

Костюк Д. А., Кутень И. С., Пушкин А. Л., Разумейчик В. С.  
Брест, Беларусь, Брестский государственный технический университет,  
Дизайн-центр электроники Promwad

## **Построение практикумов по программированию встраиваемых систем**

### **Аннотация**

Рассматриваются особенности применения процессоров семейства ARM и системы GNU/Linux для обучения студентов программированию встраиваемых электронных систем. Проанализирована направленность и особенности практических курсов, предлагаемых в БГУ и БрГТУ. Обсуждается последовательность практических заданий. Рассмотрены наборы инструментальных программных средств, включая средства разработки, отладки и виртуализации.

### **Введение**

Актуальность изучения студентами-программистами принципов разработки для встраиваемых систем (embedded systems) на основе свободного ПО показывала в последние годы устойчивый рост, связанный с очередным количественным скачком внедрения интеллектуальных электронных систем на базе универсального микроконтроллерного устройства со специализированной прошивкой. Встраиваемая система не привязана к существующей инфраструктуре прикладных программ, а потому разработчикам проще обосновать выбор свободного набора инструментов системного ПО с его модульностью и масштабируемостью.

Этот рост привёл к появлению учебных курсов по программированию встраиваемых систем на основе свободного ПО в Белорусском государственном университете и в Брестском государственном техническом университете. Оба курса имеют прикладную направленность, однако рассматривают предмет на разных уровнях аппаратной абстракции, что делает интересным их комплексное сравнение.

## Специфика практикумов

Практикум по разработке встраиваемых систем на факультете прикладной математики БГУ построен на базе опыта многолетней разработки подобных устройств в компании Promwad [1]. Вводная часть касается особенностей использования консоли GNU/Linux, базовых утилит, а также git (последнее важно из практических соображений, т. к. системы контроля версий, при их востребованности, присутствуют в учебном процессе весьма редко). При выполнении работ студенты используют язык C и, в ограниченном объеме, shell. По мере освоения необходимого материала уделяется внимание созданию модулей и взаимодействию с подсистемами ядра.

Учебный курс БрГТУ больше нацелен на особенности системы команд ARM и низкоуровневую организацию встраиваемых систем. Среди причин — исходная ориентация на специальность «Промышленная электроника» (с дополнительно адаптированным вариантом для практико-ориентированной магистратуры), а также уже имеющийся отдельный практикум по разработке модулей ядра Linux. Исходно курс предполагает как запуск ассемблерных программ непосредственно на процессоре ARM, так и выполнение программы под управлением ОС, запущенной на ARM-устройстве.

## Выбор аппаратной платформы

При разработке практикумов в качестве платформы были предсказуемо выбраны микроконтроллеры семейства ARM.

В практикуме по разработке встраиваемых систем в БГУ в качестве основной аппаратной платформы планировалось использовать комплекты BeagleBone Black [2] (ARM Cortex A8). Однако в последствии эта идея была отложена в пользу x86. Во-первых, это дало студентам возможность работать самостоятельно, а во-вторых неслож-

ная адаптация заданий практикума сделала исходный код всех работ практически полностью аппаратно-независимым.

Принцип работы с ассемблерной программой в курсе БрГТУ предполагал ее доводку в эмуляторе, а для некоторых работ — тестирование на реальном оборудовании (отладочные платы Atmel на базе ARM9). Однако на практике использование эмулятора также практически вытеснило исходную аппаратную платформу.

## Структура практикумов

Курс БГУ можно разделить на следующие этапы:

- практическое ознакомление со средствами разработки и адаптация к предъявляемым требованиям в ходе написания простейшей консольной программы на C;
- ознакомление с особенностями создания bash-скриптов;
- работа с сокетами, процессами и потоками в C;
- написание серии программ по ядру Linux (программирование char-, sys-, dev-, ргос-устройств, взаимодействие с таймерами, прерываниями и т. д.).

Требования, предъявляемые к выполнению лабораторного практикума (а также к курсовому проектированию по данной дисциплине, и далее — к дипломным проектам, имеющим отношение к компании Promwad) включают сборку с помощью make и оформление кода по Linux kernel coding style [3]. Код в обязательном порядке размещается студентами на каком-либо публичном git-хостинге (например, github.com) и имеет свободную лицензию.

Основа лабораторных работ в курсе БрГТУ — эмулятор QEMU [4], имитирующий одну из нескольких доступных отладочных плат. Для тестирования программ без ОС оттранслированный код располагается по нулевому адресу в файле-образе, соответствующем объёму памяти отладочной платы. Проверка программы в эмуляторе сводится к анализу содержимого регистров в мониторе QEMU, а также в части работ предусмотрен диалог через QEMU serial console.

Практикум включает:

- знакомство с синтаксисом ассемблера ARM и базовыми средствами кросс-компиляции;

- более детальное изучение инструментария (GAS, make) при сборке многофайлового проекта, работу с листингом, а также изучение особенностей доступа к оперативной памяти устройства;
- изучение вычислительных возможностей процессора ARM и доступа к энергонезависимой FLASH-памяти для хранения результатов;
- изучение системы прерываний, совмещение ассемблерного кода и модуля на C;
- знакомство с базовым набором инструментов для компиляции и установки GNU/Linux на ARM-устройство, изучение принципов удалённой отладки с помощью GDB.

## Программные инструменты

Оба практикума ориентированы на использование Sourcey CodeBench ARM (одна из существующих сборок GCC для кросс-компиляции). Адаптация курса БГУ к выполнению заданий на настольных ПК была выполнена подстановкой (на уровне make) компилятора из стандартного toolchain дистрибутива, что не повлекло практически никаких изменений в сборку учебных программ по сравнению с их кросс-компиляцией.

В курсе БрГТУ, помимо упоминавшихся инструментов, для заданий с ОС применён пакет автоматизированной сборки BuildRoot [5], упрощающий подготовку образа флеш-памяти с файловой системой, ядром Linux, загрузчиком uBoot и BusyBox в роли минимального набора системных утилит. При этом можно заметить, что сам набор утилит, задействованных на предлагаемых встраиваемых системах, в обоих курсах практически совпадает.

## Литература

- [1] Promwad: контрактная разработка и изготовление электроники. <http://promwad.ru/>
- [2] BeagleBone Black. <http://beagleboard.org/BLACK>
- [3] Linux kernel coding style. <https://www.kernel.org/doc/Documentation/CodingStyle>

[4] QEMU open source processor emulator. <http://wiki.qemu.org>

[5] Buildroot: Making Embedded Linux Easy. <http://buildroot.uclibc.org/>