

---

# **Разработка многопроцессных и многопоточных приложений**

---

## **Лабораторная работа**

Ревизия: 0.1



## **История изменений**

16.02.2010 – Версия 0.1. Первичный документ. Влад Ковтун

## Содержание

История изменений	2
Содержание	3
Лабораторная работа 2. Разработка многопроцессных и многопоточных приложений	4
Вопросы	4
Постановка задачи	4
Цель	4
Задачи	4
Вход	4
Пример	5
Вывод	5
Требования	5
Методические указания для самостоятельной работы	5
Алгоритм	6
Тестирование	7
Теоретические сведения	7
Литература	7
Приложение	7

# Лабораторная работа 2. Разработка многопроцессных и многопоточных приложений

## Вопросы

1. Постановка задачи.
2. Методические рекомендации.

## Постановка задачи

### Цель

Разработать консольное приложение, которое анализирует логи web-сервера Apache с использованием параллельных вычислений.

### Задачи

1. Ознакомиться с понятиями процессов и потоков в ОС.
2. Ознакомиться с понятиями межпроцессного и межпоточного взаимодействия.
3. Разработать консольное приложение для параллельного анализа нескольких файлов логов web-сервера Apache.
  - 3.1. На основе файлов логов web-сервера, следует сформировать следующую статистику:
    - количество уникальных посетителей за каждый день.
    - наиболее популярные браузеры.
    - наиболее популярные ОС.
    - наиболее популярные страницы (ссылки).
    - наиболее активные пользователи (IP).
  4. Разработать тестовое приложение, которое формирует все возможные тестовые комбинации ключей командной строки для приложения.
    - 5.1. Оценить производительность приложения, при различном количестве потоков чтения и потоков обработки. Количество потоков следует изменять от 1 до 8.
    - 5.2. Оценить производительность приложения, при различном количестве процессов чтения и анализа. Количество процессов следует изменять от 1 до 8.
    - 5.3. Оценить производительность приложения, при различном количестве процессов чтения и потоков анализа. Количество потоков и процессов следует изменять от 1 до 8.
  6. Оформить функциональную спецификацию на приложение. В функциональной спецификации обязательно указать, каким образом производится тестирование на корректность.
  7. Оформить отчет к лабораторной работе.

### Вход

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

- /id:<filedir> – полный путь к директории, в которой хранятся файлы логов для анализа. Данный параметр является обязательным, если он не указан, то происходит вывод на экран соответствующего сообщения и подсказки по использованию данного приложения.
- /if:<filenames> - перечисление имен файлов логов через пробел, которые необходимо проанализировать. Файлы должны находиться в директории указанной с помощью ключа /id. Данный ключ позволяет осуществлять выборочный анализ файлов логов.
- /o:<filename> – полный путь к файлу, либо имя файла, который будет хранить отчет. Если данный параметр не указывается, то вывод производится на экран.

- /thr:<threads count> - количество потоков, которые будут производить чтение данных из файлов. По умолчанию используется 2 потока.
- /tha:<thread count> - количество потоков, которые осуществляют анализ прочитанные записи лога и формируют статистику.
- /? - вывод информации о допустимых ключах командной строки.

## Пример

Подход 1. Анализ 3 файлов логов web-сервера Apache, результат анализа выводится в файл report.dat.

```
C:/>analyser.exe /id:"c:/logs/" /if: access-www.1.log access-www.2.log access-www.3.log /o:report.dat
```

Подход 2. Анализ 3 файлов логов web-сервера Apache, результат анализа выводится в файл report.dat. Для чтения используются 3 процесса. Для анализа используются 2 процесса.

```
C:/>analyser.exe /id:"c:/logs/" /if: access-www.1.log access-www.2.log access-www.3.log /pr:3 /pa:2 /o:report.dat
```

Подход 3. Файлы читаются отдельными процессам reader.exe, которые конфигурируются индивидуально. Информация о каналах и их количестве, по которым будет передавать информация, находится в файле mmm.cfg. Результат анализа выводится в файл report.dat.

```
C:/>analyser.exe /id:"c:/logs/" /cfg:mmm.cfg /pa:2 /o:report.dat
```

Процесс-читатель, производит анализ 3 файлов логов web-сервера Apache. Информация о каналах и их количестве, по которым будет передавать информация, находится в файле mmm.cfg.

```
C:/>reader.exe /id:"c:/logs/" /cfg:mmm.cfg
```

## Вывод

Во время работы приложения, рекомендуется выводить информацию о статусе приложения, а также о корректности его работы на консоль.

Другими словами, необходимо анализировать сколько процентов из каждого файла – проанализировано.

Допускается перенаправление вывода консоли в текстовый файл, например:

```
C:/>analyser.exe /id:"c:/logs/" /if: access-www.1.log access-www.2.log access-www.3.log /o:statistics.dat >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 текста.
- 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:

- Ознакомится с функциями для работы с процессами.

- Ознакомится с функциями для работы с потоками.
- Ознакомиться с функциями для работы с файлами.
- Ознакомиться с функциями для работы с каналами.
- Ознакомиться с функциями межпроцессного взаимодействия.
- Ознакомиться с функциями стандартной библиотеки.
- Ознакомиться с форматом файлов логов web-сервера Apache.

## Алгоритм

Данную задачу следует решить несколькими способами.

**Первый** подход, который реализует поставленную задачу, описан ниже.

### Основной поток

1. Разбор ключей командной строки.
2. Подготовка защищенной очереди данных для хранения результатов предварительного разбора записей лога.
3. Подготовка защищенной структуры данных для хранения результатов анализа.
4. Запуск потоков чтения файлов-логов и заполнения защищенной очереди.
5. Запуск потоков анализа данных, которые хранятся в защищенной очереди.
6. Вывод результатов анализа, которые находятся в защищенной структуре.

### Поток чтения

Изначально, потоку передается список файлов, которые ему необходимо проанализировать. После окончания анализа первого файла из списка, поток переходит к обработке следующего, и так до тех пор, пока не будут обработаны все файлы из списка. Такой подход планирования нагрузки не является оптимальным, однако он существенно упрощает задачу лабораторной работы.

1. Открытие файла-лога для чтения. Чтение осуществляется в НЕ ЭКСКЛЮЗИВНОМ режиме. Проверка корректности открытия.
2. Построчное считывание записей.
  - 2.1. Разбор строки.
  - 2.2. Заполнение структуры.
  - 2.3. Занесение структуры в защищенное хранилище.
3. Закрытие файла.
4. Завершение потока.

### Поток анализа

1. Эксклюзивно захватывает доступ к защищенной очереди.
2. Проверяет на наличие записей в очереди.
3. Если существует необработанная запись.
  - 3.1. Анализирует запись и заносит в защищенную структуру результатов.
  - 3.2. Удаляет запись из очереди.
4. Если очередь пуста – поток завершается.

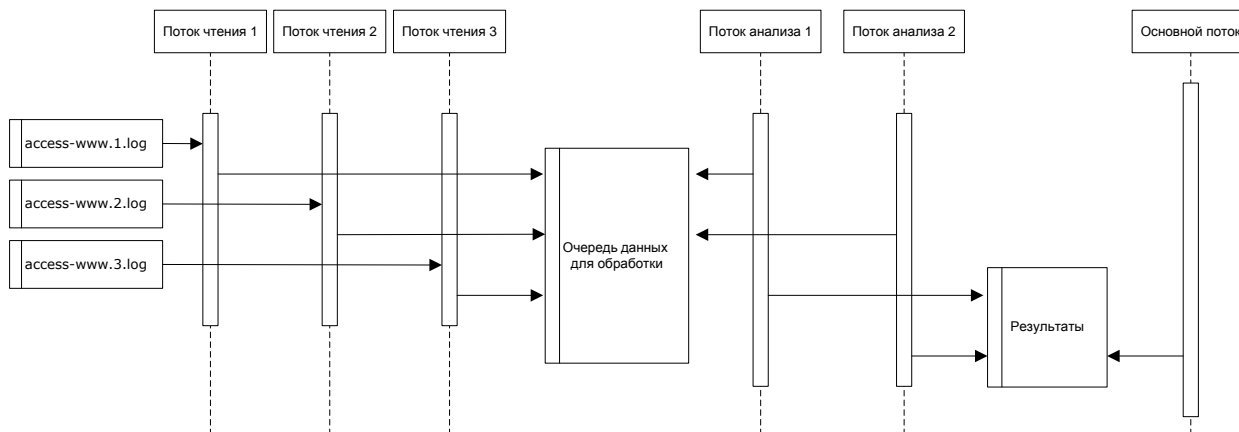


Рис. 1. Диаграмма последовательностей

**Второй** подход аналогичен первому, отличие состоит лишь в использовании процессов, вместо потоков. Причем обмен информацией между процессами следует организовать, используя средства IPC.

Рекомендуется использовать каналы (pipes), такой подход позволит разнести процессы анализа на различные компьютеры одной сети.

**Третий** подход, также аналогичен первому, отличие состоит лишь в том, что вместо потоков чтения используются процессы чтения, которые осуществляют взаимодействие с процессом анализа (владельцем нескольких потоков анализа), который формирует конечный результат.

## 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](http://www.nrjetix.com/r-and-d/lectures)
3. Описание формата лога web-сервера Apache доступно по адресу: <http://www.oglib.ru/apman/logs.html>

## Приложение

Пример лога web-сервера Apache приведен ниже:

```
82.193.97.226 - - [15/Feb/2010:17:40:20 +0200] "GET /images/bg_foot_line.gif HTTP/1.1"
200 2281 "http://natalia-ley.ya-zvezda.com/css/main.css?v103" "Mozilla/5.0 (Windows; U;
Windows NT 5.1; ru; rv:1.9.1.7) Gecko/20091221 MRA 5.5 (build 02842) Firefox/3.5.7
(.NET CLR 3.5.30729)"
```

```
82.193.97.226 - - [15/Feb/2010:17:40:20 +0200] "GET /images/bg_footer.jpg HTTP/1.1"
200 3430 "http://natalia-ley.ya-zvezda.com/css/main.css?v103" "Mozilla/5.0 (Windows; U;
Windows NT 5.1; ru; rv:1.9.1.7) Gecko/20091221 MRA 5.5 (build 02842) Firefox/3.5.7
(.NET CLR 3.5.30729)"
```

```
82.193.97.226 - - [15/Feb/2010:17:40:20 +0200] "GET /images/icons/icon_bug.gif
HTTP/1.1" 200 1149 "http://natalia-ley.ya-zvezda.com/css/main.css?v103" "Mozilla/5.0
(Windows; U; Windows NT 5.1; ru; rv:1.9.1.7) Gecko/20091221 MRA 5.5 (build 02842)
Firefox/3.5.7 (.NET CLR 3.5.30729)"
```

82.193.97.226 - - [15/Feb/2010:17:40:20 +0200] "GET /images/logo\_footer.gif HTTP/1.1" 200 4354 "http://natalia-ley.ya-zvezda.com/css/main.css?v103" "Mozilla/5.0 (Windows; U; Windows NT 5.1; ru; rv:1.9.1.7) Gecko/20091221 MRA 5.5 (build 02842) Firefox/3.5.7 (.NET CLR 3.5.30729)"

82.193.97.226 - - [15/Feb/2010:17:40:20 +0200] "GET /favicon.ico HTTP/1.1" 200 990 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; ru; rv:1.9.1.7) Gecko/20091221 MRA 5.5 (build 02842) Firefox/3.5.7 (.NET CLR 3.5.30729)"

109.184.36.140 - - [15/Feb/2010:17:40:20 +0200] "GET /index/player\_code/?v=688 HTTP/1.1" 302 26 "http://www.clubteatro.ru/" "Mozilla/5.0 (Windows; U; Windows NT 5.1; ru; rv:1.9.1.7) Gecko/20091221 MRA 5.4 (build 02652) Firefox/3.5.7"