

Лабораторная работа № 3

Тема: Оценка производительности подсистемы памяти.

Задание: разработать программу (benchmark) для оценки производительности подсистемы памяти.

1. Написать программу(функцию) на языке C/C++/C# для оценки производительности подсистемы памяти.

На вход программы подать следующие аргументы.

- 1) Подсистема памяти. Предусмотреть возможность указать подсистему для проверки производительности: RAM (оперативная память), HDD/SSD и flash.
- 2) Размер блока данных в байтах, Кб или Мб. Если размерность не указана, то в байтах, если указана, то соответственно в Кбайтах или Мбайтах.
- 3) Число испытаний, т.е. число раз повторений измерений.

Пример вызова программы: `./memory_test -m RAM -b 1024|1Kb -l 10` или
`./memory_bandwidth --memory-type RAM|HDD|SSD|flash`
`--block-size 1024|1Kb`
`--launch-count 10`

В качестве блока данных использовать одномерный массив, в котором произведение числа элементов на их размерность равна требуемому размеру блока данных. Массив инициализировать случайными значениями. Для тестирования HDD/SSD и flash создать в программе файлы в соответствующих директориях.

Измерение времени реализовать с помощью функции `clock_gettime()` или аналогичной с точность до наносекунд. Измерять время исключительно на запись элемента в память или считывание из неё, без операций генерации или преобразования данных.

На выходе программы в одну строку CSV файла со следующей структурой:

[MemoryType;BlockSize;ElementType;BufferSize;LaunchNum;Timer;WriteTime;AverageWriteTime;WriteBandwidth;AbsError(write);RelError(write);ReadTime;AverageReadTime;ReadBandwidthAbsError(read);RelError(read)];, где

MemoryType – тип памяти (RAM|HDD|SSD|flash) или модель устройства, на котором проводятся испытания;
BlockSize – размер блока данных для записи и чтения на каждом испытании;
ElementType – тип элементов используемых для заполнения массива данных;
BufferSize – размер буфера, т.е. порции данных для выполнения одной операции записи или чтения;
LaunchNum – порядковый номер испытания;
Timer – название функции обращения к таймеру (для измерения времени);
WriteTime – время выполнения отдельного испытания с номером LaunchNum [секунды];
AverageWriteTime – среднее время записи из LaunchNum испытаний [секунды];
WriteBandwidth – пропускная способность памяти $(BLOCK_SIZE/AverageWriteTime) * 10^6$ [Mb/s]
AbsError(write) – абсолютная погрешность измерения времени записи или СКО [секунды];
RelError(write) – относительная погрешность измерения времени [%];
ReadTime – время выполнения отдельного испытания LaunchNum [секунды];
AverageReadTime – среднее время записи из LaunchNum испытаний [секунды];
ReadBandwidth – пропускная способность памяти $(BLOCK_SIZE/AverageReadTime) * 10^6$ [Мб/сек.];
AbsError(read) – абсолютная погрешность измерения времени чтения или СКО [секунды];
RelError(read) – относительная погрешность измерения времени [%].

2. Написать программу(функцию) на языке C/C++/C# или скрипт (benchmark) реализующий серию испытаний программы(функции) из п.1. Оценить пропускную способность оперативной памяти при работе с блоками данных равными объёму кэш-линии, кэш-памяти L1, L2 и L3 уровня и превышающего его. Для HDD|SSD и flash провести серию из 20 испытаний с блоками данных начиная с 4 Мб с шагом 4Мб. Результаты всех испытаний сохранить в один CSV файл со структурой, описанной в п.1.

* Для HDD|SSD и flash оценить влияние размера буфера (BufferSize) на пропускную способность памяти.

3. На основе CSV файла построить сводные таблицы и диаграммы отражающие:
 - 1) Зависимость пропускной способности записи и чтения от размера блока данных (BlockSize) для разного типа памяти;
 - 2) Зависимость погрешности измерения пропускной способности от размера блока данных для разного типа памяти;
 - 3) Зависимость погрешности измерений от числа испытаний LaunchNum;
 - 4) * Зависимость пропускной способности памяти от размера буфера для HDD|SSD и flash памяти;
4. ** Оценить пропускную способность файла подкачки (windows) или раздела SWAP (linux). Сравнить с пропускной способностью RAM, HDD/SSD и flash.

Литература.

1. *Yandex.ru*
2. *Google.com*