Проектирование бортовой ОСРВ с жёстким реальным временем для космического применения

В.Ю. Чепцов <cheptsov@ispras.ru>
А.В. Хорошилов <khoroshilov@ispras.ru>

ИСП РАН

Проектирование системного встраиваемого ПО

Unix подобное
- POSIX (1988) — ОС общего назначения
- POSIX Realtime Extension (1997) — POSIX-совместимая ОСРВ
- POSIX с Vendor-specific решениями (QNX, Wind River)

Локальное в индустрии
- ARINC 653 (1997) — ОСРВ для бортовой авионики
- AUTOSAR (2005) — ОСРВ для автомобильной промышленности
- GlobalPlatform TEE (2010) — среда для доверенных вычислений

Локальное в компании
- Без операционной системы
- ОС с проприетарным интерфейсом

ИСП РАН

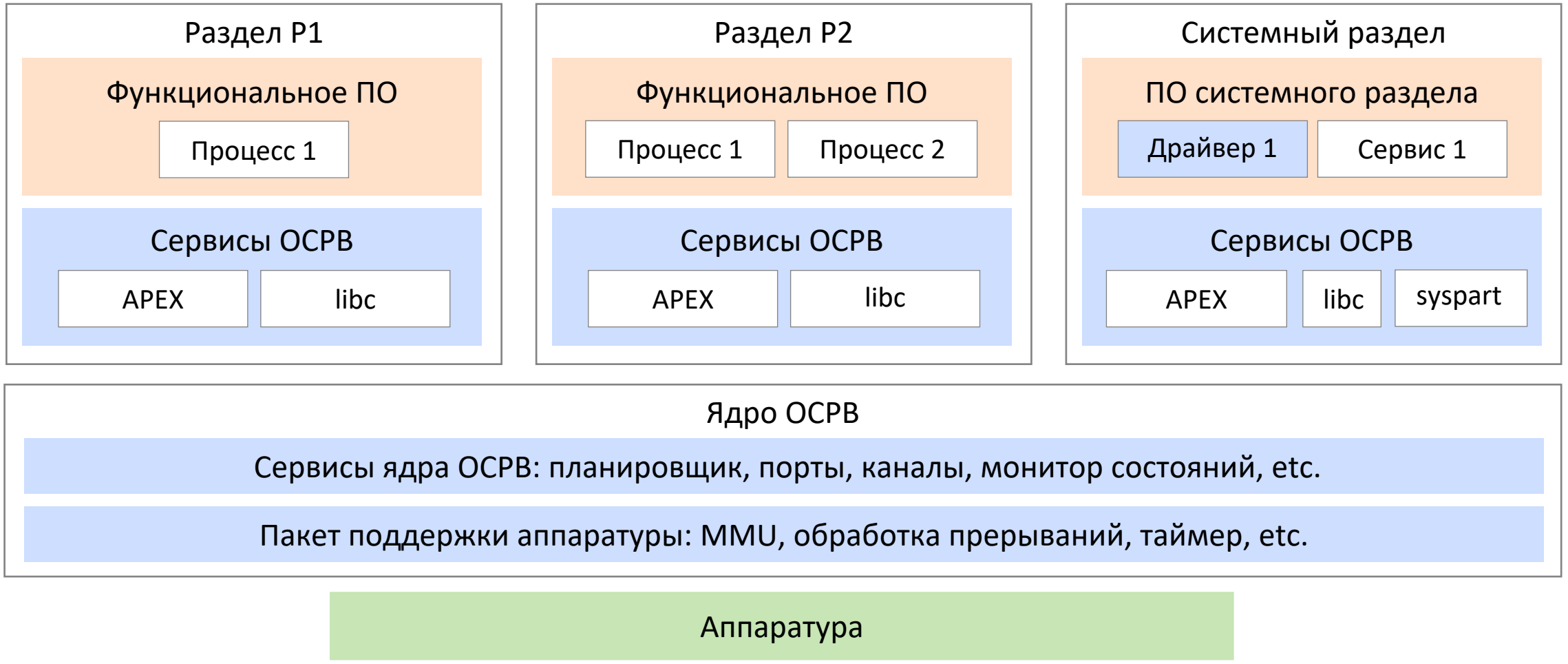
Плюсы стандартизованного системного ПО

- Переносимость функционального ПО между устройствами разных организаций.
- Единый подход по проектированию ПО.
- Возможность адаптаци большого количества инструментов, включая инструменты безопасной разработки.
- Унифицированные механизмы взаимодействия с другим функциональным ПО.
- Упрощение процессов подготовки кадров.

ARINC 653 в космосе

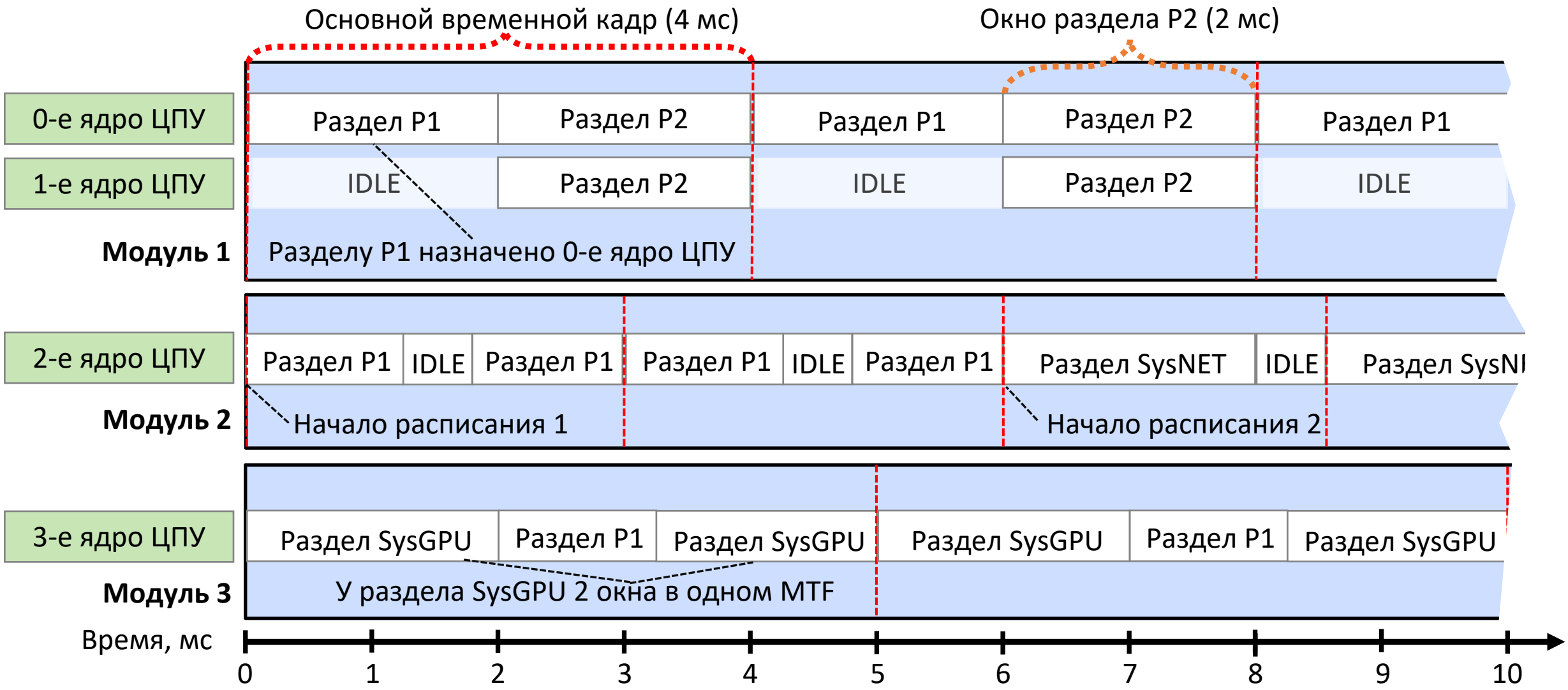
- LithOS — ОСРВ с поддержкой ARINC 653, разработанная в Валенсийском политехническом университете на базе гипервизора XtratuM в 2010 году. Используется в космических проектах с участием fentISS.
  - Спутник ANGELS (2019, CNES)
  - Спутник EyeSat 3U (2019–2020, CNES)
- AIR — ОСРВ с поддержкой ARINC 653, разработанная в Лиссабонском университете на базе RTEMS в 2007 году. Используется в проектах GMV.
  - Точный список проектов неизвестен.
  - Предположительно легла в основу новой ОСРВ XKY.

Структура ОСРВ на базе ARINC 653



ИСП РАН

Расписание ARINC 653



ИСП РАН

Ключевые особенности ARINC 653

- Функциональное ПО изолировано по времени и пространству.
- Расписания распределяют время по окнам между разделами в рамках модуля, которые повторяются каждый основной временной кадр.
- Выделенные разделам ресурсы заложены статически на уровне конфигурации.
- Единая схема взаимодействия между функциональным ПО через порты как внутри модуля, так и по бортовой сети.
- Работа с оборудованием может выполняться с пониженными привилегиями в системных разделах.

Различия вычислительных систем спутника и самолёта

- Аппаратные возможности космических вычислителей на порядки меньше:
  SPARC LEON3 50 МГц и десятки МБ ОЗУ **vs** PowerPC P3041 1000 МГц и 2 ГБ ОЗУ
- Альтернативные протоколы связи:
  SpaceWire, SpaceFibre, MIL-STD-1553 **vs** AFDX, ARINC 429, CAN
- Разная организация миссии по длительности и обслуживанию:
  Годы автономной работы без физического доступа **vs** Регулярное обслуживание
- Разная структура рисков для обеспечения надёжности основной части полёта:
  Длительная потеря управляемости не критична **vs** Немедленная реакция на сбой
  ↪ На отдельных фазах полёта (e.g. посадка), структура рисков сохраняется.

Новые вызовы для ARINC 653

1. Необходимость удалённого сервисного обслуживания (обновления).
2. Необходимость работы при сбоях вычислителей из-за агрессивной среды.
3. Необходимость не только холодного, но и горячего резервирования.
4. Необходимость синхронизации времени в бортовой сети.
5. Необходимость обработки задач в условиях недостатка аппаратных ресурсов.

Сервисное обслуживание (1/2)

Удалённое обновление необходимо для:

- Корректирования кода ПО для выполнения новых задач и исправления ошибок.
- Переконфигурации аппарата в условиях отказа оборудования.

Формат обновления может быть:

- «Холодным» — обновление применяется после перезагрузки вычислителя, что в целом вписывается в ARINC 615 и 653, но несёт риски потери вычислителя.
- «Горячим» — обновление применяется сразу, например, в конце основного временного кадра. Данные обновления могут не сохраняться в ПЗУ для снижения рисков отказа ПЗУ или возврата к рабочей версии ПО.

Сервисное обслуживание (2/2)

Для «горячего» обновления системному разделу требуются новые привилегии и сервисы для изменения обновляемого раздела:

1. Резервирование ресурсов под обновление при сборке проекта.
2. Модификация состояния:
   - Множественные расписания (ARINC 653 P2).
   - Ручная остановка/возобновление раздела (AIR, LithOS).
3. Модификация памяти:
   - Общая память через блоки памяти (ARINC 653 P2) или иные сервисы (AIR).
   - Механизмы ядра с обновлениями регионов памяти.
4. Согласование кэшей процессора (AIR).

Устойчивость к сбоям (1/2)

Внешние факторы приводят к «выбиванию» ячеек ОЗУ, инверсии условных ветвлений и некорректным вычислениям.

Доработка известных подходов

- Наличие программного BITE:
  - Проверка целостности объектов.
  - Периодическая проверка корректности вычислений.
- Наличие защитного кода с целью защиты от аппаратных сбоев.

Реализация нетиповых подходов

- Резервирование не только самих вычислителей, но и хранимых на них объектов с продолжением работы при повреждении.
- Отсутствие доверия к одному самостоятельному вычислителю.

ИСП РАН

Устойчивость к сбоям (2/2)

Типовые способы обеспечения доверия к вычислениям (NASA):

Железо
1. Аппаратные способы повышения надёжности (ECC, радстойкость).

ФПО
2. Повторение производимых вычислений несколько раз.
3. Использование эталонных и граничных значений.

ОС+ФПО
4. Сравнение вычислений между независимыми вычислителями.
5. Применение голосования для выбора верно работающего вычислителя.

Для пунктов 4 и 5 требуется нетиповая для ARINC модель бортовой сети, в которой существует N однотипных вычислителей, выполняющих одну задачу.

Синхронизация вычислений

В ARINC 653 разделы запускаются независимо, поэтому для обеспечения возможности сравнения вычислений необходимо:

- Синхронизировать время на вычислителях.
- Синхронизировать основные временные кадры на вычислителях.
- По возможности предоставить API для обмена сообщений сложнее, чем типовые порты ARINC 653.

Часы в железе даже в одинаковых вычислителях работают с разной скоростью с завода, и это только усугубляется внешними факторами условий эксплуатации.

Недостаток ресурсов

Возможностей аппаратуры потенциально недостаточно для честного разделения задач на окна в рамках расписания. Потенциальные решения:

- Группировка задач в рамках одного раздела для минимизации времени переключения адресных пространств и уменьшения потребления памяти.
- Приоритизация задач на задачи с жёстким реальным временем и мягким реальным временем. Использование FIFO очередей для последних.
- Использование пула потоков с перепланированием задач через REPLENISH и контролем временных гарантий через обработчик ошибок для сокращения потребляемой памяти стеками потоков.

Выводы

- Построение бортовой ОСРВ для космического применения на основе существующих стандартов бортового ПО вроде ARINC 653 разумно, так как даёт преимущества по унификации ПО, изоляции и масштабируемости.
- Переход от парадигмы бортовой сети из разных вычислителей к парадигме бортовой сети из разных функциональных групп вычислителей формирует новые требования, которые ARINC 653 не покрывает.
- Необходимость горячего обновления ПО вычислителей работающих в группе с последующим вводом в вычислительный цикл— нетривиальная задача.

Спасибо за внимание!

ИСП РАН