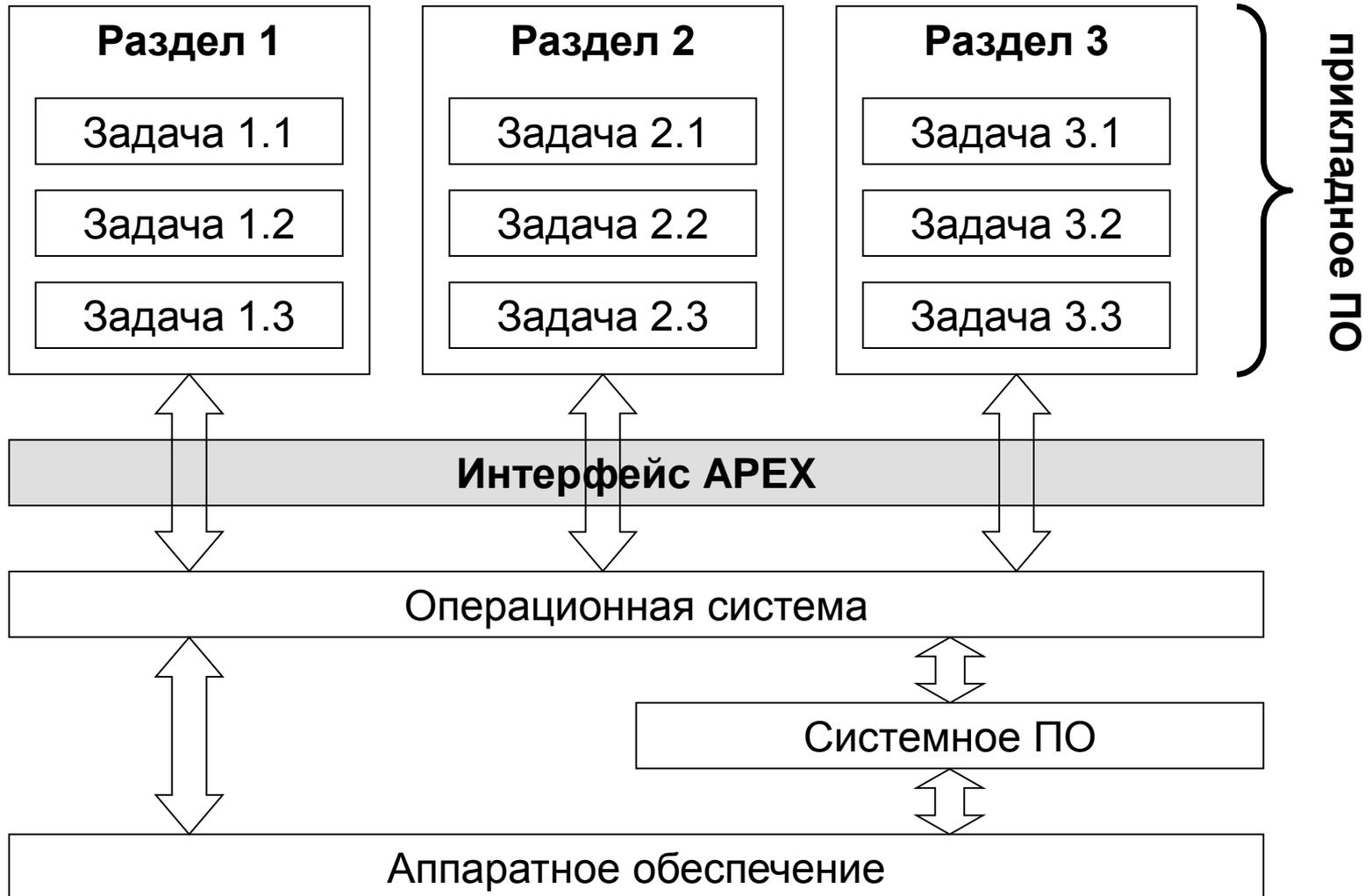


ОС РВ, использующие
статико-динамическую
модель вычислений

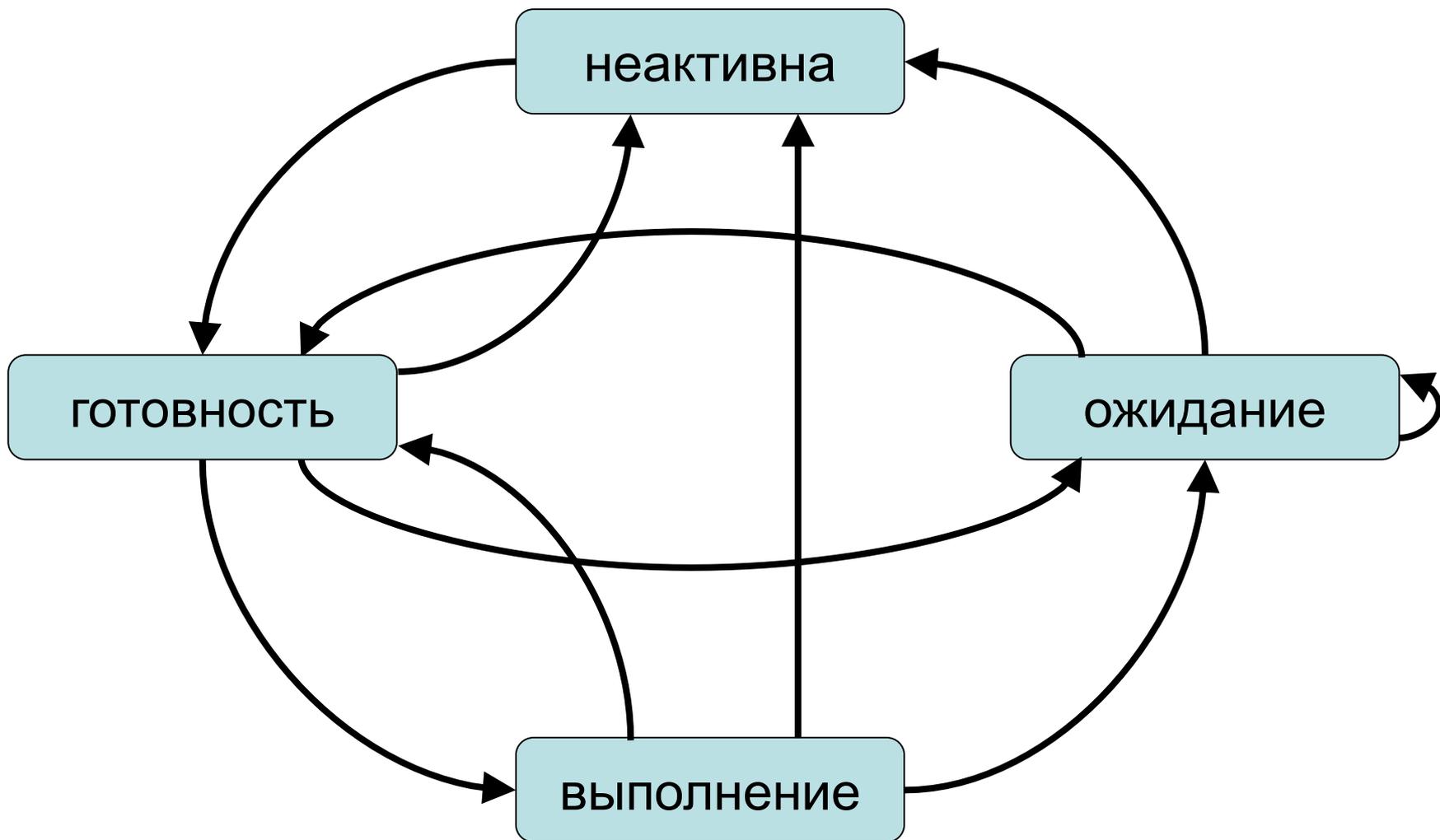
ИМА в разработке ПО

- Особенности:
 - Стандартное API со стороны ОС
 - Статическое разделение времени, памяти и ресурсов
- Преимущества:
 - Надежность
 - Переносимость
 - Возможность повторного использования
 - Модульность
 - Упрощение верификации и сертификации

Структура ПО. Интерфейс АРЕХ

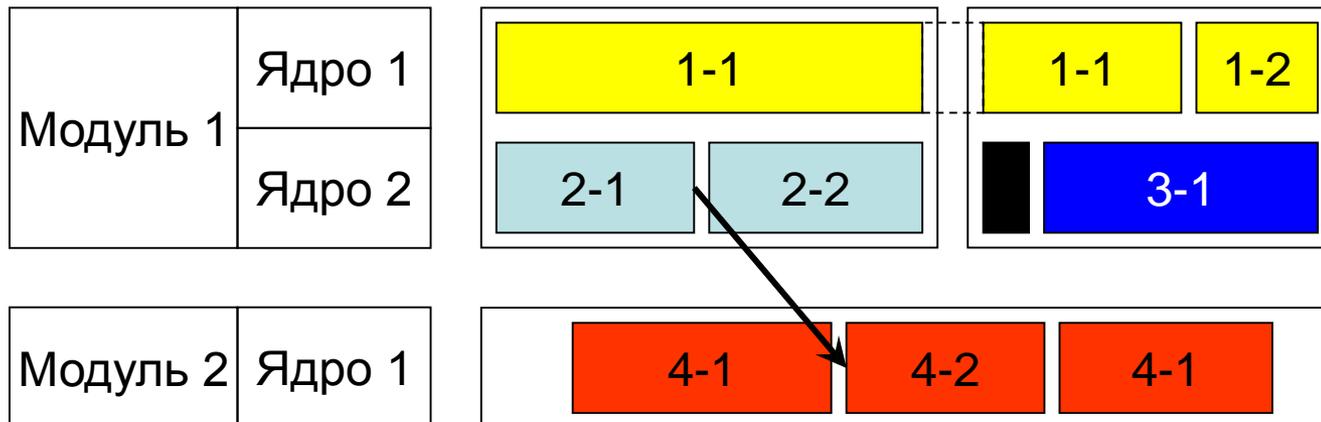


Состояния задач в ARINC653



Выполнение задач в системе

- Задачи раздела выполняются в рамках окон
- Между разделами происходит переключение контекста
- Работы выполняются в соответствии с очередью выполнения и приоритетом
- Работы выполняются с вытеснением
- Незавершенная работа может быть возобновлена в следующем окне



Взаимодействие между разделами

- Порты с очередью сообщений
- Порты с перезаписью сообщений
 - Буфер фиксированной длины
 - Сообщение в буфере перезаписывается
 - Отправка с заданным периодом

Взаимодействие внутри разделов

- Обмен данными
 - Передача сообщений
 - Общая память
- Механизмы синхронизации
 - Семафоры
 - События

Конфигурация системы

- Информация о разделах
 - Требования к памяти
 - Порты раздела
- Информация о каналах
- Расписания окон
 - Список окон
 - Привязка окон к разделам
- Информация по обработке ошибок

Исходные данные

- Вычислительные модули: $M=\{m_i\}$
- Разделы: $P=\{p_i\}$
- Задачи: $W=\{w_i\}$
 - Частота: $f(w_i)$
 - Длительность: $d_m(w_i)$
 - Приоритет: $pr(w_i)$
- Сообщения: $T=\{tr_{ij}\}$
 - Длительность (канал): $t^C(w_i, w_j)$
 - Длительность (память): $t^M(w_i, w_j)$

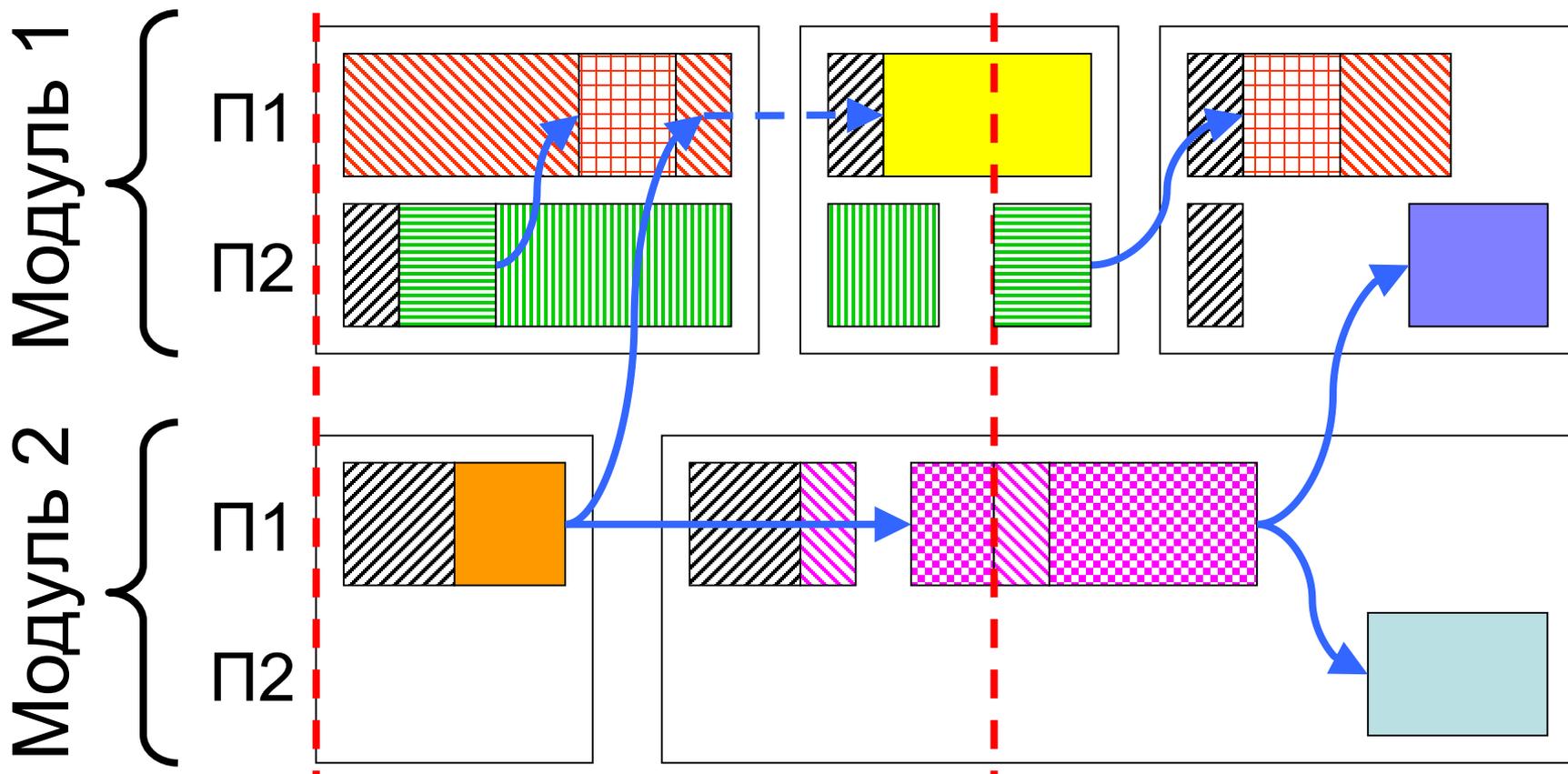
Конфигурация системы

- Расписание окон: $S = \{s_i\}$
- Привязка разделов к окнам:
 $D^S: P \rightarrow \{S^P \mid S^P \subset S\}$
- Привязка разделов к процессорам:
 $D^M(p_i)$
- Приоритет задач в рамках каждого раздела: $pr^S(w_i)$

Критерии оптимальности

- Количество размещенных работ
- Минимизация нагрузки на среду обмена (при условии размещения всех работ)

Пример расписания



Ограничения корректности (1)

- Размещение всех работ
- Ограничение на загрузку ядер
- Ограничение реального времени:
 - Интервал выполнения работы не выходит за пределы ее директивного интервала

Ограничения корректности (2)

- Раздел привязан ровно к одному процессору
- Не более одного раздела в окне
- Окна не пересекаются по времени
- Работы выполняются в рамках окон раздела
- В каждый момент времени выполняется не более одной работы
- Все работы выполняются полностью
- Выполняются зависимости по данным
- Соблюдаются приоритеты

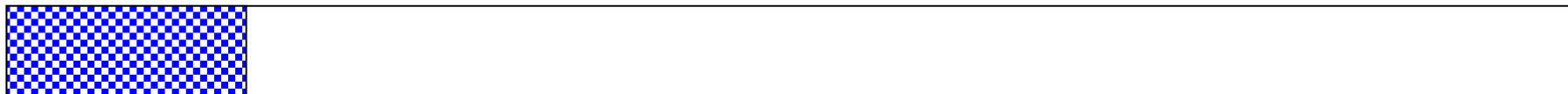
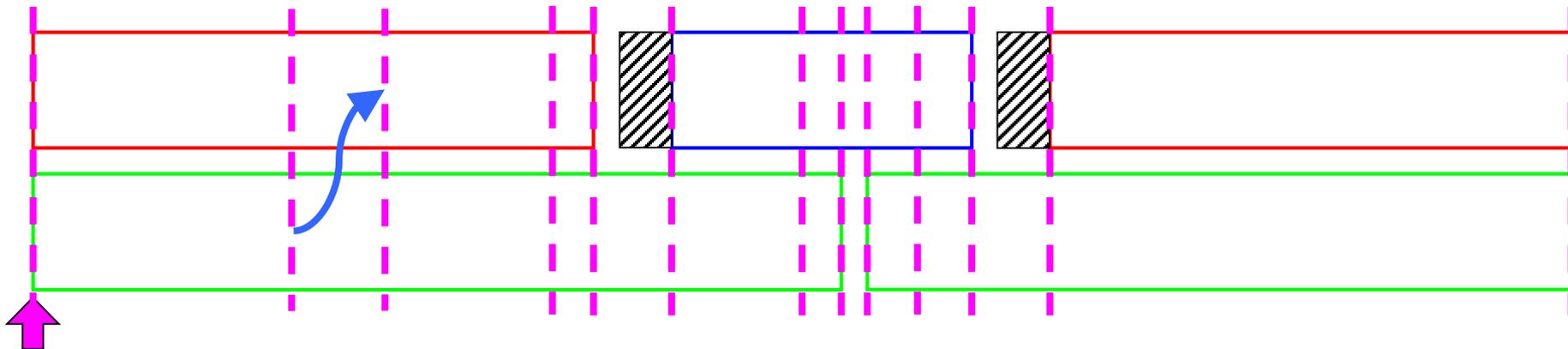
Формальная модель

- Модель основана на событиях
- Алгоритм работы:
 1. Создание начальных событий
 2. Выбор событий с минимальным временем
 3. Обновление списка готовых к выполнению работ
 4. Обработка выбранных событий
 5. Если список событий не пуст, переход к п.2

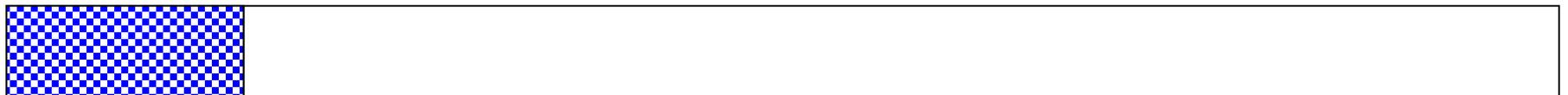
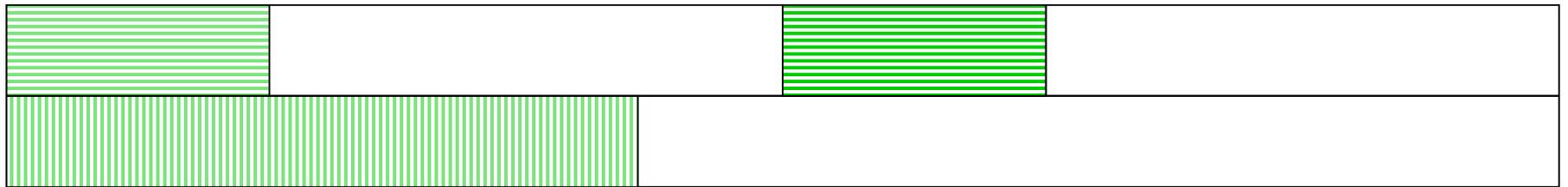
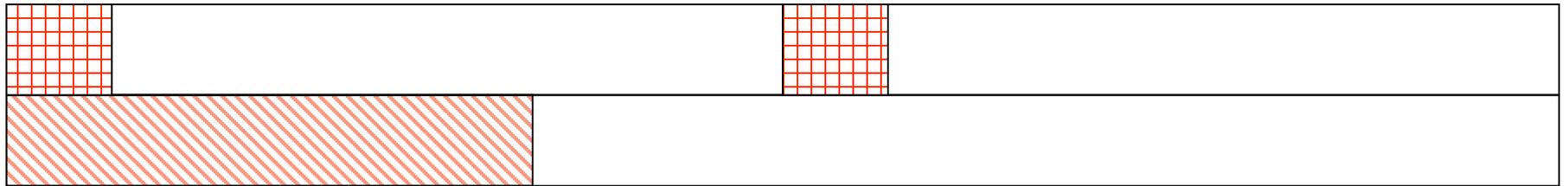
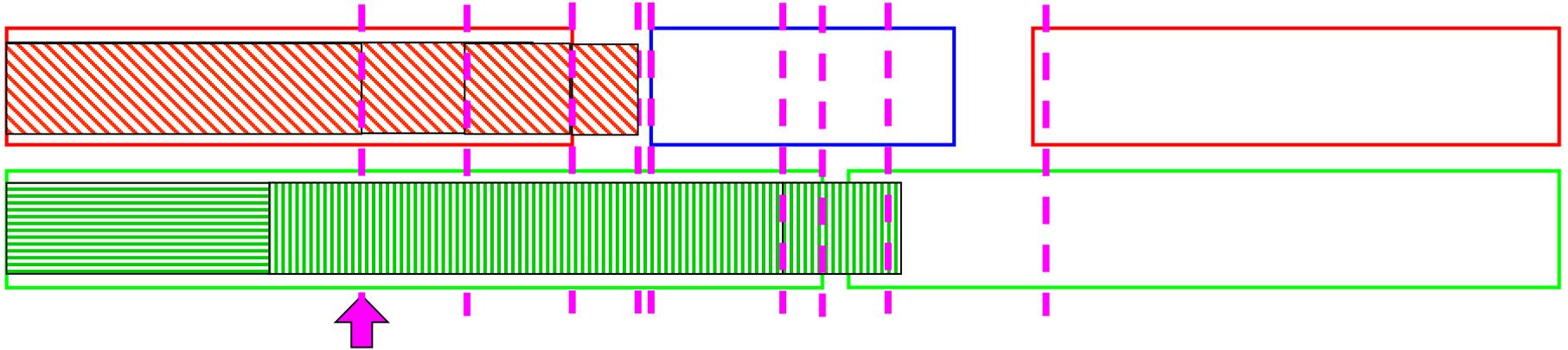
Типы событий

- Открытие окна
- Закрытие окна
- Начало директивного интервала
- Завершение работы
- Доставка сообщения

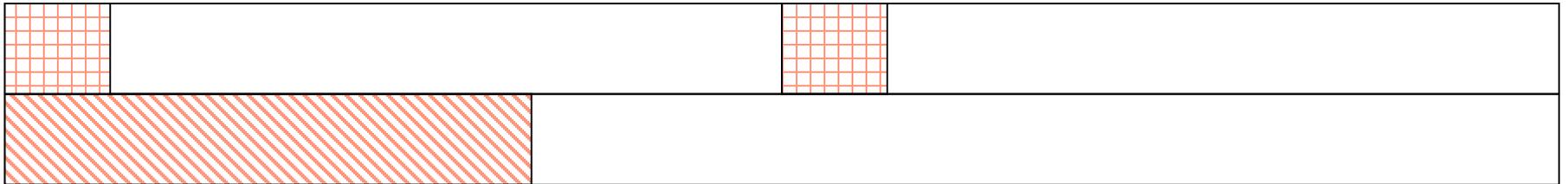
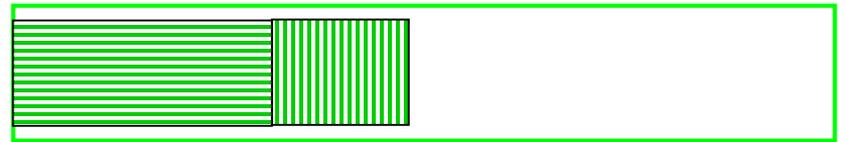
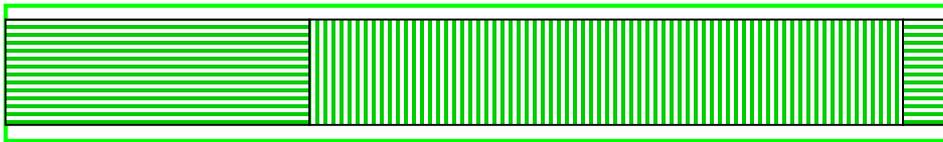
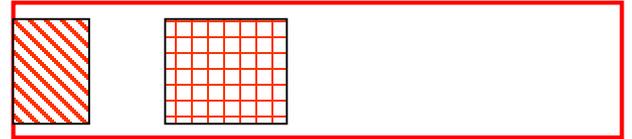
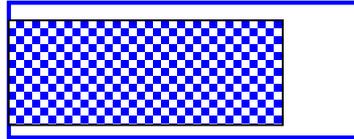
Работа модели



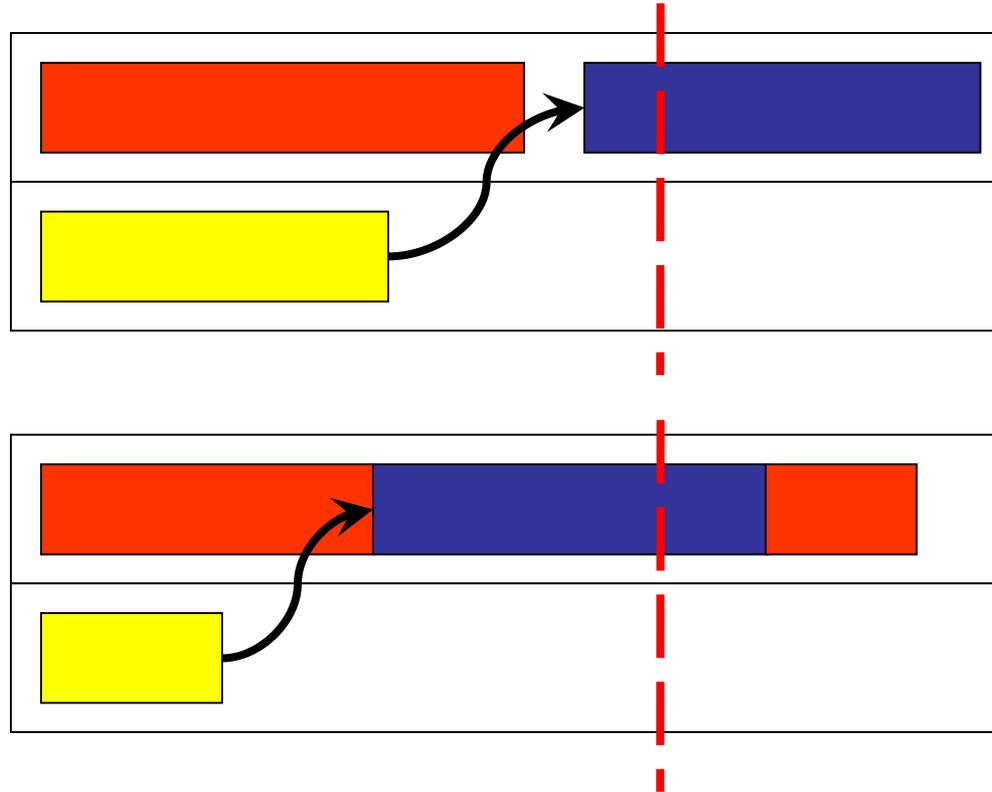
Работа модели



Работа модели



Недостатки модели



Решение:

не выполнять низкоприоритетные задачи, пока есть высокоприоритетные, ожидающие сообщений

Подход к решению задачи

Задача разбивается на две подзадачи:

- Построение привязки разделов к ядрам
 - Минимизация нагрузки на среду обмена
 - Выполнение критериев планируемости набора работ
 - Учет длительности критических путей в графе зависимостей
 - Критерии на основе загрузки:

$$\sum_{i=0}^n T_i F_i \leq n \left(2^{\frac{1}{n}} - 1 \right)$$

- Соблюдение заданного ограничения на максимальную загрузку ядра
- Построение расписания окон и приоритетов

Адаптация к изменениям

