Московский государственный
технический университет им Н.Э.
Баумана

Операционные системы

Лабораторная работа №1

Студент Юмаев А.Р.

Группа ИУ7-55

Преподаватель Рязанова Н.Ю.

Москва, 2019

Оглавление

1. Листинг дизассемблированного обработчика прерывания int 08h

```
; export.lst                          Sourcer Listing v3.07    16-Sep-19   2:27 pm   Page 1

020A:0745  FF                               db    0FFh

; Вызов подпрограммы прерывания sub_6 по адресу 020A:07B9
020A:0746  E8 0070                  call  sub_6             ; (07B9)

; Сохранение регистров в стек перед использованием
020A:0749  06                               push  es
020A:074A  1E                               push  ds
020A:074B  50                               push  ax
020A:074C  52                               push  dx
020A:074D  B8 0040                  mov   ax,40h
020A:0750  8E D8                    mov   ds,ax
020A:0752  33 C0                    xor   ax,ax             ; Zero register
020A:0754  8E C0                    mov   es,ax

; Инкремент счетчика суточного времени
; По адресу 0000:046Ch располагается значение счетчика таймера
020A:0756  FF 06 006C                       inc   word ptr ds:[6Ch]  ; (0040:006C=751Ch)

; Проверка ds:[6Ch] на начало новых суток, переход, если ZF=0 (наступили новые сутки)
020A:075A  75 04                    jnz   loc_3             ; Jump if not zero
020A:075C  FF 06 006E                       inc   word ptr ds:[6Eh]  ; (0040:006E=0Eh)

; * LOC_3 Сброс счетчика суток, если наступили новые сутки
020A:0760                   loc_3:
;                                           Сравнение на 24 часа
020A:0760  83 3E 006E 18            cmp   word ptr ds:[6Eh],18h    ; (0040:006E=0Eh)
020A:0765  75 15                            jne   loc_4             ; Jump if not equal
020A:0767  81 3E 006C 00B0                  cmp   word ptr ds:[6Ch],0B0h   ; (0040:006C=751Ch)
020A:076D  75 0D                            jne   loc_4             ; Jump if not equal
;                                           Обнуление счетчика, ax был равен нулю
020A:076F  A3 006E                  mov   word ptr ds:[6Eh],ax     ; (0040:006E=0Eh)
020A:0772  A3 006C                  mov   word ptr ds:[6Ch],ax     ; (0040:006C=751Ch)
; ~~~~~~~~~~~~~~~~~~~ Заносим 1 потому что счетчик переполнился ~~~~~~~~~~~~~~~~~~~~~~
;                                           Установка флага события
020A:0775  C6 06 0070 01            mov   byte ptr ds:[70h],1      ; (0040:0070=0)
; ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
020A:077A  0C 08                            or    al,8

; * LOC_4 Работа с двигателем НГМД
020A:077C                   loc_4:
020A:077C  50                               push  ax
; Декремент времени до отключения двигателя НГМД
020A:077D  FE 0E 0040                       dec   byte ptr ds:[40h]  ; (0040:0040=6Fh)
; Отключаем двигатель НГМД если время до отключения = 0
020A:0781  75 0B                    jnz   loc_5             ; Jump if not zero
; Установка флага отключения моторчика
020A:0783  80 26 003F F0            and   byte ptr ds:[3Fh],0F0h   ; (0040:003F=0)
020A:0788  B0 0C                    mov   al,0Ch
```

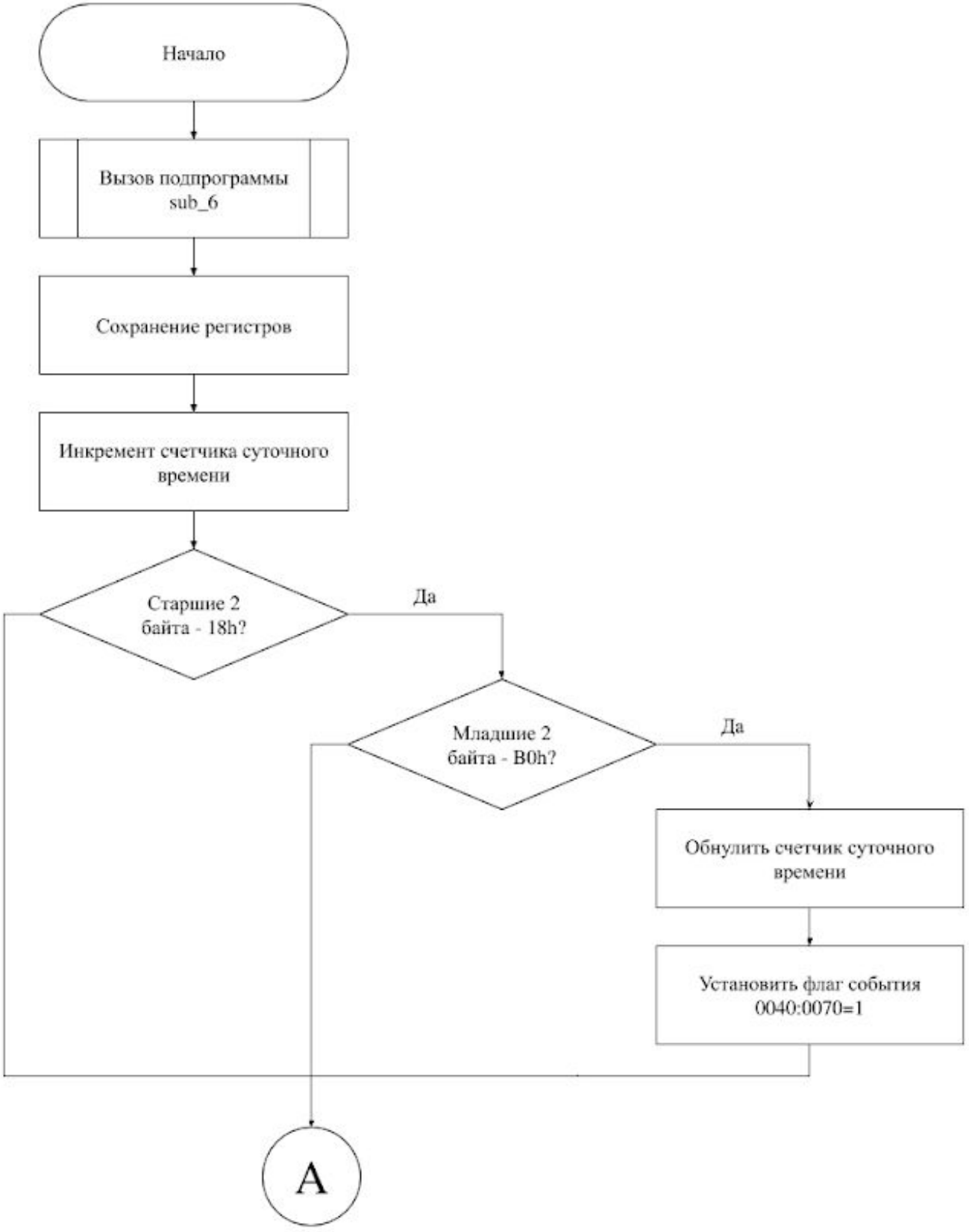
```
020A:078A  BA 03F2                  mov   dx,3F2h ; 1010d

; port 3F2h, dsk0 contrl output
020A:078D  EE                               out   dx,al

; * LOC_5 Отключение двигателя НГМД
020A:078E                loc_5:
020A:078E  58                               pop   ax
; Проверка флага IOPL 4d = 0000000000000100b
; Ур-нь привилегий ввода вывода, см. разреш. на управ. портами
020A:078F  F7 06 0314 0004                  test  word ptr ds:[314h],4     ; (0040:0314=3200h)
020A:0795  75 0C                    jnz   loc_6             ; Jump if not zero
020A:0797  9F                               lahf                    ; Load ah from flags
020A:0798  86 E0                    xchg  ah,al
020A:079A  50                               push  ax
020A:079B  26: FF 1E 0070                   call  dword ptr es:[70h] ; (0000:0070=6ADh)

020A:07A0  EB 03                    jmp   short loc_7       ; (07A5)
020A:07A2  90                               nop

; ~~~~~~~~~ * LOC_6 Вызов прерывания int 1Ch до сброса контроллера прерывания ~~~~~~~~~~~
020A:07A3                loc_6:
020A:07A3  CD 1C                    int   1Ch               ; Timer break (call each 18.2ms)
; ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

; * LOC_7 Вызов subroutine
020A:07A5                loc_7:
020A:07A5  E8 0011                  call  sub_6             ; (07B9)

; ~~~~~~~~~~~~~~~~~~~~~~~~~~ Сбросить контроллер прерываний ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
020A:07A8  B0 20                    mov   al,20h; ' '
020A:07AA  E6 20                    out   20h,al; port 20h, 8259-1 int command
                                                ;  al = 20h, end of interrupt
; ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

; Восстановление регистров из стека
020A:07AC  5A                               pop   dx
020A:07AD  58                               pop   ax
020A:07AE  1F                               pop   ds
020A:07AF  07                               pop   es

; Переход на iret по адресу 020A:064C = 020A:(07B0 - 164h)
020A:07B0  E9 FE99                  jmp   $-164h
```

2. Листинг subroutine - подпроцедуры, вызываемой при обработке прерывания от системного таймера

```
                         sub_6       proc  near
020A:07B9  1E                               push  ds
020A:07BA  50                               push  ax
020A:07BB  B8 0040                  mov   ax,40h
```

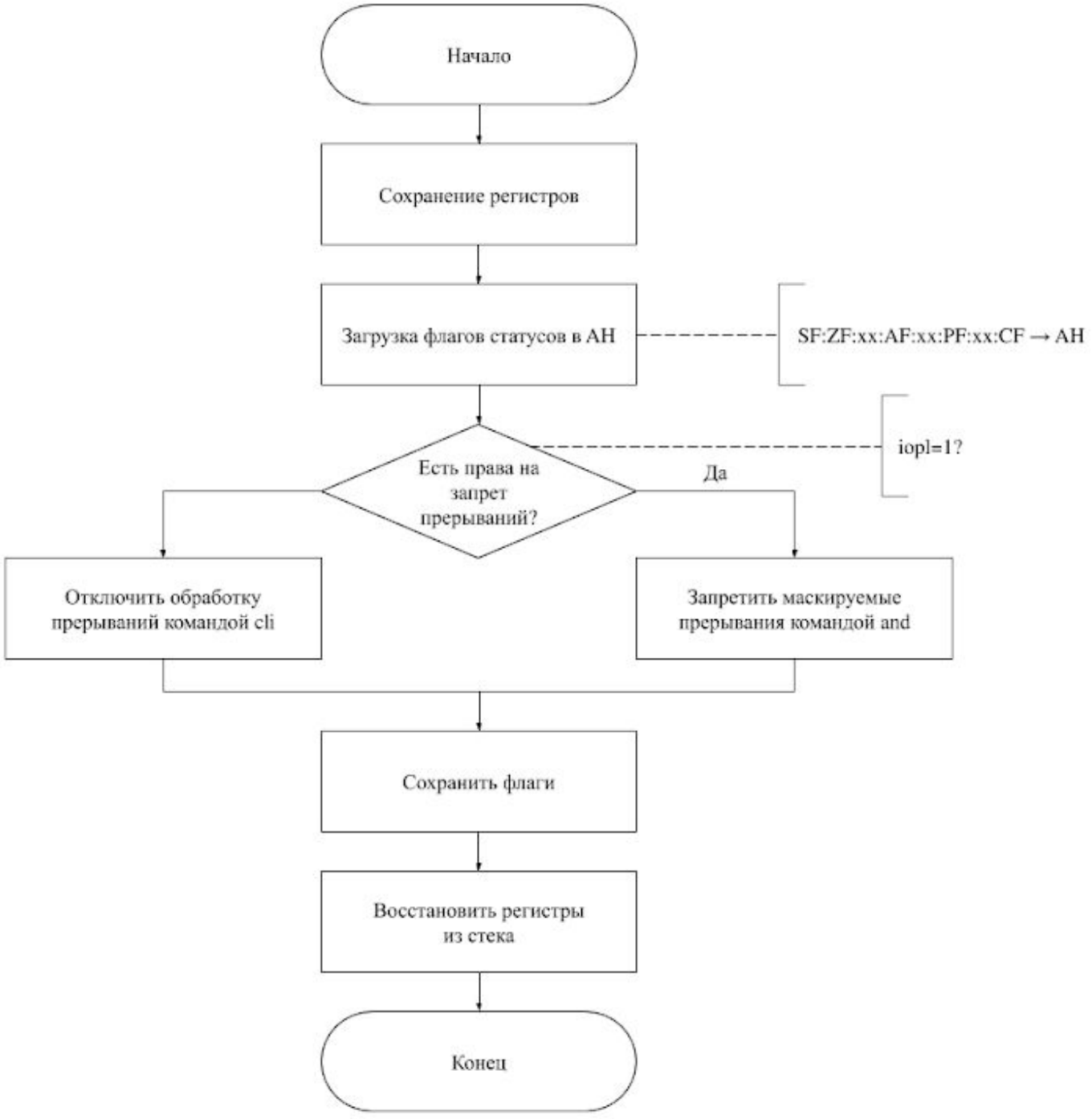
```
020A:07BE  8E D8                    mov   ds,ax
020A:07C0  9F                             lahf                    ; Load ah from flags

; (0040:0314=3200h)
; 2400h = 0010010000000000
; 3200h = 001100 → 1 ← 000000000
; test - побитовое AND, мы смотрим: разрешены ли прерывания
; Если флаг if=1 то маскируемые прерывания будут обрабатываться
020A:07C1  F7 06 0314 2400                test  word ptr ds:[314h],2400h
020A:07C7  75 0C                    jnz   loc_9             ; Jump if not zero

; (0040:0314=3200h)
; 0FDFFh = 111111 → 0 ← 111111111
; Зануляем единичку
020A:07C9  F0> 81 26 0314 FDFF lock  and   word ptr ds:[314h],0FDFFh

; * LOC_8
020A:07D0               loc_8:
020A:07D0  9E                             sahf                    ; Store ah into flags
020A:07D1  58                             pop   ax
020A:07D2  1F                             pop   ds
020A:07D3  EB 03                    jmp   short loc_10      ; (07D8)

; * LOC_9
; Interrupts will be ignored
020A:07D5               loc_9:
020A:07D5  FA                             cli                     ; Disable interrupts
020A:07D6  EB F8                    jmp   short loc_8       ; (07D0)

; * LOC_10
020A:07D8               loc_10:
020A:07D8  C3                             retn
                        sub_6       endp
```

3. Схема алгоритма обработчика прерывания int 08h




A
Декремент счетчика времени до остановки двигателя НГМД
Счетчик задержки НГМД=0?
Да
Отправить команду на отключение двигателя НГМД
out dx,al
Установка флага отключения моторчика
and byte ptr ds:[3Fh],0F0h
Вызов маск. прерываний возможен?
Да
Вызов прерывания по адресу es:[70h]
Вызов прерывания int 1Ch
Вызов подпрограммы sub_6
Сброс контроллера прерываний
Восстановление регистров из стека
Конец

4. Схема алгоритма подпроцедуры sub_6



5. Функции обработчика прерывания int 08h

1. Увеличение на единицу текущего значения 4-байтовой переменной, располагающейся в области данных BIOS по адресу 0000:046Ch - счетчик таймера. Если этот счетчик переполнится из-за того что прошло более 24 часов с момента запуска таймера, в ячейку 0000:0470h заносится значение 1.
2. Контроль за работой двигателей НГМД. Если после последнего обращения к НГМД прошло более 2 секунд, обработчик прерывания выключает двигатель. Ячейка с адресом 0000:0440h содержит время, оставшееся до выключения двигателя. Это время постоянно уменьшается обработчиком прерывания таймера. Когда оно становится равно 0, двигатель НГМД отключается.
3. Вызов прерывания INT 1Ch. После инициализации системы вектор INT 1Ch указывает на команду IRET, то есть обработчик прерывания INT 1Ch ничего не делает. Программа может установить собственный обработчик этого прерывания для того чтобы выполнять какие-либо периодические действия.