

Вопросы разработки и отладки многораздельных операционных систем для ARM архитектуры микроконтроллеров STM32

С.М. Старолетов serg_soft@mail.ru

АлмГТУ им. Ползунова / Институт Автоматики СО РАН

Написание системного кода, а особенно кода операционных систем сопряжено с трудностями прежде всего в том, что приходится работать с оборудованием по абсолютным адресам и регистрами процессора, что требует наличия оригинальной документации на процессор и периферийные устройства и даже при ее наличии существует множество скрытых моментов, которые значительно увеличивают время разработки приносят ошибки и уязвимости. Поэтому целесообразнее развиваться в направлении изучения и переписывания открытого программного кода, к которому имеются примеры и который зарекомендовал себя в сообществах разработчиков. Перспективным направлением операционных систем являются так называемые многораздельные (системы с разделами — партициями), которые разделяют процессорное время и адресное пространство между системными программами, которые к тому же могут быть отдельно настроены для выполнения только специфицированных заранее системных вызовов, выполняя какую-то специфическую работу внутри разделов в своих потоках исполнения. Примером концепта такой ОС является РОК, представленный в диссертации Жюльеном Деланжем [1]. При наличии работающего прототипа системы, который исправно планирует разделы и потоки в них, а также системных вызовов через программное прерывание можно навесить клиентский интерфейс и сделать такую системы POSIX- или ARINC-совместимой для применения в какой-либо ответственной области.

Изучая исходные коды оригинального проекта [2], было установлено, что в целом он является ориентированным на x86-архитектуру (имеются также архитектурные части и для ppc и spark) и работает с устройствами, доступными в эмуляторе QEMU. Портирование для ARM такого проекта с учебными целями сопряжено с трудностями прежде всего в том, что имеется бесчисленное множество устройств с различными конфигурациями ARM-ядер и архитектура доступных решений для устройств массового применения в целом закрыта).

В это же время, в среде разработчиков микроконтроллеров для сферы Интернета вещей и DIY решений популярны STM32 семейства плат, которые имеют лицензированные ARM ядра (Cortex M3 или M7 и даже их связки M7/M4), но тут главное прежде всего то, что компанией ST поставляется открытый код на C, адаптированный для каждого конкретного микроконтроллера, содержащий:

- BSP код инициализации устройства.

- Код для работы с контроллером прерываний NVIC.
- Код конфигурации часов/таймингов для всех подсистем на плате.
- Код для работы с MPU (модуля защиты памяти Cortex M7).
- HAL библиотеки для работы с GPIO/UART/I2C/SPI/CAN.
- Набор структур данных и функций для работы с USB Host/Device.
- Код для работы с различными LCD экранами.
- А также исходный код FreeRTOS, что снимает вопросы по коду переключений задач.

Имеется также код и примеры работы с различными DSP, а также запуска моделей ИИ на такого рода устройствах.

Конечно, все это ведет к зависимости от производителя ST (в данном случае еще и иностранного и недоступного в настоящий момент набора микросхем), но прекрасно может быть использовано для обучения и взятия лучших практик по подходам разработки под ARM.

В докладе предполагается показать шаги из опыта автора по получению работающего решения многопоточных приложений, работающих с периферией и действующих в стиле ПОК. Ранее была рассмотрена модель такой операционной системы [3], а теперь и ее реализация на основе указанного открытого кода. Предполагается рассмотреть также подход в стиле MDD по графической конфигурации в средстве STM32CubeIDE с генерацией кода (что популярно для многораздельных систем).

Что касается отладки приложений, имеющиеся средства виртуализации для такого рода процессоров и периферии на основе QEMU [4] не являются достаточными, поэтому отладку целесообразнее производить на связке аппаратного отладчика ST-Link и GDB.

В итоге предполагаемый доклад может быть полезен программистам и исследователям, прежде всего с практической точки зрения и как старт для обучения.

Литература

1. Delange J., Lec L. ПОК, an ARINC653-compliant operating system released under the BSD license //13th Real-Time Linux Workshop. – 2011. – Т. 10. – С. 181-192.
2. <https://github.com/pok-kernel/pok>
3. Staroletov S. M. A formal model of a partitioned real-time operating system in Promela (OSDAY 2020) //Труды Института системного программирования РАН. – 2020. – Т. 32. – №. 6. – С. 49-66.
4. https://github.com/beckus/qemu_stm32