Разработка приложений с активным использованием больших объемов памяти

Лабораторная работа

Ревизия: 0.1

Последняя модификация: Vlad Kovtun

Дата последней модификации: 16.03.2010 0:38:00

История изменений

03.03.2010 – Версия 0.1. Первичный документ. Влад Ковтун

Содержание

Лабораторная работа 4. Разработка приложений с активным использованием больших объемов памяти

Вопросы

1. Постановка задачи.
2. Методические рекомендации.

Постановка задачи

Цель

Разработать библиотеку управления оперативной памятью, которая позволяет снизить время выделения памяти и ее фрагментированость.

Задачи

1. Ознакомиться с понятиями управления памятью в ОС.

2. Ознакомиться с понятиями управления виртуальной памятью ОС, а также алгоритмами управления памятью.

3. Ознакомиться с особенностями управления памятью средствами C/C++/C#.

4. Ознакомиться с особенностями управления памятью средствами Microsoft Windows:

- Динамическое выделение памяти в куче.
- Выделение виртуальной памяти.
- Запись в файл отображаемый в память.

5. Разработать консольное приложение для выделения большого количества блоков оперативной памяти.

6.1. Оценить производительность приложения, при выделении блоков памяти размером: 1 кб, 10 кб, 100 кб, 1 Мб в куче. Количество блоков варьируется в диапазоне 1 млн. – 100 млн. Как соотносится время выделения памяти с превышением объема физической памяти.

6.2. Оценить производительность приложения, при выделении блоков памяти размером: 1 кб, 10 кб, 100 кб, 1 Мб в виртуальной памяти. Количество блоков варьируется в диапазоне 1 млн. – 100 млн. Как соотносится время выделения памяти с превышением объема физической памяти.

6.3. Оценить производительность приложения, при выделении блоков памяти (создании новых структур) размером: 1 кб, 10 кб, 100 кб, 1 Мб в файле, отображаемом в оперативную память. Количество блоков варьируется в диапазоне 1 млн. – 100 млн. Как соотносится время выделения памяти с превышением объема физической памяти.

7. Оформить функциональную спецификацию на приложение. В функциональной спецификации обязательно указать, каким образом производится тестирование на корректность.

8. Оформить отчет к лабораторной работе.

Вход

Для управления приложением (первый подход) предлагается использовать ключи командной строки:

- /bs:<block size> – Размер выделяемого блока памяти в килобайтах.
- /bc:<block count> - количество блоков памяти которые необходимо выделить.
- /lc:<loops count> – количество циклов тестирования, для повышения точности оценки.
- /o:<filename> - имя файла отчета, куда выводятся результаты тестирования. Если имя файла не указано, то вывод производится на консоль.
- /? - вывод информации о допустимых ключах командной строки.

Пример

Оценка производительности в выделении блоков памяти размером 1 кб, в количестве 1000 шт и количестве итераций цикла 1000 раз.

C:/>mempool.exe /bs:1 /bc:1000 /lc:1000

Вывод

Во время работы приложения с большими объемами информации, что требует длительного времени обработки, рекомендуется выводить информацию о статусе приложения, а также о корректности его работы на консоль.

Другими словами, необходимо анализировать, сколько процентов работы выполнено на текущий момент.

Допускается перенаправление вывода консоли в текстовый файл, например:

C:/>mempool.exe /bs:1 /bc:1000 /lc:1000 >report.txt

Требования

- Архитектура приложения строится по модульному принципу.
- За основу принимается стандартная библиотека C++ (в случае разработки на C++).
- Рекомендуется использовать защищенные ресурсы и указатели.
- Обязательным является обработка исключений.
- Во время работы приложения обязательным является отображение информации о статусе приложения, т.е. оно работает, обработано столько-то процентов данных.
- Следует предусмотреть возможность работы приложения в отладочном режиме, когда вся информация заносится в лог приложения. Допускается ведение персонального лога для каждого из процессов (потоков).
- Исходный код обязан быть комментирован. Для C# следует использовать нотацию Microsoft [1], для C++ следует использовать нотацию doxygen [2].

Методические указания для самостоятельной работы

При подготовке к лабораторной работе необходимо:

- Ознакомиться с понятиями управления памятью в ОС.
- Ознакомиться с понятиями управления виртуальной памятью ОС, а также алгоритмами управления памятью.
- Ознакомиться с особенностями управления памятью средствами C/C++/C#. Например:
    - new, delete.
    - Alloc/free, malloc/free, calloc/free, realloc, expand.
- Ознакомиться с особенностями управления памятью средствами Microsoft Windows:
    - Динамическое выделение памяти в куче:
        - HeapAlloc/HealFree,
        - HeapCreate/HeapDestroy.
        - HeapLock/HeapUnlock.
    - Выделение виртуальной памяти:
        - VirtualAlloc/VirtualFree,
        - VirtualLock/VirtualUnlock,
        - VirtualProtect.
    - Запись в файл отображаемый в память:
        - CreateFileMapping/OpenFileMapping/CloseHandle,

- MapViewOfFile/UnmapViewOfFile/FlushViewOfFile.

Алгоритм

Данную задачу следует решить несколькими способами.

Первый подход, который реализует поставленную задачу, описан ниже.

1. Разбор ключей командной строки.

2. Выполнение циклического выделения памяти с помощью функций динамического управления памятью, встроенными в язык программирования (без использования пула объектов).

3. Выполнение циклического выделения памяти с помощью функций динамического управления памятью, встроенными в язык программирования (с использованием пула объектов).

4. Вывод результатов оценки времени выполнения выделения памяти.

Второй подход аналогичен первому, отличие состоит лишь в использовании функций выделения памяти в куче средствами ОС.

Третий подход аналогичен первому, отличие состоит лишь в использовании функций выделения памяти в виртуальной памяти средствами ОС.

Четвертый подход аналогичен первому, отличие состоит лишь в использовании функций выделения памяти в файле отображаемом в память средствами ОС.

Тестирование

Тестирование приложения осуществляется в несколько этапов:

- Разработка Unit Test в рамках проекта. Данный подход позволит проверить корректность реализации алгоритма.

Теоретические сведения

Необходимая информация по Windows платформе доступна в [1].

Литература

1. Microsoft Developer Network. URL: http://www.msdn.com

2. Формирование документации к исходному коду с помощью средства doxygen. URL: www.nrjetix.com/r-and-d/lectures