## Активное ожидание

При активной блокировке поток циклически проверяет статус ожидаемого события, фактически поток не прекращает своей работы и не освобождает процессорное время для других потоков.

## Атомарные операторы

Атомарные операторы предназначены для потокобезопасного неблокирующего выполнения операций над данными, преимущественно целочисленного типа. Атомарность означает, что при выполнении оператора никто не вмешается в работу потока.

## Блок окружения потока

Раздел структуры потока, который содержит заголовок цепочки обработки исключений, локальное хранилище данных для потока и некоторые структуры данных, используемые интерфейсом графических устройств (GDI) и графикой OpenGL

## Взаимно-исключительный доступ

Взаимная исключительность означает, что в каждый момент времени с общим ресурсом работает только один поток, другие потоки блокируются в ожидании завершения работы первого потока.

## Декомпозиция

Под декомпозицией понимается разбиение задачи на относительно независимые части (подзадачи)

## Динамическая декомпозиция

При динамической декомпозиции каждый поток, участвующий в обработке, обращается за блоком данных. После обработки блока данных поток обращается за следующей порцией. Динамическая декомпозиция требует синхронизации доступа потоков к структуре данных. Размер блока определяет частоту обращений потоков к структуре

## Закон Амдала

Джин Амдал (Gene Amdahl) показал, что верхняя граница для ускорения определяется долей последовательных вычислений алгоритма и не превышает обратную величину к этой доле

## Классификация Джонсона

Классификация Джонсона основана на структуре памяти — общая (глобальная, global memory) или распределенная (distributed memory), и механизме коммуникаций — разделяемые переменные (shared variables) или передача сообщений (message passing).

## Классификация Флинна

Классификация вычислительных систем. Четыре класса вычислительных систем выделяются в соответствии с двумя характеристиками: множественный или одиночный поток команд, которые данная архитектура способна выполнить в единицу времени, и множественный или одиночный поток данных, которые могут быть обработаны в единицу времени.

## Конкурентные коллекции

Динамические коллекции, обеспечивающие потокобезопасность операций добавления и удаления элементов

## Критическая секция

Фрагмент кода, к которому должен быть обеспечен взаимно исключительный доступ

## Масштабируемость

Параллельный алгоритм называется масштабируемым, если при росте числа процессоров он обеспечивает увеличение ускорения при сохранении постоянного уровня эффективности использования процессоров. Масштабируемость алгоритма обеспечивается возможностью использования всех имеющихся вычислительных ресурсов в системе (процессоров или ядер процессора) для решения задачи.

## Многопроцессное выполнение

Многопроцессное выполнение подразумевает оформление каждой подзадачи в виде отдельной программы (процесса)

## Модель делегирования

Типовая модель параллельного приложения, в которой выделяется один центральный поток (управляющий, мастер) и несколько рабочих потоков. Управляющий поток запускает рабочие потоки, передает им все необходимые данные, контролирует работу и обрабатывает результаты после их завершения.

## Модель конвейерной обработки

Типовая модель параллельного приложения, в которой поток обрабатываемых данных проходит через несколько этапов. Прохождение этапов осуществляется строго последовательно. Параллелизм достигается за счет одновременной обработки разных элементов на разных этапах.

## Модель с равноправными узлами

Типовая модель параллельного приложения, в которой В модели с равноправными узлами все потоки (или задачи) участвуют в обработке; центрального узла нет. Работа узлов может осуществляться параллельно.

## Монитор

Средство синхронизации для взаимно-исключительного доступа к фрагменту кода.

## Пассивное ожидание

Пассивное ожидание реализуется с помощью операционной системы, которая сохраняет контекст потока и выгружает его, предоставляя возможность выполняться другим потокам. При наступлении ожидаемого события операционная система «будит» поток — загружает контекст потока и выделяет ему процессорное время.

## Приоритеты потоков

Приоритеты потоков определяют очередность выделения доступа к ЦП. Высокоприоритетные потоки имеют преимущество и чаще получают доступ к ЦП, чем низкоприоритетные

## Проблема гонки данных

Проблема гонки данных возникает при следующих условиях:

1. Несколько потоков работают с разделяемым ресурсом.

2. Конечный результат зависит от очередности выполнения командных последовательностей разных потоков.

## Проблема ложного разделения данных

Проблема связана с тем, что потоки работают с разными переменными, которые в оперативной памяти расположены физически близко.

## Семафор

Средство синхронизации с внутренним счетчиком. При захвате семафора счетчик атомарно уменьшается на единицу. При освобождении семафора счетчик атомарно увеличивается на единицу. Если счетчик равен нулю, при попытке захвата семафора поток блокируется и ожидает освобождения семафора.

## Сигнальные сообщения

Средство синхронизации для условной блокировки. Один поток блокируется в ожидании сигнала от другого потока

## Синхронизация

Синхронизация необходима для координации выполнения потоков, то есть для согласования порядка выполнения потоков или для согласования доступа потоков к разделяемому ресурсу.

## Системы класса MIMD

Вычислительные системы с множественным потоком команд и множественных потоком данных (класс «MIMD» — Multiple Instructions, Multiple Data).

## Системы класса MISD

Вычислительные системы, в которых существует множественный поток команд и одиночный поток данных (класс «MISD» — Multiple Instructions, Single Data).

## Системы класса SIMD

Вычислительные системы с одиночным потоком команд и с множественным потоком данных (класс «SIMD» — Single Instruction, Multiple Data). Подобный класс составляют многопроцессорные системы, в которых в каждый момент времени может выполняться одна и та же команда для обработки нескольких информационных элементов.

## Системы класса SISD

Вычислительные системы, в которых существует одиночный поток команд и одиночный поток данных (класс «SISD» — SingleInstruction, SingleData). В каждый момент времени процессор обрабатывает одиночный поток команд над одиночным потоком данных. К данному типу систем относятся последовательные персональные компьютеры с одноядерными процессорами.

## Статическая декомпозиция

При статической декомпозиции фрагменты данных назначаются потокам до начала обработки и, как правило, содержат одинаковое число элементов для каждого потока

## Стек пользовательского режима

Раздел структуры потока, который используется для передаваемых в методы локальных переменных и аргументов

## Стек режима ядра

Раздел структуры потока, который используется, когда код приложения передает аргументы в функцию операционной системы, находящуюся в режиме ядра.

## Ускорение

Ускорение параллельного алгоритма по сравнению с последовательным вариантом выполнения определяется как отношение времени выполнения последовательного алгоритма к времени выполнения параллельного алгоритма.

## Эффективность

Оценка эффективности параллельного алгоритма определяется как отношение времени выполнения последовательного алгоритма к произведению числа процессоров и времени выполнения параллельного алгоритма

## Ядро потока

Раздел структуры потока, который содержит информацию о текущем состоянии потока: приоритет потока, программный и стековый указатели. Программный и стековые указатели образуют контекст потока и позволяют восстановить выполнение потока на процессоре.