

**Федеральное государственное автономное образовательное учреждение  
высшего образования «Самарский национальный исследовательский университет  
имени академика С.П. Королева»**

## **МОДЕЛИРОВАНИЕ И АНАЛИЗ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ**

Задания на лабораторные работы

Самара 2020

## Содержание

1. Лабораторная работа №1 - Синхронизация параллельных процессов. Производитель-потребитель.
2. Лабораторная работа №2 - Синхронизация параллельных процессов. Сравнение файлов.
3. Лабораторная работа №3 - Модели взаимодействия параллельных процессов. Нахождение простых чисел.
4. Лабораторная работа №4 - Модели взаимодействия параллельных процессов. Гравитационная задача N тел.

## **Лабораторная работа №1**

### **Синхронизация параллельных процессов. Производитель-потребитель.**

**Цель работы:** Изучение влияния синхронизации на производительность параллельных программ

#### ***1 Задание на лабораторную работу***

1. Разработать параллельную программу, выполняющую копирование данных из одного массива (А) в другой массив (В) с использованием промежуточного буфера. Программа должна состоять из двух потоков. Первый поток – производитель – содержит исходный массив А и копирует данные из него в буфер по одному элементу. Второй поток содержит массив В и производит чтение данных из буфера по одному элементу и запись их в массив В. Запись в буфер возможна при наличии в нем хотя бы одного свободного элемента. Чтение из буфера возможно при наличии в нем хотя бы одного несчитанного элемента. С буфером связаны два счетчика: первый счетчик содержит индекс первого доступного элемента для чтения (начало данных), второй счетчик содержит индекс первого доступного для записи элемента. Если счетчики равны, буфер пуст. Буфер циклический, то есть второй счетчик может быть меньше первого. Если второй счетчик меньше первого на 1, то буфер заполнен. Альтернативным подходом может служить учет только занятых элементов буфера. В этом случае первый счетчик содержит индекс первого занятого элемента, второй счетчик содержит индекс последнего занятого элемента.
2. Измерить время работы программы для различных значений длины буфера. Размер массивов должен таким, чтобы массивы А, В и буфер заполняли максимум свободной оперативной памяти. Минимальный размер массива А –  $10^8$  элементов. Размер буфера – от 10 элементов до половины длины массива А. Рекомендуется использовать не менее 10 различных значений размера буфера при проведении вычислительных экспериментов.
3. Объяснить наблюдаемые изменения длительности работы программы.
4. Сравнить время работы параллельной программы с временем работы ее последовательного аналога, осуществляющего последовательное копирование путем чередования фаз заполнения и освобождения буфера.
5. Объяснить наблюдаемые закономерности.
6. Составить отчет по результатам работы.

#### ***2 Содержание отчета***

1. Постановка задачи.
2. Словесное описание работы параллельного алгоритма.
3. Исходный текст различных вариантов программы.

4. Результаты вычислительных экспериментов *с объяснением полученных зависимостей*. Результаты должны содержать таблицы с длительностью различных вариантов программы и ускорением. Графики достаточно построить только для ускорения.
5. Выводы по работе.

### **3 Контрольные вопросы**

1. В чем заключается задача критической секции?
2. Что такое мелко модульная неделимость, условие «не больше одного»?
3. Как реализуются критические секции с помощью активных блокировок, операций test and set?

### **СПИСОК ЛИТЕРАТУРЫ**

1. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления - СПб.: БХВ-Петербург, 2002. - 608 с.
2. Грегори Р. Эндрюс Основы многопоточного, параллельного и распределенного программирования - Вильямс, 2002. - 512 с.

## Лабораторная работа №2

### Синхронизация параллельных процессов. Сравнение файлов.

**Цель работы:** Изучение влияния синхронизации на производительность параллельных программ

#### *1 Задание на лабораторную работу*

1. Разработать параллельную программу, выполняющую сравнение двух файлов (по аналогии с утилитой Linux diff). Программа должна построчно сравнивать два текстовых файла и выводить на стандартный вывод номера и значения отличающихся строк. Если один из файлов длиннее другого, то программа должна вывести каждую дополнительную строку более длинного файла. Программа должна состоять из трех потоков: по одному потоку на построчное чтение каждого файла и один поток на сравнение двух считанных строк. Строки из разных файлов могут читаться одновременно, но сравнение происходит, когда считаны обе строки с одинаковыми номерами. Пока процесс, сравнивающий строки, не выполнит сравнение, процессы-читатели ждут и не читают очередные строки.
2. Измерить время работы программы для различных файлов. Файлы должны иметь различную длину строк и количество строк. Длина строк изменяется от нескольких символов до нескольких тысяч символов. Количество строк изменяется от нескольких десятков до нескольких десятков тысяч. Рекомендуется использовать не менее 10 различных значений для длин строк и количества строк в файлах при проведении экспериментов.
3. Объяснить наблюдаемые изменения времени работы программы.
4. Сравнить длительность работы параллельной программы с длительностью работы ее последовательного аналога, осуществляющего последовательное чтение по одной строке из каждого файла и их сравнение.
5. Объяснить наблюдаемые закономерности.
6. Составить отчет по результатам работы.

#### *2 Содержание отчета*

1. Постановка задачи.
2. Словесное описание параллельного алгоритма.
3. Исходный текст различных вариантов программы.
4. Результаты вычислительных экспериментов **с объяснением полученных зависимостей**. Результаты должны содержать таблицы с длительностью различных вариантов программы и ускорением. Графики достаточно построить только для ускорения.
5. Выводы по работе.

### **3 Контрольные вопросы**

1. Что такое состояние программы, свойства безопасности и живучести?
2. Опишите алгоритм реализации критических секций: алгоритм разрыва узла.
3. Опишите алгоритмы реализации критических секций: алгоритм билета, алгоритм поликлиники.

### **СПИСОК ЛИТЕРАТУРЫ**

1. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления - СПб.: БХВ-Петербург, 2002. - 608 с.
2. Грегори Р. Эндрюс Основы многопоточного, параллельного и распределенного программирования - Вильямс, 2002. - 512 с.

**Лабораторная работа №3**  
**Модели взаимодействия параллельных процессов.**  
**Нахождение простых чисел.**

**Цель работы:** Изучение различных парадигм параллельного программирования и моделей взаимодействия параллельных процессов

**1 Задание на лабораторную работу**

1. С использованием одной из моделей взаимодействия параллельных процессов разработать параллельную программу нахождения всех простых чисел в диапазоне от 1 до  $N$ . Целое число  $x > 1$  называется простым, если оно делится без остатка только на  $x$  и на 1. Любое целое число  $y > 1$  можно представить в виде произведения простых чисел.
2. Программа должна работать с любым количеством потоков, задаваемым пользователем. Программа должна реализовывать следующий алгоритм:
  - Диапазон  $[1, N]$  разбивается на  $P$  частей, где  $P$  — количество потоков или процессов параллельной программы.
  - Каждый поток находит простые числа в своем диапазоне, последовательно проверяя все числа внутри этого диапазона.
  - Все потоки используют общий буфер (одномерный массив) для хранения найденных простых чисел. Каждый поток при нахождении очередного простого числа добавляет его в буфер.
  - Определение того, является очередное проверяемое число простым или нет, выполняется его делением на уже найденные простые числа, находящиеся в общем буфере. При этом возникает задача синхронизации: при анализе очередного числа  $y$  поток должен убедиться, что буфер содержит все числа, меньшие  $y$ , которые могут быть его делителями. Поскольку некоторые из этих чисел добавляются в буфер другими потоками, текущий поток должен приостановиться и подождать, когда остальные потоки добавят все необходимые ему числа.
3. Измерить длительность работы программы для различного количества потоков. Число  $N$  выбрать таким, чтобы последовательная программа работала не менее минуты. Количество потоков взять от 1 до 64, используя до 10 различных значений.
4. Добиться того, чтобы параллельная программа имела ускорение больше 1.
5. Объяснить наблюдаемые изменения длительности работы программы при изменении количества потоков от 1 до  $P$ .
6. Составить отчет по результатам работы.

## **2 Содержание отчета**

1. Постановка задачи.
2. Описание используемой модели взаимодействия параллельных процессов, словесное описание параллельного алгоритма и исходный текст программы.
3. Результаты вычислительных экспериментов **с объяснением полученных зависимостей**. Результаты должны содержать таблицы с длительностью различных вариантов программы и ускорением. Графики достаточно построить только для ускорения.
4. Выводы по работе.

## **3 Контрольные вопросы**

1. Опишите модели (парадигмы) параллельного программирования: итеративный и рекурсивный параллелизм, производители и потребители.
2. Опишите модели (парадигмы) параллельного программирования: клиенты и серверы, взаимодействующие равные.
3. Опишите различные модели взаимодействия параллельных процессов: управляющий-рабочие, алгоритмы пульсации, конвейерные алгоритмы, алгоритмы типа «зонд-эхо».

## **СПИСОК ЛИТЕРАТУРЫ**

1. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления - СПб.: БХВ-Петербург, 2002. - 608 с.
2. Грегори Р. Эндрюс Основы многопоточного, параллельного и распределенного программирования - Вильямс, 2002. - 512 с.



**Лабораторная работа №4**  
**Модели взаимодействия параллельных процессов.**  
**Гравитационная задача N тел.**

**Цель работы:** Изучение различных парадигм параллельного программирования и моделей взаимодействия параллельных процессов

**1 Задание на лабораторную работу**

1. С использованием одной из моделей взаимодействия параллельных процессов разработать параллельную программу моделирования гравитационного взаимодействия  $N$  тел.

Каждое тело имеет массу, начальное положение и скорость. Гравитация вызывает перемещение и ускорение тел. Движение системы  $N$  тел имитируется пошагово с помощью дискретных отрезков времени. На каждом временном шаге вычисляются силы, действующие на каждое тело, и обновляются скорости и положения тел. Имитация гравитационного взаимодействия имеет следующую структуру:

```
// инициализировать тела
for ( t = t_start; t < t_finish; t = t + dt ) {
    // вычислить силы
    // переместить тела
}
```

Значение  $dt$  является временным шагом.

Величина силы гравитации между двумя телами  $i$  и  $j$  вычисляется по формуле:

$$F = G * m_i * m_j / r^2, \quad (1)$$

где  $m_i$  и  $m_j$  – массы тел, а  $r$  – расстояние между ними. Гравитация является чрезвычайно слабым взаимодействием, поэтому значение гравитационной постоянной  $G$  очень мало:  $6.67 * 10^{-11}$ .

Предположим, что все тела расположены на одной плоскости, т.е. задача является двумерной. В этом случае силы, ускорения, скорости и положения тел являются двумерными векторами, то есть их значения задаются парами чисел, каждое из которых определяет проекцию вектора на соответствующую координатную ось. Например, ускорение тела  $i$   $\mathbf{a}_i = [a_{ix}, a_{iy}]$ .

Направление силы, действующей на тело  $i$  со стороны тела  $j$ , задается единичным вектором, направленным от тела  $i$  в сторону тела  $j$ , а силы воздействия тела  $i$  на тело  $j$  – противоположным вектором. Величины сил, действующих между любыми двумя телами, равны, а направления противоположны. После вычисления модуля силы по формуле (1) необходимо вычислить величину проекций вектора силы на координатные оси, исходя из направления действия силы. Вычисленные значения будут компонентами вектора силы  $\mathbf{F} = [F_x, F_y]$ .

Общая сила, действующая на тело, равна векторной сумме сил воздействия со стороны всех остальных тел.

Гравитационные силы, действующие на тело, вызывают его ускорение и перемещение. Ускорение тела  $i$  равно отношению общей силы  $F$ , действующей на тело, к массе тела:

$$a = F/m_i$$

Если за малый интервал времени  $dt$  ускорение  $a_i$  тела  $i$  остается практически постоянным, то изменение скорости приблизительно равно:

$$dv_i = a_i * dt$$

Изменение положения тела есть интеграл его скорости и ускорения на интервале времени  $dt$ , который приблизительно равен:

$$dp_i = v_i * dt + (a_i/2) * dt^2 = (v_i + dv_i/2) * dt$$

Эта формула устроена по так называемой скачкообразной (leapfrog) схеме, в которой одна половина изменения в положении тела обусловлена предыдущей скоростью, а другая – новой.

При использовании приведенных выше формул следует различать векторные и скалярные величины. Компоненты соответствующих векторов следует вычислять отдельно или использовать операции над векторами.

2. Программа должна визуально изображать изменение взаимного положения тел в течение некоторого интервала времени (нескольких дискретных шагов времени) под действием гравитационных сил. Размерами тел можно пренебречь, представив их точками.
3. Измерить длительность работы программы для различного количества тел и различного количества потоков. Количество тел должно задаваться пользователем. Количество потоков взять от 1 до 64, используя до 10 различных значений. **Длительность работы последовательной программы должна быть не менее одной минуты.**
4. Объяснить наблюдаемые изменения длительности работы программы.
5. Составить отчет по результатам работы.

**ЗАМЕЧАНИЕ:** Использование приведенных выше формул вызывает резкое увеличение ускорения тел при их сближении за счет малого значения квадрата расстояния в знаменателе формулы вычисления силы. Для исключения этого эффекта возможно использование следующего подхода:

при сближении двух тел до расстояния, меньшего, чем некоторое заданное значение, считать силу взаимодействия между данными телами константой (например, нулем, то есть не рассматривать взаимодействие между данными телами).

Если тела имеют фиксированный размер, то при сближении они могут соударяться и отталкиваться друг от друга.

## **2 Содержание отчета**

1. Постановка задачи.
2. Описание используемой модели взаимодействия параллельных процессов, словесное описание параллельного алгоритма и исходный текст программы.
3. Результаты вычислительных экспериментов **с объяснением полученных зависимостей**. Результаты должны содержать таблицы с длительностью различных вариантов программы и ускорением. Графики достаточно построить только для ускорения.
4. Выводы по работе.

## **3 Контрольные вопросы**

1. Опишите модели (парадигмы) параллельного программирования: итеративный и рекурсивный параллелизм, производители и потребители.
2. Опишите модели (парадигмы) параллельного программирования: клиенты и серверы, взаимодействующие равные.
3. Опишите различные модели взаимодействия параллельных процессов: управляющий-рабочие, алгоритмы пульсации, конвейерные алгоритмы, алгоритмы типа «зонд-эхо».

## **СПИСОК ЛИТЕРАТУРЫ**

1. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления - СПб.: БХВ-Петербург, 2002. - 608 с.
2. Грегори Р. Эндрюс Основы многопоточного, параллельного и распределенного программирования - Вильямс, 2002. - 512 с.