

## ЗАДАНИЕ №3

### Практические приемы построения многопоточных приложений

**Цели и задачи:** Изучить работу с потоками. Научиться разбивать задачу на части, для последующего их выполнения различными потоками.

#### Общие сведения

Потоки предоставляют возможность проведения параллельных или псевдопараллельных (в случае одного ядра) вычислений. Потоки могут порождаться во время работы программы, процесса или другого потока. Основное отличие потоков от процессов заключается в том, что различные потоки имеют различные пути выполнения, но при этом пользуются общей памятью. Таким образом, несколько порожденных в программе потоков, могут пользоваться глобальными переменными, и любое изменение данных одним потоком, будет доступно и для всех остальных.

Существует несколько моделей построения многопоточных приложений. Среди них можно отметить следующие.

**Итеративный параллелизм** используется для реализации нескольких потоков (часто идентичных), каждый из которых содержит циклы. Потоки программы, описываются итеративными функциями и работают совместно над решением одной задачи.

**Рекурсивный параллелизм** используется в программах с одной или несколькими рекурсивными процедурами, вызов которых независим. Это технология «разделяй-и-властвуй» или «перебор-с-возвратами».

**Производители и потребители** – это парадигма взаимодействующих неравноправных потоков. Одни потоки «производят» данные, другие их «потребляют». Часто такие потоки организуются в **конвейер**, через который проходит информация. Каждый поток конвейера потребляет выход своего предшественника и производит входные данные для своего последователя. Другой распространенный способ организации потоков – древовидная структура или сети слияния, на этом основан, в частности, метод **дихотомии**.

**Клиенты и серверы** – еще один способ взаимодействия неравноправных потоков. Клиентский поток запрашивает сервер и ждет ответа. Серверный поток ожидает запроса от клиента, затем действует в соответствии с поступившим запросом.

**Управляющий и рабочие** – модель организации вычислений, при которой существует поток, координирующий работу всех остальных потоков. Как правило, управляющий поток распределяет данные, собирает и анализирует результаты.

**Взаимодействующие равные** – модель, в которой исключен не занимающийся непосредственными вычислениями управляющий поток. Распределение работ в таком приложении либо фиксировано заранее, либо динамически определяется во время выполнения. Одним из распространенных способов динамического распределения работ является «портфель задач». Портфель задач, как правило, реализуется с помощью разделяемой переменной, доступ к которой в один момент времени имеет только один процесс.

### Порядок выполнения задания

1. Выбрать модель приложения, наиболее точно отвечающую специфике задачи или применить указанную. Изучить используемую модель по дополнительным источникам информации.

2. Разработать алгоритм решения задания, с учетом разделения вычислений между несколькими потоками. Избегать ситуаций неуправляемого изменения одних и тех же общих данных несколькими потоками. Если же избежать этого невозможно, необходимо использовать мьютексы и критические секции.

3. Разработать программу с применением функций библиотеки **POSIX Threads** или **стандартной библиотеки C++** и протестировать ее на нескольких примерах.

### Дополнительные сведения

1. Отчет, программа, тестовые наборы, примеры выполнения выкладываются на Гит.

2. Отчет должен содержать титульный лист с идентификацией студента, номер варианта, описание задачи.

3. В отчете необходимо подробно описать используемую модель вычислений. Привести источники информации, в которых описана данная модель.

4. Для удобства ввод-вывод данных можно организовать с использованием файлов.

5. Допускается использование командной строки для задания имен файлов, размерностей и других скалярных аргументов. При использовании командной строки в отчете необходимо указать ее формат.

6. Для порождения массивов данных можно использовать генераторы случайных чисел.

## **Варианты заданий**

1. Вычислить векторное произведение квадратных матриц  $A$  и  $B$ . Входные данные: произвольные квадратные матрицы  $A$  и  $B$  одинаковой размерности. Размер матриц задается входным параметром. Количество потоков является входным параметром, при этом размерность матриц может быть не кратна количеству потоков.

2. Найти определитель матрицы  $A$ . Входные данные: целое положительное число  $n$ , произвольная матрица  $A$  размерности  $n \times n$ . Количество потоков является входным параметром, при этом размерность матриц может быть не кратна количеству потоков.

3. Найти алгебраическое дополнение для каждого элемента матрицы. целое положительное число  $n$ , произвольная матрица  $A$  размерности  $n \times n$ . Количество потоков является входным параметром, при этом размерность матриц может быть не кратна количеству потоков.

4. Найти обратную матрицу для матрицы  $A$ . Входные данные: целое положительное число  $n$ , произвольная матрица  $A$  размерности  $n \times n$ . Количество потоков является входным параметром, при этом размерность матриц может быть не кратна количеству потоков.

5. Определить ранг матрицы. Входные данные: целое положительное число  $n$ , произвольная матрица  $A$  размерности  $n \times n$ . Количество потоков является входным параметром, при этом размерность матриц может быть не кратна количеству потоков.

6. Вычислить прямое произведение множеств  $A_1, A_2, A_3, A_4$ . Входные данные: множества чисел  $A_1, A_2, A_3, A_4$ , мощности множеств могут быть не равны между собой и мощность каждого множества больше или равна 1. Количество потоков является входным параметром.

7. Вычислить прямое произведение множеств  $A_1, A_2, A_3 \dots A_n$ . Входные данные: целое положительное число  $n$ , множества чисел  $A_1, A_2, A_3 \dots A_n$ , мощности множеств равны между собой и мощность каждого множества больше или равна 1. Количество потоков является входным параметром.

8. Используя формулы Крамера, найти решение системы линейных уравнений.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 = b_3 \\ a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 = b_4 \end{cases}$$

Предусмотреть возможность деления на ноль. Входные данные: коэффициенты системы. Оптимальное количество потоков выбрать самостоятельно.

9. Определить, является ли множество  $C$  объединением множеств  $A$  и  $B$  ( $A \cup B$ ), пересечением множеств ( $A \cap B$ ), разностью множеств  $A$  и  $B$  ( $A \setminus B$ ), разностью множеств  $B$  и  $A$  ( $B \setminus A$ ). Входные данные: множества целых положительных чисел  $A, B, C$ . Оптимальное количество потоков выбрать самостоятельно.

10. Найти все возможные тройки компланарных векторов. Входные данные: множество не равных между собой векторов  $(x, y, z)$ , где  $x, y, z$  – числа. Оптимальное количество потоков выбрать самостоятельно.

11. Определить индексы  $i, j$  ( $i \neq j$ ), для которых выражение  $A[i] - A[i+1] + A[i+2] - A[i+3] + \dots \pm A[j]$  имеет максимальное значение. Входные данные: массив чисел  $A$ , произвольной длины большей 10. Количество потоков не является параметром задачи.

12. Определить индексы  $i, j$ , для которых существует наиболее длинная последовательность  $A[i] < A[i+1] < A[i+2] < A[i+3] < \dots < A[j]$ . Входные данные: массив чисел  $A$ , произвольной длины большей 1000. Количество потоков является входным параметром.

13. Определить множество индексов  $i$ , для которых  $(A[i] - B[i])$  или  $(A[i] + B[i])$  являются простыми числами. Входные данные: массивы целых положительных чисел  $A$  и  $B$ , произвольной длины  $\geq 1000$ . Количество потоков является входным параметром.

14. Определить множество индексов  $i$ , для которых  $A[i]$  и  $B[i]$  не имеют общих делителей (единицу в роли делителя не рассматривать). Входные данные: массивы целых положительных чисел  $A$  и  $B$ , произвольной длины  $\geq 1000$ . Количество потоков является входным параметром.

15. Вывести список всех целых чисел, содержащих от 4 до 9 значащих цифр, которые после умножения на  $n$ , будут содержать все те же самые цифры в произвольной последовательности и в произвольном количестве. Входные данные: целое положительное число  $n$ , больше единицы и меньше десяти. Количество потоков является входным параметром.

16. *Задача об инвентаризации по рядам.* После нового года в библиотеке университета обнаружилась пропажа каталога. После поиска и наказания виноватых, ректор дал указание восстановить каталог силами студентов. Фонд библиотека представляет собой прямоугольное помещение, в котором находится  $M$  рядов по  $N$  шкафов по  $K$  книг в каждом шкафу. Требуется создать многопоточное приложение, составляющее каталог. При решении задачи использовать метод «портфель задач», причем в качестве отдельной задачи задается составление каталога одним студентом для одного ряда.

17. *Задача об инвентаризации по книгам.* После нового года в библиотеке университета обнаружилась пропажа каталога. После поиска и наказания, виноватых ректор дал указание восстановить каталог силами студентов. Фонд библиотека представляет собой прямоугольное помещение, в котором находится  $M$  рядов по  $N$  шкафов по  $K$  книг в каждом шкафу. Требуется создать многопоточное приложение, составляющее каталог. При решении задачи использовать метод «портфель задач», причем в качестве отдельной задачи задается внесение в каталог записи об отдельной книге.

18. *Задача о наследстве.* У старого дона Энрике было два сына, у каждого из сыновей – еще по два сына, каждый из которых имел еще по два сына. Умирая, дон Энрике завещал все свое богатство правнукам в разных

долях. Адвокат дон Хосе выполнил задачу дележа наследства в меру своих способностей. Правнуки заподозрили адвоката в укрывательстве части наследства. Требуется создать многопоточное приложение, которое при известных сумме завещания дон Эрике и доле каждого наследника, проверяет честность адвоката. При решении использовать принцип дихотомии.

19. У одной очень привлекательной студентки есть  $N$  поклонников. Традиционно в день св. Валентина очень привлекательная студентка проводит романтический вечер с одним из поклонников. Счастливый избранник заранее не известен. С утра очень привлекательная студентка получает  $N$  «валентинок» с различными вариантами романтического вечера. Выбрав наиболее заманчивое предложение, студентка извещает счастливого о своем согласии, а остальных – об отказе. Требуется создать многопоточное приложение, моделирующее поведение студентки. При решении использовать парадигму «клиент-сервер» с активным ожиданием.

20. *Задача о производстве булавок.* В цехе по заточке булавок все необходимые операции осуществляются тремя рабочими. Первый из них берет булавку и проверяет ее на предмет кривизны. Если булавка не кривая, то рабочий передает ее своему напарнику. Напарник осуществляет собственно заточку и передает заточенную булавку третьему рабочему, который осуществляет контроль качества операции. Требуется создать многопоточное приложение, моделирующее работу цеха. При решении использовать парадигму «производитель-потребитель».

21. *Задача про экзамен.* Преподаватель проводит экзамен у группы студентов. Каждый студент заранее знает свой билет и готовит по нему ответ. Подготовив ответ, он передает его преподавателю. Преподаватель просматривает ответ и сообщает студенту оценку. Требуется создать многопоточное приложение, моделирующее действия преподавателя и студентов. При решении использовать парадигму «клиент-сервер».

22. *Первая задача о Винни-Пухе, или неправильные пчелы.* Неправильные пчелы, подсчитав в конце месяца убытки от наличия в лесу Винни-Пуха, решили разыскать его и наказать в назидание всем другим любителям сладкого. Для поисков медведя они поделили лес на участки, каждый из которых прочесывает одна стая неправильных пчел. В случае нахождения медведя на своем участке стая проводит показательное наказание и возвращается в улей. Если участок прочесан, а Винни-Пух на

нем не обнаружен, стая также возвращается в улей. Требуется создать многопоточное приложение, моделирующее действия пчел. При решении использовать парадигму портфеля задач.

23. *Первая военная задача.* Темной-темной ночью прапорщики Иванов, Петров и Нечепорчук занимаются хищением военного имущества со склада родной военной части. Будучи умными людьми и отличниками боевой и строевой подготовки, прапорщики ввели разделение труда: Иванов выносит имущество со склада, Петров грузит его в грузовик, а Нечепорчук подсчитывает рыночную стоимость добычи. Требуется составить многопоточное приложение, моделирующее деятельность прапорщиков. При решении использовать парадигму «производитель-потребитель».

24. *Задача о Пути Кулака.* На седых склонах Гималаев стоят два древних буддистских монастыря: Гуань-Инь и Гуань-Янь. Каждый год в день сошествия на землю боддисатвы Араватти монахи обоих монастырей собираются на совместное празднество и показывают свое совершенствование на Пути Кулака. Всех соревнующихся монахов разбивают на пары, победители пар бьются затем между собой и так далее, до финального поединка. Монастырь, монах которого победил в финальном бою, забирает себе на хранение статую боддисатвы. Реализовать многопоточное приложение, определяющего победителя. В качестве входных данных используется массив, в котором хранится количество энергии Ци каждого монаха. При решении использовать принцип дихотомии.

25. *Первая задача об Острове Сокровищ.* Шайка пиратов под предводительством Джона Сильвера высадилась на берег Острова Сокровищ. Не смотря на добытую карту старого Флинта, местоположение сокровищ по-прежнему остается загадкой, поэтому искать клад приходится практически на ощупь. Так как Сильвер ходит на деревянной ноге, то самому бродить по джунглям ему не с руки. Джон Сильвер поделил остров на участки, а пиратов на небольшие группы. Каждой группе поручается искать клад на одном из участков, а сам Сильвер ждет на берегу. Пираты, обшарив свою часть острова, возвращаются к Сильверу и докладывают о результатах. Требуется создать многопоточное приложение с управляющим потоком, моделирующее действия Сильвера и пиратов.

26. *Вторая задача об Острове Сокровищ.* Шайка пиратов под предводительством Джона Сильвера высадилась на берег Острова Сокровищ. Не смотря на добытую карту старого Флинта, местоположение сокровищ по-

прежнему остается загадкой, поэтому искать клад приходится практически на ощупь. Так как Сильвер ходит на деревянной ноге, то самому бродить по джунглям ему не с руки. Джон Сильвер поделил остров на участки, а пиратов на небольшие группы. Каждой группе поручается искать клад на нескольких участках, а сам Сильвер ждет на берегу. Группа пиратов, обшарив одну часть острова, переходит к другой, еще необследованной части. Закончив поиски, пираты возвращаются к Сильверу и докладывают о результатах. Требуется создать многопоточное приложение с управляющим потоком, моделирующее действия Сильвера и пиратов. При решении использовать парадигму портфеля задач.

27. *Пляшущие человечки.* На тайном собрании глав преступного мира города Лондона председатель собрания профессор Мориарти постановил: отныне вся переписка между преступниками должна вестись тайнописью. В качестве стандарта были выбраны «пляшущие человечки», шифр, в котором каждой букве латинского алфавита соответствует хитроумный значок. Реализовать многопоточное приложение, шифрующее исходный текст (в качестве ключа используется кодовая таблица, устанавливающая однозначное соответствие между каждой буквой и каким-нибудь числом). Каждый поток шифрует свои кусочки текста. При решении использовать парадигму портфеля задач.

28. *И снова пляшущие человечки.* Узнав о планах преступников озвученных в задаче 33, Шерлок Холмс предложил лондонской полиции специальную машину для дешифровки сообщений злоумышленников. Реализовать многопоточное приложение, дешифрующее кодированный текст. В качестве ключа используется известная кодовая таблица, устанавливающая однозначное соответствие между каждой буквой и каким-нибудь числом. Процессом узнавания кода в решении задачи пренебречь. Каждый поток дешифрует свои кусочки текста. При решении использовать парадигму портфеля задач.

29. Вычислить интеграл:

$$\int_a^b f(x) dx,$$

используя метод прямоугольников. Входные данные: вещественные числа  $a$  и  $b$ , функция  $f(x)$  задается с использованием описания в программе в виде отдельной функции. При суммировании использовать принцип дихотомии. Протестировать на различных функциях.



30. *Задача для агронома.* Председатель дачного кооператива Сидоров В.И. получил указание, что в связи с составлением единого земельного кадастра, необходимо представить справку о площади занимаемых земель. Известно, что территория с запада и востока параллельна меридианам, на севере ограничена параллелью, а с юга выходят к реке, описываемой функцией  $f(x)$ . Требуется создать многопоточное приложение, вычисляющее площадь угодий методом адаптивной квадратуры. При решении использовать парадигму рекурсивного параллелизма. Замечание: кривизну Земли из-за малой занимаемой площади не учитывать.