

ОТЧЕТ

По лабораторной работе №1 (часть 1)

По курсу «Операционные системы»

Тема: «Прерывание таймера INT 08h и его функции»

Студент:

Щербатюк Д.С.

Группа:

ИУ7-54

Преподаватель:

Рязанова Н. Ю.

Листинг прерывания INT 08H

; Необходимые константы

; The following equates show data references outside the range of the program.

```
= 0070      data_1e      equ    70h              ; (0000:0070=6ADh)
= 003F      data_2e      equ    3Fh              ; (0040:003F=0)
= 0040      data_3e      equ    40h              ; (0040:0040=0B0h)
= 006C      data_4e      equ    6Ch              ; (0040:006C=0D787h)
= 006E      data_5e      equ    6Eh              ; (0040:006E=15h)
= 0070      data_6e      equ    70h              ; (0040:0070=0)
= 0314      data_7e      equ    314h             ; *(0040:0314=3200h)
= B000      data_14e     equ    0B000h           ; (B000:B000=20h)
```

; Вход в прерывание, вызов sub_1 по адресу 07B9, пролог функции (сохранение регистров)

```
020C:0746 E8 0070          call    sub_1          ; (07B9)
020C:0749 06              push    es
020C:074A 1E              push    ds
020C:074B 50              push    ax
020C:074C 52              push    dx
020C:074D B8 0040          mov     ax,40h
020C:0750 8E D8          mov     ds,ax
020C:0752 33 C0          xor     ax,ax          ; Zero register
020C:0754 8E C0          mov     es,ax
```

; Инкремент счётчика суточного времени и переход на проверку начала новых суток

```
020C:0756 FF 06 006C          inc     word ptr ds:data_4e      ; (0040:006C=0D787h)
020C:075A 75 04          jnz     loc_1                  ; Jump if not zero
020C:075C FF 06 006E          inc     word ptr ds:data_5e      ; (0040:006E=15h)
```

; Сброс счётчика суточного времени, если наступили новые сутки

```
020C:0760          loc_1:                      ; xref 020C:075A
020C:0760 83 3E 006E 18      cmp     word ptr ds:data_5e,18h ; (0040:006E=15h)
020C:0765 75 15          jne     loc_2                  ; Jump if not equal
020C:0767 81 3E 006C 00B0  cmp     word ptr ds:data_4e,0B0h ; (0040:006C=0D787h)
020C:076D 75 0D          jne     loc_2                  ; Jump if not equal
020C:076F A3 006E          mov     ds:data_5e,ax          ; (0040:006E=15h)
020C:0772 A3 006C          mov     ds:data_4e,ax          ; (0040:006C=0D787h)
020C:0775 C6 06 0070 01      mov     byte ptr ds:data_6e,1   ; (0040:0070=0)
020C:077A 0C 08          or      al,8
```

; Декремент счётчика времени до остановки дисководов

```
020C:077C          loc_2:                      ; xref 020C:0765, 076D
020C:077C 50              push    ax
020C:077D FE 0E 0040      dec     byte ptr ds:data_3e      ; (0040:0040=0B0h)
020C:0781 75 0B          jnz     loc_3                  ; Jump if not zero
```

; Посылка сигнала отключения моторчика дисководов

```
20C:0783 80 26 003F F0 and    byte ptr ds:data_2e,0F0h      ; (0040:003F=0)
020C:0788 B0 0C          mov     al,0Ch
020C:078A BA 03F2      mov     dx,3F2h
020C:078D EE              out     dx,al          ; port 3F2h, disk0 contrl output
```

; Проверка на возможность вызова маскируемых прерываний

```
020C:078E          loc_3:                ; xref 020C:0781
020C:078E  58                pop     ax
020C:078F  F7 06 0314 0004    test   word ptr ds:data_7e,4    ; (0040:0314=3200h)
020C:0795  75 0C                jnz    loc_4                    ; Jump if not zero
020C:0797  9F                lahf                     ; Load ah from flags
020C:0798  86 E0                xchg   ah,al
020C:079A  50                push   ax
020C:079B  26: FF 1E 0070      call   dword ptr es:data_1e    ; (0000:0070=6ADh)
020C:07A0  EB 03                jmp     short loc_5            ; (07A5)
020C:07A2  90                nop
```

; Вызов пользовательского прерывания по таймеру

```
020C:07A3          loc_4:                ; xref 020C:0795
020C:07A3  CD 1C                int     1Ch                    ; Timer break (call each 18.2ms)
020C:07A5          loc_5:                ; xref 020C:07A0
020C:07A5  E8 0011            call    sub_1                  ; (07B9)
```

; Отправка сигнала «end of interrupt» контроллеру прерываний

```
020C:07A8  B0 20                mov     al,20h                ; ' '
020C:07AA  E6 20                out     20h,al                ; port 20h, 8259-1 int command
                                ; al = 20h, end of interrupt
```

; Завершение обработки прерывания

```
020C:07AC  5A                pop     dx
020C:07AD  58                pop     ax
020C:07AE  1F                pop     ds
020C:07AF  07                pop     es
```

; Переход по метке в сторону завершения работы прерывания

```
020C:07B0  E9 FE99            jmp     $-164h
020C:07B3  C4 C4 0E E9 93 FE  db     0C4h,0C4h, 0Eh,0E9h, 93h,0FEh (iret)
```

Листинг подпрограммы sub_1

```
                                sub_1      proc near
020C:07B9  1E                push    ds
020C:07BA  50                push    ax
020C:07BB  B8 0040            mov     ax,40h
020C:07BE  8E D8             mov     ds,ax
020C:07C0  9F                lahf                     ; Load ah from flags
```

; Проверка на возможность вызова маскируемых прерываний

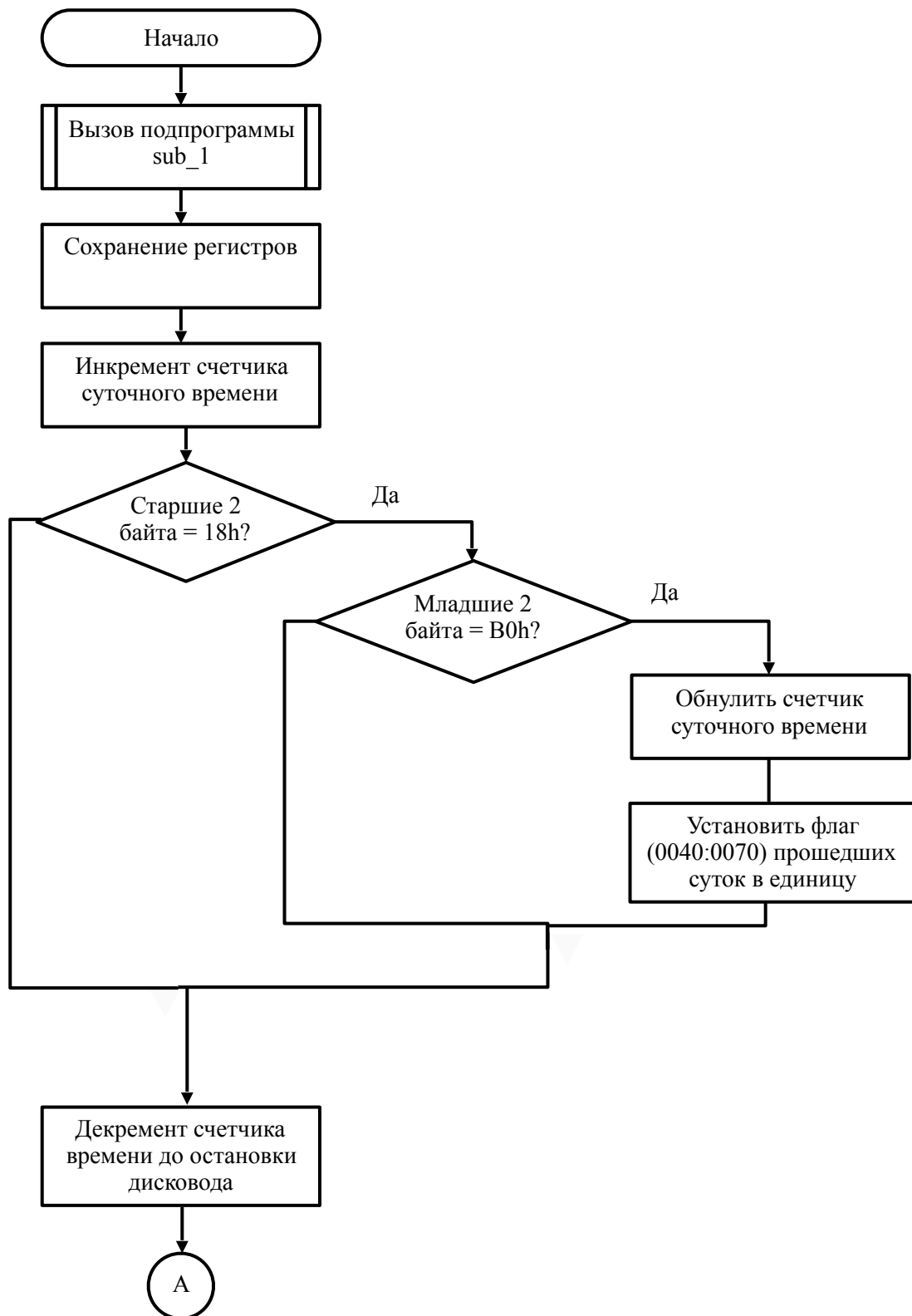
```
020C:07C1  F7 06 0314 2400    test   word ptr ds:data_7e,2400h; (0040:0314=3200h)
020C:07C7  75 0C                jnz    loc_7                    ; Jump if not zero
020C:07C9  F0> 81 26 0314 FDFF lock and word ptr ds:data_7e,0FDFFh ;
(0040:0314=3200h)
```

```
020C:07D0          loc_6:                ; xref 020C:07D6
020C:07D0  ú9E              sahf                     ; Store ah into flags
020C:07D1  58                pop     ax
020C:07D2  1F                pop     ds
020C:07D3  EB 03                jmp     short loc_ret_8        ; (07D8)
```

; Запрет на вызов прерываний

```
020C:07D5          loc_7:                ; xref 020C:07C7
020C:07D5  FA                cli                     ; Disable interrupts
020C:07D6  EB F8                jmp     short loc_6            ; (07D0)
020C:07D8          loc_ret_8:            ; xref 020C:07D3
020C:07D8  C3                retn
                                sub_1      endp
```

Схема работы обработчика прерывания INT 08H



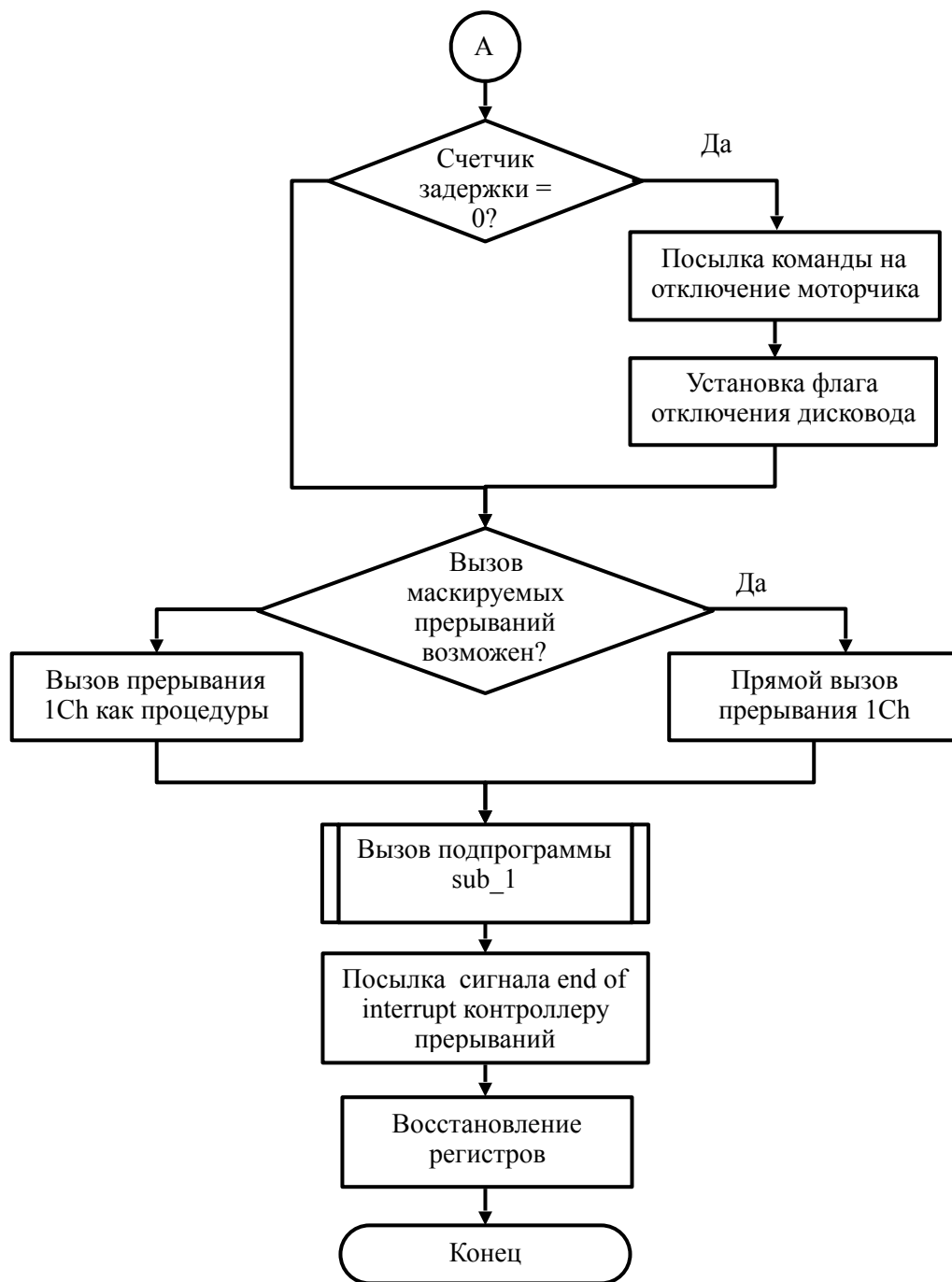
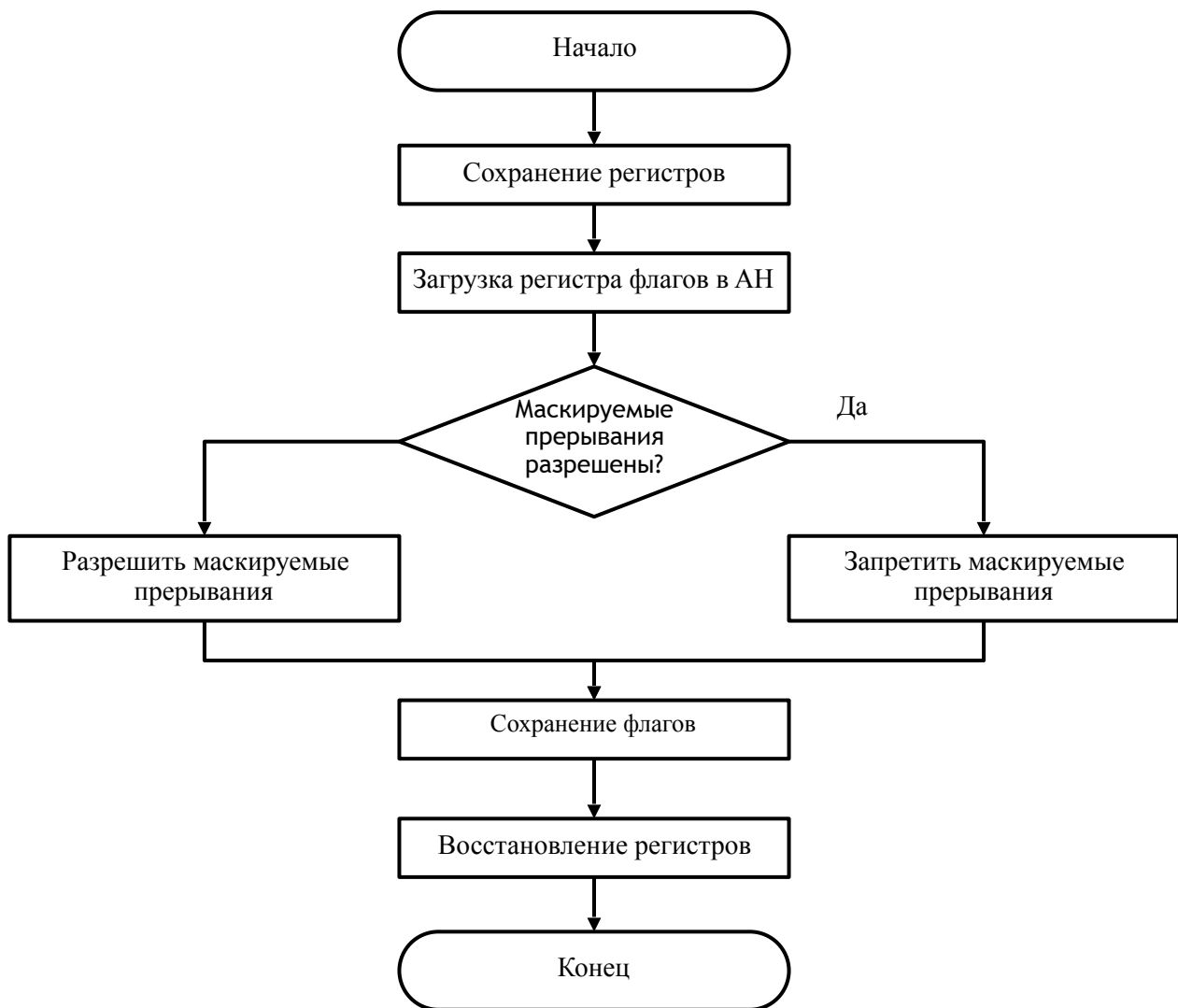


Схема работы подпрограммы sub_1



Функции обработчика прерывания 08h в DOS (реальный режим)

- Увеличивает текущее значение четырехбайтовой переменной, располагающейся в области данных BIOS по адресу 0000:046Ch (счетчик тиков таймера). Если этот счетчик переполняется (прошло более 24 часов с момента запуска таймера), в ячейку 0000:0470h заносится 1.
- Контроль над работой моторчика дисководов. Если после последнего обращения к дисководу прошло более 2 секунд, обработчик прерывания выключает моторчик. Ячейка с адресом 0000:0440h содержит время, оставшееся до выключения моторчика, которое уменьшается обработчиком прерывания. Когда оно становится равно 0, обработчик выключает двигатель дисководов.

- Вызов пользовательского прерывания 1Ch. Его стандартный обработчик состоит из одной команды IRET. Во время выполнения прерывания INT 1CH все аппаратные прерывания запрещены.

Вывод

В реальном режиме прерывание 08h отвечает за изменение счётчика системного времени и управление контроллером дисководов с целью минимизировать время работы моторчика дисководов, а также является способом периодического вызова пользовательского прерывания.