

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

Студент гр. 9381

Камакин Д.В.

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.

TETR_TO_HEX — перевод значения 4-ёх младших битов в регистре AL в цифру 16-ой CC (остаётся в регистре AL).

BYTE_TO_HEX — перевод значения байта из регистра AL в число 16-ой CC. Результат записывается в AL и AH.

WRD_TO_HEX — перевод значения слова в регистре AX в число 16-ой CC и запись в виде 4 символов по адресу, записанному в регистре DI.

BYTE_TO_DEC — перевод байта в регистре AL в 10-ую CC. Символы записываются по адресу, записанному в регистре SI.

OUTPUT — вызов функции 09h прерывания int 21h (посылает строку на стандартный вывод).

HEX_TO_BYTE — запись и перевод числа 16-ой cc из ax в 10-ую по адресу, записанному в di.

MEM_AVAILABLE — записывает количество доступной памяти в адрес, записанный в di.

MEM_EXTENDED — записывает количество расширенной памяти в адрес, записанный в di.

OUTPUT_MCB — выводит цепочку блоков управления памятью.

MEM_FREE — освобождает неиспользуемую программой память.

MEM_REQUEST — запрашивает 1000h (64кб) памяти и выводит результат работы.

Последовательность действий.

Записываем смещения строк, в которые требуется поместить информацию, в регистр DI, после чего вызываем соответствующие функции.

Ход работы

1. Написан модуль типа .COM, который выбирает и распечатывает следующую информацию: количество доступной памяти, размер расширенной памяти и выводит цепочку блоков управления памятью.

```
F:\>lb3.com
Available memory: 648912 bytes
Extended memory: 15370 bytes
MCB table:
PSP address: 0008 Size: 16 SC/SD:
PSP address: 0000 Size: 64 SC/SD:
PSP address: 0040 Size: 256 SC/SD:
PSP address: 0192 Size: 144 SC/SD:
PSP address: 0192 Size: 648912 SC/SD: LB3
```

2. Программа была изменена таким образом, чтобы она освобождала память, которую она не занимает

```
F:\>lb3.com
Available memory: 648912 bytes
Extended memory: 15370 bytes
MCB table:
PSP address: 0008 Size: 16 SC/SD:
PSP address: 0000 Size: 64 SC/SD:
PSP address: 0040 Size: 256 SC/SD:
PSP address: 0192 Size: 144 SC/SD:
PSP address: 0192 Size: 6432 SC/SD: LB3
PSP address: 0000 Size: 642464 SC/SD: .i6p
```

3. Программа была изменена таким образом, чтобы после освобождения памяти программа запрашивала 64Кб памяти функцией 48h прерывания 21h

```
F:\>lb3.com
Available memory: 648912 bytes
Extended memory: 15370 bytes
Memory request succeeded
MCB table:
PSP address: 0008 Size: 16 SC/SD:
PSP address: 0000 Size: 64 SC/SD:
PSP address: 0040 Size: 256 SC/SD:
PSP address: 0192 Size: 144 SC/SD:
PSP address: 0192 Size: 6432 SC/SD: LB3
PSP address: 0192 Size: 65536 SC/SD: LB3
PSP address: 0000 Size: 576912 SC/SD:
```

4. Программа была изменена таким образом, чтобы она запрашивала 64Кб памяти функцией 48h прерывания 21h до освобождения памяти.

```
F:\>lb3.com
Memory request failed
Available memory: 648912 bytes
Extended memory: 15370 bytes
MCB table:
PSP address: 0008 Size: 16 SC/SD:
PSP address: 0000 Size: 64 SC/SD:
PSP address: 0040 Size: 256 SC/SD:
PSP address: 0192 Size: 144 SC/SD:
PSP address: 0192 Size: 6432 SC/SD: LB3
PSP address: 0000 Size: 642464 SC/SD: .i6p
```

Результаты исследования проблем.

1. Что означает «доступный объём памяти»?

Доступный объём памяти — это максимальный объём памяти, который доступен для запуска и выполнения программ.

2. Где МСВ блок Вашей программы в списке?

Блок МСВ программы имеет поле PSP address равное 0192.

3. Какой размер памяти занимает программа в каждом случае?

1. Весь доступный объём (648912 байт)

2. Программа занимает только необходимый ей объём (6432 байт)
3. $6432 + \text{запрошенные } 65536 = 71968 \text{ байт}$
4. 6432 байт (только то, что необходимо для работы программы)

Выводы

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