

**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
ПАМЯТЬЮ.**

Факультет: КТИ

Дата выполнения работы: 25.03.2021

Студент гр. 9381

Семенов А. Н.

Преподаватель

Ефремов М. А.

Санкт-Петербург
2021

Цель работы.

Исследование организации управления памятью, ориентируясь на тип основной памяти, реализованный в компьютере и способ организации, принятый в ОС. Рассмотрение нестраничной памяти и способа управления динамическими разделами. Исследование структур данных и работы функций управления памятью ядра операционной системы.

Функции и структуры данных.

TETR_TO_HEX – переводит значение 4 младших битов регистра `al` в цифру 16-ричной системы счисления в виде символа, которая кладется в регистр `al`.

BYTE_TO_HEX – переводит значение байта, содержащегося в регистре `al` в двухсимвольное число в шестнадцатеричной системе счисления, которые кладется в регистр `ax`: код первого символа в `al`, второго в `ah`.

WRD_TO_HEX – переводит значение регистра `AX` в шестнадцатеричное число в виде 4 символов, которые кладутся в память по адресу `di`.

WRD_TO_DEC – переводит значение регистра `AX` в десятичное число в виде символов, которые кладутся в память по адресу `di`.

Ход работы

1. Написание исходного текста `.COM` модуля на языке ассемблера, файл: `LAB_3-1.ASM`. Данная программа поэтапно выводит следующую информацию: количество доступной памяти, размер расширенной памяти, цепочку блоков управления памятью.
2. Трансляция исходного кода командой: `masm LAB_3-1.ASM`, отладка и получение объектного модуля `LAB_3-1.OBJ`.
3. Сборка объектного модуля командой: `link LAB_3-1.OBJ`, и получение загрузочного модуля `LAB_3-1.EXE`.
4. Получение `.COM` модуля: `LAB_3-1.COM`, с помощью команды: `exe2bin LAB_3-1.EXE`.

5. Запуск программы в терминале ДОС и перенаправление выходных данных в файл *output.txt* командой: *lab_3-1.com > output.txt*. Результаты работы программы представлены на рисунке 1.

```
Количество доступной памяти: 633 килобайт 720 байт
Размер расширенной памяти: 15360 килобайт
-----
МСВ-адрес: 016F
МСВ-тип: 4D
PSP-адрес владельца участка памяти: 0008
Размер участка: 0 килобайт 16 байт
Зарезервировано:
-----
МСВ-адрес: 0171
МСВ-тип: 4D
PSP-адрес владельца участка памяти: 0000
Размер участка: 0 килобайт 64 байт
Зарезервировано:
-----
МСВ-адрес: 0176
МСВ-тип: 4D
PSP-адрес владельца участка памяти: 0040
Размер участка: 0 килобайт 256 байт
Зарезервировано:
-----
МСВ-адрес: 0187
МСВ-тип: 4D
PSP-адрес владельца участка памяти: 0192
Размер участка: 0 килобайт 144 байт
Зарезервировано:
-----
МСВ-адрес: 0191
МСВ-тип: 5A
PSP-адрес владельца участка памяти: 0192
Размер участка: 633 килобайт 720 байт
Зарезервировано: LAB_3-1
```

Рис. 1. Запуск программы LAB_3-1.COM

6. Затем исходный текст .COM модуля на языке ассемблера модифицируется так, чтобы программа освобождала лишнюю, не используемую ею память. Измененный текст сохраняется в файле: *LAB_3-2.ASM*. Данный файл компилируется, линкуется и запускается аналогично первому файлу. Результаты работы программы представлены на рисунке 2.

```

Количество доступной памяти: 633 килобайт 720 байт
Размер расширенной памяти: 15360 килобайт
-----
МСВ-адрес: 016F
МСВ-тип: 4D
PSP-адрес владельца участка памяти: 0008
Размер участка: 0 килобайт 16 байт
Зарезервировано:
-----
МСВ-адрес: 0171
МСВ-тип: 4D
PSP-адрес владельца участка памяти: 0000
Размер участка: 0 килобайт 64 байт
Зарезервировано:
-----
МСВ-адрес: 0176
МСВ-тип: 4D
PSP-адрес владельца участка памяти: 0040
Размер участка: 0 килобайт 256 байт
Зарезервировано:
-----
МСВ-адрес: 0187
МСВ-тип: 4D
PSP-адрес владельца участка памяти: 0192
Размер участка: 0 килобайт 144 байт
Зарезервировано:
-----
МСВ-адрес: 0191
МСВ-тип: 4D
PSP-адрес владельца участка памяти: 0192
Размер участка: 4 килобайт 140 байт
Зарезервировано: LAB_3-2
-----
МСВ-адрес: 0292
МСВ-тип: 5A
PSP-адрес владельца участка памяти: 0000
Размер участка: 629 килобайт 704 байт
Зарезервировано: АСК1ш-ЕТХ.Г>

```

Рис. 2. Запуск программы LAB_3-2.COM

7. Далее исходный текст .COM модуля *LAB_3-2.ASM* на языке ассемблера модифицируется так, чтобы программа после освобождения лишней, не используемой ею памяти, запрашивала 64 килобайта дополнительной памяти. Измененный текст сохраняется в файле: *LAB_3-3.ASM*. Данный файл компилируется, линкуется и запускается аналогично предыдущим файлам. Результаты работы программы представлены на рисунке 3.

```

Количество доступной памяти: 633 килобайт 720 байт
Размер расширенной памяти: 15360 килобайт
-----
МСВ-адрес: 016F
МСВ-тип: 4D
PSP-адрес владельца участка памяти: 0008
Размер участка: 0 килобайт 16 байт
Зарезервировано:
-----
МСВ-адрес: 0171
МСВ-тип: 4D
PSP-адрес владельца участка памяти: 0000
Размер участка: 0 килобайт 64 байт
Зарезервировано:
-----
МСВ-адрес: 0176
МСВ-тип: 4D
PSP-адрес владельца участка памяти: 0040
Размер участка: 0 килобайт 256 байт
Зарезервировано:
-----
МСВ-адрес: 0187
МСВ-тип: 4D
PSP-адрес владельца участка памяти: 0192
Размер участка: 0 килобайт 144 байт
Зарезервировано:
-----
МСВ-адрес: 0191
МСВ-тип: 4D
PSP-адрес владельца участка памяти: 0192
Размер участка: 4 килобайт 140 байт
Зарезервировано: LAB_3-3
-----
МСВ-адрес: 0292
МСВ-тип: 4D
PSP-адрес владельца участка памяти: 0192
Размер участка: NOI64 килобайт 140 байт
Зарезервировано: ACKi
LAB_3-3
-----
МСВ-адрес: 1293
МСВ-тип: 5A
PSP-адрес владельца участка памяти: 0000
Размер участка: 565 килобайт 688 байт
Зарезервировано:

```

Рис. 3. Запуск программы LAB_3-3.COM

8. Наконец, исходный текст .COM модуля *LAB_3-3.ASM* на языке ассемблера модифицируется так, чтобы программа сначала запрашивала 64 килобайта дополнительной памяти, а затем освобождала лишнюю, не используемую ею память. Измененный текст сохраняется в файле: *LAB_3-4.ASM*. Данный файл

компилируется, линкуется и запускается аналогично предыдущим файлам. Результаты работы программы представлены на рисунке 4.

```
Количество доступной памяти: 633 килобайт 720 байт  
Размер расширенной памяти: 15360 килобайт  
Ошибка с кодом: 8
```

Рис. 4. Запуск программы LAB_3-4.COM

Результаты исследования проблем.

Сравнивая результаты вызовов модулей *lab_3-1.com* и *lab_3-2.com*, в последнем появился свободный блок, размер которого стал равен чуть больше 4 Кб.

Сравнивая результаты вызовов модулей *lab_3-2.com* и *lab_3-3.com*, в последнем появился еще один свободный блок памяти, размером 64 Кб.

Сравнивая результаты вызовов модулей *lab_3-3.com* и *lab_3-4.com*, в последнем программа завершилась с ошибкой после попытки выделить дополнительные 64 Кб памяти в связи с тем, что свободной памяти не хватило. Код такой ошибки: 8.

Ответы на вопросы.

1) Что означает «Доступный объем памяти»? Объем памяти, которую программа может использовать при выполнении.

2) Где МСВ блок вашей программы в списке? У всех МСВ-блоков данной программы поле «PSP-адрес владельца участка памяти» имеет значение 0192.

3) Какой размер памяти занимает программа в каждом случае?

Lab_3-1.com: 633 Кб 720 байт;

Lab_3-2.com: 4 Кб 140 байт;

Lab_3-3.com: 4 Кб 140 байт + 64 Кб = 68 Кб 140 байт;

Lab_3-4.com: 633 Кб 720 байт.

Вывод.

В ходе лабораторной работы было проведено исследование организации управления памятью, структур данных и работы функций управления памятью ядра операционной системы. Была разработана, протранслирована и протестирована собственная программа на языке Ассемблера, печатающая информацию о количестве доступной памяти, размере расширенной памяти и выводящая цепочку блоков управления памятью.