Минобрнауки России  
Федеральное государственное автономное образовательное  
Учреждение высшего образования  
«Санкт-Петербургский государственный электротехнический  
Университет им. В.И. Ульянова (Ленина)»  
(СПГЭТУ «ЛЭТИ»)  
Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

отчет

**по лабораторной работе №2**

**по дисциплине «Операционные системы»**

Тема: Управление памятью

Выполнил студент группы 9308: Дементьев Д.П.

Принял: к.т.н., доцент Тимофеев А.В.

Санкт-Петербург

2021

**Цель работы:** исследовать способы управления памятью с помощью программного интерфейса Win32.

**Исходный код (GitHub):**

https://github.com/CamelNotFemale/5\_sem\_OS/tree/main/lab2

**Задание 2.1.** Исследовать виртуальное адресное пространство процесса.

Главное меню программы:

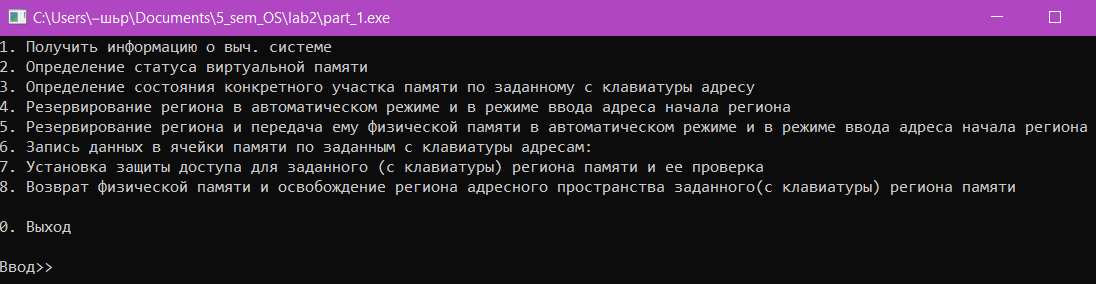


Рисунок 1. Главное меню

Получение информации о вычислительной системе:

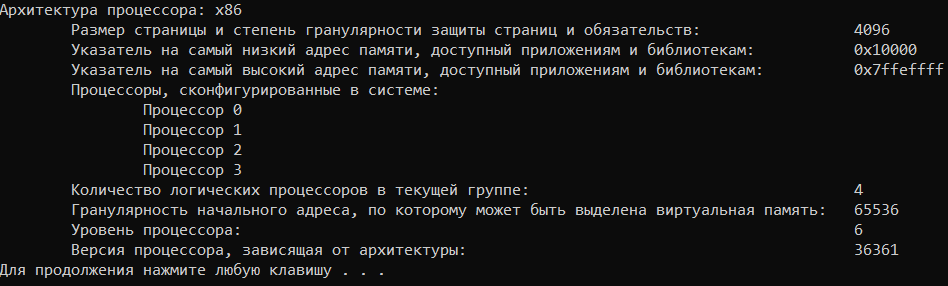


Рисунок 2. Получение информации о вычислительной системе

Определение статуса виртуальной памяти:

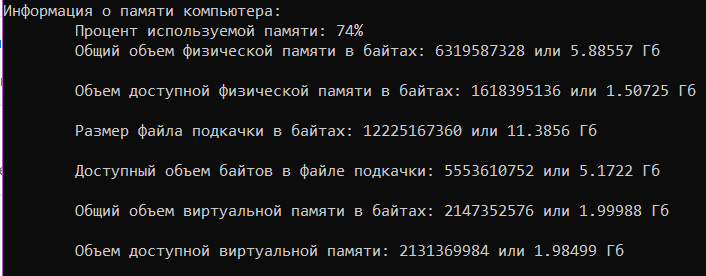


Рисунок 3. Определение статуса виртуальной памяти.

Определение состояния конкретного участка памяти (по адресу):

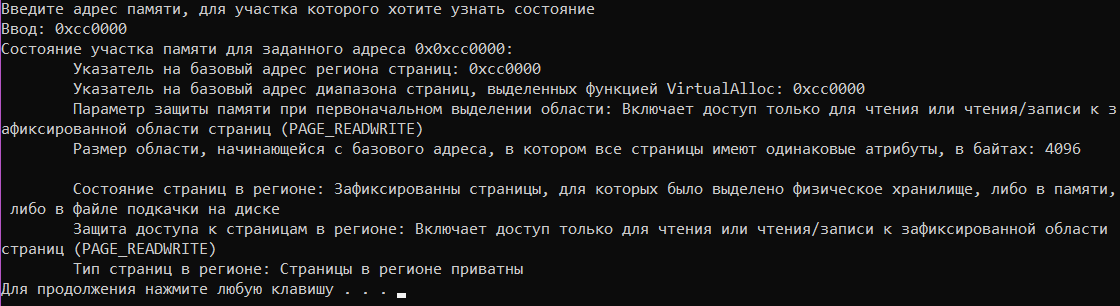


Рисунок 4. Определение состояния конкретного участка памяти по адресу 0x024A0000

Резервирование региона в автоматическом режиме:

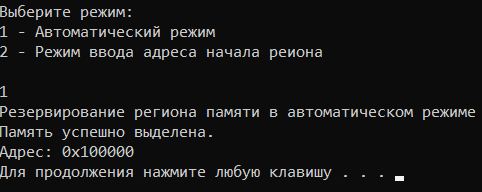


Рисунок 5. Резервирование в автоматическом режиме

Резервирование региона и передача ему физической памяти в режиме ввода адреса начала региона:

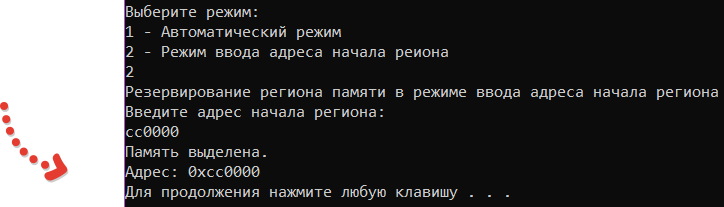


Рисунок 6. Резервирование региона и передача ему физической памяти в режиме ввода адреса начала (0xcc0000)

Запись данных в ячейки памяти по заданным с клавиатуры адресам:

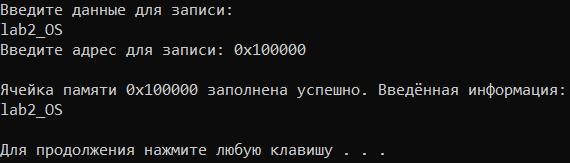


Рисунок 7. Запись данных в ячейки памяти по заданным с клавиатуры адресам (0x100000)

Установка защиты доступа для заданного региона памяти и её проверка:

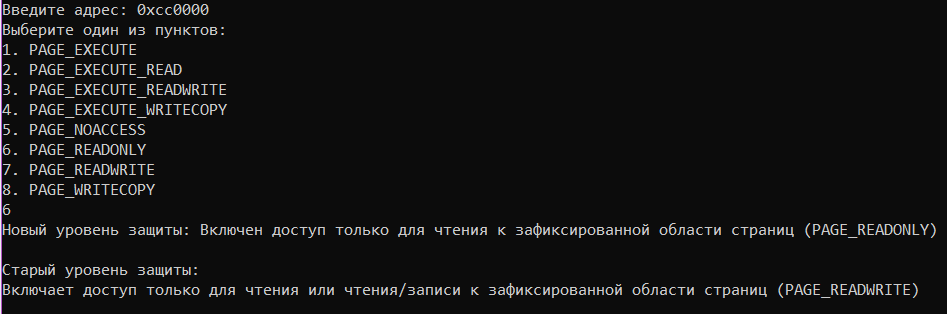


Рисунок 8. Установка защиты доступа для заданного региона памяти и её проверка (0xcc0000)

Возврат физической памяти и освобождение региона адресного пространства заданного региона памяти:



Рисунок 9. Возврат физической памяти и освобождение региона адресного (0xcc0000)

**Выводы (задание 2.1.):** в ходе выполнения программы проверили корректность её работы. Резервирование и освобождение для регионов памяти выполняется правильно, за исключением ситуаций, когда регион уже зарезервирован ранее. Успешность изменения уровня доступа памяти была проверена последовательностью действий: VirtualQuery(..), VirtualProtect(), VirtualQuery(..) – изменения были корректно отображены.

Автоматические режимы резервирования регионов памяти и передачи им физ. памяти были реализованы при помощи передачи в функцию VirtualAlloc() параметра NULL в качестве начального адреса региона для выделения. При резервировании памяти с заданием начального адреса важно учитывать округление исходного адреса до ближайшей степени гранулярности выделения памяти (в нашем случае 4096 бит).

**Задание 2.2**. Использование проецируемых файлов для обмена данными между процессами.

1. Запустим программу-писатель.
2. Введём имя файла.
3. Введём имя отображения.
4. Введём данные, который будут записаны в файл.

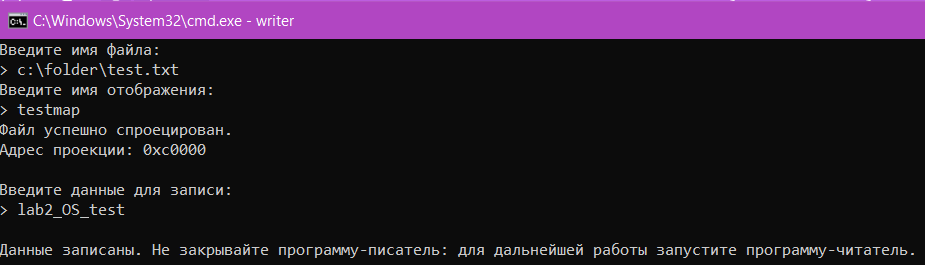


Рисунок 10. Работа программы-писателя.

1. Теперь запустим программу-читатель.
2. Введём имя отображения (оно должно совпадать с именем отображения из программы-писателя).
3. Выведем информацию, введённую в программе-писателе.

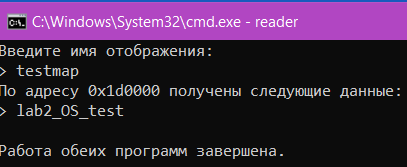


Рисунок 11. Работа программы-читателя.

1. Закроем обе программы.

**Выводы (задание 2.2.):** судя по вышеприведённому алгоритму действий и скриншотам, программы работают успешно. Обе программы ссылаются на одно и то же отображение, созданное программой-писателем.

Видно, что адреса проекций в программе-писателе и программе-читателе различаются. Происходит это потому, что каждая из программ проецирует созданный объект на своё ВАП и используют эту часть адресного пространства как разделяемую область данных.

**Общий вывод:** были исследованы способы управления памятью в Win32, в частности, работой с виртуальной памятью и проецируемыми файлами.

В ходе выполнения первой части была изучена работа с виртуальной памятью. Использование виртуальной памяти позволяет предоставлять бОльшие объемы памяти по сравнению с физической памятью, доступной системе, путём использования вторичной памяти.

Также ВП позволяет изолировать выполняющиеся программы друг от друга благодаря присвоению каждому процессу своего набора страниц, что в свою очередь обеспечивает безопасность выполнения процессов.

Как работает виртуальная память:

- занимаемая процессором память разбивается на несколько частей (они же страницы);

- логический адрес, к которому обращается процесс, динамически транслируется в физический адрес;

- если страница не находится в физической памяти, необходимо совершить подкачку с диска;

В ходе выполнения второй части было изучено проецирование файлов в память. Проецируемые файлы, как и виртуальная память, позволяют резервировать регион адресного пространства и передавать ему физическую память. Различие между этими механизмами состоит в том, что в последнем случае физическая память не выделяется из системного страничного файла, а берется из файла, уже находящегося на диске.