

*Осипова Т.А. Компоненты Linux на ARM архитектуре // Академия педагогических идей «Новация». Серия: Студенческий научный вестник. – 2018. – №7 (июль). – АРТ 441-эл. – 0,2 п.л. - URL: <http://akademnova.ru/page/875550>*

## **РУБРИКА: ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

**УДК 004**

**Осипова Татьяна Александровна**

студентка 4 курса, факультет инженерного бизнеса и менеджмента  
*Научный руководитель:* Соколянский В.В., к.м.н., доцент  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования “Московский государственный технический  
университет имени Н. Э. Баумана” (национальный исследовательский  
университет)  
г. Москва, Российская Федерация  
e-mail: yatusu@mail.ru

## **КОМПОНЕНТЫ LINUX НА ARM АРХИТЕКТУРЕ**

*Аннотация:* В статье рассмотрены особенности имплементации решений на основе ОС Linux для архитектуры.

*Ключевые слова:* ARM, Linux, информационные системы

**Osipova Tatiana**

4nd year student, Faculty of Engineering Business and Management  
Supervisor: V.Sokolyanskiy, PhD, Associate Professor  
FGBOU VPO "Bauman Moscow State Technical University"  
Moscow, Russian Federation

## **LINUX COMPONENT ON ARM ARCHITECTURE**

*Abstract:* The article describes the main features of implementation OS Linux on architecture.

*Keywords:* ARM, Linux, information systems.

ARM архитектура получила широкое распространение с приходом в нашу жизнь сотовых телефонов. Её главной особенностью можно назвать RISC – Reduced Instruction Set Computer. В отличие от ПК-процессоров на базе Intel, имеющих архитектуру CISC - Complex Instruction Set Computing, ARM архитектура предъявляет меньшие требования к вычислительной мощности и энергопотреблению. Вместе с тем современные ARM решения поддерживают мультипроцессорные технологии (SMP - Symmetric Multiprocessing, симметричная мультипроцессорность). Так, для SMP-систем уже недостаточно так называемых baremetal решений на базе freertos, ecos, aodos. Для SMP-систем используются Linux-решения. Компоненты Linux на ARM-архитектуре отличаются от знакомых нам компонентов для ПК. Современная ARM архитектура базируется на процессорах семейства Cortex-A, широко используемых в сотовых телефонах, роутерах, планшетах и прочей аппаратуре.

Среди отличий между ARM системами и привычными ПК следует отметить существенные ограничения ARM систем. Так, в минимальной ARM системе могут отсутствовать видеоподсистема, аудио, клавиатура, мышь, жёсткий диск; что, в свою очередь, объясняет низкие требования (по отношению к ПК), предъявляемые к быстродействию и выполняемым задачам. В качестве примера приведём Ethernet-маршрутизатор на базе процессора iMX-6, содержащий несколько Ethernet-контроллеров, последовательный порт и встроенную NAND-флэш.

Первый компонент – начальный загрузчик. К функциям начального загрузчика относятся считывание бинарных образов компонентов ОС, размещение их в оперативной памяти и запуск их согласно установленным скриптам. В ARM-архитектуре в качестве начального загрузчика обычно используется U-Boot. U-Boot учитывает особенности архитектуры: тип

процессора, памяти, устройство хранения данных (USB-флэш, жёсткий диск, CD-карта), периферийные устройства.

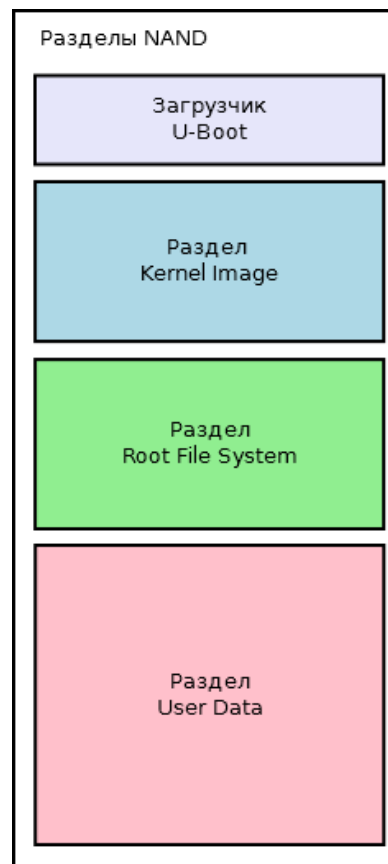


Рисунок 1 – Структура разделов NAND

U-Boot располагается в начальных секторах устройства хранения данных. Следом за ним находится бинарный образ Linux-ядра. Бинарный образ Linux-ядра, также так называемый DTB блок – Device Tree, отвечающий за конфигурации настройки ядра Linux. К данным настройкам относятся:

- настройки памяти;
- конфигуратор прерываний;
- адреса регистров периферийных устройств;
- дополнительные параметры ядра.

Затем – Root File System (RFS), которая в ARM архитектурах содержит

корневой раздел Linux со всеми системными программами. В отличие от традиционной ПК-архитектуры, RFS имеет небольшой размер и постоянно присутствует в оперативной памяти. RFS доступна в режиме “Только для чтения” (Read-Only). Как правило, в этой связи тип файловой системы выбирается наиболее простым: squashfs, cramfs, cromfs. Системы типа ext2 не применяются из-за избыточности.

Основной элемент RFS – busybox, представляющий собой все основные Linux-приложения, собранные как отдельный исполняемый модуль. Такое решение позволяет существенно упростить структуру файловой системы и сделать её минимальной (по размеру). Также в RFS присутствуют приложения, отвечающие за основную функциональность рассматриваемой ARM системы. Так, для приведённого выше Ethernet-маршрутизатора такими приложениям являются:

- TCP/IP стек;
- маршрутизатор iptables;
- криптосервис openssl.

Linux-скрипты объединяют данные приложения, запуская их согласно конфигурации. Как показано на картине, после RFS располагаются секторы пользовательских данных. Они используются для хранения пользовательских настроек, конфигурации, журнальных файлов, регистрации событий в системе. При рассмотрении данного маршрутизатора (где используется устройство NAND, не обладающее встроенным контролем BAD-блоков) для хранения пользовательских данных применяется журналируемая файловая система UBIFS, основной особенностью которой является менеджмент BAD-блоков.

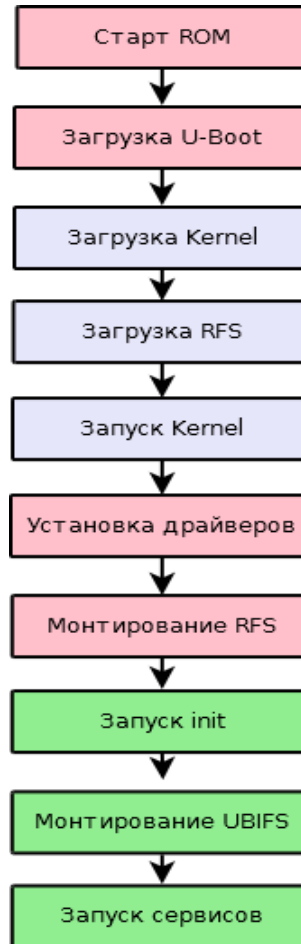


Рисунок 2 – Стартовая последовательность системы

Рассмотрим функционирование описанной системы. При запуске процессора управление передаётся встроенному в ROM микрокоду, который:

- настраивает системную память;
- активирует NAND;
- считывает из NAND начальный загрузчик U-boot;
- U-boot, исполняя загрузочный скрипт, считывает из NAND Linux-ядро и образ RFS;
- после этого U-boot передаёт управление ядру, сообщая ему начальную конфигурацию системы;

- ядро разворачивает в памяти основные подсистемы, монтирует RFS и передаёт управление в Userspace, запуская процесс init;

- процесс init монтирует пользовательский раздел ubfs, поднимает TCP/IP сервисы, настраивает службы, переводя систему в режим приёма передачи данных.

Система развёрнута. Будучи представителем семейства Linux, эта система (несмотря на её минимализм) может работать месяцами без перебоев и перезагрузок. Перечисленные компоненты представлены как в виде готовых бинарных модулей, так и в виде свободно распространяемого исходного кода для широкого спектра ARM устройств. Поэтому сборка, конфигурация и запуск такой системы могут быть выполнены с помощью свободно доступных в Интернете дистрибутивов и документации.

#### **Список использованной литературы:**

1. В.Костромин, Свободная система для свободных людей (обзор истории операционной системы Linux), март 2005 г.
2. Paul Cobbaut, "Фундаментальные основы Linux". Перевод выполнен А.Паниным.
3. О.И.Цилюрик, "Инструменты Linux для Windows-программистов", редакция 2.46 от 10.08.2011г.

***Дата поступления в редакцию: 23.07.2018 г.***

***Опубликовано: 23.07.2018 г.***

***© Академия педагогических идей «Новация». Серия «Студенческий научный вестник», электронный журнал, 2018***

***© Осипова Т.А., 2018***