



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет имени
Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа №1 по дисциплине "Операционные системы"

Тема Дизассемблирование INT 8h

Студент Рядинский К. В.

Группа ИУ7-53Б

Преподаватель Рязанова Н.Ю.

Москва — 2021 г.

1. Полученный дизассемблированный код

1.1. Листинг INT8h

```
1      020A:0746      ;*      call      sub_1      ; (07B9)
2 020A:0746      db      0E8h, 70h, 00h
3
4 ; сохранение регистров
5 020A:0749      push     es
6 020A:074A      push     ds
7 020A:074B      push     ax
8 020A:074C      push     dx
9 020A:074D      mov      ax,40h
10 020A:0750      mov      ds,ax
11 020A:0752      xor      ax,ax      ; Zero register
12 020A:0754      mov      es,ax
13
14 ; инкремент счетчика таймера реального времени
15 020A:0756      inc      word ptr ds:[6Ch]
16 ; (0040:006C=2E56h), по этому адресу располагается счетчик реального времени
17 020A:075A      jnz      loc_1      ; Jump if not zero
18 ;если значение в 0040:006C равно нулю, то инкрементируются старшие 2 байта
19 020A:075C      inc      word ptr ds:[6Eh]      ; (0040:006E=2)
20
21 ; сброс счетчика таймера реального времени, если наступили новые сутки
22 020A:0760      loc_1:
23 020A:0760      cmp      word ptr ds:[6Eh],18h      ; (0040:006E=2)
24 020A:0765      jne      loc_2      ; Jump if not equal
25 020A:0767      cmp      word ptr ds:[6Ch],0B0h      ; (0040:006C=2E56h)
26 020A:076D      jne      loc_2      ; Jump if not equal
27 ; прошло более 24 часов с момента запуска таймера, обнуление счетчика
28 020A:076F      mov      word ptr ds:[6Eh],ax      ; (0040:006E=2)
29 020A:0772      mov      word ptr ds:[6Ch],ax      ; (0040:006C=2E56h)
30 020A:0775      mov      byte ptr ds:[70h],1      ; (0040:0070=0)
31 ; в 0040:0070 хранится перепополнения таймера (переход через 24 часа)
32 020A:077A      or      al,8
33
34
35
36 ; декремент значения времени до выключения моторчика дисководов
37 020A:077C      loc_2:
38 020A:077C      push     ax
39 020A:077D      dec      byte ptr ds:[40h]      ; (0040:0040=5Bh)
40 ; ячейка с адресом 0040:0040 содержит время, оставшееся до выключения двигателя
41 020A:0781      jnz      loc_3      ; Jump if not zero
42 ; посыл команды на отключение моторчика дисководов
43 020A:0783      and      byte ptr ds:[3Fh],0F0h      ; (0040:003F=0)
44 ; в 0040:003F хранится состояние моторчика дисководов
45 020A:0788      mov      al,0Ch
46 020A:078A      mov      dx,3F2h
47 020A:078D      out      dx,al      ; port 3F2h, disk control output
48
49 ; проверка на возможность вызова маскируемых прерываний
50 020A:078E      loc_3:
51 020A:078E      pop      ax
52 020A:078F      test     word ptr ds:[314h],4      ; (0040:0314=3200h)
53 ; ячейка с адресом 0040:0314 содержит информацию о значениях флагов (Проверка
    parity flag)
```

```

54 020A:0795          jnz loc_4          ; Jump if not zero
55 020A:0797          lahf              ; Load ah from flags
56 ; загрузка младшего байта регистра флагов в AH
57 ; обмен ah и al
58 020A:0798          xchg     ah, al
59 020A:079A          push     ax
60 ; косвенный вызов пользовательского прерывания
61 ; Вызов 1Ch с помощью адреса в таблице векторов прерывания
62 ; При вызове через int произойдет сохранения регистра флагов в стек,
63 ; а в случае вызова через call на его месте будет лежать сохраненный до ax
64 ; который по выходе из 1Ch будет установлен в флаги через iret
65 020A:079B          call     dword ptr es:[70h] ; (0000:0070=6ADh)
66 020A:07A0          jmp     short loc_5          ; (07A5)
67 020A:07A2          nop
68
69 ; вызов пользовательского прерывания
70 020A:07A3          loc_4:
71 020A:07A3          int     1Ch              ; Timer break (call each 18.2ms)
72
73 ; сброс контроллера прерываний
74 020A:07A5          loc_5:
75 020A:07A5          call     sub_1              ; (07B9)
76 020A:07A8          mov     al, 20h            ; ' '
77 020A:07AA          out     20h, al           ; port 20h, 8259-1 int command
78                                     ; al = 20h, end of interrupt
79 ; восстановление регистров
80 020A:07AC          pop     dx
81 020A:07AD          pop     ax
82 020A:07AE          pop     ds
83 020A:07AF          pop     es
84 ; переход по метке для завершения работы прерывания
85 020A:07B0          jmp     $-164h
86 020A:07B3          db     0C4h
87                                     ;* No entry point to code
88 020A:07B4          les     cx, dword ptr ds:[93E9h] ; (0000:93E9=5A14h) Load 32 bit
89 ptr
89 020A:07B8          db     0FEh

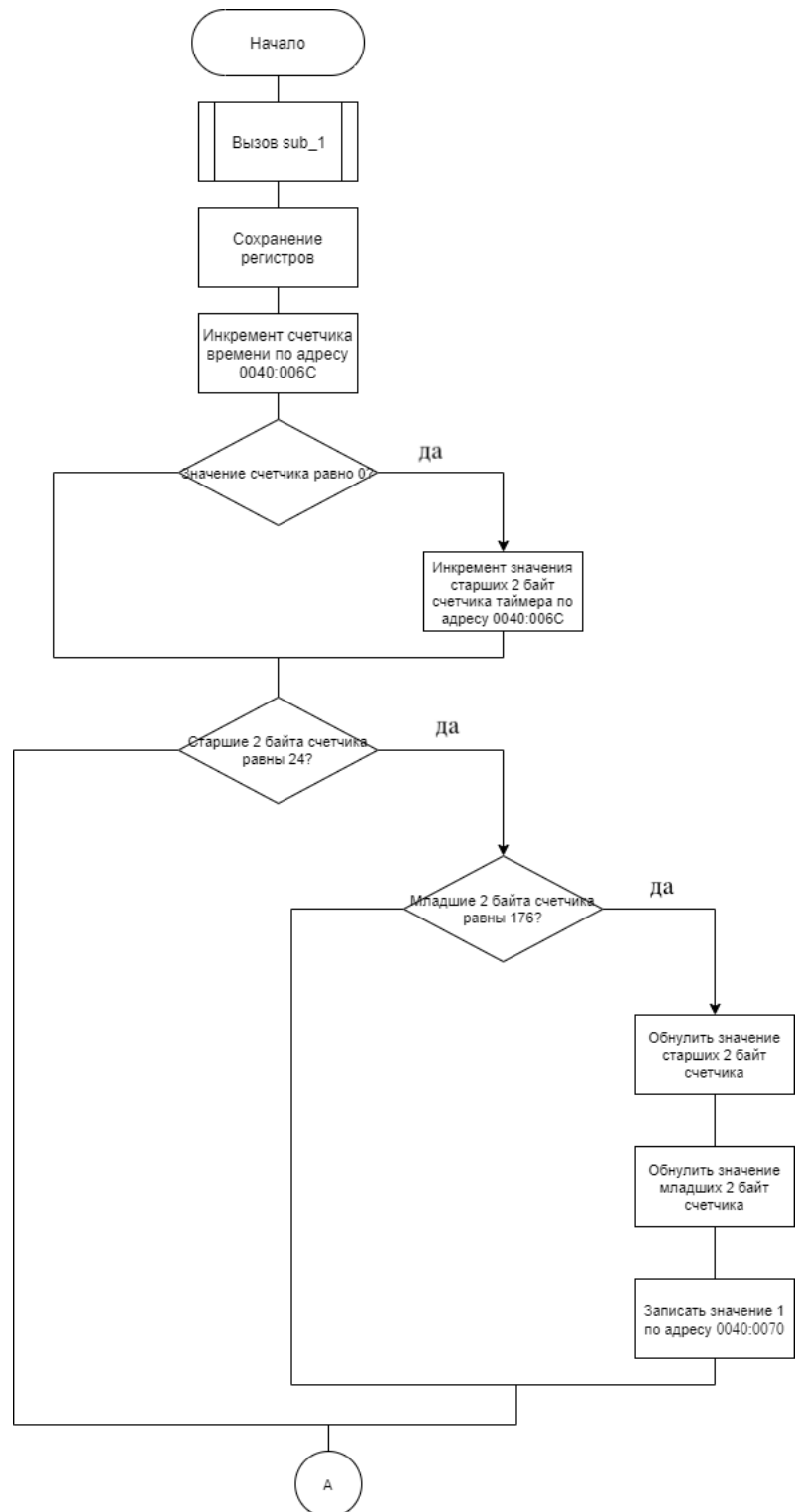
```

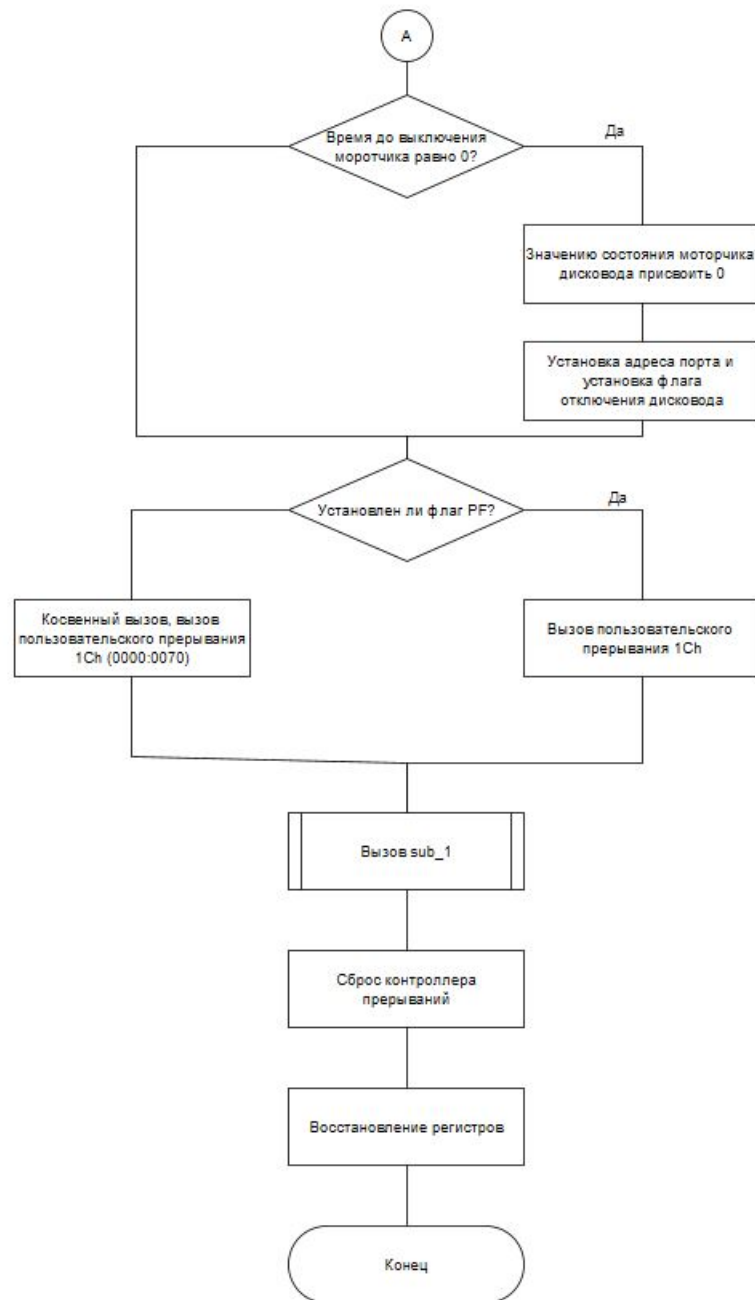
1.2. Листинг процедуры sub_1

```
1  sub_1      proc    near
2      ; сохранение регистров и загрузка регистра флагов
3      020A:07B9      push    ds
4      020A:07BA      push    ax
5      020A:07BB      mov     ax,40h
6      020A:07BE      mov     ds,ax
7      020A:07C0      lahf                    ; Load ah from flags
8
9      ; проверка на возможность вызова маскируемых прерываний
10     020A:07C1      test     word ptr ds:[314h],2400h; (0040:0314=3200h)
11     020A:07C7      jnz     loc_7          ; Jump if not zero
12     ; Сброс IEF (9 бит) lock для того, чтобы команда была неделимой
13     020A:07C9      lock and word ptr ds:[314h],0FDFFh;
14     (0040:0314=3200h)
15     ; установка флага IF в ноль
16
17     ; сохранение флагов и восстановление регистров
18     020A:07D0      loc_6:
19     020A:07D0      sahf                    ; Store ah into flags
20     020A:07D1      pop     ax
21     020A:07D2      pop     ds
22     020A:07D3      jmp     short loc_8      ; (07D8)
23
24     ; запрет на вызов маскируемых прерываний
25     020A:07D5      loc_7:
26     020A:07D5      cli                    ; Disable interrupts
27     ; сбрасывает interrupt flag (IF). Когда этот флаг сброшен процессор игнориру
28     ; прерывания (кроме NMI) от внешних устройств.
29     020A:07D6      jmp     short loc_6      ; (07D0)
30
31     ; завершение процедуры
32     020A:07D8      loc_8:
33     020A:07D8      retn
34     sub_1      endp
```

2. Схема алгоритмов

2.1. Схема алгоритма обработчика INT8h





2.2. Схема алгоритма процедуры sub_1

