

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

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

Студент гр. 7381

Адамов Я.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2019

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.

Описание функций.

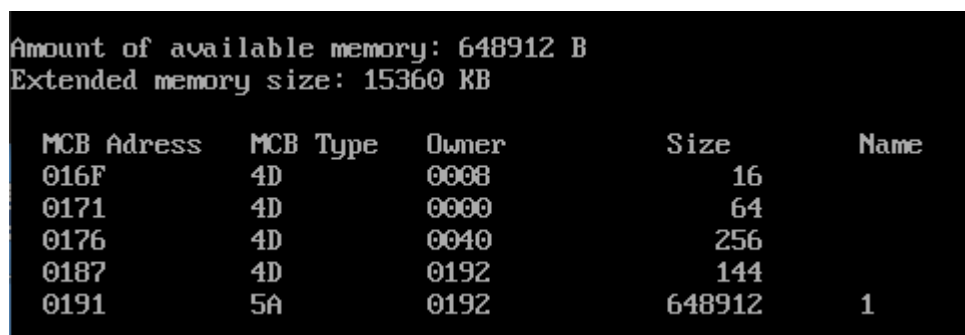
Название функции	Описание
TETR_TO_HEX	Перевод 4-битного числа в код символа в 16 с/с. Вспомогательная функция для BYTE_TO_HEX.
BYTE_TO_HEX	Байт в AL переводится в два символа шестн. числа в AX.
WRD_TO_HEX	Перевод в 16 с/с 16-ти разрядного числа, в AX – число, DI – адрес последнего символа.
PrintMsg	Печать строки, адрес которой помещен в DX.
PrintAvailableMemory	Вывод на экран количества доступной памяти.
PrintExtendedMemorySize	Вывод на экран размера расширенной памяти.
PrintMCB	Вывод на экран данных о блоке управления памятью.
PrintMemoryManagementUnits	Вывод на экран цепочки блоков управления памятью.

Описание структур данных.

Название	Тип	Описание
AvailableMemory	db	Строка для вывода количества доступной памяти.
ExtendedMemorySize	db	Строка для вывода размера расширенной памяти.
TableHead	db	Шапка таблицы блоков управления памятью.
MCB	db	Строка для вывода данных о блоке управления памятью.

Описание работы утилиты.

Программа выводит на экран количество доступной памяти, размер расширенной памяти и цепочку блоков управления памятью. Результат работы программы представлен на рис. 1. Результат работы версии программы, освобождающей не занимаемую память, представлен на рис. 2. Результат работы модификации второй версии программы, запрашивающей 64 КБ памяти, представлен на рис. 3. Результат работы версии программы, которая сначала запрашивает 64 КБ памяти, а затем освобождает неиспользуемую, представлен на рис. 4.



```
Amount of available memory: 648912 B
Extended memory size: 15360 KB

MCB Adress  MCB Type  Owner      Size      Name
016F        4D        0008       16
0171        4D        0000       64
0176        4D        0040      256
0187        4D        0192      144
0191        5A        0192     648912    1
```

Рисунок 1 – результат работы первой версии программы.

```
Amount of available memory: 648912 B
Extended memory size: 15360 KB
```

MCB Address	MCB Type	Owner	Size	Name
016F	4D	0008	16	
0171	4D	0000	64	
0176	4D	0040	256	
0187	4D	0192	144	
0191	4D	0192	13216	2
04CC	5A	0000	635680	

Рисунок 2 – результат работы второй версии программы.

```
Amount of available memory: 648912 B
Extended memory size: 15360 KB
```

MCB Address	MCB Type	Owner	Size	Name
016F	4D	0008	16	
0171	4D	0000	64	
0176	4D	0040	256	
0187	4D	0192	144	
0191	4D	0192	13328	3
04D3	4D	0192	65536	3
14D4	5A	0000	570016	&ë▲♀ &ë>

Рисунок 3 – результат работы третьей версии программы.

```
Amount of available memory: 648912 B
Extended memory size: 15360 KB

Requesting of 64 KB of memory is failed.
```

MCB Address	MCB Type	Owner	Size	Name
016F	4D	0008	16	
0171	4D	0000	64	
0176	4D	0040	256	
0187	4D	0192	144	
0191	4D	0192	14224	4
050B	5A	0000	634672	

Рисунок 4 – результат работы четвертой версии программы.

Вывод.

В ходе выполнения лабораторной работы были исследованы структуры данных и работы функций управления памятью ядра операционной системы.

Ответы на контрольные вопросы.

Что означает "доступный объём памяти"?

Объём оперативной памяти, которую может использовать программа.

Где МСВ блок Вашей программы в списке?

МСВ блок первой версии программы – последний; второй – предпоследней, так как за ним следует блок освобождённой памяти; третьей – пятый, за ним идём запрошенный блок памяти, а после – освобождённый; четвёртой – аналогично второй версии, так как не удалось выделить 64 КБ памяти из-за её отсутствия. Перед данным блоком в каждом случае имеется ещё один, владельцем которого является программа, он предназначен для хранения переменных среды.

Какой размер памяти занимает программа в каждом случае?

В первом случае программа занимает весь доступный объём памяти; во втором – только объём, занимаемый самой программой; в третьем – как и во втором, плюс 64 КБ выделенной памяти; в четвёртом – аналогично второму.

Приложение А. 1.asm.

TESTPC SEGMENT

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

ORG 100H

START: JMP BEGIN

; _____
; Данные

AvailableMemory db 0dh,0ah,'Amount of available memory:

B',0dh,0ah,'\$'

ExtendedMemorySize db 'Extended memory size: KB',0dh,0ah,'\$'

TableHead db 0dh,0ah,' MCB Adress MCB Type Owner Size
Name ',0dh,0ah,'\$'

MCB db '
,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

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
    push ax
    mov cx,10
loop_wd:
    div cx
    or dl,30h
    mov [si],dl
    dec si
    xor dx,dx
    cmp ax,10
    jae loop_wd
    cmp ax,00h
    jbe end_1_2
    or al,30h
    mov [si],al
end_1_2:
    pop ax
    pop dx
    pop cx
    ret
```

```
WRD_TO_DEC ENDP
```

```
PrintMsg PROC near
```

```
    push ax
    mov ah,09h
    int 21h
    pop ax
    ret
```

```
PrintMsg ENDP
```

```
PrintAvailableMemory PROC near
```

```
    push ax
    push bx
    push dx
    push si
```



```

mov ah,04Ah
mov bx,0FFFFh
int 21h
mov ax,10h
mul bx
lea si,AvailableMemory
add si,35
call WRD_TO_DEC
lea dx,AvailableMemory
call PrintMsg

```

```

pop si
pop dx
pop bx
pop ax
ret

```

PrintAvailableMemory ENDP

PrintExtendedMemorySize PROC near

```

push ax
push bx
push dx
push si

```

```

mov al,30h
out 70h,al
in al,71h
mov bl,al
mov al,31h
out 70h,al
in al,71h
mov ah,al
mov al,bl
sub dx,dx
lea si,ExtendedMemorySize
add si,26
call WRD_TO_DEC
lea dx,ExtendedMemorySize
call PrintMsg

```

```
pop si
pop dx
pop bx
pop ax
ret
```

```
PrintExtendedMemorySize ENDP
```

```
PrintMCB PROC near
```

```
    ; Address
    lea di,MCB
    mov ax,es
    add di,5
    call WRD_TO_HEX
```

```
    ; Type
    lea di,MCB
    add di,15
    xor ah,ah
    mov al,es:[0]
    call BYTE_TO_HEX
    mov [di],al
    inc di
    mov [di],ah
```

```
    ; Owner
    lea di,MCB
    mov ax,es:[1]
    add di,29
    call WRD_TO_HEX
```

```
    ; Size
    lea di,MCB
    mov ax,es:[3]
    mov bx,10h
    mul bx
    add di,46
    push si
    mov si,di
    call WRD_TO_DEC
```

```

    pop si

    ; Name
    lea di,MCB
    add di,53
    mov bx,0
print_:
    mov dl,es:[bx+8]
    mov [di],dl
    inc di
    inc bx
    cmp bx,8
    jne print_
    mov ax,es:[3]
    mov bl,es:[0]
    ret
PrintMCB ENDP

PrintMemoryManagementUnits PROC near
    lea dx,TableHead
    call PrintMsg
    mov ah,52h
    int 21h
    sub bx,2h
    mov es,es:[bx]
metka_1:
    call PrintMCB
    lea dx,MCB
    call PrintMsg
    mov cx,es
    add ax,cx
    inc ax
    mov es,ax
    cmp bl,4Dh
    je metka_1
    ret
PrintMemoryManagementUnits ENDP

```

```

; _____

```

; Код

BEGIN:

```
    call PrintAvailableMemory  
    call PrintExtendedMemorySize  
    call PrintMemoryManagementUnits
```

```
    xor al,al  
    mov ah,4ch  
    int 21h
```

```
TESTPC ENDS  
    END START
```

Приложение Б. 2.asm.

TESTPC SEGMENT

ASSUME CS:TESTPC, DS:TESTPC, ES:NOTHING, SS:NOTHING
ORG 100H

START: JMP BEGIN

; _____
; Данные

AvailableMemory db 0dh,0ah,'Amount of available memory:
B',0dh,0ah,'\$'
ExtendedMemorySize db 'Extended memory size: KB',0dh,0ah,'\$'
TableHead db 0dh,0ah,' MCB Adress MCB Type Owner Size
Name ',0dh,0ah,'\$'
MCB db '
' ,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
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
    push ax
    mov cx,10
loop_wd:
    div cx
    or dl,30h
    mov [si],dl
    dec si
    xor dx,dx
    cmp ax,10
    jae loop_wd
    cmp ax,00h
    jbe end_1_2
    or al,30h
    mov [si],al
end_1_2:
    pop ax
    pop dx
    pop cx
    ret
```

```
WRD_TO_DEC ENDP
```

```
PrintMsg PROC near
```

```
    push ax
    mov ah,09h
    int 21h
    pop ax
    ret
```

```
PrintMsg ENDP
```

```
PrintAvailableMemory PROC near
```

```
    push ax
    push bx
    push dx
    push si
```

```

mov ah,04Ah
mov bx,0FFFFh
int 21h
mov ax,10h
mul bx
lea si,AvailableMemory
add si,35
call WRD_TO_DEC
lea dx,AvailableMemory
call PrintMsg

```

```

pop si
pop dx
pop bx
pop ax
ret

```

PrintAvailableMemory ENDP

PrintExtendedMemorySize PROC near

```

push ax
push bx
push dx
push si

```

```

mov al,30h
out 70h,al
in al,71h
mov bl,al
mov al,31h
out 70h,al
in al,71h
mov ah,al
mov al,bl
sub dx,dx
lea si,ExtendedMemorySize
add si,26
call WRD_TO_DEC
lea dx,ExtendedMemorySize
call PrintMsg

```



```
pop si
pop dx
pop bx
pop ax
ret
```

```
PrintExtendedMemorySize ENDP
```

```
PrintMCB PROC near
```

```
    ; Address
    lea di,MCB
    mov ax,es
    add di,5
    call WRD_TO_HEX
```

```
    ; Type
    lea di,MCB
    add di,15
    xor ah,ah
    mov al,es:[0]
    call BYTE_TO_HEX
    mov [di],al
    inc di
    mov [di],ah
```

```
    ; Owner
    lea di,MCB
    mov ax,es:[1]
    add di,29
    call WRD_TO_HEX
```

```
    ; Size
    lea di,MCB
    mov ax,es:[3]
    mov bx,10h
    mul bx
    add di,46
    push si
    mov si,di
    call WRD_TO_DEC
```

```

    pop si

    ; Name
    lea di,MCB
    add di,53
    mov bx,0
print_:
    mov dl,es:[bx+8]
    mov [di],dl
    inc di
    inc bx
    cmp bx,8
    jne print_
    mov ax,es:[3]
    mov bl,es:[0]
    ret
PrintMCB ENDP

PrintMemoryManagementUnits PROC near
    lea dx,TableHead
    call PrintMsg
    mov ah,52h
    int 21h
    sub bx,2h
    mov es,es:[bx]
metka_1:
    call PrintMCB
    lea dx,MCB
    call PrintMsg
    mov cx,es
    add ax,cx
    inc ax
    mov es,ax
    cmp bl,4Dh
    je metka_1
    ret
PrintMemoryManagementUnits ENDP

```

```

; _____

```

; Код

BEGIN:

call PrintAvailableMemory
call PrintExtendedMemorySize

mov ah,4ah
lea bx,EndOfProgram
int 21h

call PrintMemoryManagementUnits

xor al,al
mov ah,4ch
int 21h

EndOfProgram db 0

TESTPC ENDS
END START

Приложение В. 3.asm.

TESTPC SEGMENT

ASSUME CS:TESTPC, DS:TESTPC, ES:NOTHING, SS:NOTHING
ORG 100H

START: JMP BEGIN

; _____
; Данные

AvailableMemory db 0dh,0ah,'Amount of available memory:
B',0dh,0ah,'\$'
ExtendedMemorySize db 'Extended memory size: KB',0dh,0ah,'\$'
TableHead db 0dh,0ah,' MCB Adress MCB Type Owner Size
Name ',0dh,0ah,'\$'
MCB db '
' ,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
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
    push ax
    mov cx,10
loop_wd:
    div cx
    or dl,30h
    mov [si],dl
    dec si
    xor dx,dx
    cmp ax,10
    jae loop_wd
    cmp ax,00h
    jbe end_1_2
    or al,30h
    mov [si],al
end_1_2:
    pop ax
    pop dx
    pop cx
    ret
```

```
WRD_TO_DEC ENDP
```

```
PrintMsg PROC near
```

```
    push ax
    mov ah,09h
    int 21h
    pop ax
    ret
```

```
PrintMsg ENDP
```

```
PrintAvailableMemory PROC near
```

```
    push ax
    push bx
    push dx
    push si
```

```

mov ah,04Ah
mov bx,0FFFFh
int 21h
mov ax,10h
mul bx
lea si,AvailableMemory
add si,35
call WRD_TO_DEC
lea dx,AvailableMemory
call PrintMsg

```

```

pop si
pop dx
pop bx
pop ax
ret

```

PrintAvailableMemory ENDP

PrintExtendedMemorySize PROC near

```

push ax
push bx
push dx
push si

```

```

mov al,30h
out 70h,al
in al,71h
mov bl,al
mov al,31h
out 70h,al
in al,71h
mov ah,al
mov al,bl
sub dx,dx
lea si,ExtendedMemorySize
add si,26
call WRD_TO_DEC
lea dx,ExtendedMemorySize
call PrintMsg

```

```
pop si
pop dx
pop bx
pop ax
ret
```

```
PrintExtendedMemorySize ENDP
```

```
PrintMCB PROC near
```

```
    ; Address
    lea di,MCB
    mov ax,es
    add di,5
    call WRD_TO_HEX
```

```
    ; Type
    lea di,MCB
    add di,15
    xor ah,ah
    mov al,es:[0]
    call BYTE_TO_HEX
    mov [di],al
    inc di
    mov [di],ah
```

```
    ; Owner
    lea di,MCB
    mov ax,es:[1]
    add di,29
    call WRD_TO_HEX
```

```
    ; Size
    lea di,MCB
    mov ax,es:[3]
    mov bx,10h
    mul bx
    add di,46
    push si
    mov si,di
    call WRD_TO_DEC
```



```

    pop si

    ; Name
    lea di,MCB
    add di,53
    mov bx,0
print_:
    mov dl,es:[bx+8]
    mov [di],dl
    inc di
    inc bx
    cmp bx,8
    jne print_
    mov ax,es:[3]
    mov bl,es:[0]
    ret
PrintMCB ENDP

```

```

PrintMemoryManagementUnits PROC near
    lea dx,TableHead
    call PrintMsg
    mov ah,52h
    int 21h
    sub bx,2h
    mov es,es:[bx]
metka_1:
    call PrintMCB
    lea dx,MCB
    call PrintMsg
    mov cx,es
    add ax,cx
    inc ax
    mov es,ax
    cmp bl,4Dh
    je metka_1
    ret
PrintMemoryManagementUnits ENDP

```

```

; _____

```

; Код

BEGIN:

```
    call PrintAvailableMemory  
    call PrintExtendedMemorySize
```

```
    mov ah,4ah  
    lea bx,EndOfProgram  
    int 21h
```

```
    mov ah,48h  
    mov bx,1000h  
    int 21h
```

```
    call PrintMemoryManagementUnits
```

```
    xor al,al  
    mov ah,4ch  
    int 21h
```

```
    EndOfProgram db 0
```

```
TESTPC ENDS  
    END START
```

Приложение Г. 4.asm.

TESTPC SEGMENT

ASSUME CS:TESTPC, DS:TESTPC, ES:NOTHING, SS:NOTHING
ORG 100H

START: JMP BEGIN

; _____
; Данные

AvailableMemory db 0dh,0ah,'Amount of available memory:
B',0dh,0ah,'\$'
ExtendedMemorySize db 'Extended memory size: KB',0dh,0ah,'\$'
TableHead db 0dh,0ah,' MCB Adress MCB Type Owner Size
Name ',0dh,0ah,'\$'
MCB db '
' ,0dh,0ah,'\$'
ErrorMessage db 0dh,0ah,'Requesting of 64 KB of memory is
failed.',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
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_1
    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
      push ax
      mov cx,10
loop_wd:
      div cx
      or dl,30h
      mov [si],dl
      dec si
      xor dx,dx
      cmp ax,10
      jae loop_wd
      cmp ax,00h
      jbe end_1_2
      or al,30h
      mov [si],al
end_1_2:
      pop ax
      pop dx
      pop cx
      ret
```

```
WRD_TO_DEC ENDP
```

```
PrintMsg PROC near
```

```
      push ax
      mov ah,09h
      int 21h
      pop ax
      ret
```

```
PrintMsg ENDP
```

```
PrintAvailableMemory PROC near
```

```
      push ax
      push bx
```

```

push dx
push si

mov ah,04Ah
mov bx,0FFFFh
int 21h
mov ax,10h
mul bx
lea si,AvailableMemory
add si,35
call WRD_TO_DEC
lea dx,AvailableMemory
call PrintMsg

pop si
pop dx
pop bx
pop ax
ret

```

PrintAvailableMemory ENDP

PrintExtendedMemorySize PROC near

```

push ax
push bx
push dx
push si

mov al,30h
out 70h,al
in al,71h
mov bl,al
mov al,31h
out 70h,al
in al,71h
mov ah,al
mov al,bl
sub dx,dx
lea si,ExtendedMemorySize
add si,26
call WRD_TO_DEC

```

```
    lea dx,ExtendedMemorySize
    call PrintMsg
```

```
    pop si
    pop dx
    pop bx
    pop ax
    ret
```

```
PrintExtendedMemorySize ENDP
```

```
PrintMCB PROC near
```

```
    ; Address
    lea di,MCB
    mov ax,es
    add di,5
    call WRD_TO_HEX
```

```
    ; Type
    lea di,MCB
    add di,15
    xor ah,ah
    mov al,es:[0]
    call BYTE_TO_HEX
    mov [di],al
    inc di
    mov [di],ah
```

```
    ; Owner
    lea di,MCB
    mov ax,es:[1]
    add di,29
    call WRD_TO_HEX
```

```
    ; Size
    lea di,MCB
    mov ax,es:[3]
    mov bx,10h
    mul bx
    add di,46
    push si
```

```

    mov si,di
    call WRD_TO_DEC
    pop si

    ; Name
    lea di,MCB
    add di,53
    mov bx,0
print_:
    mov dl,es:[bx+8]
    mov [di],dl
    inc di
    inc bx
    cmp bx,8
    jne print_
    mov ax,es:[3]
    mov bl,es:[0]
    ret
PrintMCB ENDP

```

```

PrintMemoryManagementUnits PROC near
    lea dx,TableHead
    call PrintMsg
    mov ah,52h
    int 21h
    sub bx,2h
    mov es,es:[bx]
metka_1:
    call PrintMCB
    lea dx,MCB
    call PrintMsg
    mov cx,es
    add ax,cx
    inc ax
    mov es,ax
    cmp bl,4Dh
    je metka_1
    ret
PrintMemoryManagementUnits ENDP

```



```
; _____  
; Код
```

BEGIN:

```
    call PrintAvailableMemory  
    call PrintExtendedMemorySize
```

```
    mov ah,48h  
    mov bx,1000h  
    int 21h
```

```
    jc m_1  
    jmp m_2
```

```
m_1:  
    lea dx, ErrorMessage  
    call PrintMsg
```

```
M_2:  
    mov ah,4Ah  
    lea bx,EndOfProgram  
    int 21h
```

```
    call PrintMemoryManagementUnits
```

```
    xor al,al  
    mov ah,4ch  
    int 21h
```

```
    EndOfProgram db 0
```

```
TESTPC ENDS  
    END START
```