АЛГОРИТМ ФУНКЦИОНИРОВАНИЯ ПРОГРАММНО-АППАРАТНОГО ОБЕСПЕЧЕНИЯ ОРГАНИЗАЦИИ РАБОТЫ СЕРВИСНЫХ ЦЕНТРОВ ПО РЕМОНТУ И ОБСЛУЖИВАНИЮ ЭЛЕКТРОННОЙ ТЕХНИКИ

Капельян А.Ю.

Институт информационных технологий БГУИР,
г. Минск, Республика Беларусь

Скудняков Ю.А.- доцент каф. ИСиТ, к.т.н,, доцент

В работе рассмотрен алгоритм взаимодействия программной и аппаратной частей сервисной системы в процессе организации ремонта и обслуживания современной электронной техники.

На данный момент большинство электронных устройств представляет собой программно-аппаратного комплекс. Такое решение позволяет расширить функционал устройства и предоставлять пользователю максимально комфортную среду для работы, удобный и продуманный интерфейс. Логика системы обеспечивает понятную и быструю навигацию по ней, не углубляясь в ее внутреннее строение.

В данной работе рассматривается алгоритм взаимодействие аппаратной и программной частей на примере мобильного телефона под управлением системы Android.

При подключении источника питания (АКБ) к материнской плате, контроллер питания производит тестирование периферийных устройств на наличие неисправностей. При отсутствии сигналов POWER_GOOD контроллер питания отключает источник питания от платы путем блокирования контроллера АКБ, таким образом осуществляется защита устройства по дежурным цепям питания. Если внутреннее тестирование прошло успешно, сигнал POWER_GOOD поступил от периферийных устройств, то контроллер питания генерирует дежурное напряжение для включения устройства и переходит в режим ожидания инициализации материнской платы.

При нажатии на кнопку включения контроллер питания осуществляет генерацию всех основных напряжений питания для периферийных устройств, таких как: центральный процессор, флеш-память, контроллер дисплейного модуля и т.д. При поступлении питания центральный процессор считывает с флеш-памяти такие значения как PIT и CID, и сравнивает их с записанными во внутреннюю память. При их несовпадении центральный процессор отправляет команду выключения на контроллер питания и инициализации не происходит.

Данная процедура позволяет исключить возможность подмены флеш-памяти от другого устройства без специального оборудования. При совпадении значений центральный процессор начинает считывать данные из флеш-памяти в соответствующих разделах. Под каждую модель процессора существует уникальная разметка (разделение) памяти в соответствии с внутренним строением самого процессора.

Первичным этапом осуществляется инициализация Boot Loader, который копирует файлы в рабочую память устройства и передает управление коду, расположенному в разделе boot.Boot, запускает настройку кэша, защищенную память, планировщик задач и загружает драйверы. Когда ядро завершит настройку и запуск своих подсистем, инициализируется корневой процесс ***init().*** Процесс ***init()*** подключает директории /sys, /dev, /proc и запускает службы, которые указаны в файле ***init.rc***.

Формат ***init.rc*** представляет собой набор команд, разделенных на блоки. Каждый блок определяет стадию загрузки. Так же ***init()*** инициализирует запуск ***среды выполнения Android*** путем включения службы ***Zygote***. Таким образом, производится инициализация всех разделов системы Android: Bootloader, Boot, System, Data, User.

Так же существует еще два варианта инициализации устройства, служащих для редактирования программной части. Режим Recovery позволяет внести изменения в существующую операционную систему, записанную в флеш-память. Режим Fastboot позволяет изменять любые разделы памяти

устройства, не считая Bootloader. Переход в данные режимы загрузки осуществляется путем замыкания контактов Testpoin и кнопки включения одновременно на выключенном устройстве.

Использование рассмотренного алгоритма функционирования программно-аппаратного обеспечения современных мобильных устройств позволяет осуществлять автоматизированный, гибкий и оперативный процесс организации работы сервисных центров по ремонту и обслуживанию электронной техники.