

# ARINC 653 и языки программирования

## Метод адаптации ARINC 653 - совместимой ОСРВ для ПО, написанного на разных языках программирования

Pavel Isachenko

MIPT

7 мая 2024 г.

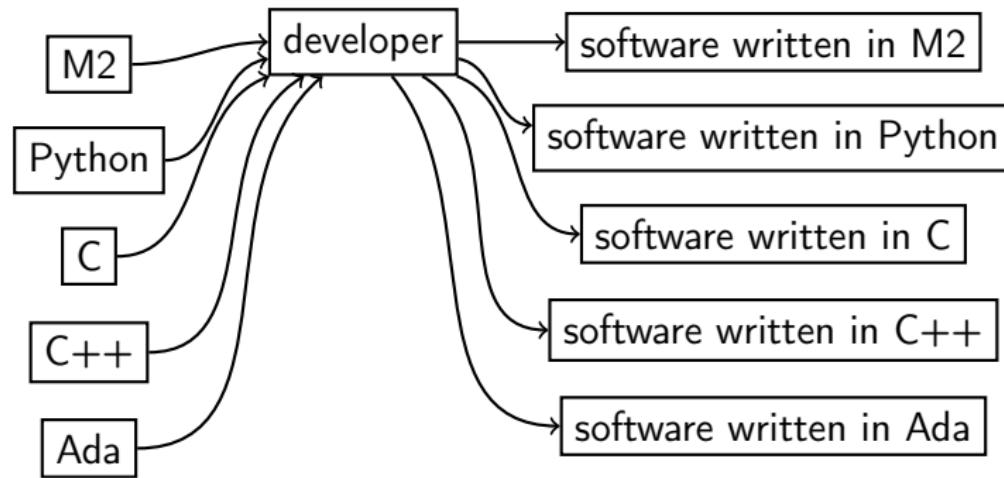
## Определения

Операционная система реального времени (OCPB) — операционная система, способная обеспечить требуемый уровень сервиса в определённый промежуток времени. Бортовые OCPB используются для управления бортовым оборудованием.

ARINC 653 — стандарт на интерфейс прикладного программного обеспечения для применения в авионике.

ARINC 653 определяет APEX interface (API = Application Program Interface) между основным программным обеспечением компьютера авионики и прикладным программным обеспечением.

# Мотивация

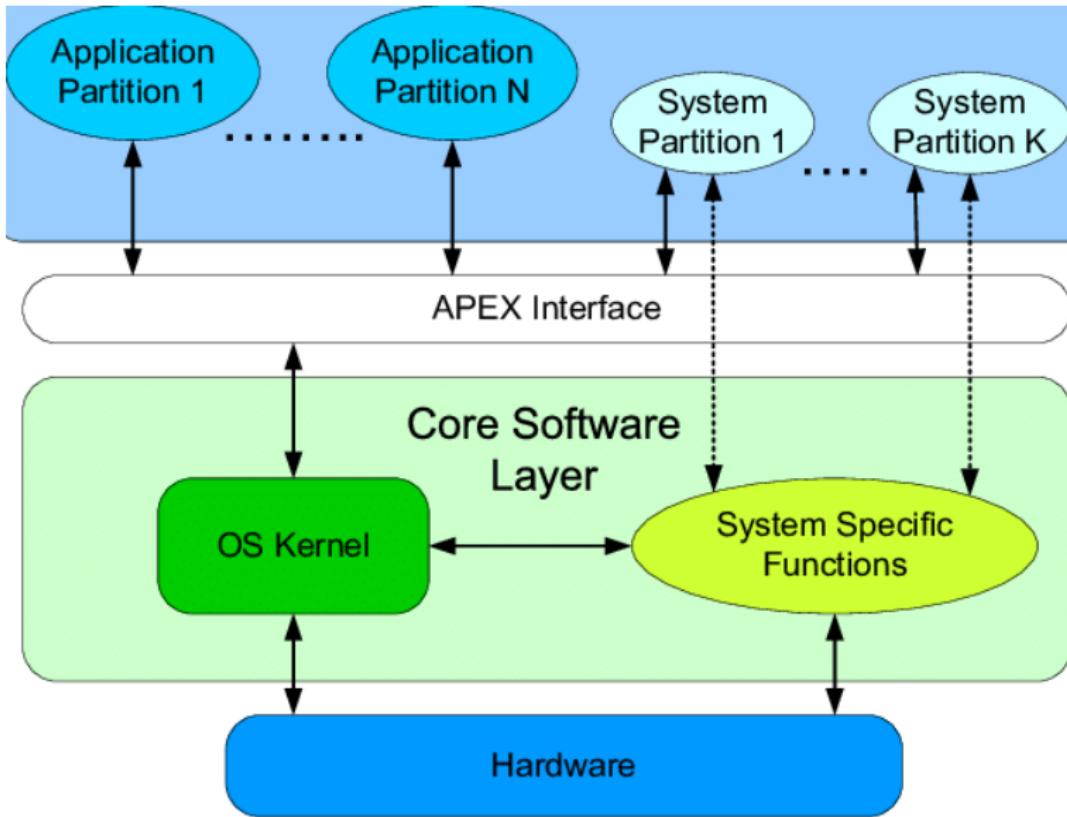


Возникает необходимость переносить уже написанное программное обеспечение

## Цель и задачи

Разработать метод адаптации компилируемого языка  
программирования для разработки функционального ПО в ARINC 653  
совместимых ОСРВ

- ① Анализ требований к окружению и программным интерфейсам со стороны ARINC 653
- ② Проектирование метода добавления поддержки компилируемого языка в ARINC 653 совместимую ОС
- ③ Разработка прототипа на языке Ada в соответствии с методом



# Анализ: Что дает нам стандарт

Набор интерфейсов для:

- ① Многопоточность
- ② Межпроцессный обмен данными
- ③ Обработка ошибок

# Проектирование: Задачи общие для любого ЯП

В результате анализа можно выделить следующие шаги:

- ① Описать и реализовать APEX interface на ЯП
- ② Создать специфичный тулчейн для ЯП
- ③ Адаптировать особенности ЯП к APEX окружению
- ④ Адаптировать runtime ЯП

## Пример

ARINC 653 Р2 дает нам определение: The types FILE\_ID\_TYPE and DIRECTORY\_ID\_TYPE are used to define identifiers for open files and directories. Identifiers for open files and directories remain open until closed or a partition restart occurs.... Мы же в свою очередь должны реализовать это определение на выбранном нами ЯП.

```
type FILE_MODE_TYPE is (READ, READ_WRITE);
type MESSAGE_SIZE_TYPE is a numeric type; -- number of bytes
type FILE_ERRNO_TYPE is numeric type;
type FILE_NAME_TYPE is a n-character string;
type FILE_ID_TYPE is numeric type;
type DIRECTORY_ID_TYPE is numeric type;
```

## Пример

В соответствии с ARINC 653 ресурсы выделяются только на стадии инициализации, после не освобождаются

# Проектирование: О тулчайне

В простом случае это компилятор и линковщик, приспособленные для создания исполняемых файлов под отличную архитектуру. При выборе компилятора стоит смотреть на версии с открытым исходным кодом, ввиду возможных модификаций.

# Проектирование: Почему Ada

- ❶ ГЛАВНОЕ: APEX interface описан в спецификации ARINC 653, остается его реализовать

## Реализация: О тулчайне с поддержкой Ada

Для Ada есть множество копмиляторов, как наиболее хорошо поддерживаемый был выбран GNAT для апробации метода. GNAT — свободный компилятор языка Ада, является частью GNU Compiler Collection. GCC собирается с флагами `--disable-libada --enable-languages=ada`

## Реализация: О libada

Библиотека времени выполнения Ada отвечает за реализацию стандартной библиотеки. Собирается для каждой платформы и соответственно архитектуры отдельно, содержит различные настройки к примеру: разрешены ли исключения, рекурсия, нужно ли проводить проверки стека. Компилируемая программа статически связывается с библиотекой времени выполнения для создания окончательного исполняемого файла.

# Результаты

- ① Собран тулчейн с поддержкой Ada на базе GNAT.
- ② Реализована базовая Ada Runtime Library для powerpc (прототип), stm32f429 (на базе runtime AdaCore).
- ③ Адаптированы основные особенности сборки проектов на Ada 95 под ОС, т. е. можно писать проекты с некоторыми ограничениями

# Выводы

- ➊ Разработан метод адаптации компилируемого языка программирования для разработки функционального ПО в ARINC 653 совместимых ОСРВ