

Дмитрий Иванов, JetBrains

Многозадачность

- Многозадачность **vs** Многопоточность
- Кооперативная **vs** Вытесняющая
- Процессы (Processes), потоки (Threads), задачи (Tasks)

Примитивы синхронизации

- `lock()`, Monitor, [Synchronized]
- EventWaitHandle: Manual, Auto
- Mutex – межпроцессное взаимодействие
- Barrier, CountdownEvent, Semaphore
- Slim, SpinWait – для короткого времени ожидания
- WaitHandle, Kernel32Dll

The top of the slide features a dark background with a pattern of small, colorful, hand-drawn icons. These icons include a lightbulb, a gear, a pyramid, a hammer, a camera, a hand, a flame, a lightning bolt, and various geometric shapes.

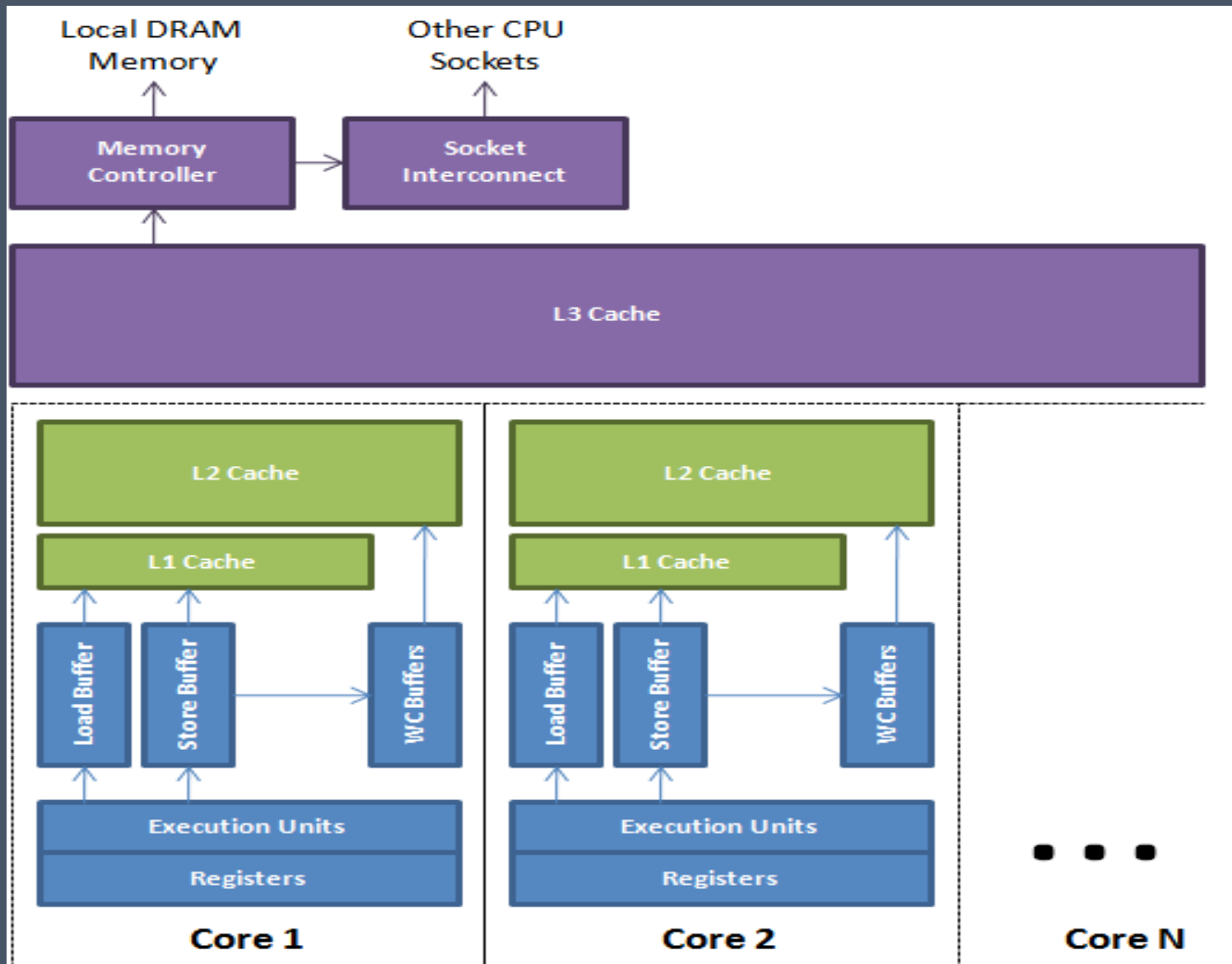
Демонстрация

ReaderWriterLock

ReadWriteLock

- Всегда используйте Slim (или велосипед =)
- Upgradable – только один тред
- Fairness, starvation
- **using** вместо **try... finally**
- TryEnterReadLock, TryEnterWriteLock

Memory model



Memory barriers

- Значения “оседают” в регистрах, Store buffer, Write Combining buffer
- Instruction reordering
- Процессорные инструкции: **mfence**, **lfence**, **sfence**
- Thread.MemoryBarrier(), **volatile**, Volatile.Read(), Volatile.Write()



Демонстрация

Double-checked locking

Неблокирующие алгоритмы

- Процессорные инструкции: `stprxchg`, `xadd`, префикс `lock`
- **Interlocked**: `Read(Int64)`, `Increment`, `Exchange`, `CompareExchange`
- Зачем:
 - Пишем свою структуру данных
 - Весело =)
- Тесты, тесты и еще раз тесты !!!

The top header features a dark background with a repeating pattern of colorful, stylized icons. These icons include a lightbulb, a gear, a hand, a hammer, a camera, a pyramid, a magnifying glass, a flame, a lightning bolt, and an eye, among others.

Демонстрация

BlockingQueue

The top header features a dark background with a repeating pattern of colorful, stylized icons. These icons include a lightbulb, a gear, a hand, a hammer, a camera, a pyramid, a magnifying glass, a flame, a lightning bolt, and an eye, among others.

Демонстрация

BlockingQueue

Модели многопоточного приложения

- Транзакционная
 - Immutable state
 - no locking
 - Commit/Rollback
- Пессимистичная
 - Глобальный RwLock
 - Прерываемые Read-ы
- Акторы, COM

COM-объекты

- Apartments: STA, MTA, Neutral

STA-thread

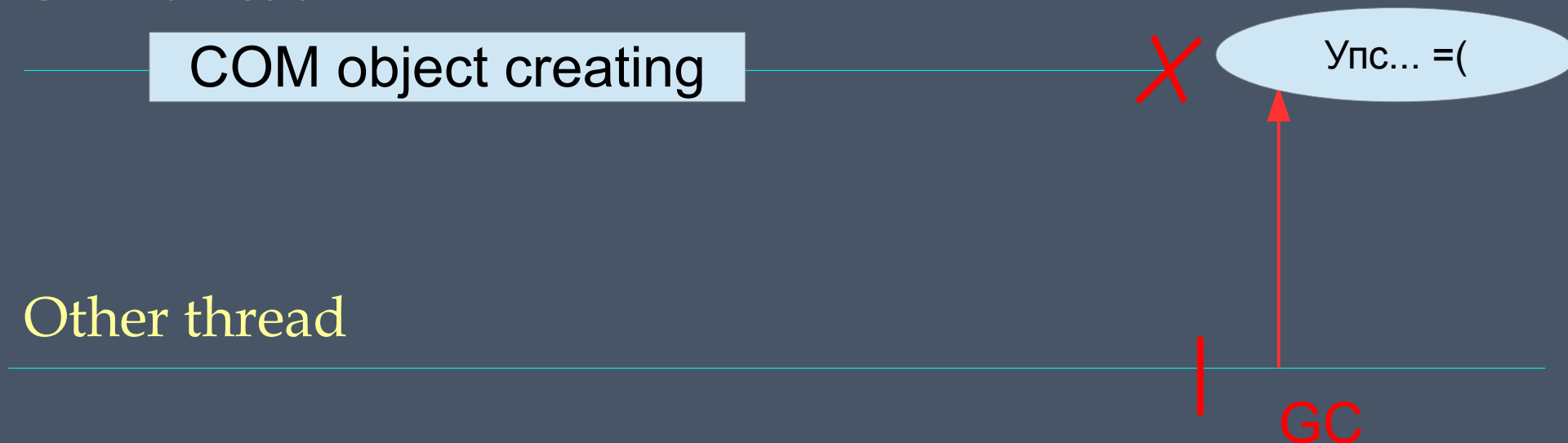
COM object creating



Упс... =(

Other thread

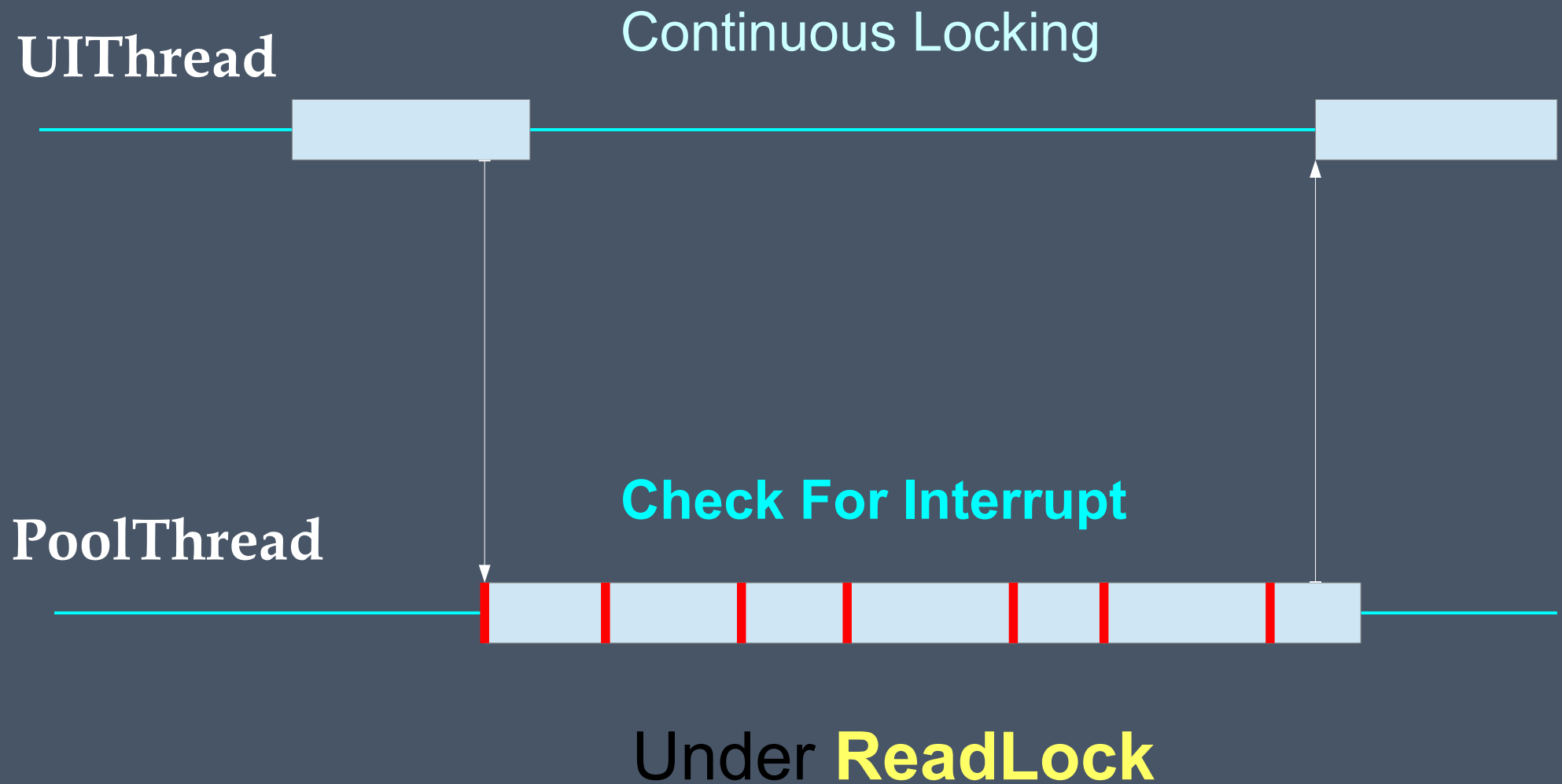
GC



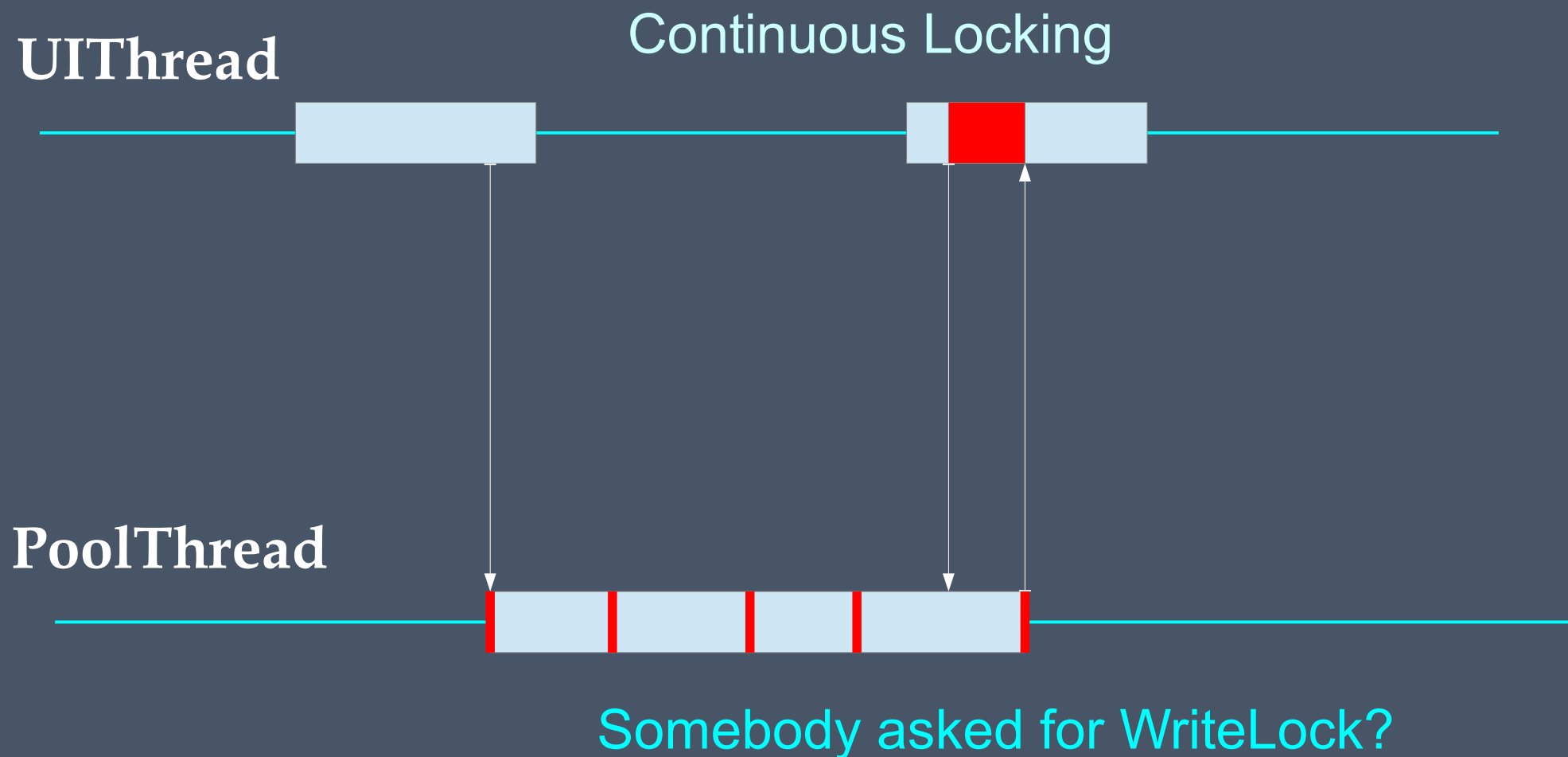
Модель ReSharper

- Main thread
 - Разрешен AcquireWrite
 - Разрешен Upgrade
 - Разрешен AcquireRead
- Pool thread
 - Разрешён TryAcquireRead
 - Везде CheckForInterrupt (<200ms)
 - Запрещён TryAcquireWrite

InterruptedException



Прерывание



ReentrancyGuard

- Reentrancy in UI Thread
- ExecuteOrQueue
- QueueWithReadLock, QueueWithWriteLock
- Dispatcher – вызов без гарда

UIThread

Read

Write

GroupingEvent

prolongate



no prolongate



Другие конструкции

- SingleThreadExecutorView - Actor
 - Очередь задач
 - Возможно синхронное ожидание
 - Stealing задач
- TaskBarrier (IDisposable)
 - Задачи могут порождать задачи
 - Топологическая сортировка зависимых задач

Timer API

- QueueAt, QueueRecurring
- Время: UtcNow
- Самое быстрое: Environment.TickCount

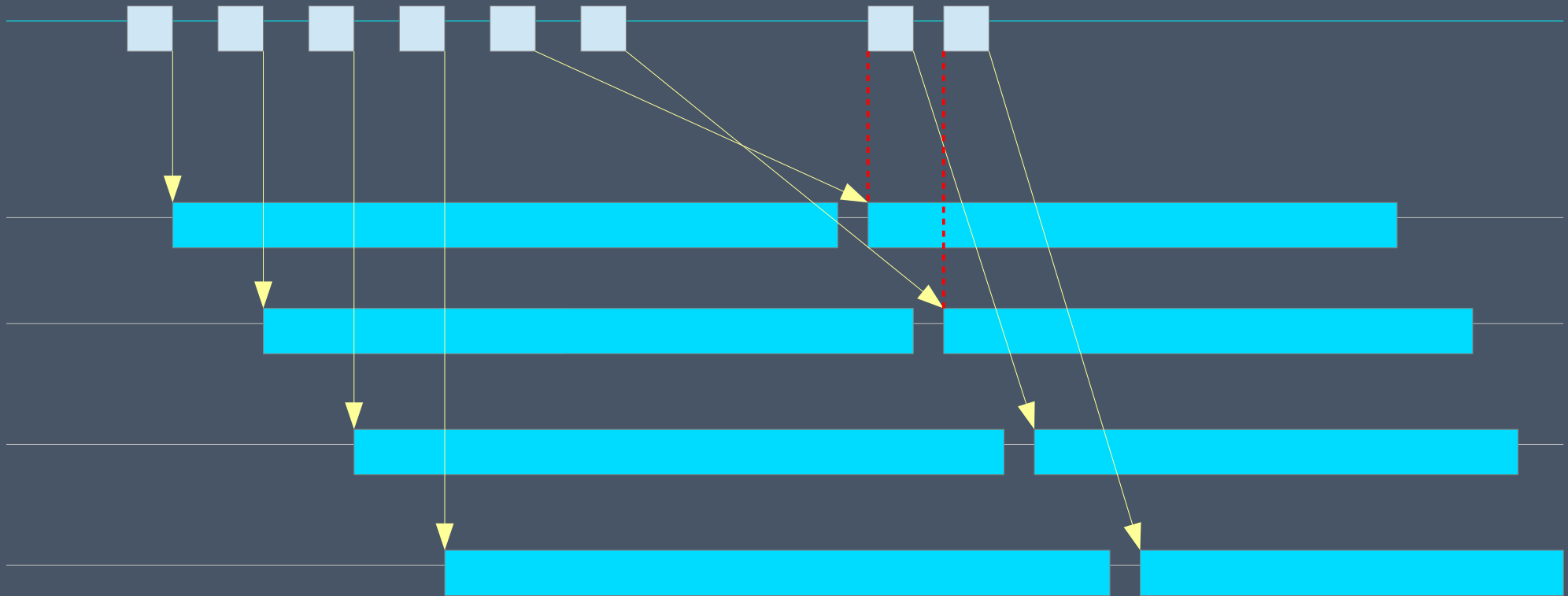
IProperty<T>

- Isignal<T>, Advise
- FlowInto, CreateAnd(bool)
- FlowIntoGrouped - GroupingEvent
- FlowIntoUiThread - marshallng

File IO

- PreprocessSingleThreadedAndParallelize

Buffer = 2



Средства анализа

- Видимые тормоза, дёрганность, лаги при тайпинге → подвисание `UIThread` (не качается очередь сообщений)
- Проверяем: между `CheckForInterrupt` < 200ms
- Проверяем: `WriteLock.Acquire` < 200ms
- DotTrace **Timeline**

SynchronizationContext

- [ThreadStatic]
- Send(), Post() (default → ThreadPool)
- Wait() (через WaitHelper)
- WPF →
DispatcherSynchronizationContext

Tasks

- Task(Action), Task<T>(Func<T>)
- short **vs** long
- Lifecycle (status): created → started → executing → completed (canceled, failed)
- Interruptable: CancellationToken
- Блокирующее API: Wait (One/ All/ Any)
- Task.Start(), Schedulers

The top header features a dark background with a pattern of colorful, faint icons. These icons include a lightbulb, a gear, a hand, a hammer, a camera, a pyramid, a magnifying glass, a flame, a lightning bolt, and an eye, among others.

Демонстрация

Tasks, Cancellation,
WPF context & scheduler

Старое API

- APM
 - `IAsyncResult`, `BeginXxx(AsyncCallback)`, `EndXxx()`
- EAP
 - `XxxAsync()`, `XxxCompletedEventHandler`, `XxxAsyncCancel()` или `CancelAsync()`
- Адаптер для IOBound tasks → `TaskCompletionSource`

Неблокирующее API

- Task.Delay(), TaskCompletionSource
- ContinueWith(), ContinueWhenAll()
- async / await – сахар для ContinueWith
- parent / child - неблокирующее ожидание
(TaskCreationOptions.AttachedToParent)

Демонстрация

Эмулируем Async

Вопросы и ответы

?

?

?

?

?

?

?

?

?

?