**31. Системы реального времени. Метрики систем реального времени. Проблемы реализации MRE реального времени.**

* Жесткое реальное время (Hard RT)
  + Завершение операции после наступления крайнего срока является бесполезным.
  + управление оборудованием, протоколы передачи данных низкого уровня, обработка звуковых сигналов.
  + Несоблюдение крайнего срока может привести к поломке оборудования, потере данных, неприемлемым искажениям сигнала.
* Мягкое реальное время (Soft RT)
  + Завершение операции после наступления крайнего срока является допустимым при условии ограниченного количества таких событий
  + Сетевые сервисы, потоковое видео, видеоигры
  + Существуют механизмы обработки задержек: пропуск кадра, повторный запрос и т. д.

Real-Time OS:

* Примеры: RTLinux, QNX, RTOS, VxWorks
* Особенности
  + Невытесняющее планирование нитей
  + Строгая семантика приоритетов нитей
  + Упорядоченная синхронизация (fair locks)
  + Запрет выгрузки памяти
  + Наследование приоритетов
* Недостатки
  + Дополнительные затраты на синхронизацию
  + Ограниченная масштабируемость
  + Убегающие нити (Runaway threads)

Метрики:

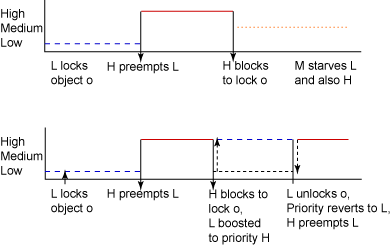
* Минимальная/Средняя Утилизация Мутатора (Minimum Mutator Utilization)
  + Минимальное/среднее соотношение времени исполнения пользовательского и системного кода за максимальное/среднее время отклика
  + Независимая от приложения метрика производительности
* Максимальное/Среднее время отклика
  + максимальное/среднее время реакции на внешнее событие
  + Сумма времени обработки в пользовательском коде и задержки создаваемой системой
* Минимальная память
  + Количество памяти необходимое системе для сохранения временных метрик

Проблемы:

* Детерминированное планирование нитей
  + Код нити не может прерываться в произвольных местах
  + Строгий порядок передачи управления между нитями
* Синхронизация
  + Детерминированная передача управления при синхронизации
  + Строгий порядок очередности ожидающих нитей
  + Защита от инверсии приоритетов
* Управление памятью
* Детерминированное время выделения памяти
* Детерминированное время обращения к памяти
* Защита от фрагментации и выгрузки памяти

Невытесняющее планирование: Нити с более высоким приоритетом стоят в очереди несколько раз. Порядок передачи нити строго детерменирован.

Может возникнуть инверсия приоритетов: Через блокировку ресурса низкоприоритетный процесс «понижает» приоритет более высокоприоритетного процесса



Стандартные техники программирования:

* Статическое выделение памяти
  + Все объекты выделяются перед стартом основного цикла обработки событий
* Обработка пропусков крайнего срока (watchdog)
  + Обнаружение повисания системы
  + Рестарт нитей обработки событий
* Обработка повышенной загрузки ЦП одной нитью
  + Обнаружение убегающих нитей
  + Рестарт нити

Проблемы MRE:

* Сборка мусора
  + Задержки при останавливающей сборке мусора
  + Невозможность выделить достаточный объем памяти без сборки мусора
* JIT компиляция
  + Недетерминированная производительность
  + Задержки компиляции
* Загрузка классов
  + Ленивая загрузка и верификация классов во время исполнения