Микроядерная сертифицируемая операционная система реального времени

Николай Пакулин, ИСП РАН
npak@ispras.ru

JetOS кратко

- JetOS – операционная система для бортового оборудования гражданской авиации
- JetOS начинался как клон проекта POK (pok.tuxfamily.org)
  - Переработано более 80% кода
- Поддерживаются PowerPC, ARM, MIPS, x86
- Язык C99 с ограничениями
- Участники проекта: ГосНИИАС, ИСП РАН, ДС Барс, DZ Systems

Особенности проекта JetOS

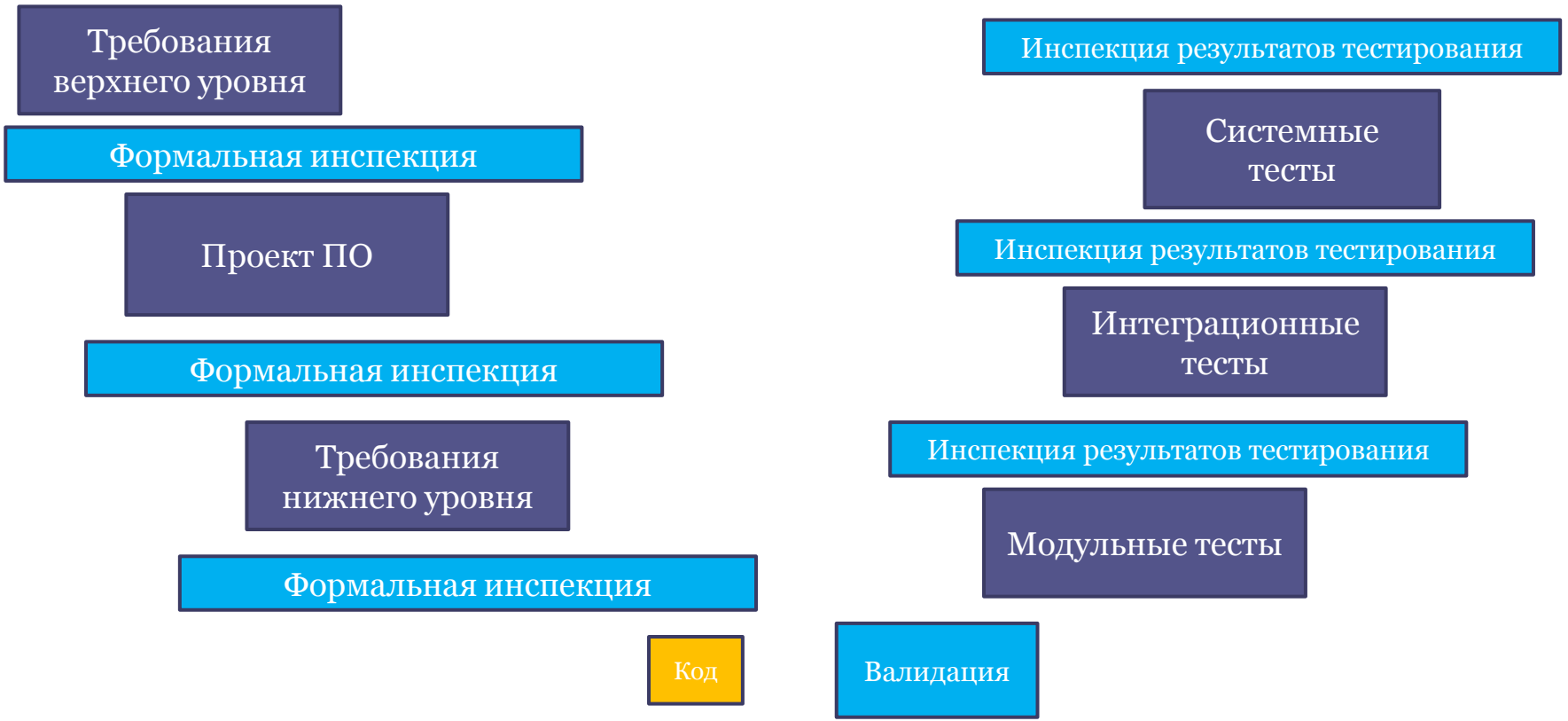
Жесткое реальное время

- Статическое разделение времени и ресурсов
- Гарантированное время реакции
  - Real time vs. Real fast
- Специализированное API
  - ARINC 653
- Отказоустойчивость
- Специальная аппаратура
  - PowerPC / MIPS / *ARM(?)*

Сертифицируемость

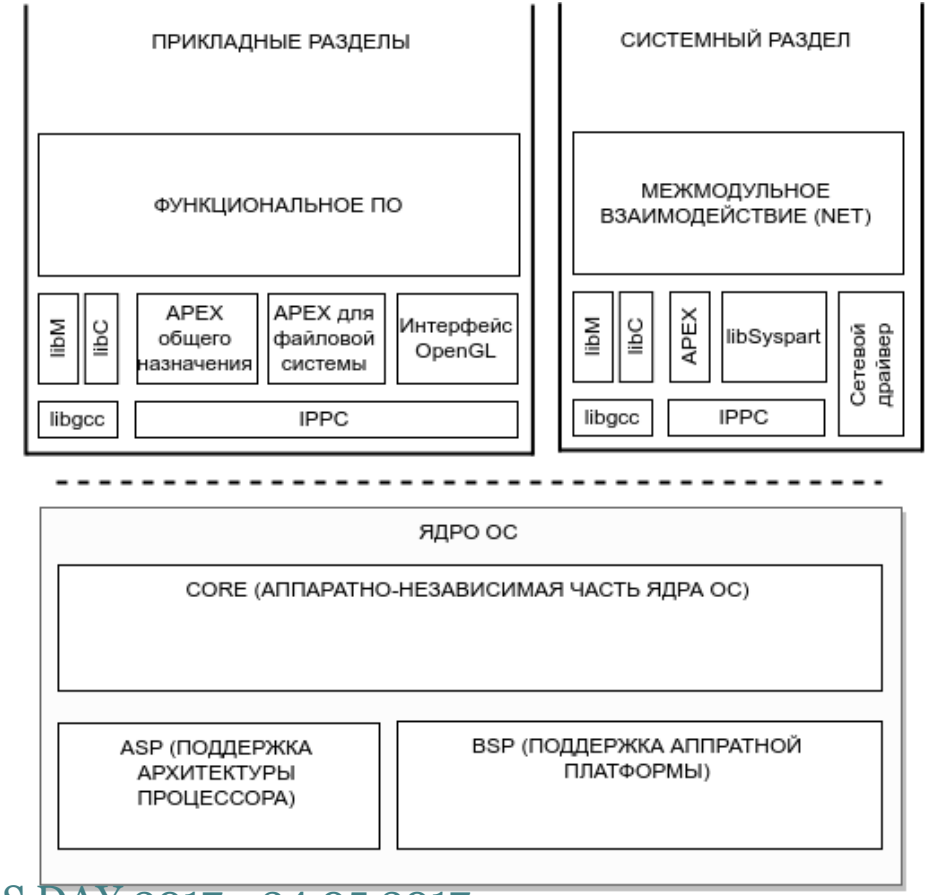
- Жесткий цикл разработки
  - Кодирование < 10%
- Все изменения через «Запрос на изменение»
- Крупные модули – масштабный и сложный процесс изменений
- Квалифицированные инструменты

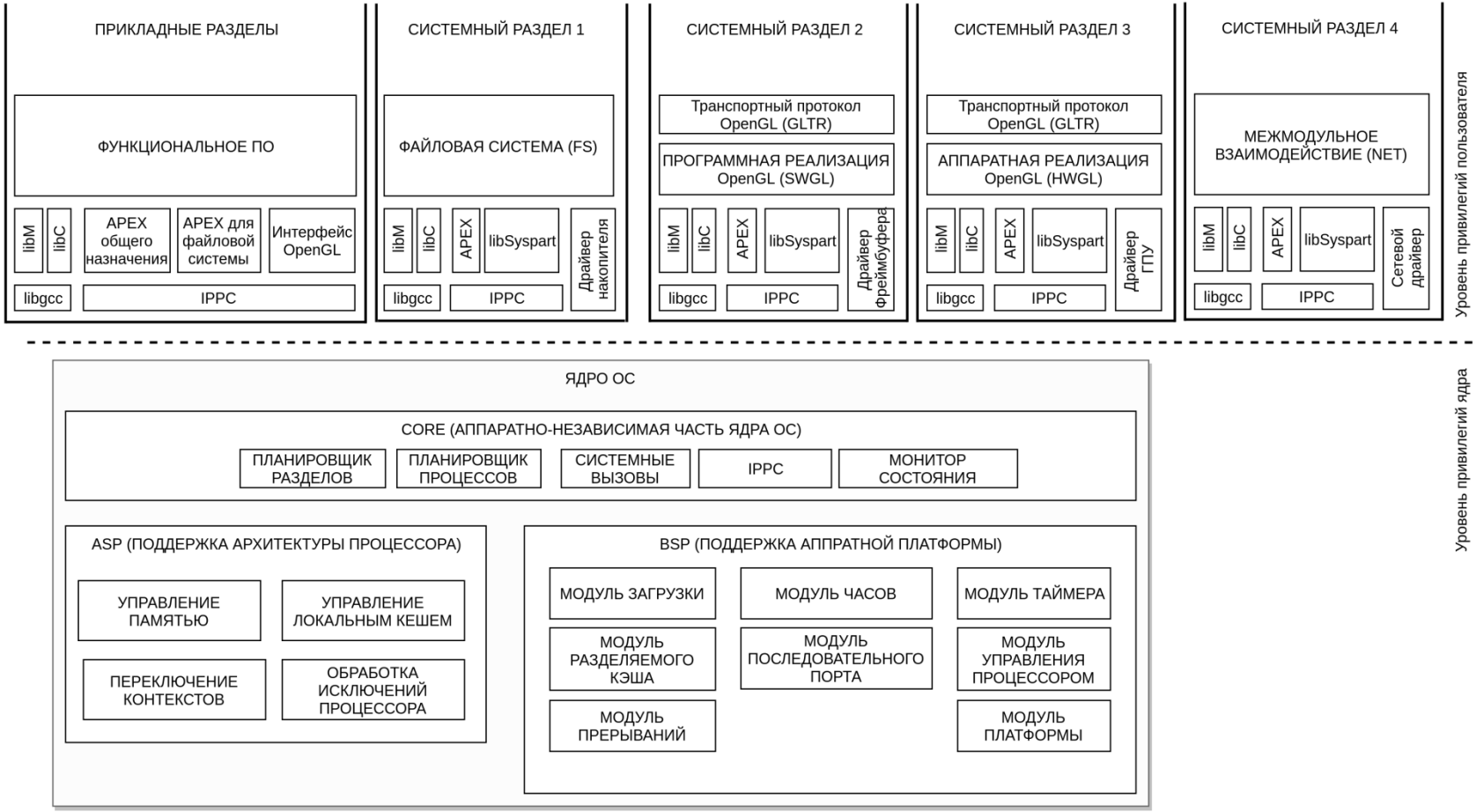
Сертифицируемость …



Архитектура

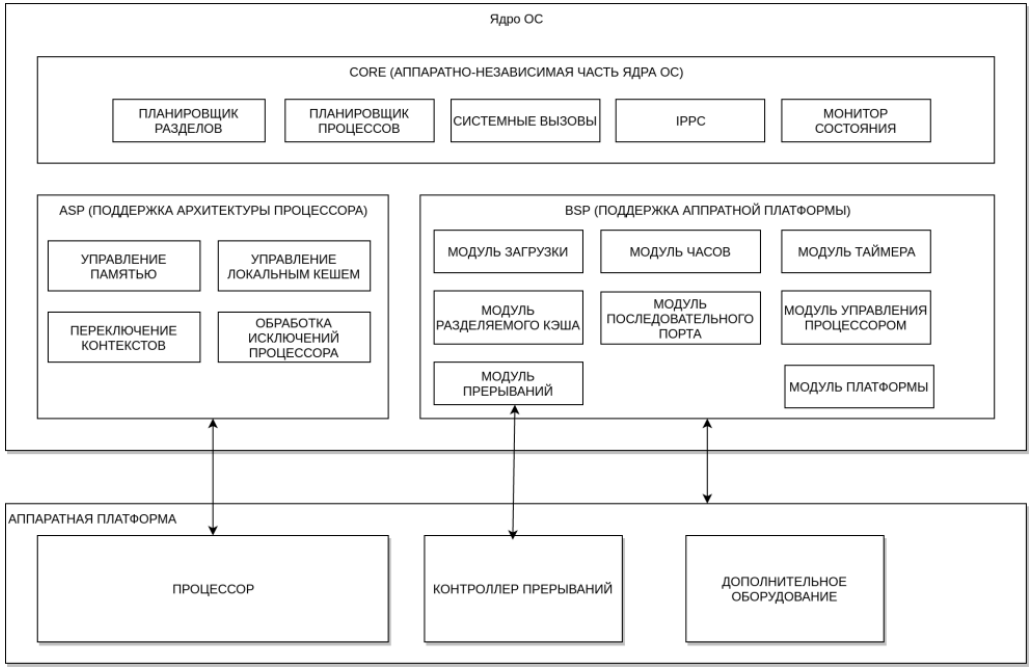
- Минимальное ядро
- Разделы
  - Адресные пространства
- Статическая аллокация ресурсов
- Драйверы вне ядра
  - В том числе сетевой стек
- Взаимодействие между разделами
  - Interpartition process call




ПРИКЛАДНЫЕ РАЗДЕЛЫ
ФУНКЦИОНАЛЬНОЕ ПО
libM
libC
APEX общего назначения
APEX для файловой системы
Интерфейс OpenGL
libgcc
IPPC
СИСТЕМНЫЙ РАЗДЕЛ 1
ФАЙЛОВАЯ СИСТЕМА (FS)
libM
libC
APEX
libSyspart
Драйвер накопителя
libgcc
IPPC
СИСТЕМНЫЙ РАЗДЕЛ 2
Транспортный протокол OpenGL (GLTR)
ПРОГРАММНАЯ РЕАЛИЗАЦИЯ OpenGL (SWGL)
libM
libC
APEX
libSyspart
Драйвер фреймбуфера
libgcc
IPPC
СИСТЕМНЫЙ РАЗДЕЛ 3
Транспортный протокол OpenGL (GLTR)
АППАРАТНАЯ РЕАЛИЗАЦИЯ OpenGL (HWGL)
libM
libC
APEX
libSyspart
Драйвер ГПУ
libgcc
IPPC
СИСТЕМНЫЙ РАЗДЕЛ 4
МЕЖМОДУЛЬНОЕ ВЗАИМОДЕЙСТВИЕ (NET)
libM
libC
APEX
libSyspart
Сетевой драйвер
libgcc
IPPC
Уровень привилегий пользователя
Уровень привилегий ядра
ЯДРО ОС
CORE (АППАРАТНО-НЕЗАВИСИМАЯ ЧАСТЬ ЯДРА ОС)
ПЛАНИРОВЩИК РАЗДЕЛОВ
ПЛАНИРОВЩИК ПРОЦЕССОВ
СИСТЕМНЫЕ ВЫЗОВЫ
IPPC
МОНИТОР СОСТОЯНИЯ
ASP (ПОДДЕРЖКА АРХИТЕКТУРЫ ПРОЦЕССОРА)
УПРАВЛЕНИЕ ПАМЯТЬЮ
УПРАВЛЕНИЕ ЛОКАЛЬНЫМ КЕШЕМ
ПЕРЕКЛЮЧЕНИЕ КОНТЕКСТОВ
ОБРАБОТКА ИСКЛЮЧЕНИЙ ПРОЦЕССОРА
BSP (ПОДДЕРЖКА АППРАТНОЙ ПЛАТФОРМЫ)
МОДУЛЬ ЗАГРУЗКИ
МОДУЛЬ ЧАСОВ
МОДУЛЬ ТАЙМЕРА
МОДУЛЬ РАЗДЕЛЯЕМОГО КЭША
МОДУЛЬ ПОСЛЕДОВАТЕЛЬНОГО ПОРТА
МОДУЛЬ УПРАВЛЕНИЯ ПРОЦЕССОРОМ
МОДУЛЬ ПРЕРЫВАНИЙ
МОДУЛЬ ПЛАТФОРМЫ

Микроядро

- Универсальная часть
  - Прерывания
    - Только таймер и watchdog
  - Планировщик ARINC 653
  - IPPC
- Поддержка платформы
  - Загрузчик
  - «Материнка» – часы, кэш L3, контроллер прерываний
- Поддержка процессора
  - MMU, кэш
  - Переключение контекстов
  - Приём прерываний



«Кибербезопасность»

С учетом особенностей ARINC-653

Пример для понимания

POSIX

```
int sock;
sock = socket(AF_INET,
  SOCK_STREAM, 0);
connect(sock, &serv_addr,
  sizeof(serv_addr));
n = write(sock, msg,
  msg_size);;
/* Произвольный адрес
  назначения
  Потенциально –
  взаимодействие со всем
  миром */
```

ARINC 653

```
SAMPLING_PORT_ID_TYPE prt;
CREATE_QUEUING_PORT("QP1",
  64, 10, SOURCE, FIFO,
  &prt, &ret);
SEND_QUEUING_MESSAGE(prt,
  &msg, msg_size, 0, &ret);
/* НЕТ адреса назначения
   Статическая конфигурация
*/
```

ARINC 653 и безопасность

- Строго регламентирован набор механизмов взаимодействия
- Статически задаются endpoints в конфигурации
- Безопасность by construction

За скобками проекта

- Доверенная загрузка
- Обновление системы
- Поддержка криптографических средств

Заключение

Текущий статус

- Реализовано:
  - Микроядро для одноядерного режима процессора
  - стандарт ARINC 653 часть 1 (rev. 3)
  - libc / C99 (подмножество)
  - поддержка PowerPC, MIPS, ARM
  - Удаленный отладчик
- Разрабатывается
  - реализация стандарта ARINC 653 часть 1 (rev. 4) – многоядерный режим
  - реализация стандарта ARINC 653 часть 2
  - **Сертификационный набор**

Творческие планы

- Реализация POSIX (подмножество)
- Реализация OpenGL с аппаратным ускорением
- Реализация файловой системы
- Обновление системы (Dataloader)
- Квалифицированные инструменты

СПАСИБО!