

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Отчет

Лабораторная работа №6

Выполнил студент гр. 150502:

Бейнар А.В.

Проверил:

Поденок Л.П.

Минск 2023

Тема работы:

Работа с файлами, отображенными в память.

Цель работы:

Кооперация потоков для высокопроизводительной обработки больших файлов. Изучаемые системные вызовы: `pthread_create()`, `pthread_exit()`, `pthread_join()`, `pthread_yield()`, `pthread_cancel()`, `pthread_barrier_init()`, `pthread_barrier_destroy()`, `pthread_barrier_wait()`, `mmap()`, `munmap()`.

Задание:

Написать многопоточную программу `sort_index` для сортировки вторичного индексного файла таблицы базы данных, работающую с файлом в двух режимах: `read()`/`write()` и с использованием отображение файлов в адресное пространство процесса. Программа должна запускаться следующим образом:

```
sort_index memsize granul threads filename
```

Параметры командной строки:

`memsize` := размер рабочего буфера, кратный размеру страницы (`getpagesize()`)

`blocks` := порядок разбиения буфера

`threads` := количество потоков (от `k` до `N`)

`k` := количество ядер

`N` := максимальное количество потоков (8k??)

`filename` := имя файла

Количество блоков должно быть степенью двойки и превышать количество потоков.

Для целей тестирования написать программу генерации неотсортированного индексного файла.

Алгоритм программы генерации

Генерируемый файл представляет собой вторичный индекс по времени и состоит из заголовка и индексных записей фиксированной длины.

Индексная запись имеет следующую структуру:

```
struct index_s {
    double
    time_mark; // временная метка (модифицированная юлианская
дата)
    uint64_t recno; // первичный индекс в таблице БД
} index_record;
Заголовок представляет собой следующую структуру
struct index_hdr_s {
    uint64_t
    records; // количество записей
    struct index_s idx[]; // массив записей в количестве
records
}
```

Временная метка определяется в модифицированный юлианских днях. Целая часть лежит в пределах от 15020.0 (1900.01.01-0:0:0.0) до «вчера»1. Дробная – это часть дня (0.5 – 12:0:0.0). Для генерации целой и дробной

частей временной метки используется системный генератор случайных чисел (random(3)).

Первичный индекс, как вариант, может заполняться последовательно, начиная с 1, но может быть случайным целым > 0 (в программе сортировки не используется). Размер индекса в записях должен быть кратен 256 и кратно превышать планируемую выделенную память для отображения. Размер индекса и имя файла указывается при запуске программы генерации.

Алгоритм программы сортировки

1) Основной поток запускает threads потоков, сообщая им адрес буфера, размер блока memsize/blocks, и их номер от 1 до threads - 1, используя возможность передачи аргумента для start_routine. Порожденные потоки останавливаются на барьере, ожидая прихода основного.

2) Основной поток с номером 0 открывает файл, отображает его часть размером memsize на память и синхронизируется на барьере. Барьер «открывается» и все threads потоков входят на равных в фазу сортировки.

3) Фаза сортировки

С каждым из блоков связана карта (массив) отсортированных блоков, в которой изначально блоки с 0 по threads-1 отмечены, как занятые.

Поток n начинает с того, что выбирает из массива блок со своим номером и его сортирует, используя qsort(3). После того, как поток отсортировал свой первый блок, он на основе конкурентного захвата мьютекса, связанного с картой, получает к ней эксклюзивный доступ, отмечает следующий свободный блок, как занятый, освобождает мьютекс и приступает к его сортировке.

Если свободных блоков нет, синхронизируется на барьере. После прохождения барьера все блоки будут отсортированы.

4) Фаза слияния

Поскольку блоков степень двойки, слияния производятся парами в цикле.

Поток 0 сливает блоки 0 и 1, поток 1 – блоки 2 и 3, и так далее.

Для отметки слитых пар и не слитых используется половина карты. Если для потока нет пары слияния, он синхронизируется на барьере.

В результате слияния количество блоков, подлежащих слиянию сокращается в два раза, а размер их в два раза увеличивается.

После очередного прохождения барьера количество блоков, подлежащих слиянию, станет меньше количества потоков. В этом случае распределение блоков между потоками осуществляется на основе конкурентного захвата мьютекса, связанного с картой. Потоки, которым не досталось блока, синхронизируются на барьере.

Когда осталась последняя пара, все потоки с номером не равным нулю синхронизируются на барьере, а поток с номером 0 выполняет слияние последней пары.

После слияния буфер становится отсортирован и подлежит сбросу в файл (munmap()).

Если не весь файл обработан, продолжаем с шага 2).

Если весь файл обработан, основной поток отправляет запрос отмены порожденным потокам, выполняет слияние отсортированных частей файла и завершается.

Вывод в консоли после компиляции

Вывод сгенерированного индексного файла:

```
Number of records: 4096
Value: 16396.320856, Index: 1
Value: 15140.346061, Index: 2
Value: 17472.987639, Index: 3
Value: 17477.602341, Index: 4
Value: 15358.199028, Index: 5
Value: 18391.202512, Index: 6
Value: 18103.920202, Index: 7
Value: 16906.933449, Index: 8
Value: 15599.259703, Index: 9
Value: 15964.055327, Index: 10
Value: 19208.167559, Index: 11
Value: 15107.188429, Index: 12
Value: 18923.353600, Index: 13
Value: 15800.366852, Index: 14
Value: 17574.083226, Index: 15
Value: 18954.415549, Index: 16
Value: 15656.332742, Index: 17
Value: 17761.907380, Index: 18
Value: 18670.172278, Index: 19
Value: 18202.009588, Index: 20
Value: 17716.708970, Index: 21
Value: 19404.809938, Index: 22
...
```

Вывод отсортированного индексного файла:

```
Value: 15022.086914, Index: 975
Value: 15022.297642, Index: 2709
Value: 15025.997790, Index: 925
Value: 15026.267307, Index: 1363
Value: 15030.968635, Index: 2630
Value: 15033.375723, Index: 1710
Value: 15033.783909, Index: 3772
Value: 15034.630040, Index: 337
Value: 15035.297959, Index: 2563
```

Value: 15035.631760, Index: 2126
Value: 15038.834225, Index: 59
Value: 15039.876298, Index: 2166
...
Value: 19492.224624, Index: 146
Value: 19492.851765, Index: 2880
Value: 19493.726437, Index: 1512
Value: 19495.559466, Index: 3797
Value: 19496.226133, Index: 290
Value: 19498.026689, Index: 782
Value: 19499.272655, Index: 3813
Value: 19499.367772, Index: 1025
Value: 19500.003928, Index: 1720
Value: 19500.331080, Index: 414
Value: 19501.438115, Index: 720
Number of records: 4096