

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Операционные системы»
Тема: Исследование организации управления основной памяти

Студент гр. 9381

Гурин С.Н.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Для исследования организации управления памятью необходимо ориентироваться на тип основной памяти, реализованный в компьютере и способ организации, принятый в ОС. В лабораторной работе рассматривается нестраничная память и способ управления динамическими разделами. Для реализации управления памятью в этом случае строится список занятых и свободных участков памяти. Функции ядра, обеспечивающие управление основной памятью, просматривают и преобразуют этот список.

Порядок выполнения работы.

Шаг 1. Для выполнения лабораторной работы необходимо написать и отладить программный модуль типа .COM, который выбирает и распечатывает следующую информацию:

- 1) Количество доступной памяти.
- 2) Размер расширенной памяти.
- 3) Выводит цепочку блоков управления памятью.

Адреса при выводе представляются шестнадцатеричными числами. Объем памяти функциями управления памятью выводится в параграфах. Необходимо преобразовать его в байты и выводить в виде десятичных чисел. Последние восемь байт MSB выводятся как символы, не следует преобразовывать их в шестнадцатеричные числа.

Запустите программу и внимательно оцените результаты. Сохраните результаты, полученные программой, и включите их в отчет в виде скриншота.

Шаг 2. Измените программу таким образом, чтобы она освобождала память, которую она не занимает. Для этого используйте функцию 4Ah прерывания 21h (пример в разделе «Использование функции 4AH»). Повторите эксперимент, запустив модифицированную программу. Сравните выходные данные с результатами, полученными на предыдущем шаге. Сохраните результаты, полученные программой, и включите их в отчет в виде скриншота.

Шаг 3. Измените программу еще раз таким образом, чтобы после освобождения памяти, программа запрашивала 64Кб памяти функцией 48H

прерывания 21H. Повторите эксперимент, запустив модифицированную программу. Сравните выходные данные с результатами, полученными на предыдущих шагах. Сохраните результаты, полученные программой, и включите их в отчет в виде скриншота.

Шаг 4. Измените первоначальный вариант программы, запросив 64Кб памяти функцией 48H прерывания 21H до освобождения памяти. Обязательно обрабатывайте завершение функций ядра, проверяя флаг CF. Сохраните результаты, полученные программой, и включите их в отчет в виде скриншота.

Шаг 5. Оцените результаты, полученные на предыдущих шагах. Ответьте на контрольные вопросы и оформите отчет.

Выполнение работы.

Была реализована программа, которая выводит требуемую информацию с помощью процедур `print_available_mem_size`, `print_extended_mem_size`, `print_meb_list`. Так же были реализованы варианты программы, которые:

- 1) Освобождают неиспользуемую память
- 2) После освобождения неиспользуемой памяти запрашивает 64 КБ
- 3) До освобождения неиспользуемой памяти запрашивает 64 КБ

Результаты выполнения программы были продемонстрированы далее.

```
C:\USERS\SIMON\DESKTOP\AEE\LAB_1\SRC>LAB3_1.COM
available memory size: 648912 bytes
extended memory size: 246720 bytes
MCB #1: address: 016F PSP address: 0008 size: 16 SC/SD:
MCB #2: address: 0171 PSP address: 0000 size: 64 SC/SD:
MCB #3: address: 0176 PSP address: 0040 size: 256 SC/SD:
MCB #4: address: 0187 PSP address: 0192 size: 144 SC/SD:
MCB #5: address: 0191 PSP address: 0192 size: 648912 SC/SD: LAB3_1
```

Рис. 1

```
C:\USERS\SIMON\DESKTOP\AEE\LAB_1\SRC>LAB3_2.COM
available memory size: 648912 bytes
extended memory size: 246720 bytes
MCB #1: address: 016F PSP address: 0008 size: 16 SC/SD:
MCB #2: address: 0171 PSP address: 0000 size: 64 SC/SD:
MCB #3: address: 0176 PSP address: 0040 size: 256 SC/SD:
MCB #4: address: 0187 PSP address: 0192 size: 144 SC/SD:
MCB #5: address: 0191 PSP address: 0192 size: 848 SC/SD: LAB3_2
MCB #6: address: 01C7 PSP address: 0000 size: 648048 SC/SD:
```

Рис. 2

```

C:\USERS\SIMON\DESKTOP\AEF\LAB_1\SRC>LAB3_3.COM
available memory size: 648912 bytes
extended memory size: 246720 bytes
New memory has been added!
MCB #1: address: 016F PSP address: 0008 size: 16 SC/SD:
MCB #2: address: 0171 PSP address: 0000 size: 64 SC/SD:
MCB #3: address: 0176 PSP address: 0040 size: 256 SC/SD:
MCB #4: address: 0187 PSP address: 0192 size: 144 SC/SD:
MCB #5: address: 0191 PSP address: 0192 size: 944 SC/SD: LAB3_3
MCB #6: address: 01CD PSP address: 0192 size: 65536 SC/SD: LAB3_3
MCB #7: address: 11CE PSP address: 0000 size: 582400 SC/SD:

```

Рис. 3

```

C:\USERS\SIMON\DESKTOP\AEF\LAB_1\SRC>LAB3_4.COM
available memory size: 648912 bytes
extended memory size: 246720 bytes
New memory hasn't been added!
MCB #1: address: 016F PSP address: 0008 size: 16 SC/SD:
MCB #2: address: 0171 PSP address: 0000 size: 64 SC/SD:
MCB #3: address: 0176 PSP address: 0040 size: 256 SC/SD:
MCB #4: address: 0187 PSP address: 0192 size: 144 SC/SD:
MCB #5: address: 0191 PSP address: 0192 size: 944 SC/SD: LAB3_4
MCB #6: address: 01CD PSP address: 0000 size: 647952 SC/SD:

```

Рис. 4

Ответы на контрольные вопросы

1) Что означает “доступный объем памяти”?

Доступный объем памяти – это область памяти, которая не занята процессами системы и может выделиться для использования.

2) Где МСВ блок Вашей программы в списке?

МСВ блок программы находится в конце списка МСВ блоков, где перед ним будет находится среда, принадлежащая программе, блок можно отследить по одинаковому сегменту адреса PSP.

3) Какой размер памяти занимает программа в каждом случае?

1. Весь объем памяти
2. 848 байт
3. 65536 байт + 944 байт = 66480 байт
4. 944 байт

Вывод

В результате выполнения данной работы были исследованы структуры данных и работа функций управления памятью ядра операционной системы.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab3_1.asm

```
TESTPC                                SEGMENT
                                     ASSUME     CS:TESTPC, DS:TESTPC, ES:NOTHING,
                                     SS:NOTHING

                                     ORG     100H

START:                                JMP     BEGIN

; ДАННЫЕ

N                                     DB         0DH, 0AH, '$'
AREA_SIZE                            DB         "SIZE:          ", '$'
SC_SD                                DB         "SC/SD: ", '$'
ADDRESS                              DB         "ADDRESS:          ", '$'
PSP_ADDRESS                          DB         "PSP ADDRESS:          ", '$'
EXTENDED_MEM_SIZE                    DB         "EXTENDED MEMORY SIZE:          BYTES",
0DH, 0AH, '$'
AVAILABLE_MEM_SIZE                   DB         "AVAILABLE MEMORY SIZE:          BYTES",
0DH, 0AH, '$'
MCB_NUMBER                           DB         "MCB #", '$'
DEC_NUMBER                           DB         "    ", '$'

; ПРОЦЕДУРЫ

; -----

TETR_TO_HEX                          PROC NEAR

                                     AND     AL, 0FH
                                     CMP     AL, 09
                                     JBE     NEXT
                                     ADD     AL, 07
                                     NEXT:    ADD     AL, 30H
                                     RET

TETR_TO_HEX                          ENDP

; -----

BYTE_TO_HEX                          PROC NEAR
; БАЙТ В AL ПЕРЕВОДИТСЯ В ДВА СИМВОЛА ШЕСТН. ЧИСЛА В AX
                                     PUSH    CX
                                     MOV     AH, AL
                                     CALL    TETR_TO_HEX
                                     XCHG    AL, AH
```

```

MOV CL,4
SHR AL,CL
CALL TETR_TO_HEX ;В AL СТАРШАЯ ЦИФРА
POP CX ;В АН МЛАДШАЯ
RET

BYTE_TO_HEX ENDP
;-----
WRD_TO_HEX PROC NEAR
;ПЕРЕВОД В 16 С/С 16-ТИ РАЗРЯДНОГО ЧИСЛА
; В АХ - ЧИСЛО, DI - АДРЕС ПОСЛЕДНЕГО СИМВОЛА
PUSH BX
MOV BH,АH
CALL BYTE_TO_HEX
MOV [DI],АH
DEC DI
MOV [DI],AL
DEC DI
MOV AL,BH
CALL BYTE_TO_HEX
MOV [DI],АH
DEC DI
MOV [DI],AL
POP BX
RET
WRD_TO_HEX ENDP
;-----
BYTE_TO_DEC PROC NEAR
; ПЕРЕВОД В 10С/С, SI - АДРЕС ПОЛЯ МЛАДШЕЙ ЦИФРЫ
PUSH CX
PUSH DX
XOR AH,АH
XOR DX,DX
MOV CX,10
LOOP_BD:
DIV CX
OR DL,30H
MOV [SI],DL
DEC SI

```

```

                                XOR    DX,DX
                                CMP     AX,10
                                JAE     LOOP_BD
                                CMP     AL,00H
                                JE       END_L
                                OR       AL,30H
                                MOV     [SI],AL
END_L:
                                POP      DX
                                POP      CX
                                RET
BYTE_TO_DEC                     ENDP
;-----
; КОД
PARAM_TO_BYTE                   PROC
                                PUSH     AX
                                PUSH     BX
                                PUSH     CX
                                PUSH     DX
                                PUSH     SI

                                MOV      BX, 10H
                                MUL      BX
                                MOV      BX, 0AH
                                XOR      CX, CX

DIV_LOOP:
                                DIV      BX
                                PUSH     DX
                                INC      CX
                                SUB      DX, DX
                                CMP      AX, 0H
                                JNZ      DIV_LOOP

PRINT_SYM:
                                POP      DX
                                ADD      DL, 30H
                                MOV      [SI], DL

```

```

                                INC    SI
                                LOOP  PRINT_SYM

                                POP    SI
                                POP    DX
                                POP    CX
                                POP    BX
                                POP    AX
                                RET

PARAM_TO_BYTE      ENDP

PRINT_N            PROC NEAR
                                PUSH  AX
                                PUSH  DX

                                MOV    DX, OFFSET N
                                MOV    AH, 9H
                                INT     21H

                                POP    DX
                                POP    AX
                                RET

PRINT_N            ENDP

PRINT_AV_MEM_SIZE  PROC NEAR
                                MOV     AH, 4AH
                                MOV     BX, 0FFFFH
                                INT      21H
                                MOV     AX, BX
                                MOV     SI, OFFSET AVAILABLE_MEM_SIZE
                                ADD     SI, 23
                                CALL    PARAM_TO_BYTE
                                MOV     DX, OFFSET AVAILABLE_MEM_SIZE

                                MOV     AH, 09H
                                INT      21H

                                RET

```


PRINT_AV_MEM_SIZE ENDP

```
PRINT_EXT_MEM_SIZE      PROC NEAR
    MOV     AL,30H ; ЗАПИСЬ АДРЕСА ЯЧЕЙКИ CMOS
    OUT     70H,AL
    IN      AL,71H ; ЧТЕНИЕ МЛАДШЕГО БАЙТА
    MOV     BL,AL ; РАЗМЕРА РАСШИРЕННОЙ ПАМЯТИ
    MOV     AL,31H ; ЗАПИСЬ АДРЕСА ЯЧЕЙКИ CMOS
    OUT     70H,AL
    IN      AL,71H ; ЧТЕНИЕ СТАРШЕГО БАЙТА
    ; РАЗМЕРА РАСШИРЕННОЙ ПАМЯТИ
    MOV     AH, AL
    MOV     SI, OFFSET EXTENDED_MEM_SIZE
    ADD     SI, 22
    CALL    PARA_TO_BYTE
    MOV     DX, OFFSET EXTENDED_MEM_SIZE

    MOV     AH, 09H
    INT     21H

    RET
PRINT_EXT_MEM_SIZE      ENDP
```

```
PRINT_MCB              PROC NEAR
    PUSH    AX
    PUSH    DX
    PUSH    SI
    PUSH    DI
    PUSH    CX

    MOV     AX, ES;MCB
    MOV     DI, OFFSET ADDRESS
    ADD     DI, 12
    CALL    WRD_TO_HEX
    MOV     DX, OFFSET ADDRESS

    MOV     AH, 09H
    INT     21H
```

```

MOV  AX, ES:[1] ;PSP
MOV  DI, OFFSET PSP_ADDRESS
ADD  DI, 16
CALL WRD_TO_HEX
MOV  DX, OFFSET PSP_ADDRESS

MOV  AH, 09H
INT  21H

MOV  AX, ES:[3] ;SIZE OF PARA
MOV  SI, OFFSET AREA_SIZE
ADD  SI, 6
CALL PARA_TO_BYTE
MOV  DX, OFFSET AREA_SIZE

MOV  AH, 09H
INT  21H

MOV  BX, 8 ;SC SD
MOV  DX, OFFSET SC_SD

MOV  AH, 09H
INT  21H

MOV  CX, 7
PRINT_SC_SD_LOOP:
MOV  DL, ES:[BX]
MOV  AH, 02H
INT  21H
INC  BX
LOOP PRINT_SC_SD_LOOP

POP  CX
POP  DI
POP  SI
POP  DX
POP  AX

```

```

                                RET
PRINT_MCB                      ENDP

OFFSET_DEC                     PROC NEAR
OFFSET_DEC_LOOP:
                                CMP     BYTE PTR [SI], ' '
                                JNE      END_OFFSET_DEC
                                INC      SI
                                JMP      OFFSET_DEC_LOOP
END_OFFSET_DEC:
                                RET
OFFSET_DEC                     ENDP

PRINT_MCB_LIST                 PROC NEAR
                                PUSH     AX
                                PUSH     BX
                                PUSH     ES
                                PUSH     DX

                                MOV      AH, 52H
                                INT      21H
                                MOV      AX, ES:[BX-2]
                                MOV      ES, AX
                                MOV      CL, 1

PRINT_LIST:
                                MOV      DX, OFFSET MCB_NUMBER

                                MOV      AH, 09H
                                INT      21H

                                MOV      AL, CL
                                MOV      SI, OFFSET DEC_NUMBER
                                ADD      SI, 2
                                CALL     BYTE_TO_DEC
                                CALL     OFFSET_DEC
                                MOV      DX, SI

                                MOV      AH, 09H

```

```

                                INT    21H

                                MOV     DL, ':'
                                MOV     AH, 02H
                                INT     21H
                                MOV     DL, ' '
                                MOV     AH, 02H
                                INT     21H
                                CALL    PRINT_MCB
                                CALL    PRINT_N
                                MOV     AL, ES:[0]
                                CMP     AL, 5AH
                                JE       END_MCB_LIST

                                MOV     BX, ES:[3]
                                MOV     AX, ES
                                ADD     AX, BX
                                INC     AX
                                MOV     ES, AX
                                INC     CL
                                JMP     PRINT_LIST

                                END_MCB_LIST:
                                    POP     DX
                                    POP     ES
                                    POP     BX
                                    POP     AX
                                    RET

PRINT_MCB_LIST                 ENDP

BEGIN:
; . . . . .

                                CALL    PRINT_AV_MEM_SIZE
                                CALL    PRINT_EXT_MEM_SIZE
                                CALL    PRINT_MCB_LIST
; . . . . .

```

```

; ВЫХОД В DOS

                                XOR    AL,AL
                                MOV     AH,4CH
                                INT     21H
TESTPC                          ENDS
                                END     START

```

Название файла: lab3_2.asm

```

TESTPC                          SEGMENT
                                ASSUME   CS:TESTPC, DS:TESTPC, ES:NOTHING,
                                SS:NOTHING

                                ORG      100H

START:                          JMP      BEGIN

; ДАННЫЕ

N                                DB       0DH, 0AH, '$'
AREA_SIZE                      DB       "SIZE:      ", '$'
SC_SD                          DB       "SC/SD: ", '$'
ADDRESS                        DB       "ADDRESS:    ", '$'
PSP_ADDRESS                    DB       "PSP ADDRESS:      ", '$'
EXTENDED_MEM_SIZE              DB       "EXTENDED MEMORY SIZE:      BYTES",
0DH, 0AH, '$'
AVAILABLE_MEM_SIZE              DB       "AVAILABLE MEMORY SIZE:      BYTES",
0DH, 0AH, '$'
MCB_NUMBER                     DB       "MCB #", '$'
DEC_NUMBER                     DB       "      ", '$'

; ПРОЦЕДУРЫ

; -----
TETR_TO_HEX                    PROC NEAR

                                AND      AL,0FH
                                CMP      AL,09
                                JBE      NEXT
                                ADD      AL,07
NEXT:                          ADD      AL,30H
                                RET

TETR_TO_HEX                    ENDP

; -----
BYTE_TO_HEX                    PROC NEAR
; БАЙТ В AL ПЕРЕВОДИТСЯ В ДВА СИМВОЛА ШЕСТН. ЧИСЛА В AX

```

```

                                PUSH CX
                                MOV  AH,AL
                                CALL TETR_TO_HEX
                                XCHG AL,AH
                                MOV  CL,4
                                SHR  AL,CL
                                CALL TETR_TO_HEX ;В AL СТАРШАЯ ЦИФРА
                                POP  CX ;В AH МЛАДШАЯ
                                RET

```

```

BYTE_TO_HEX                    ENDP

```

```

;-----

```

```

WRD_TO_HEX                    PROC NEAR
;ПЕРЕВОД В 16 С/С 16-ТИ РАЗРЯДНОГО ЧИСЛА
; В AX - ЧИСЛО, DI - АДРЕС ПОСЛЕДНЕГО СИМВОЛА

```

```

                                PUSH BX
                                MOV  BH,AH
                                CALL BYTE_TO_HEX
                                MOV  [DI],AH
                                DEC  DI
                                MOV  [DI],AL
                                DEC  DI
                                MOV  AL,BH
                                CALL BYTE_TO_HEX
                                MOV  [DI],AH
                                DEC  DI
                                MOV  [DI],AL
                                POP  BX
                                RET

```

```

WRD_TO_HEX                    ENDP

```

```

;-----

```

```

BYTE_TO_DEC                    PROC NEAR
; ПЕРЕВОД В 10С/С, SI - АДРЕС ПОЛЯ МЛАДШЕЙ ЦИФРЫ

```

```

                                PUSH CX
                                PUSH DX
                                XOR  AH,AH
                                XOR  DX,DX
                                MOV  CX,10

```

```

                                LOOP_BD:

```

```

                                DIV    CX
                                OR      DL, 30H
                                MOV     [SI], DL
                                DEC     SI
                                XOR     DX, DX
                                CMP     AX, 10
                                JAE     LOOP_BD
                                CMP     AL, 00H
                                JE       END_L
                                OR      AL, 30H
                                MOV     [SI], AL
END_L:
                                POP     DX
                                POP     CX
                                RET
BYTE_TO_DEC                     ENDP
; -----
; КОД
PARAM_TO_BYTE                   PROC
                                PUSH    AX
                                PUSH    BX
                                PUSH    CX
                                PUSH    DX
                                PUSH    SI

                                MOV     BX, 10H
                                MUL     BX
                                MOV     BX, 0AH
                                XOR     CX, CX

DIV_LOOP:
                                DIV     BX
                                PUSH    DX
                                INC     CX
                                SUB     DX, DX
                                CMP     AX, 0H
                                JNZ     DIV_LOOP

```

```

PRINT_SYM:
    POP    DX
    ADD    DL, 30H
    MOV    [SI], DL
    INC    SI
    LOOP   PRINT_SYM

    POP    SI
    POP    DX
    POP    CX
    POP    BX
    POP    AX
    RET

PARAM_TO_BYTE    ENDP

PRINT_N          PROC NEAR
    PUSH    AX
    PUSH    DX

    MOV     DX, OFFSET N
    MOV     AH, 9H
    INT     21H

    POP     DX
    POP     AX
    RET

PRINT_N          ENDP

PRINT_AV_MEM_SIZE  PROC NEAR
    MOV     AH, 4AH
    MOV     BX, 0FFFFH
    INT     21H
    MOV     AX, BX
    MOV     SI, OFFSET AVAILABLE_MEM_SIZE
    ADD     SI, 23
    CALL    PARAM_TO_BYTE
    MOV     DX, OFFSET AVAILABLE_MEM_SIZE

```



```

MOV AH, 09H
INT 21H

RET

PRINT_AV_MEM_SIZE ENDP

PRINT_EXT_MEM_SIZE PROC NEAR
MOV AL, 30H ; ЗАПИСЬ АДРЕСА ЯЧЕЙКИ CMOS
OUT 70H, AL
IN AL, 71H ; ЧТЕНИЕ МЛАДШЕГО БАЙТА
MOV BL, AL ; РАЗМЕРА РАСШИРЕННОЙ ПАМЯТИ
MOV AL, 31H ; ЗАПИСЬ АДРЕСА ЯЧЕЙКИ CMOS
OUT 70H, AL
IN AL, 71H ; ЧТЕНИЕ СТАРШЕГО БАЙТА
; РАЗМЕРА РАСШИРЕННОЙ ПАМЯТИ
MOV AH, AL
MOV SI, OFFSET EXTENDED_MEM_SIZE
ADD SI, 22
CALL PARA_TO_BYTE
MOV DX, OFFSET EXTENDED_MEM_SIZE

MOV AH, 09H
INT 21H

RET

PRINT_EXT_MEM_SIZE ENDP

PRINT_MCB PROC NEAR
PUSH AX
PUSH DX
PUSH SI
PUSH DI
PUSH CX

MOV AX, ES;MCB
MOV DI, OFFSET ADDRESS
ADD DI, 12
CALL WRD_TO_HEX

```

```

MOV    DX, OFFSET ADDRESS

MOV    AH, 09H
INT    21H

MOV    AX, ES:[1] ;PSP
MOV    DI, OFFSET PSP_ADDRESS
ADD    DI, 16
CALL   WRD_TO_HEX
MOV    DX, OFFSET PSP_ADDRESS

MOV    AH, 09H
INT    21H

MOV    AX, ES:[3] ;SIZE OF PARA
MOV    SI, OFFSET AREA_SIZE
ADD    SI, 6
CALL   PARA_TO_BYTE
MOV    DX, OFFSET AREA_SIZE

MOV    AH, 09H
INT    21H

MOV    BX, 8 ;SC SD
MOV    DX, OFFSET SC_SD

MOV    AH, 09H
INT    21H

MOV    CX, 7
PRINT_SC_SD_LOOP:
MOV    DL, ES:[BX]
MOV    AH, 02H
INT    21H
INC    BX
LOOP   PRINT_SC_SD_LOOP

POP    CX

```

```

                                POP    DI
                                POP    SI
                                POP    DX
                                POP    AX
                                RET
PRINT_MCB                      ENDP

OFFSET_DEC                    PROC NEAR
OFFSET_DEC_LOOP:
                                CMP     BYTE PTR [SI], ' '
                                JNE     END_OFFSET_DEC
                                INC     SI
                                JMP     OFFSET_DEC_LOOP
END_OFFSET_DEC:
                                RET
OFFSET_DEC                    ENDP

PRINT_MCB_LIST                PROC NEAR
                                PUSH    AX
                                PUSH    BX
                                PUSH    ES
                                PUSH    DX

                                MOV     AH, 52H
                                INT     21H
                                MOV     AX, ES:[BX-2]
                                MOV     ES, AX
                                MOV     CL, 1
PRINT_LIST:
                                MOV     DX, OFFSET MCB_NUMBER

                                MOV     AH, 09H
                                INT     21H

                                MOV     AL, CL
                                MOV     SI, OFFSET DEC_NUMBER
                                ADD     SI, 2
                                CALL    BYTE_TO_DEC

```

```

        CALL OFFSET_DEC
        MOV  DX, SI

        MOV  AH, 09H
        INT  21H

        MOV  DL, ':'
        MOV  AH, 02H
        INT  21H
        MOV  DL, ' '
        MOV  AH, 02H
        INT  21H
        CALL PRINT_MCB
        CALL PRINT_N
        MOV  AL, ES:[0]
        CMP  AL, 5AH
        JE   END_MCB_LIST

        MOV  BX, ES:[3]
        MOV  AX, ES
        ADD  AX, BX
        INC  AX
        MOV  ES, AX
        INC  CL
        JMP  PRINT_LIST

END_MCB_LIST:
        POP  DX
        POP  ES
        POP  BX
        POP  AX
        RET

PRINT_MCB_LIST      ENDP

DEL_FREE_MEMORY     PROC NEAR
                    PUSH AX
                    PUSH BX
                    PUSH CX

```

```

                                PUSH  DX

                                LEA   AX, FINAL_END
                                MOV    BX, 10H
                                SUB    DX, DX
                                DIV    BX
                                INC    AX
                                MOV    BX, AX
                                MOV    AL, 0
                                MOV    AH, 4AH
                                INT     21H

                                POP     DX
                                POP     CX
                                POP     BX
                                POP     AX
                                RET

DEL_FREE_MEMORY                ENDP

BEGIN:
; . . . . .

                                CALL PRINT_AV_MEM_SIZE
                                CALL PRINT_EXT_MEM_SIZE
                                CALL DEL_FREE_MEMORY
                                CALL PRINT_MCB_LIST

; . . . . .
; ВЫХОД В DOS

                                XOR     AL, AL
                                MOV     AH, 4CH
                                INT      21H

                                FINAL_END:

TESTPC                          ENDS

                                END     START

```

Название файла: lab3_3.asm

```

TESTPC                          SEGMENT
                                ASSUME  CS:TESTPC, DS:TESTPC, ES:NOTHING,
                                SS:NOTHING

                                ORG     100H

```

```

START:                                JMP  BEGIN
; ДАННЫЕ
N                                     DB      0DH, 0AH, '$'
AREA_SIZE                           DB      "SIZE:      ", '$'
SC_SD                               DB      "SC/SD: ", '$'
ADDRESS                             DB      "ADDRESS:    ", '$'
PSP_ADDRESS                         DB      "PSP ADDRESS:  ", '$'
EXTENDED_MEM_SIZE                   DB      "EXTENDED MEMORY SIZE:      BYTES",
0DH, 0AH, '$'
AVAILABLE_MEM_SIZE                   DB      "AVAILABLE MEMORY SIZE:      BYTES",
0DH, 0AH, '$'
MCB_NUMBER                           DB      "MCB #", '$'
DEC_NUMBER                           DB      "      ", '$'
MEM_ACCEPT                           DB      "NEW MEMORY HAS BEEN ADDED!", 0DH, 0AH,
'$'
MEM_FAIL                             DB      "NEW MEMORY HASN'T BEEN ADDED!", 0DH, 0AH, '$'

; ПРОЦЕДУРЫ
; -----
TETR_TO_HEX                          PROC  NEAR
                                     AND   AL, 0FH
                                     CMP   AL, 09
                                     JBE   NEXT
                                     ADD   AL, 07
NEXT:                                ADD   AL, 30H
                                     RET
TETR_TO_HEX                          ENDP
; -----
BYTE_TO_HEX                          PROC  NEAR
; БАЙТ В AL ПЕРЕВОДИТСЯ В ДВА СИМВОЛА ШЕСТН. ЧИСЛА В АХ
                                     PUSH  CX
                                     MOV   AH, AL
                                     CALL  TETR_TO_HEX
                                     XCHG  AL, AH
                                     MOV   CL, 4
                                     SHR   AL, CL
                                     CALL  TETR_TO_HEX ; В AL СТАРШАЯ ЦИФРА
                                     POP   CX ; В АН МЛАДШАЯ

```

```

                                RET
BYTE_TO_HEX                    ENDP
;-----
WRD_TO_HEX                     PROC NEAR
; ПЕРЕВОД В 16 С/С 16-ТИ РАЗРЯДНОГО ЧИСЛА
; В AX - ЧИСЛО, DI - АДРЕС ПОСЛЕДНЕГО СИМВОЛА
                                PUSH BX
                                MOV  BH, AH
                                CALL BYTE_TO_HEX
                                MOV  [DI], AH
                                DEC  DI
                                MOV  [DI], AL
                                DEC  DI
                                MOV  AL, BH
                                CALL BYTE_TO_HEX
                                MOV  [DI], AH
                                DEC  DI
                                MOV  [DI], AL
                                POP  BX
                                RET
WRD_TO_HEX                     ENDP
;-----
BYTE_TO_DEC                    PROC NEAR
; ПЕРЕВОД В 10С/С, SI - АДРЕС ПОЛЯ МЛАДШЕЙ ЦИФРЫ
                                PUSH CX
                                PUSH DX
                                XOR  AH, AH
                                XOR  DX, DX
                                MOV  CX, 10
                                LOOP_BD:
                                DIV  CX
                                OR    DL, 30H
                                MOV  [SI], DL
                                DEC  SI
                                XOR  DX, DX
                                CMP  AX, 10
                                JAE  LOOP_BD
                                CMP  AL, 00H

```

```

                                JE      END_L
                                OR      AL, 30H
                                MOV     [SI], AL
END_L:
                                POP     DX
                                POP     CX
                                RET
BYTE_TO_DEC      ENDP
; -----
; КОД
PARAM_TO_BYTE    PROC
                                PUSH    AX
                                PUSH    BX
                                PUSH    CX
                                PUSH    DX
                                PUSH    SI

                                MOV     BX, 10H
                                MUL     BX
                                MOV     BX, 0AH
                                XOR     CX, CX

DIV_LOOP:
                                DIV     BX
                                PUSH    DX
                                INC     CX
                                SUB     DX, DX
                                CMP     AX, 0H
                                JNZ     DIV_LOOP

PRINT_SYM:
                                POP     DX
                                ADD     DL, 30H
                                MOV     [SI], DL
                                INC     SI
                                LOOP    PRINT_SYM

                                POP     SI

```



```

                                POP    DX
                                POP    CX
                                POP    BX
                                POP    AX
                                RET
PARAM_TO_BYTE      ENDP

PRINT_N            PROC  NEAR
                                PUSH  AX
                                PUSH  DX

                                MOV    DX, OFFSET N
                                MOV    AH, 9H
                                INT     21H

                                POP    DX
                                POP    AX
                                RET
PRINT_N            ENDP

PRINT_AV_MEM_SIZE  PROC  NEAR
                                MOV     AH, 4AH
                                MOV     BX, 0FFFFH
                                INT      21H
                                MOV     AX, BX
                                MOV     SI, OFFSET AVAILABLE_MEM_SIZE
                                ADD     SI, 23
                                CALL    PARAM_TO_BYTE
                                MOV     DX, OFFSET AVAILABLE_MEM_SIZE

                                MOV     AH, 09H
                                INT      21H

                                RET
PRINT_AV_MEM_SIZE  ENDP

PRINT_EXT_MEM_SIZE PROC  NEAR
                                MOV     AL, 30H ; ЗАПИСЬ АДРЕСА ЯЧЕЙКИ CMOS

```

```

                                OUT    70H,AL
                                IN      AL,71H ; ЧТЕНИЕ МЛАДШЕГО БАЙТА
                                MOV     BL,AL ; РАЗМЕРА РАСШИРЕННОЙ ПАМЯТИ
                                MOV     AL,31H ; ЗАПИСЬ АДРЕСА ЯЧЕЙКИ CMOS
                                OUT     70H,AL
                                IN      AL,71H ; ЧТЕНИЕ СТАРШЕГО БАЙТА
; РАЗМЕРА РАСШИРЕННОЙ ПАМЯТИ
                                MOV     AH, AL
                                MOV     SI, OFFSET EXTENDED_MEM_SIZE
                                ADD     SI, 22
                                CALL    PARA_TO_BYTE
                                MOV     DX, OFFSET EXTENDED_MEM_SIZE

                                MOV     AH, 09H
                                INT     21H

                                RET
PRINT_EXT_MEM_SIZE    ENDP

PRINT_MCB             PROC NEAR
                                PUSH    AX
                                PUSH    DX
                                PUSH    SI
                                PUSH    DI
                                PUSH    CX

                                MOV     AX, ES;MCB
                                MOV     DI, OFFSET ADDRESS
                                ADD     DI, 12
                                CALL    WRD_TO_HEX
                                MOV     DX, OFFSET ADDRESS

                                MOV     AH, 09H
                                INT     21H

                                MOV     AX, ES:[1] ;PSP
                                MOV     DI, OFFSET PSP_ADDRESS
                                ADD     DI, 16

```

```

CALL WRD_TO_HEX
MOV DX, OFFSET PSP_ADDRESS

MOV AH, 09H
INT 21H

MOV AX, ES:[3] ;SIZE OF PARA
MOV SI, OFFSET AREA_SIZE
ADD SI, 6
CALL PARA_TO_BYTE
MOV DX, OFFSET AREA_SIZE

MOV AH, 09H
INT 21H

MOV BX, 8 ;SC SD
MOV DX, OFFSET SC_SD

MOV AH, 09H
INT 21H

MOV CX, 7
PRINT_SC_SD_LOOP:
MOV DL, ES:[BX]
MOV AH, 02H
INT 21H
INC BX
LOOP PRINT_SC_SD_LOOP

POP CX
POP DI
POP SI
POP DX
POP AX
RET

PRINT_MCB      ENDP

OFFSET_DEC     PROC NEAR

```

```

OFFSET_DEC_LOOP:
    CMP  BYTE PTR [SI], ' '
    JNE  END_OFFSET_DEC
    INC  SI
    JMP  OFFSET_DEC_LOOP
END_OFFSET_DEC:
    RET
OFFSET_DEC
    ENDP

PRINT_MCB_LIST    PROC NEAR
    PUSH AX
    PUSH BX
    PUSH ES
    PUSH DX

    MOV  AH, 52H
    INT  21H
    MOV  AX, ES:[BX-2]
    MOV  ES, AX
    MOV  CL, 1

PRINT_LIST:
    MOV  DX, OFFSET MCB_NUMBER

    MOV  AH, 09H
    INT  21H

    MOV  AL, CL
    MOV  SI, OFFSET DEC_NUMBER
    ADD  SI, 2
    CALL BYTE_TO_DEC
    CALL OFFSET_DEC
    MOV  DX, SI

    MOV  AH, 09H
    INT  21H

    MOV  DL, ':'
    MOV  AH, 02H

```

```

        INT     21H
        MOV     DL, ' '
        MOV     AH, 02H
        INT     21H
        CALL    PRINT_MCB
        CALL    PRINT_N
        MOV     AL, ES:[0]
        CMP     AL, 5AH
        JE      END_MCB_LIST

        MOV     BX, ES:[3]
        MOV     AX, ES
        ADD     AX, BX
        INC     AX
        MOV     ES, AX
        INC     CL
        JMP     PRINT_LIST

END_MCB_LIST:
        POP     DX
        POP     ES
        POP     BX
        POP     AX
        RET

PRINT_MCB_LIST      ENDP

DEL_FREE_MEMORY     PROC NEAR
                    PUSH AX
                    PUSH BX
                    PUSH CX
                    PUSH DX

                    LEA  AX, FINAL_END
                    MOV  BX, 10H
                    SUB  DX, DX
                    DIV  BX
                    INC  AX
                    MOV  BX, AX

```

```

                                MOV    AL,0
                                MOV    AH,4AH
                                INT     21H

                                POP     DX
                                POP     CX
                                POP     BX
                                POP     AX
                                RET

DEL_FREE_MEMORY      ENDP

MEMORY_REQUEST      PROC  NEAR
                                PUSH    AX
                                PUSH    BX
                                PUSH    DX

                                MOV     BX, 1000H
                                MOV     AH, 48H
                                INT      21H
                                JC       MEM_FAILED
                                JMP      MEM_ACCEPTED

                                MEM_FAILED:
                                MOV     DX, OFFSET MEM_FAIL
                                MOV     AH, 09H
                                INT      21H
                                JMP     END_MEMORY_REQUEST

                                MEM_ACCEPTED:
                                MOV     DX, OFFSET MEM_ACCEPT
                                MOV     AH, 09H
                                INT      21H

                                END_MEMORY_REQUEST:

                                POP     DX
                                POP     BX
                                POP     AX
                                RET

```

```

MEMORY_REQUEST      ENDP
BEGIN:
; . . . . .

CALL PRINT_AV_MEM_SIZE
CALL PRINT_EXT_MEM_SIZE
CALL DEL_FREE_MEMORY
CALL MEMORY_REQUEST
CALL PRINT_MCB_LIST

; . . . . .
; ВЫХОД В DOS

XOR  AL,AL
MOV  AH,4CH
INT  21H

FINAL_END:
TESTPC      ENDS
END  START

```

Название файла: lab3_4.asm

```

TESTPC      SEGMENT
ASSUME      CS:TESTPC, DS:TESTPC, ES:NOTHING,
SS:NOTHING

ORG  100H

START:      JMP  BEGIN

; ДАННЫЕ
N           DB          0DH, 0AH, '$'
AREA_SIZE   DB          "SIZE:      ", '$'
SC_SD       DB          "SC/SD: ", '$'
ADDRESS     DB          "ADDRESS:   ", '$'
PSP_ADDRESS DB          "PSP ADDRESS:      ", '$'
EXTENDED_MEM_SIZE DB      "EXTENDED MEMORY SIZE:      BYTES",
0DH, 0AH, '$'
AVAILABLE_MEM_SIZE DB      "AVAILABLE MEMORY SIZE:      BYTES",
0DH, 0AH, '$'
MCB_NUMBER  DB          "MCB #", '$'
DEC_NUMBER  DB          "      ", '$'
MEM_ACCEPT  DB  "NEW MEMORY HAS BEEN ADDED!", 0DH, 0AH,
'$'
MEM_FAIL     DB  "NEW MEMORY HASN'T BEEN ADDED!", 0DH, 0AH, '$'

```

; ПРОЦЕДУРЫ

;-----

```
TETR_TO_HEX      PROC NEAR
                   AND    AL,0FH
                   CMP    AL,09
                   JBE    NEXT
                   ADD    AL,07
NEXT:              ADD    AL,30H
                   RET
```

```
TETR_TO_HEX      ENDP
```

;-----

```
BYTE_TO_HEX      PROC NEAR
; БАЙТ В AL ПЕРЕВОДИТСЯ В ДВА СИМВОЛА ШЕСТН. ЧИСЛА В AX
                   PUSH CX
                   MOV    AH,AL
                   CALL   TETR_TO_HEX
                   XCHG   AL,AH
                   MOV    CL,4
                   SHR    AL,CL
                   CALL   TETR_TO_HEX ;В AL СТАРШАЯ ЦИФРА
                   POP    CX ;В AH МЛАДШАЯ
                   RET
```

```
BYTE_TO_HEX      ENDP
```

;-----

```
WRD_TO_HEX      PROC NEAR
; ПЕРЕВОД В 16 С/С 16-ТИ РАЗРЯДНОГО ЧИСЛА
; В AX - ЧИСЛО, DI - АДРЕС ПОСЛЕДНЕГО СИМВОЛА
                   PUSH BX
                   MOV    BH,AH
                   CALL   BYTE_TO_HEX
                   MOV    [DI],AH
                   DEC    DI
                   MOV    [DI],AL
                   DEC    DI
                   MOV    AL,BH
                   CALL   BYTE_TO_HEX
                   MOV    [DI],AH
                   DEC    DI
```



```

MOV     [DI],AL
POP     BX
RET
WRD_TO_HEX      ENDP
;-----
BYTE_TO_DEC     PROC NEAR
; ПЕРЕВОД В 10С/С, SI - АДРЕС ПОЛЯ МЛАДШЕЙ ЦИФРЫ
PUSH    CX
PUSH    DX
XOR     AH,AH
XOR     DX,DX
MOV     CX,10
LOOP_BD:
DIV     CX
OR      DL,30H
MOV     [SI],DL
DEC     SI
XOR     DX,DX
CMP     AX,10
JAE     LOOP_BD
CMP     AL,00H
JE      END_L
OR      AL,30H
MOV     [SI],AL
END_L:
POP     DX
POP     CX
RET
BYTE_TO_DEC     ENDP
;-----
; КОД
PARA_TO_BYTE     PROC
PUSH    AX
PUSH    BX
PUSH    CX
PUSH    DX
PUSH    SI

```

```

        MOV    BX, 10H
        MUL    BX
        MOV    BX, 0AH
        XOR    CX, CX

DIV_LOOP:
        DIV    BX
        PUSH   DX
        INC    CX
        SUB    DX, DX
        CMP    AX, 0H
        JNZ    DIV_LOOP

PRINT_SYM:
        POP    DX
        ADD    DL, 30H
        MOV    [SI], DL
        INC    SI
        LOOP   PRINT_SYM

        POP    SI
        POP    DX
        POP    CX
        POP    BX
        POP    AX
        RET

PARAM_TO_BYTE    ENDP

PRINT_N          PROC NEAR
        PUSH   AX
        PUSH   DX

        MOV    DX, OFFSET N
        MOV    AH, 9H
        INT    21H

        POP    DX
        POP    AX

```

```

                                RET
PRINT_N                        ENDP

PRINT_AV_MEM_SIZE             PROC NEAR
                                MOV  AH, 4AH
                                MOV  BX, 0FFFFH
                                INT  21H
                                MOV  AX, BX
                                MOV  SI, OFFSET AVAILABLE_MEM_SIZE
                                ADD  SI, 23
                                CALL PARA_TO_BYTE
                                MOV  DX, OFFSET AVAILABLE_MEM_SIZE

                                MOV  AH, 09H
                                INT  21H

                                RET
PRINT_AV_MEM_SIZE             ENDP

PRINT_EXT_MEM_SIZE            PROC NEAR
                                MOV  AL,30H ; ЗАПИСЬ АДРЕСА ЯЧЕЙКИ CMOS
                                OUT  70H,AL
                                IN   AL,71H ; ЧТЕНИЕ МЛАДШЕГО БАЙТА
                                MOV  BL,AL ; РАЗМЕРА РАСШИРЕННОЙ ПАМЯТИ
                                MOV  AL,31H ; ЗАПИСЬ АДРЕСА ЯЧЕЙКИ CMOS
                                OUT  70H,AL
                                IN   AL,71H ; ЧТЕНИЕ СТАРШЕГО БАЙТА
                                ; РАЗМЕРА РАСШИРЕННОЙ ПАМЯТИ
                                MOV  AH, AL
                                MOV  SI, OFFSET EXTENDED_MEM_SIZE
                                ADD  SI, 22
                                CALL PARA_TO_BYTE
                                MOV  DX, OFFSET EXTENDED_MEM_SIZE

                                MOV  AH, 09H
                                INT  21H

                                RET

```

```
PRINT_EXT_MEM_SIZE    ENDP
```

```
PRINT_MCB              PROC NEAR
                        PUSH AX
                        PUSH DX
                        PUSH SI
                        PUSH DI
                        PUSH CX

                        MOV  AX, ES;MCB
                        MOV  DI, OFFSET ADDRESS
                        ADD  DI, 12
                        CALL WRD_TO_HEX
                        MOV  DX, OFFSET ADDRESS

                        MOV  AH, 09H
                        INT  21H

                        MOV  AX, ES:[1] ;PSP
                        MOV  DI, OFFSET PSP_ADDRESS
                        ADD  DI, 16
                        CALL WRD_TO_HEX
                        MOV  DX, OFFSET PSP_ADDRESS

                        MOV  AH, 09H
                        INT  21H

                        MOV  AX, ES:[3] ;SIZE OF PARA
                        MOV  SI, OFFSET AREA_SIZE
                        ADD  SI, 6
                        CALL PARA_TO_BYTE
                        MOV  DX, OFFSET AREA_SIZE

                        MOV  AH, 09H
                        INT  21H

                        MOV  BX, 8 ;SC SD
                        MOV  DX, OFFSET SC_SD
```

```

                                MOV  AH, 09H
                                INT  21H

                                MOV  CX, 7
PRINT_SC_SD_LOOP:
                                MOV  DL, ES:[BX]
                                MOV  AH, 02H
                                INT  21H
                                INC  BX
                                LOOP PRINT_SC_SD_LOOP

                                POP  CX
                                POP  DI
                                POP  SI
                                POP  DX
                                POP  AX
                                RET
PRINT_MCB                      ENDP

OFFSET_DEC                    PROC NEAR
OFFSET_DEC_LOOP:
                                CMP  BYTE PTR [SI], ' '
                                JNE  END_OFFSET_DEC
                                INC  SI
                                JMP  OFFSET_DEC_LOOP
END_OFFSET_DEC:
                                RET
OFFSET_DEC                    ENDP

PRINT_MCB_LIST                PROC NEAR
                                PUSH  AX
                                PUSH  BX
                                PUSH  ES
                                PUSH  DX

                                MOV  AH, 52H
                                INT  21H

```

```

MOV  AX, ES:[BX-2]
MOV  ES, AX
MOV  CL, 1
PRINT_LIST:
MOV  DX, OFFSET MCB_NUMBER

MOV  AH, 09H
INT  21H

MOV  AL, CL
MOV  SI, OFFSET DEC_NUMBER
ADD  SI, 2
CALL BYTE_TO_DEC
CALL OFFSET_DEC
MOV  DX, SI

MOV  AH, 09H
INT  21H

MOV  DL, ':'
MOV  AH, 02H
INT  21H
MOV  DL, ' '
MOV  AH, 02H
INT  21H
CALL PRINT_MCB
CALL PRINT_N
MOV  AL, ES:[0]
CMP  AL, 5AH
JE   END_MCB_LIST

MOV  BX, ES:[3]
MOV  AX, ES
ADD  AX, BX
INC  AX
MOV  ES, AX
INC  CL
JMP  PRINT_LIST

```

```

                                END_MCB_LIST:
                                    POP    DX
                                    POP    ES
                                    POP    BX
                                    POP    AX
                                    RET
PRINT_MCB_LIST                ENDP

DEL_FREE_MEMORY               PROC NEAR
                                PUSH    AX
                                PUSH    BX
                                PUSH    CX
                                PUSH    DX

                                LEA     AX, FINAL_END
                                MOV     BX, 10H
                                SUB     DX, DX
                                DIV     BX
                                INC     AX
                                MOV     BX, AX
                                MOV     AL, 0
                                MOV     AH, 4AH
                                INT     21H

                                POP     DX
                                POP     CX
                                POP     BX
                                POP     AX
                                RET
DEL_FREE_MEMORY               ENDP

MEMORY_REQUEST                PROC NEAR
                                PUSH    AX
                                PUSH    BX
                                PUSH    DX

                                MOV     BX, 1000H

```

```

                                MOV    AH, 48H
                                INT     21H
                                JC      MEM_FAILED
                                JMP     MEM_ACCEPTED

MEM_FAILED:
                                MOV     DX, OFFSET MEM_FAIL
                                MOV     AH, 09H
                                INT     21H
                                JMP     END_MEMORY_REQUEST

MEM_ACCEPTED:
                                MOV     DX, OFFSET MEM_ACCEPT
                                MOV     AH, 09H
                                INT     21H

END_MEMORY_REQUEST:
                                POP     DX
                                POP     BX
                                POP     AX
                                RET

MEMORY_REQUEST      ENDP

BEGIN:
; . . . . .

                                CALL    PRINT_AV_MEM_SIZE
                                CALL    PRINT_EXT_MEM_SIZE
                                CALL    MEMORY_REQUEST
                                CALL    DEL_FREE_MEMORY
                                CALL    PRINT_MCB_LIST

; . . . . .
; ВЫХОД В DOS

                                XOR     AL,AL
                                MOV     AH,4CH
                                INT     21H

FINAL_END:
TESTPC              ENDS

                                END     START

```