

Пояснительная записка к тестовому заданию

Постановка задачи

В качестве тестового задания было предложено реализовать программу, имитирующую работу жесткого диска, и оценить среднее время, затрачиваемое на операцию чтения/-записи.

При моделировании были сделаны допущения:

- носитель имеет один диск с одной рабочей поверхностью;
- скорость перемещения головки между соседними дорожками постоянна для любой пары дорожек;
- количество секторов на дорожке не зависит от того, является ли дорожка внешней или внутренней, и является постоянной величиной;
- контроллер жесткого диска имеет ограниченный буфер, работа с которым должна производиться по одному из следующих алгоритмов:
 - FCFS;
 - SSF;
 - Elevator algorithm;
- запросы на чтение/запись являются случайными событиями, образующими стационарный поток Пуассона;
- передача запросов на чтение/запись из интерфейса в буфер, из буфера на выполнение и возврат результата осуществляются мгновенно.

FCFS (First Come, First Served) – модель обработки, при которой запросы выполняются по одному в порядке их поступления в очередь.

SSF (Shortest Seek First) – модель, при которой обрабатывается запрос к ближайшему цилиндру (относительно текущего положения).

Elevator algorithm – модель, при которой фиксируется направление «движения» (к внутренней дорожке или к внешней) и сканирование очереди запросов выполняется с учетом этого направления. Будем иметь в виду, что нумерация дорожек производится начиная с внешней, тогда если зафиксировано направление к внутренней дорожке, то следующим обрабатываемым запросом станет тот, который обращен к ближайшему цилиндру (относительно текущего положения), номер которого больше номера текущего цилиндра. Если такие запросы отсутствуют, то направление меняется на противоположное.

Исходные данные и реализация модели

В качестве источника исходных данных было выбрано описание продуктовой линейки жестких дисков Seagate Barracuda [1]. В модели были приняты следующие константы [1, сс. 11-14]:

- число оборотов диска в минуту – 7200;
- количество цилиндров – 16383;
- размер сектора – 512 байт.

Существует оценка параметра «Track-to-track» [1, сс. 16]: операция чтения - 1 мс, операция записи - 1.2 мс. Этот параметр представляет собой среднее время всех возможных переходов между любыми двумя цилиндрами диска для чтения или записи, соответственно. Будем предполагать, что число операций чтения и записи равно, тогда справедливо положить

$$\nu = 2 \frac{tract_to_track}{N_c}, \quad \text{здесь } tract_to_track = \frac{1 + 1.2}{2} - \text{средняя скорость чтения/записи,}$$

N_c – количество цилиндров, а ν – постоянная скорость перемещения головки между двумя соседними дорожками.

Исходя из допущения, что работа чтение/запись производится только с одной стороны одного диска, среднее количество секторов на дорожке может быть определено по формуле

$$N_s = \frac{S_d}{N_c S_s},$$

здесь, как и выше, N_c – количество цилиндров, а S_d и S_s – размеры диска и сектора в байтах соответственно. Таким образом, если на вход программы подается число, определяющее объем диска в гигабайтах, следует иметь в виду, что 1 Гбайт = 10^9 байт [1, сс. 2], тогда результирующая формула для определения среднего количества секторов может быть записана в виде

$$N_s = \frac{V}{N_c S_s} 10^9, \quad V - \text{объем диска в ГБ.}$$

Эта формула справедлива для всякой дорожки диска в силу допущения о постоянстве количества секторов на всех дорожках.

Вероятность возникновения за некоторое время k событий в стационарном процессе Пуассона определяется формулой [2]

$$P\{\xi = k\} = \frac{a^k}{k!} e^{-a}, \quad k = 0, 1, 2, \dots; \quad a = \lambda\tau, \quad (1)$$

здесь τ – рассматриваемый интервал времени, а λ – интенсивность потока. Вероятность P возникновения хотя бы одного события в рассматриваемом интервале τ :

$$P = 1 - P\{\xi = 0\} = 1 - e^{-\lambda\tau},$$

отсюда согласно (1) следующее событие наступит через

$$\tau = -\frac{1}{\lambda} \ln(r),$$

здесь r - случайная величина с равномерным распределением в интервале $(0; 1)$.

В реализации программы для определения временного интервала τ между событиями и номеров дорожек и секторов запросов на чтение/запись использовались функции генерирования псевдослучайных чисел. При этом, если в случае с определением временного промежутка оказалось достаточно стандартной функции «rand()», то для номеров секторов, а в общем случае и для дорожек (может быть в будущем), отрезка $[0; 32767]$ не достаточно. Поэтому для последних был использован стандартный класс «std::default_random_engine», который генерирует числа в отрезке $[1; 2147483646]$.

Каждый предложенный алгоритм работы с буфером контроллера порождает свой класс, имитирующий работу жесткого диска и позволяющий оценить среднее время операции чтения/записи. С точки зрения проектирования эту задачу можно разрешить применением наследования или реализацией класса-шаблона с последующей специализацией функций работы с буфером контроллера.

Был выбран промежуточный вариант. Каждый производный класс реализует собственный алгоритм работы с буфером контроллера с помощью выделенного для этого виртуального метода «get_io_task_from_que()». Для остальных методов, так или иначе обращающихся к очереди контроллера, реализованы шаблонные функции, специализации которых используются в каждом классе самостоятельно. UML диаграмма представлена на Рис. 1.

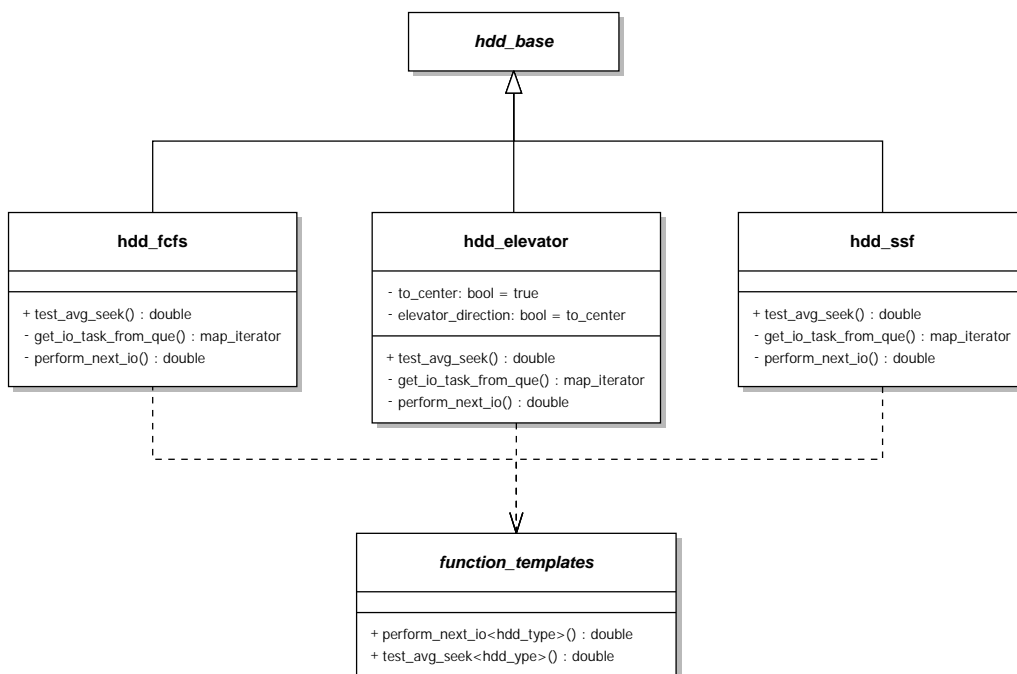


Рис. 1: UML диаграмма классов

Параграф

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Barracuda: Product manual. Seagate Technology LLC, PN: 100686584, Oct. 2012.
2. Королук В.С. Справочник по теории вероятностей и математической статистике. М.: Наука, 1985.