

ТЕМА 5 СИНХРОНИЗАЦИЯ РАБОТЫ НИТЕЙ В ЯЗЫКЕ C#

5.1 Цель лабораторной работы

Получить практические навыки по синхронизации нитей процессов компьютера.

5.2 Методические указания к лабораторной работе

Изучить материал 10, 11 и 12 лекций дисциплины.

Изучить материал, рекомендуемый в силлабусе: стр. 739-793 [3] – сокращенный вариант представлен:

« Блокировка сама по себе очень быстра: она требует десятков наносекунд, если собственно блокирования не происходит. Если требуется блокирование, то последующее переключение задач занимает уже микросекунды или даже миллисекунды на перепланировку потоков. Однако сравните это с часами, которые вы должны будете потратить, не поставив lock там, где надо.

При неправильном использовании у блокировки могут быть и негативные последствия – уменьшение возможности параллельного исполнения потоков, взаимоблокировки, гонки блокировок. Возможности для параллельного исполнения уменьшаются, когда слишком много кода помещено в конструкцию lock, заставляя другие потоки простаивать все время, пока этот код исполняется. Взаимоблокировка наступает, когда каждый из двух потоков ожидает на блокировке другого потока и, таким образом, ни тот, ни другой не могут двинуться дальше. Гонкой блокировок называется ситуация, когда любой из двух потоков может первым получить блокировку, однако программа ломается, если первым это сделает “неправильный” поток.

Взаимоблокировка – общий синдром многих объектов синхронизации. Хорошее правило, помогающее избегать взаимоблокировок, состоит в том, чтобы начинать с блокировки минимального количества объектов, и увеличивать степень детализации блокировок, когда размер кода в блокировке чрезмерно увеличивается.»

5.3 Домашнее задание на лабораторную работу

Синхронизировать три нити одного процесса с помощью оператор lock. Каждая нить поочередно формирует и печатает три числа a, b и c. Числа первой нити формируются в диапазоне от 10 до 19, второй нити – от 20 до 29 и третьей нити – от 30 до 39. Печать чисел каждой нитью производится по диагонали сверху вниз и слева направо (печать каждой нити занимает три строки) в продолжение печати предыдущей нити.

5.4 Индивидуальные задания для СРС

5.4.1 Синхронизировать три нити одного процесса с помощью глобальных переменных. Каждая нить формирует 3 числа для одноименной строки матрицы 3x3. Числа первой нити формируются в диапазоне от 10 до 19, второй нити – от 20 до 29 и третьей нити – от 30 до 39. Процесс управляет всеми нитями и выводит на экран результат их работы.

5.4.2 Синхронизировать три нити одного процесса с помощью метода Sleep. Первая нить выводит на экран монитора имя студента, вторая нить – его отчество и третья нить – его фамилию. Вывод должен выполняться в указанном порядке. Процесс синхронизирует работу всех нитей.

5.4.3 Синхронизировать три нити одного процесса с помощью метода Join. Первая нить выводит на экран монитора текущее время в минутах, вторая нить – текущее время в секундах, а третья нить – текущее время в миллисекундах. Вывод должен выполняться в указанном порядке. Процесс синхронизирует работу всех нитей.

5.4.4 Синхронизировать три нити одного процесса с помощью метода Join. Каждая нить формирует 3 числа для одноименной строки матрицы 3x3. Числа первой нити формируются в диапазоне от 10 до 19, второй нити – от 20 до 29 и третьей нити – от 30 до 39. Процесс управляет всеми нитями и выводит на экран результат их работы.

5.4.5 Синхронизировать три нити одного процесса с помощью оператора lock. Каждая нить выводит на экран монитора в заданном порядке три слова некоторого известного предложения. Процесс синхронизирует работу всех нитей.

5.4.6 Синхронизировать три нити одного процесса с помощью глобальных переменных. Первая нить выводит на экран монитора текущий месяц, вторая нить – день, а третья нить – час работы программы. Вывод должен выполняться в указанном порядке. Процесс синхронизирует работу всех нитей.

5.4.7 Синхронизировать три нити одного процесса с помощью метода Sleep. Каждая нить формирует 4 числа для одноименной строки матрицы 3x4. Числа первой нити формируются в диапазоне от 10 до 19, второй нити – от 20 до 29 и третьей нити – от 30 до 39. Процесс управляет всеми нитями и выводит на экран результат их работы.

5.4.8 Синхронизировать три нити одного процесса с помощью метода Join. Каждая нить выводит на экран монитора одно слово некоторого известного предложения. Процесс синхронизирует работу всех нитей.

5.4.9 Синхронизировать три нити одного процесса с помощью метода Sleep. Первая нить выводит на экран монитора текущее время в часах, вторая нить – текущее время в минутах, а третья нить – текущее время в секундах. Вывод должен выполняться в указанном порядке. Процесс синхронизирует работу всех нитей.

5.4.10 Синхронизировать три нити одного процесса с помощью оператора lock. Каждая нить формирует 5 чисел для одноименной строки матрицы 3x5. Числа первой нити формируются в диапазоне от 10 до 19, второй нити – от

20 до 29 и третьей нити – от 30 до 39. Процесс управляет всеми нитями и выводит на экран результат их работы.

5.4.11 Синхронизировать три нити трех процессов с помощью мьютекса. Первая нить первого процесса выводит на экран монитора фамилию студента, первая нить второго процесса – его имя и первая нить третьего процесса – его отчество.

5.4.12 Синхронизировать три нити одного процесса с помощью оператора lock. Первая нить выводит на экран монитора текущий месяц, вторая нить – день, а третья нить – час работы программы. Вывод должен выполняться в указанном порядке. Процесс синхронизирует работу всех нитей.

5.4.13 Синхронизировать три нити одного процесса с помощью автоматической синхронизации. Каждая нить формирует 3 числа для одноименной строки матрицы 3x3. Числа первой нити формируются в диапазоне от 10 до 19, второй нити – от 20 до 29 и третьей нити – от 30 до 39. Процесс управляет всеми нитями и выводит на экран результат их работы.

5.4.14 Синхронизировать три нити трех процессов с помощью семафора. Первая нить первого процесса выводит на экран монитора фамилию студента, первая нить второго процесса его оценки по математике и физике и первая нить третьего процесса его оценку по системному программированию.

5.4.15 Синхронизировать три нити трех процессов с помощью мьютекса. Первая нить первого процесса выводит на экран монитора текущее время в часах, первая нить второго процесса – текущее время в минутах, а первая нить третьего процесса – текущее время в секундах.

5.4.16 Синхронизировать три нити трех процессов с помощью мьютекса. Каждый процесс формирует 3 числа для одноименной строки матрицы 3x3. Первая нить первого процесса формирует и выводит на экран монитора числа от 10 до 19, первая нить второго процесса – от 20 до 29 и первая нить третьего процесса – от 30 до 39.

5.4.17 Синхронизировать три нити одного процесса с помощью глобальных переменных. Каждая нить выводит на экран монитора два слова некоторого известного предложения. Процесс синхронизирует работу всех нитей.

5.4.18 Синхронизировать три нити трех процессов с помощью семафора. Первая нить первого процесса выводит на экран монитора сторону треугольника А, первая нить второго процесса – сторону треугольника В, а первая нить третьего процесса – сторону треугольника С. Первый процесс определяет можно ли из полученных значений сторон построить треугольник и печатает в консольном окне соответствующие комментарии.

5.4.19 Синхронизировать три нити трех процессов с помощью семафора. Каждый процесс формирует 4 числа для одноименной строки матрицы 3x4. Первая нить первого процесса формирует и выводит на экран монитора числа от 10 до 19, первая нить второго процесса – от 20 до 29 и первая нить третьего процесса – от 30 до 39.

5.4.20 Синхронизировать три нити одного процесса с помощью автоматической синхронизации. Каждая нить выводит на экран монитора два

слова некоторого известного предложения. Процесс синхронизирует работу всех нитей.

5.5 Контрольные вопросы для защиты отчета на СРСП

5.5.1 Понятие действия?

5.5.2 Понятие синхронизации нитей?

5.5.3 Классификация средств синхронизации нитей?

5.5.4 Как с помощью глобальной переменной можно синхронизировать работу нитей одного процесса?

5.5.5 Как с помощью метода Sleep можно синхронизировать работу нитей одного процесса?

5.5.6 Как с помощью метода Join можно синхронизировать работу нитей одного процесса?

5.5.7 Как с помощью оператора lock можно синхронизировать работу нитей одного процесса?

5.5.8 Как с помощью объекта синхронизации Mutex можно синхронизировать работу нитей?

5.5.9 Как с помощью объекта синхронизации Semaphore можно синхронизировать работу нитей?

5.5.10 Понятие автоматической синхронизации нитей.