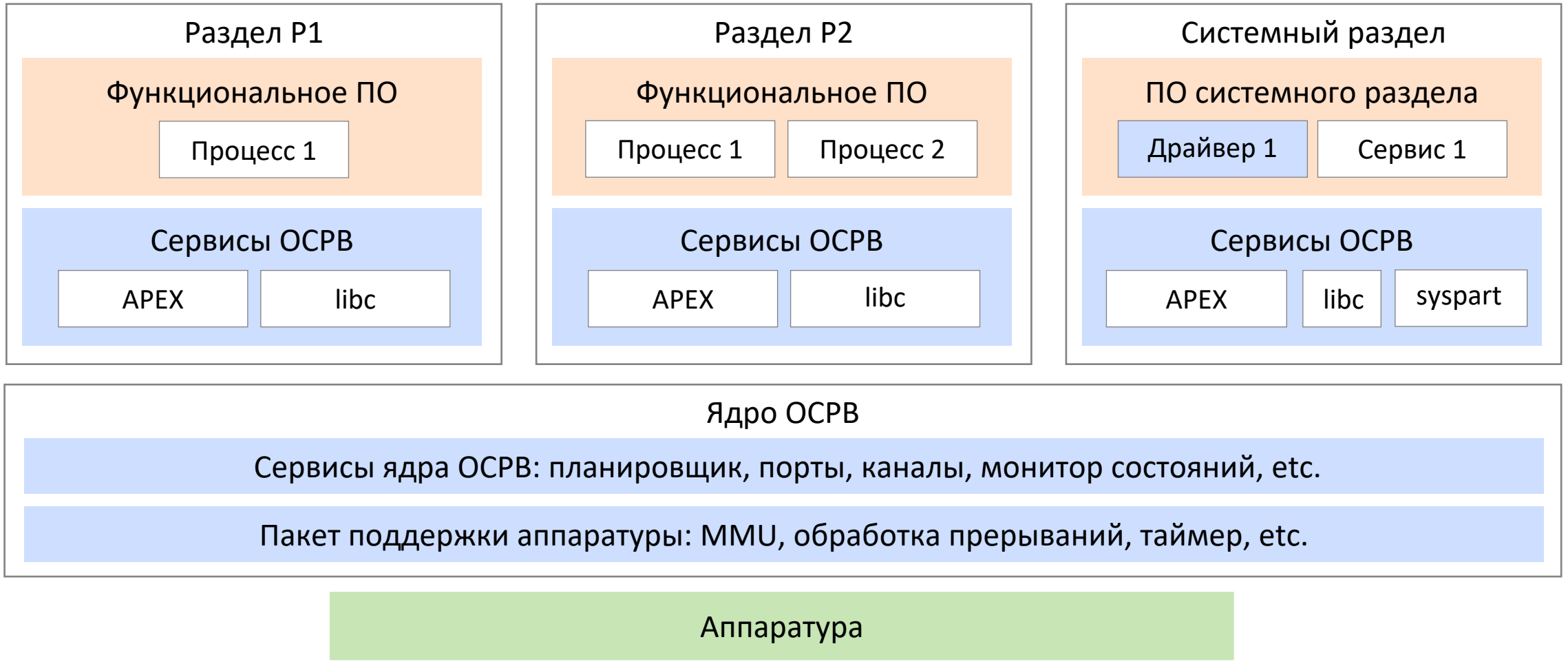
Метод надёжной временной изоляции для ARINC 653 OCPB

В. Ю. Чепцов <cheptsov@ispras.ru>

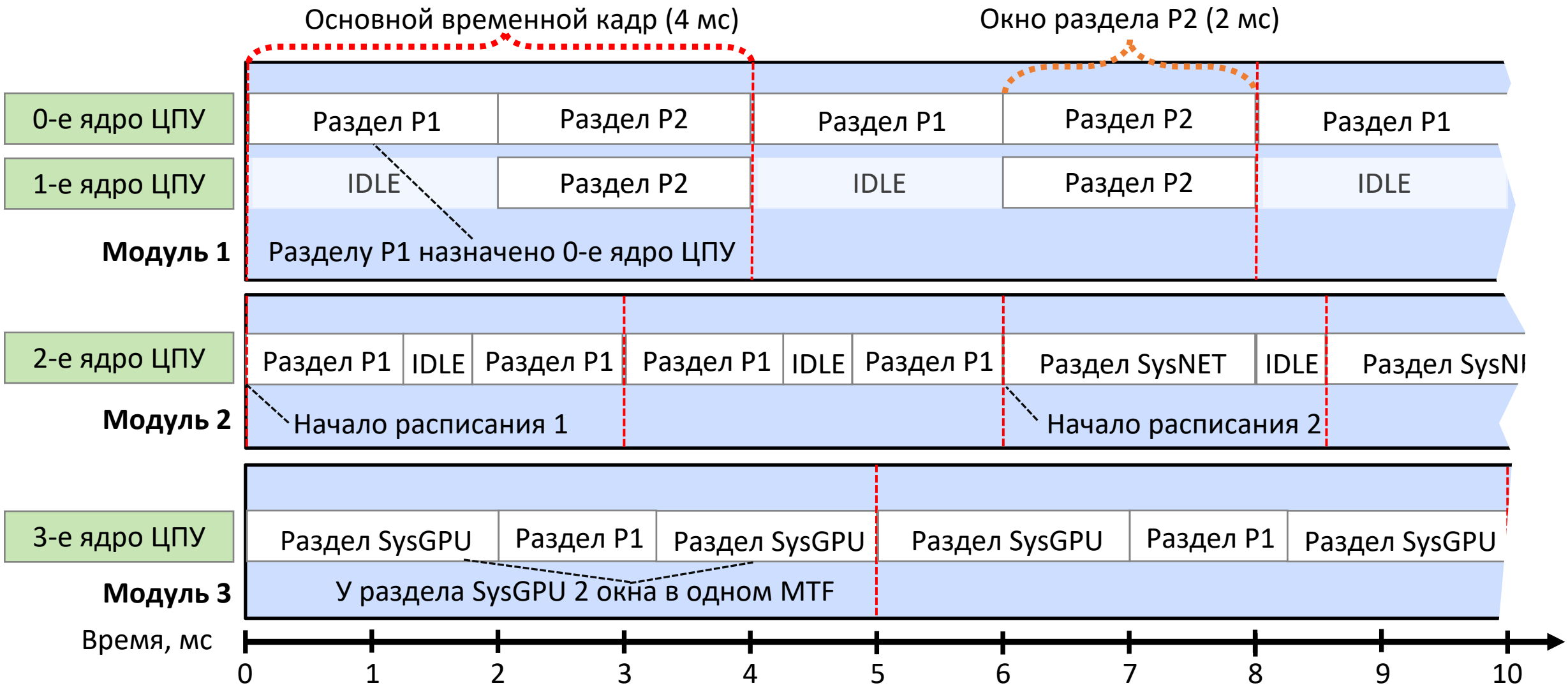
OS DAY 2024-06-20

Структура ОСРВ на базе ARINC 653



ИСП РАН

Расписание ARINC 653



ИСП РАН

Требования к изоляции из ARINC 653

Рамочные требования сформулированы в DO-248 и CAST-32A:

- Раздел не может повредить области памяти кода, данных или ввода-вывода других разделов.
- Раздел не может использовать больше разделяемых ресурсов, чем ему было выделено.
- Сбои в одном разделе не могут приводить к негативным последствиям в других разделах.
- Раздел не может выполняться на процессорном ядре более длительное время, чем ему было выделено, вне зависимости от активности или неактивности других разделов на других процессорных ядрах.

В реальности есть дополнительные требования по защищённости, латентности, отказоустойчивости и пр.

Предложенные требования к временной изоляции

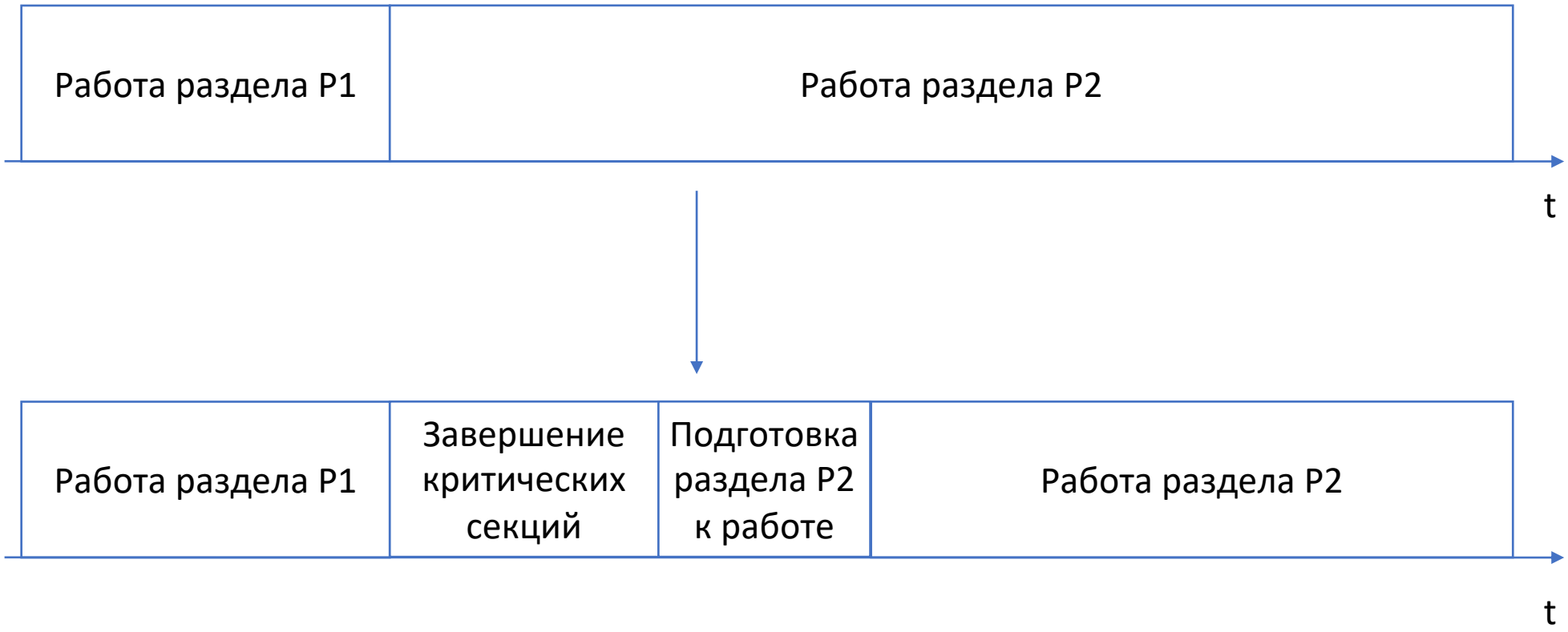
- Время начала работы раздела в новом окне должно соответствовать временной отметке начала окна в статическом расписании с заданной погрешностью.
- Выполнение раздела на процессорном ядре должно быть ограничено в соответствии с конфигурацией, и не может быть превышено.
- Выполнение раздела на процессорном ядре не должно прерываться, кроме как для выполнения переключения между разделами или выполнения действий, затребованных самим разделом.
- Работа предыдущих разделов не приводит к отложенному влиянию на время работы кода текущего раздела.

Особенности временной изоляции в ARINC 653

Архитектура ARINC 653 обеспечивает базовые механизмы временной изоляции:

- Механизм расписаний обеспечивает изоляцию разделов друг от друга, не позволяя двум независимым приложениям выполняться одновременно.
- Механизм планирования потоков с фиксированными приоритетами и ограничение вложенности блокировок обеспечивают предсказуемое выполнение потоков внутри раздела и защиту от инверсии приоритетов.
- Статическое бюджетирование времени в расписаниях позволяет проводить оценку наихудшего времени выполнения кода.

Переключение окон в ARINC 653 OCPB



ИСП РАН

Обеспечение временной изоляции в ARINC 653 OCPB

Качество реализации временной изоляции в ARINC 653 OCPB зависит от:

- Наихудшего времени выполнения (WCET) критических секций, *синхронно* блокирующих переключение адресных пространств:
  - Обслуживающих алгоритмов (межпроцессных обменов, обновления расписаний, проверки сроков завершения потоков и пр.)
  - Аппаратно-зависимых алгоритмов (межъядерной синхронизации (rendezvous), переключения адресных пространств, обработчиков прерываний и пр.)
- Наличия и наихудшего времени выполнения (WCET) критических секций, *асинхронно* блокирующих переключение адресных пространств (прерываний от таймеров и других устройств).

Реализация временной изоляции в ARINC 653 ОСРВ

Оптимизация WCET синхронных событий специфична для конкретной системы и плохо поддаётся обобщению даже для систем одного класса, таких как ARINC 653.

В нашей реализации ARINC 653 ОСРВ работа с устройствами вынесена в системные разделы, поэтому основная задача обеспечения временной изоляции в части асинхронных событий ложится на планировщик процессов.

Архитектурное решение обработки событий от таймеров определяет стратегию оптимизации WCET асинхронных событий и латентность выполнения кода под управлением ОСРВ.

ИСП РАН

Стратегии механизмов планирования

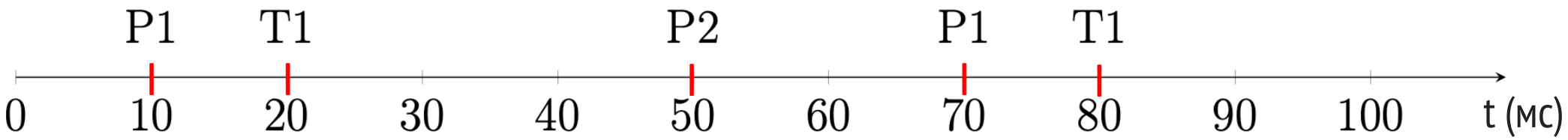
- Периодический механизм планирования, при котором события таймеров обрабатываются с некоторой фиксированной частотой. В зависимости от возможностей аппаратуры:
  - Таймер взводится при каждом тике в обработчике прерывания.
  - Таймер взводится автоматически в момент генерации прерывания.

  Часто встречается в ОСРВ, включая ARINC 653 ОСРВ (FreeRTOS, VxWorks, OSE).
- Непериодический механизм планирования, при котором генератор прерывания от таймеров устанавливается на ближайшее событие.

  Распространён в системах общего назначения (Windows, NoHz Linux) и микроконтроллерах как *средство снижения энергопотребления*.

Периодический механизм планирования

Периодический планировщик с квантом 10 мс:



Планирование разделов начинается с 10 мс с момента запуска ОСРВ, разделы P1 и P2 работают друг за другом и имеют длительность 40 мс и 20 мс соответственно.

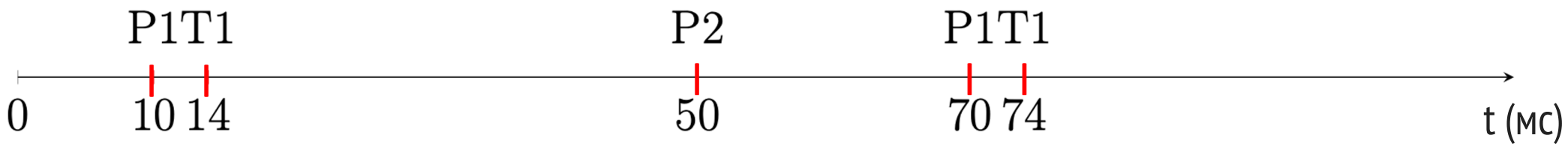
В разделе P1 сразу после передачи управления устанавливается таймер T1 на 4 мс. В периодическом планировщике он сработает через 10 мс.

Недостатки периодического механизма планирования

- При выполнении приложения процесс может быть прерван в произвольный момент времени для обработки прерывания.
- При расчёте наихудшего времени выполнения кода необходимо учитывать максимальную длительность и минимальный период обработки прерывания.
- Для приближения оценки наихудшего времени выполнения кода к реальной необходимо вычислять время выполнения на достаточно длинных трассах, чтобы не считать время обработки прерывания несколько раз.
- Низкая частота обработки прерываний приводит к высокой латентности системных сервисов, когда же высокая частота приводит к повышению наихудшего времени выполнения кода.

Непериодический механизм планирования

Непериодический планировщик с разрешением 1 мс:



Планирование разделов начинается с 10 мс с момента запуска ОСРВ, разделы P1 и P2 работают друг за другом и имеют длительность 40 мс и 20 мс соответственно.

В разделе P1 сразу после передачи управления устанавливается таймер T1 на 4 мс. В непериодическом планировщике он сработает через 4 мс.

Преимущества непериодического механизма планирования

- Повышение производительности кода за счёт отсутствия «холостых» срабатываний таймера во время выполнения активного процесса.
- Повышение времени отклика системы за счёт возможности установки прерывания от таймера на более ранний срок.
- Минимизация энергопотребления за счёт отсутствия «холостых» срабатываний таймера во время бездействия системы.

Недостатки непериодического механизма планирования

Основным препятствием во внедрении непериодических таймеров в ОСРВ является предельная частота возникновения прерываний.

Если у функционального ПО есть возможность взводить таймеры (в ARINC 653 такие интерфейсы есть в каждом наборе сервисов), то пользовательский код может сымитировать ситуацию, в которой асинхронные прерывания будут возникать слишком часто, что приведёт к деградации производительности и ухудшению времени наихудшего случая.

Разрешающая способность таймера обычно высока (< 10 мкс) и может превышать время обработки прерывания. Т. е. установка прерывания перед переключением на высокоприоритетный поток или другой раздел приводит к увеличению латентности на время обработки прерывания.

Совмещение непериодического механизма с периодическим

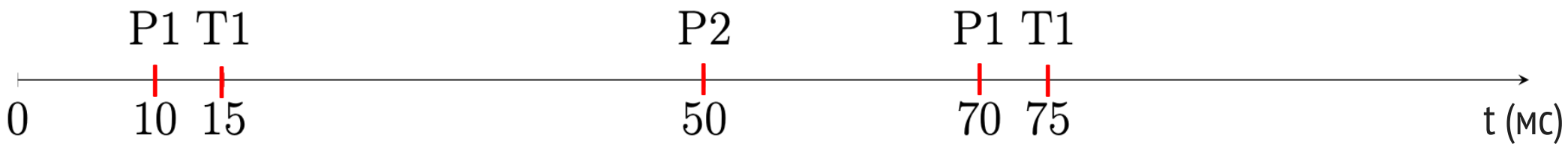
Предложим совместить подходы периодического и непериодического таймера, используя модель непериодического таймера, но установив минимальное время срабатывания таймера не чаще, чем раз в квант:

- Квант времени ограничивает частоту возникновения прерываний.
- Непериодический таймер исключает срабатывание «холостых» прерываний.

Латентность сохраняется на предсказуемом уровне за счёт корректировки кванта планировщика под задачу и целевой вычислитель.

Непериодический механизм планирования с квантованием

Непериодический планировщик с квантом 5 мс:



Планирование разделов начинается с 10 мс с момента запуска ОСРВ, разделы P1 и P2 работают друг за другом и имеют длительность 40 мс и 20 мс соответственно.

В разделе P1 сразу после передачи управления устанавливается таймер T1 на 4 мс. В непериодическом планировщике с квантованием он сработает через 5 мс.

Результаты

- Механизм непериодического планирования с квантованием решает проблему «холостых» срабатываний таймера во время выполнения активного процесса и позволяет ограничивать латентность от возникновения прерываний.
- Подход был успешно применён в рамках КЛОС на микропроцессорах с архитектурами ARMv7-A, ARMv8-A, MIPS32, PowerPC, RISC-V.
- В результате применения подхода удалось упростить формулу расчёта WCET, исключив из неё время на обработку прерываний от таймера.

Спасибо за внимание!

ИСП РАН