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

ЛАБОРАТОРНАЯ РАБОТА № 3

по дисциплине «Операционные системы»

ТЕМА: Исследование организации управления основной памятью.

Студентка гр. 9381	Москаленко Е.М
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.

В лабораторной работе исследуются структуры данных и работа функций управления памятью ядра операционной системы.

Функции:

Название	Описание
PRINT_SYMB	Вывод символа на экран (используя функцию DOS 02h,
	прерывания 21h).
TETR_TO_HEX	Перевод четырех младших битов регистре AL в 16- ричную цифру.
BYTE_TO_HEX	Перевод байта из AL в число 16-ной с.с. Символы записываются в регистры AL и AH.
WRD_TO_HEX	Перевод слова из АН в число в 16-ной с.с. Записывается в виде 4 символов по адресу из DI.
BYTE_TO_DEC	Перевод байта из AL в 16-ной с.с в число в 10-ной с.с Записывается по адресу, на который указывает SI (младшая цифра).
PRINT	Вывод строки на экран при помощи функции 9h прерывания 21h.

Выполнение работы.

- 1) Был написан и отлажен программный модуль .СОМ, который выбирает и распечатывает следующую информацию:
 - Количество доступной памяти

- Размер расширенной памяти
- Выводит цепочку битов управления памятью

```
F:\> lab3_1.com
Available memory:
                  648912 Ь
Extended memory:
                  15360 КЪ
List of MCB:
MCB type: 4Dh
                PSP adress: 0008h
                                        Size:
                                                    16 b
MCB type: 4Dh
                PSP adress: 0000h
                                        Size:
                                                    64 b
1CB type: 4Dh
                PSP adress: 0040h
                                        Size:
                                                   256 Ъ
MCB type: 4Dh
                PSP adress: 0192h
                                        Size:
                                                   144 Ь
MCB type: 5Ah
                PSP adress: 0192h
                                        Size:
                                                648912 Ъ
                                                                LAB3 1
```

2) Далее программа была изменена таким образом, что она освобождает память, которую она не занимает. Для этого использована функция 4Ah прерывания 21H. Результат программы представлен ниже.

```
F:\> lab3 2.com
                  648912 Ь
Available memory:
                   15360 КЪ
Extended memory:
List of MCB:
MCB type: 4Dh
               PSP adress: 0008h
                                       Size:
                                                   16 b
MCB type: 4Dh
               PSP adress: 0000h
                                       Size:
                                                   64 b
MCB type: 4Dh PSP adress: 0040h
                                                  256 Ъ
                                       Size:
MCB type: 4Dh PSP adress: 0192h
                                       Size:
                                                  144 Ъ
MCB type: 4Dh PSP adress: 0192h
                                       Size:
                                                  800 Ь
                                                               LAB3 2
MCB type: 5Ah PSP adress: 0000h
                                                               Q<u>o</u>ª ïθY=
                                       Size:
                                               648096 Ъ
```

3) Затем программа была изменена еще раз таким способом, что после освобождения памяти она запрашивает 64Кб памяти функцией 48Н прерывания 21Н. Результат программы представлен ниже.

```
F: \mathbb{N} \setminus Ab3_3.com
Available memory:
                   648912 Ъ
Memory request successed
Extended memory:
                    15360 КЪ
List of MCB:
                                          Size:
MCB type: 4Dh
                PSP adress: 0008h
                                                      16 b
MCB type: 4Dh
                PSP adress: 0000h
                                          Size:
                                                      64 b
MCB type: 4Dh
                                                     256 Ъ
                PSP adress: 0040h
                                          Size:
MCB type: 4Dh
                PSP adress: 0192h
                                          Size:
                                                     144 Ь
MCB type: 4Dh
                PSP adress: 0192h
                                          Size:
                                                     896 Ъ
                                                                   LAB3 3
MCB type: 4Dh
                PSP adress: 0192h
                                          Size:
                                                   65536 Ъ
                                                                   LAB3 3
                                          Size:
                                                  582448 Ъ
                                                                   ght (C)
MCB type: 5Ah
                PSP adress: 0000h
```

4) Программа была изменена таким образом, что запрашивает 64Кб памяти функцией 48Н прерывания 21Н до освобождения памяти. Результат программы представлен ниже.

```
F:\> lab3_4.com
Available memory: 648912 b
Memory request failed
Extended memory:
                   15360 КЪ
List of MCB:
KB type: 4Dh
               PSP adress: 0008h
                                        Size:
                                                    64 b
1CB type: 4Dh
               PSP adress: 0000h
                                        Size:
1CB tupe: 4Dh
               PSP adress: 0040h
                                        Size:
                                                   256 Ъ
1CB type: 4Dh
              PSP adress: 0192h
                                                   144 Ъ
                                        Size:
               PSP adress: 0192h
1CB type: 4Dh
                                        Size:
                                                   896 Ъ
                                                                 LAB3 4
KB type: 5Ah
               PSP adress: 0000h
                                                                 •B 3 ôff≈
                                        Size:
                                                648000 Ъ
```

Выводы:

В ходе выполнения лабораторной работы были освоены и применены функции управления памятью ядра операционной системы и изучены способы управления динамическими разделами памяти.

Ответы на контрольные вопросы:

1. Что означает "доступный объем памяти"?

Доступный объем памяти – это область основной памяти, выделенная программе.

2. Где МСВ блок вашей программы в списке?

MCB блок моей программы PSP адрес (сегментный адрес владельца участка памяти) = 0192h.

3.Какой размер памяти занимает программа в каждом случае?

В 1-ой версии программы она занимает всю доступную память: 648912 6 + 144 6 = 649056 **б.**

Во 2-ой — только объем, занимаемый самой программой: 800~6+1446= **944 б**.

В 3-ей – 65536 + 896 б + 144 б = 66576 б, 4кб выделили

В 4-ой – выделить 64кб невозможно: 896 б + 144 б = **1040 б**.

ПРИЛОЖЕНИЕ А

исходный код

lab3_1.asm

```
TESTPC segment
         ASSUME CS: TESTPC, DS: TESTPC, ES:nothing, SS:nothing
     ORG 100h
Start:
    jmp Begin
    Available memory db 'Available memory:
                                       Kb$'
    Extended memory db 'Extended memory:
                     db 'List of MCB:$'
    MCB
    MCBtype db 'MCB type: 00h$'
    PSP address db 'PSP adress: 0000h$'
    size_s db 'Size:
                        b$'
 Endl db 13, 10, '$'
 Tab db 9,'$'
TETR TO HEX PROC near
   and al, OFh
   cmp al, 09
   jbe next
   add al, 07
Next:
   add al, 30h
   ret
  TETR TO HEX endp
BYTE TO HEX PROC near
   push cx
   mov ah, al
   call TETR_TO_HEX
```

```
xchg al, ah
   mov cl, 4
    shr al, cl
    call TETR TO HEX
   pop cx
    ret
  BYTE TO HEX endp
WRD TO HEX PROC near
  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
  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_1:
  pop DX
  pop CX
  ret
BYTE TO DEC ENDP
WRD TO DEC PROC near
   push cx
   push dx
   mov cx, 10
wloop_bd:
   div cx
   or dl, 30h
   mov [si], dl
   dec si
   xor dx, dx
   cmp ax, 10
   jae
        wloop_bd
       al, 00h
   cmp
   jе
             wend_l
             al, 30h
   or
         [si], al
   mov
wend_1:
   pop
        dx
   pop
       CX
```

```
ret
WRD_TO_DEC endp
;вывод строки
PRINT proc near
   push ax
   mov ah, 09h
   int 21h
   pop ax
   ret
  PRINT endp
;print a symbol
Print_symb proc near
    push ax
    mov ah, 02h
    int
             21h
    pop
        ax
    ret
Print_symb endp
Begin:
    mov ah, 4Ah ;amount of available memory
    mov bx, OFFFFh ; max amount
    int 21h
    mov dx, 0
    mov ax, bx
    mov cx, 10h
    mul cx
    lea si, Available_memory
 add si, 24
    call wrd to dec
```

```
lea dx, Available memory
   call print
   lea dx, Endl
   call print
   mov al, 30h ; amount of extended memory
   out 70h, al
   in al, 71h
   mov bl, al
   mov al, 31h
   out 70h, al
   in al, 71h
   mov ah, al
   mov al, bl
   lea si, Extended memory
add si, 24
   mov dx, 0
   call wrd_to_dec
   lea dx, Extended memory
  call print
   lea dx, Endl
   call print
lea dx, MCB ; MCB blocks
call
      print
   lea dx, Endl
  call print
mov
       ah, 52h ; getting access to MCB
       21h
int
      ax, es:[bx-2]
mov
mov es, ax
```

```
;type of MCB
tag1:
    mov al, es:[0000h]
   call BYTE TO HEX
   lea di, MCBtype
   add di, 10
   mov [di], ax
   lea dx, MCBtype
   call print
   lea dx, Tab
   call print
   mov ax, es:[1h] ; segment address PSP owner of piece of memory
   lea di, PSP address
      di, 15
   add
   call WRD TO HEX
   lea dx, PSP_address
   call print
   lea dx, Tab
   call print
   mov ax, es:[0003h];
   mov cx, 10h
   mul cx
   lea si, size_s
   add si, 13
   call WRD TO DEC
         dx, Size_s
   lea
   call Print
   lea dx, Tab
```

```
call Print
   ;последние 8 байт
   push ds
   push es
  pop ds
  mov dx, 8h
  mov di, dx
  mov cx, 8h
tag2:
    cmp cx,0
   je tag3
  mov dl, byte PTR [di]
   call Print_symb
   dec cx
   inc di
      tag2
   jmp
tag3:
   pop ds
    lea dx, Endl
   call Print
   cmp byte ptr es:[0000h], 5Ah ; check is the last block or not
      Quit
   jе
       ax, es ; address of next block
   mov
   add ax, es:[3h]
      ax
   inc
   mov es, ax
   jmp
      tag1
Quit:
  mov ax, 0
  mov ah, 4ch ;finish
```

int 21h

TESTPC ENDS

END Start

lab3 2.asm

TESTPC segment

ASSUME CS: TESTPC, DS: TESTPC, ES:nothing, SS:nothing

ORG 100h

Start:

jmp Begin

Available memory db 'Available memory: b\$'

Extended memory db 'Extended memory: Kb\$'

MCB db 'List of MCB:\$'

MCBtype db 'MCB type: 00h\$'

PSP_address db 'PSP adress: 0000h\$'

size s db 'Size: b\$'

Endl db 13, 10, '\$'

Tab db 9,'\$'

TETR_TO_HEX PROC near

and al, OFh

cmp al, 09

jbe next

add al, 07

Next:

add al, 30h

ret

TETR TO HEX endp

BYTE_TO_HEX PROC near

push cx

mov ah, al

call TETR TO HEX

xchg al, ah

mov cl, 4

shr al, cl

call TETR TO HEX

pop cx

ret

BYTE TO HEX endp

WRD TO HEX PROC near

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

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_1:
pop DX
pop CX
ret
BYTE TO DEC ENDP
WRD_TO_DEC PROC near
push cx
push dx
mov cx, 10
wloop_bd:
div cx
or dl, 30h
mov [si], dl
dec si
xor dx, dx
cmp ax, 10
jae wloop_bd
cmp al, 00h
je wend_1
    al, 30h
or
mov [si], al
wend 1:
pop dx
```

pop cx

```
ret
WRD TO DEC endp
;вывод строки
PRINT proc near
push ax
mov ah, 09h
int 21h
pop ax
ret
PRINT endp
;print a symbol
Print_symb proc near
push ax
mov ah, 02h
int
        21h
pop ax
ret
Print_symb endp
Begin:
mov ah, 4Ah ;amount of available memory
mov bx, OFFFFh ; max amount
int 21h
mov dx, 0
mov ax, bx
mov cx, 10h
mul cx
lea si, Available memory
add si, 24
call wrd to dec
```

```
lea dx, Available memory
call print
lea dx, Endl
call print
; clear memory
lea ax, ProgEnds
mov bx, 10h
   dx, 0
mov
div
   bx
inc
   ax
mov bx, ax
mov al, 0
mov ah, 4Ah
   21h
int
    al, 30h
mov
            ; amount of extended memory
out
   70h, al
    al, 71h
in
mov bl, al
mov al, 31h
out 70h, al
in al, 71h
mov ah, al
mov al, bl
lea si, Extended_memory
add si, 24
mov dx, 0
call wrd_to_dec
lea dx, Extended memory
call print
lea dx, Endl
```

```
lea dx, MCB ; MCB blocks
call print
lea dx, Endl
call print
mov ah, 52h ;getting access to MCB
int 21h
mov ax, es: [bx-2]
mov es, ax
;type of MCB
tag1:
mov al, es:[0000h]
call BYTE TO HEX
lea di, MCBtype
add di, 10
mov [di], ax
lea dx, MCBtype
call print
lea dx, Tab
call print
mov ax, es:[1h] ; segment address PSP owner of piece of memory
lea di, PSP_address
add di, 15
call WRD_TO_HEX
lea dx, PSP_address
call print
lea dx, Tab
call print
```

call print

```
mov ax, es:[0003h];
mov cx, 10h
mul cx
lea si, size_s
add si, 13
call WRD TO DEC
lea dx, Size_s
call Print
lea dx, Tab
call Print
;последние 8 байт
push ds
push es
pop ds
mov dx, 8h
mov di, dx
mov cx, 8h
tag2:
    cx,0
cmp
je tag3
mov dl, byte PTR [di]
call Print symb
dec cx
inc di
jmp tag2
tag3:
pop ds
lea dx, Endl
call Print
cmp byte ptr es:[0000h], 5Ah ; check is the last block or not
```

```
je Quit
mov ax, es ; address of next block
add ax, es:[3h]
inc ax
mov es, ax
jmp tag1
Quit:
mov ax, 0
mov ah, 4ch ; finish
int 21h
ProgEnds:
TESTPC ENDS
END Start
                           lab3_3.asm
TESTPC segment
         ASSUME CS: TESTPC, DS: TESTPC, ES:nothing, SS:nothing
     ORG 100h
Start:
    jmp Begin
    Available memory db 'Available memory:
                                                  b$'
                                        Kb$'
    Extended memory db 'Extended memory:
                      db 'List of MCB:$'
    MCBtype db 'MCB type: 00h$'
    PSP address db 'PSP adress: 0000h$'
    size s db 'Size:
                               b$'
 Endl db 13, 10, '$'
 Tab db 9,'$'
 Fail db 'Memory request failed$'
 Success db 'Memory request successed$'
TETR TO HEX PROC near
```

```
and al, OFh
   cmp al, 09
   jbe next
   add
       al, 07
Next:
       al, 30h
   add
   ret
   TETR TO HEX endp
BYTE TO HEX PROC near
   push cx
   mov ah, al
   call TETR_TO_HEX
   xchg al, ah
   mov cl, 4
   shr al, cl
   call TETR_TO_HEX
   pop cx
   ret
  BYTE_TO_HEX endp
WRD_TO_HEX PROC near
  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
  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
WRD_TO_DEC PROC near
   push cx
   push dx
   mov cx, 10
wloop_bd:
```

div cx

```
or dl, 30h
   mov [si], dl
   dec
       si
   xor dx, dx
   cmp ax, 10
   jae
       wloop_bd
       al, 00h
   cmp
            wend_l
   jе
            al, 30h
   or
   mov [si], al
wend 1:
       dx
   pop
   pop
       CX
   ret
WRD TO DEC endp
PRINT proc near
   push ax
   mov ah, 09h
   int 21h
   pop
       ax
  ret
  PRINT endp
;print a symbol
Print_symb proc near
    push ax
    mov ah, 2h
             21h
    int
    pop
             ax
    ret
Print_symb endp
```

Begin:

```
ah, 4Ah ;amount of available memory
   mov
   mov bx, OFFFFh ; max amount
   int 21h
  mov dx, 0
   mov ax, bx
   mov cx, 10h
   mul cx
   lea si, Available memory
add si, 24
  call wrd to dec
   lea dx, Available memory
  call print
  lea dx, Endl
  call print
; clear memory
leaax, ProgEnds
mov bx, 10h
mov dx, 0
   bx
div
inc ax
mov bx, ax
      al, 0
mov
      ah, 4Ah
mov
int
      21h
;requst of memory
  mov
           ax, 0
           bx, 1000h ; 64kb
mov ah, 48h
int
      21h
```

```
jnc Success Ex
 lea dx, Fail
 call Print
    lea dx, Endl
    call Print
Success Ex:
 lea dx, Success
 call
       Print
       dx, Endl
 lea
 call
       Print
    mov al, 30h ; amount of extended memory
    out 70h, al
    in al, 71h
    mov bl, al
    mov al, 31h
    out 70h, al
    in al, 71h
    mov ah, al
    mov al, bl
    lea si, Extended memory
 add si, 24
    mov dx, 0
    call wrd to dec
    lea dx, Extended_memory
    call print
    lea dx, Endl
    call print
 lea dx, MCB ; MCB blocks
 call print
    lea dx, Endl
    call print
```

```
mov ah, 52h ; getting access to MCB
 int
       21h
       ax, es:[bx-2]
 mov
       es, ax
 mov
  ;type of MCB
tag1:
     mov al, es:[0000h]
   call BYTE_TO_HEX
        di, MCBtype
   lea
   add di, 10
   mov [di], ax
   lea
         dx, MCBtype
   call print
   lea dx, Tab
   call print
   mov ax, es:[1h] ; segment address PSP owner of piece of memory
       di, PSP address
   lea
      di, 15
   add
   call WRD TO HEX
   lea dx, PSP address
   call print
   lea dx, Tab
   call print
   mov ax, es:[0003h];
   mov cx, 10h
   mul
       CX
   lea si, size_s
```

```
add si, 13
   call WRD_TO_DEC
   lea dx, Size_s
   call Print
   lea dx, Tab
 call Print
  push ds
  push es
  pop ds
  mov dx, 8h
  mov di, dx
  mov cx, 8h
tag2:
    cmp
           cx,0
     je
           tag3
  mov dl, byte PTR [di]
  call Print_symb
   dec cx
   inc di
   jmp
           tag2
tag3:
   pop ds
    lea dx, Endl
   call Print
       byte ptr es:[0000h], 5Ah ; check is the last block or not
   cmp
        Quit
   jе
   mov ax, es ; address of next block
   add
      ax, es:[3h]
   inc
       ax
   mov es, ax
   jmp tag1
```

```
Quit:
   mov ax, 0
   mov ah, 4ch ; finish
   int 21h
ProgEnds:
TESTPC ENDS
    END Start
                            lab3_4.asm
TESTPC
         segment
         ASSUME CS: TESTPC, DS: TESTPC, ES:nothing, SS:nothing
     ORG 100h
Start:
    jmp Begin
    Available_memory db 'Available memory:
                                                    b$'
    Extended memory db 'Extended memory:
                                               Kb$'
                    db 'List of MCB:$'
    MCBtype db 'MCB type: 00h$'
    PSP address db 'PSP adress: 0000h$'
    size_s db 'Size:
                          b$'
 Endl db 13, 10, '$'
         db 9,'$'
 Tab
 Fail db 'Memory request failed$'
 Success db 'Memory request successed$'
TETR TO HEX PROC near
   and al, OFh
   cmp al, 09
       next
al, 07
   jbe
   add
Next:
   add al, 30h
   ret
  TETR TO HEX endp
BYTE TO HEX PROC near
   push cx
         ah, al
   mov
   call
         TETR TO HEX
   xchg al, ah
   mov cl, 4
   shr
        al, cl
   call TETR TO HEX
        CX
   pop
   ret
 BYTE TO HEX endp
WRD TO HEX PROC near
  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
     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 1:
     pop DX
     pop CX
     ret
  BYTE_TO_DEC ENDP
  WRD TO DEC PROC near
      push cx
      push dx
             cx, 10
      mov
  wloop_bd:
      div
             CX
             dl, 30h
      or
             [si], dl
      mov
      dec
             si
       xor
             dx, dx
             ax, 10
      cmp
      jae
             wloop bd
             al, 00h
      cmp
      jе
                   wend l
                  al, 30h
      or
      mov
            [si], al
```

```
wend 1:
   pop dx
   pop cx
   ret
WRD TO DEC endp
PRINT proc near
   push ax
   mov ah, 09h
         21h
   int
   pop ax
   ret
  PRINT endp
;print a symbol
Print_symb proc near
     push ax
     mov ah, 2h
     int
              21h
     pop
              ax
     ret
Print symb endp
Begin:
     mov ah, 4Ah ; amount of available memory mov bx, 0FFFFh ; max amount
     int 21h
     mov dx, 0
     mov ax, bx
     mov cx, 10h
     mul cx
     lea si, Available memory
 add si, 24
     call wrd to dec
     lea dx, Available memory
     call print
     lea dx, Endl
     call print
  ;requst of memory
   mov ax, 0
mov bx, 1000h; 64kb
 mov ah, 48h int 21h
          Success Ex
   jnc
  lea dx, Fail call Print
    lea
              dx, Endl
    call Print
  jmp Clear
```

```
Success Ex:
 lea dx, Success call Print lea dx, Endl call Print
 ; clear memory
Clear:
 leaax, ProgEnds
 mov bx, 10h
 mov dx, 0
 div bx
 inc ax
mov bx, ax
mov al, 0
mov ah, 4Ah
int 21h
     mov al, 30h ; amount of extended memory
     out 70h, al
          al, 71h
     in
     mov bl, al
     mov al, 31h
     out 70h, al
          al, 71h
     in
     mov ah, al
     mov al, bl
     lea si, Extended memory
 add si, 24
     mov dx, 0
     call wrd_to_dec
     lea dx, Extended memory
     call print
     lea dx, Endl
     call print
  lea dx, MCB ; MCB blocks call print
     lea dx, Endl
     call print
          ah, 52h ;getting access to MCB
          21h
 int
 mov
         ax, es:[bx-2]
         es, ax
 mov
   ;type of MCB
tag1:
    mov al, es:[0000h]
   call BYTE_TO_HEX
lea di, MCBtype
   add di, 10
   mov [di], ax
```

```
lea dx, MCBtype
        call print
                 dx, Tab
        lea
        call print
             ax, es:[1h] ; segment address PSP owner of piece of
        mov
memory
        lea
             di, PSP address
        add di, 15
        call WRD TO HEX
             dx, PSP address
        call print
        lea dx, Tab
        call print
        mov ax, es:[0003h];
             cx, 10h
        mov
        mul
              CX
        lea si, si
add si, 13
             si, size_s
        call WRD TO DEC
        lea dx, Size_s
        call Print
              dx, Tab
        lea
        call Print
        push ds
        push es
        pop
              ds
        mov dx, 8h
        mov di, dx
mov cx, 8h
    tag2:
         cmp cx,0
je tag3
nov dl, byte PTR [di]
        mov
        call Print_symb
        dec cx
        inc
                  di
        jmp
                  tag2
    tag3:
        pop ds
lea dx, Endl
        call Print
             byte ptr es:[0000h], 5Ah ; check is the last block or
        cmp
not
        jе
                  Quit
             ax, es ; address of next block
        mov
             ax, es:[3h]
        add
        inc
             ax
        mov
             es, ax
```

```
jmp tag1

Quit:
    mov ax, 0
    mov ah, 4ch ;finish
    int 21h

ProgEnds:
TESTPC ENDS
    END Start
```