ЛАБОРАТОРНАЯ РАБОТА №1

Курс «Параллельные методы и алгоритмы»

Тема: Разработка многопоточных приложений.

Цель: Научиться писать и отлаживать приложения с продвинутыми элементами многопоточности и асинхронности на любом из высокоуровневых языков программирования и с использованием библиотеки OpenMP.

Темы для предварительной проработки [УСТНО]**:**

- Конкурентность и параллелизм.
- Потоки и пул потоков.
- Асинхронность.
- Синхронизация потоков.
- Библиотека OpenMP.

Индивидуальные задания [КОД] **:**

1. Написать консольное приложение одновременной загрузки данных с набора URL-адресов (список из не менее 10 адресов должен считываться из текстового файла). При старте и окончании загрузки с каждого адреса приложение должно вывести соответствующее сообщение на новой строке в консоли:

```
Start downloading: https://www.rabbitmq.com/tutorials/tutorial-two-dotnet.html
Start downloading: https://dotnet.microsoft.com/apps/aspnet/web-apps
Start downloading: http://tech.jonathangardner.net/wiki/Why_Java_Sucks
Start downloading: https://github.com/ar1st0crat
Start downloading: https://highlyscalable.wordpress.com/2012/02/01/mapreduce-patterns/
Start downloading: https://developer.nvidia.com/blog/even-easier-introduction-cuda/
Start downloading: https://github.com/Yelp/mrjob
Start downloading: https://mpi4py.readthedocs.io/en/stable/overview.html
Download complete: https://www.rabbitmq.com/tutorials/tutorial-two-dotnet.html
Download complete: https://dotnet.microsoft.com/apps/aspnet/web-apps
Download complete: https://mpi4py.readthedocs.io/en/stable/overview.html
Download complete: https://developer.nvidia.com/blog/even-easier-introduction-cuda/
Download complete: https://github.com/Yelp/mrjob
Download complete: http://tech.jonathangardner.net/wiki/Why_Java_Sucks
Download complete: https://highlyscalable.wordpress.com/2012/02/01/mapreduce-patterns/
Download complete: https://github.com/ar1st0crat
Downloading images...
Done
Adding images to archive...
Done
```

Загрузка с каждого адреса должна производиться в отдельном потоке. После полной загрузки данных со всех адресов приложение должно запустить отдельный поток, который произведет парсинг всех результатов и для всех гиперссылок с адресами изображений запустит потоки загрузки соответствующих изображений с последующим добавлением их в

архив images.zip и сохранением его в указанной директории. Создать также приложение с асинхронной версией реализации вышеуказанной функциональности.
2. Написать консольное приложение одновременной загрузки данных с набора URL-адресов (список из не менее 10 адресов должен считываться из текстового файла). При старте и окончании загрузки с каждого адреса приложение должно вывести соответствующее сообщение на новой строке в консоли. Реализовать паттерн "Производитель/потребитель" (Producer/Consumer). N потоков занимаются загрузкой данных с URL-адресов и парсингом результатов, M потоков занимаются скачиванием изображений по всем найденным гиперссылкам и сохранением их в указанной директории.
3. Реализовать многопоточный вариант алгоритма быстрой сортировки.
4. Реализовать многопоточный вариант алгоритма быстрой сортировки с помощью OpenMP.
5. Реализовать многопоточный вариант алгоритма сортировки слиянием.
6. Реализовать многопоточный вариант алгоритма сортировки слиянием с помощью OpenMP.
7. Написать многопоточное приложение для расчета числа π по формуле ряда: 1 - 1/3 + 1/5 - 1/7 + 1/9 - ...
8. Написать многопоточное приложение для расчета числа π по методу Монте-Карло с помощью OpenMP.
9. Написать многопоточное приложение, демонстрирующее проблему "обедающих философов" без взаимной блокировки.
10. Написать многопоточное приложение, демонстрирующее проблему "обедающих философов" со взаимной блокировкой.
11. Написать многопоточное приложение, демонстрирующее проблему "спящего парикмахера".
12. Написать многопоточное приложение, реализующее рекурсивное копирование содержимого директории, передаваемой в виде параметра командной строки. Использовать пул N потоков (число N задается вторым параметром командной строки).

Примечание. Привести результаты замеров времени выполнения openMP-версий приложений и значение ускорения относительно однопоточной реализации.

Контрольные вопросы [ОТЧЕТ]**:**

1. В чем состоит отличие конкурентности от параллелизма?
2. Что такое пул потоков? Как он используется в TPL .NET?
3. Как связаны между собой futures, promises и tasks?
4. Что такое «состояние гонок»? Приведите примеры.
5. Что означает starvation (голодание) в многопоточных приложениях?
6. Что такое deadlock (взаимная блокировка)? Приведите примеры.
7. Опишите основные примитивы синхронизации потоков.
8. Опишите классические задачи синхронизации потоков и способы их решения.
9. Назначение библиотеки OpenMP. Особенности компиляции приложений.
10. Опишите основные прагмы OpenMP.
11. Укажите особенности отладки приложений с элементами многопоточности и асинхронности.

Рекомендуемые источники:

[1] Гергель В.П. Теория и практика параллельных вычислений: учеб. пособие / В.П. Гергель. – М.: Интернет-Ун-т информ. технологий: БИНОМ. Лаб. знаний, 2007. – 423 с.

[2] Уильямс Э. Параллельное программирование на C++ в действии. Практика разработки многопоточных программ / Э.Уильямс. – М.: ДМК Пресс, 2012. – 672с.

[3] Herlihy M. The Art of Multiprocessor Programming / M. Herlihy, N. Shavit. – Morgan Kaufmann, 2012. – 536 p.

[4] Joe Albahari. Threading in C#. URL: http://www.albahari.com/threading.