----------------|--|
| ??DATE            |          | Text "05/04/20"  |           |                |                 |                |  |
| ??FILENAME        | Text     | "LR6"            |           |                |                 |                |  |
| ??TIME            | Text     | "10:20:52"       |           |                |                 |                |  |
| ??VERSION         | Number   | 030A             |           |                |                 |                |  |
| @CPU              | Text     | 0101H            |           |                |                 |                |  |
| @CURSEG           | Text     | MYCODE           |           | #1             |                 |                |  |
| @FILENAME         | Text     | LR6              |           |                |                 |                |  |
| @WORDSIZE         | Text     | 2                |           | #1             |                 |                |  |
| CIRCLE            |          | Near MYCODE:00A9 |           | #71            | 84              |                |  |
| CLR               | Byte     | MYCODE:0069      | #21       | 24             |                 |                |  |
| CLRF              | Near     | MYCODE:006C      | #23       | 122            | 134             |                |  |
| CLRSCR            |          | Near MYCODE:007E |           | #42            | 58              |                |  |
| COOLSTR           |          | Byte MYCODE:0057 |           | #18            | 102             |                |  |
| ERROR             | Near     | MYCODE:00EF      | 94 97 100 | #107           |                 |                |  |
| FIN               | Near     | MYCODE:012A      | 63 #137   |                |                 |                |  |
| GETCH             | Near     | MYCODE:0074      | #30       | 135            |                 |                |  |
| GOAWAY            |          | Near MYCODE:010C |           | 117            | #121            |                |  |
| GOOD              | Near     | MYCODE:009B      | 62 #64    |                |                 |                |  |
| LOOP1             | Near     | MYCODE:00B5      | 75 #77    |                |                 |                |  |
| LOOP2             | Near     | MYCODE:00FD      | #114      | 120            |                 |                |  |
| ME                | Byte     | MYCODE:0021      | #10       | 91             |                 |                |  |
| MYSTR             | Byte     | MYCODE:0026      | #11       | 70 92          | 113             |                |  |
| NCOOLSTR          | Byte     | MYCODE:003B      | #14       | 108            |                 |                |  |
| NEXT              | Near     | MYCODE:00C2      | 81 #83    |                |                 |                |  |
| NOK               | Near     | MYCODE:0124      | 129 #133  |                |                 |                |  |
| NOSTR             | Byte     | MYCODE:0000      | #4        | 127            |                 |                |  |
| OK                | Near     | MYCODE:011E      | 126 #130  |                |                 |                |  |
| OUTSTR            |          | Near MYCODE:0084 |           | #48            | 103 109 128 132 |                |  |
| PUTCH             | Near     | MYCODE:0079      | #36       | 105 111 118    |                 |                |  |
| SRV               | Near     | MYCODE:00D7      | #95       | 98             |                 |                |  |
| START             | Near     | MYCODE:0089      | #55       | 143            |                 |                |  |
| YES               | Near     | MYCODE:00FA      | 106 #112  |                |                 |                |  |
| YESSTR            |          | Byte MYCODE:0010 |           | #7             | 131             |                |  |
| Groups & Segments | Bit Size | Align            | Combine   | Class          | Cref            | (defined at #) |  |
| MYCODE            | 16       | 0130             | Para none | CODE           | #1 2 2          |                |  |



## 6. Результаты работы программы



## 7. Выводы по ЛР №6

В ходе этой лабораторной работы я научился работать с параметрами командной строки в ассемблере.