

ОПЕРАЦИОННАЯ СИСТЕМА

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

ФУНКЦИИ

СОСТАВ

ЯДРО, ПЛАНИРОВЩИК И МНОГОЗАДАЧНОСТЬ

ПРОЦЕСС

КЛАССИФИКАЦИЯ

ОПЕРАЦИОННАЯ СИСТЕМА ОБЩЕГО НАЗНАЧЕНИЯ

ОПЕРАЦИОННАЯ СИСТЕМА РЕАЛЬНОГО ВРЕМЕНИ

ОПЕРАЦИОННАЯ СИСТЕМА СЕТЕВАЯ

ОПЕРАЦИОННАЯ СИСТЕМА МОБИЛЬНАЯ

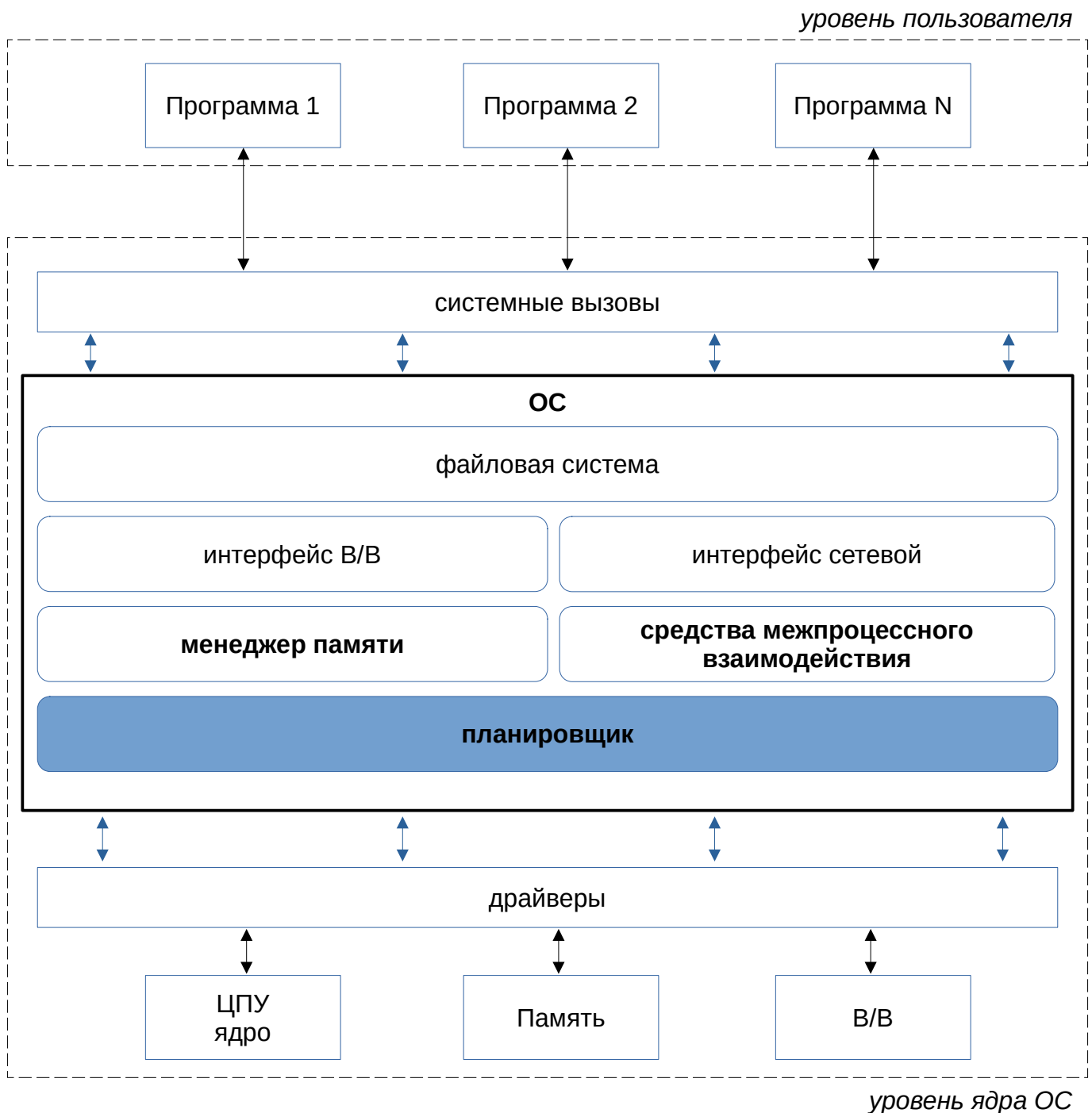
ВВЕДЕНИЕ

Операционная система (ОС) / Operating System (OS)

это программное обеспечение, которое

- управляет вычислительным устройством (компьютером или микроконтроллером)
- позволяет выполнять несколько прикладных программ (многозадачность)
- предоставляет средства (механизмы):
 - связь программ / процессов / потоков друг с другом (межпроцессное взаимодействие)
 - связь программ / процессов / потоков с аппаратными компонентами (оборудованием)
 - распределение общих ресурсов (память, каналы В/В)

ОС позволяют разработчикам программ абстрагироваться от деталей реализации и функционирования конкретных устройств и программ.



ФУНКЦИИ

Управление программами (многозадачность)

Планировщик (Scheduler)

- запуск / приостановка / завершение / изменение приоритета программы / задачи

Управление памятью

Менеджер памяти (Memory Manager)

- индивидуальное адресное пространство для каждого процесса (стек, куча)
- динамическое выделение памяти (memory allocation)
- виртуальная память

Различные уровни абстракции

Единый интерфейс для работы с различным оборудованием, позволяющий разработчикам программ абстрагироваться от деталей реализации и функционирования оборудования и других программ

- процессы и потоки (thread)
- файлы и файловые системы
- драйверы
- сокеты, системные вызовы, прерывания

Изоляция ошибок

Ошибки одной программы не влияют на работу других программ и ОС в целом

ОС представляет собой интерфейс («прослойку») между пользователем ресурсов вычислительного устройства и самими ресурсами. Этот интерфейс управляет взаимодействиями: пользователь-ресурс, пользователь-пользователь, пользователь-оборудование, оборудование-оборудование.

СОСТАВ

- **Ядро**
 - + планировщик
 - + средства межпроцессного взаимодействия
 - + средства управления памятью
 - + средства поддержки одной или нескольких файловых систем
 - + драйверы оборудования
 - + подсистема В/В
 - + сетевая подсистема
- **Системная оболочка**
 - + библиотеки
 - + утилиты (например, системная консоль)



положение ОС в организационной иерархии вычислительного устройства

Средства межпроцессного взаимодействия

- общая память
- очереди
- семафоры, мьютексы
- события, сигналы
- сокеты

ЯДРО, ПЛАНИРОВЩИК И МНОГОЗАДАЧНОСТЬ

Ядро

- центральная часть ОС (по сути, ядро и есть ОС)
- реакционный механизм — реакция на события с их последующей обработкой
- загружается в память вычислительного устройства после его включения

Планировщик

- одновременное исполнение нескольких задач (многозадачность)
- работает (тактируется) по прерыванию таймера вычислительного устройства

Реализация многозадачности

1. Включение устройства

- инициализация памяти
- инициализация периферии (включая периферии процессора / микроконтроллера)
- инициализация компонентов ядра ОС
- инициализация и запуск тактового таймера планировщика

2. Запуск планировщика

3. Работа ОС

- срабатывает прерывание тактового таймера через заданный период (квант времени)
 - работа передается планировщику, который переключает контекст:
 - сохраняет состояние (контекст) задачи, исполняемой до прерывания
 - загружается контекст следующей по приоритету задачи
 - работа передается задаче
 - запускается тактовый таймер
- срабатывает прерывание от какого-то оборудования
 - работа передается обработчику события
 - по завершении работы обработчика, работа передается прерванной задаче

Задача (Task)

- функция с конечным или бесконечным циклом, выполняющая какой-либо запрограммированный алгоритм (иное название - программа)

Контекст

- значения локальных переменных (включая входные аргументы)
- указатель на последнюю исполняемую инструкцию
- состояние

Переключение контекста

- смена контекста одной задачи на контекст другой без потери данных
- контекст хранится в стеке (для каждой программы свой стек в ОЗУ)
- за предоставленный квант времени выполняется набор инструкций задачи (последовательно, начиная с указанной при переключении)

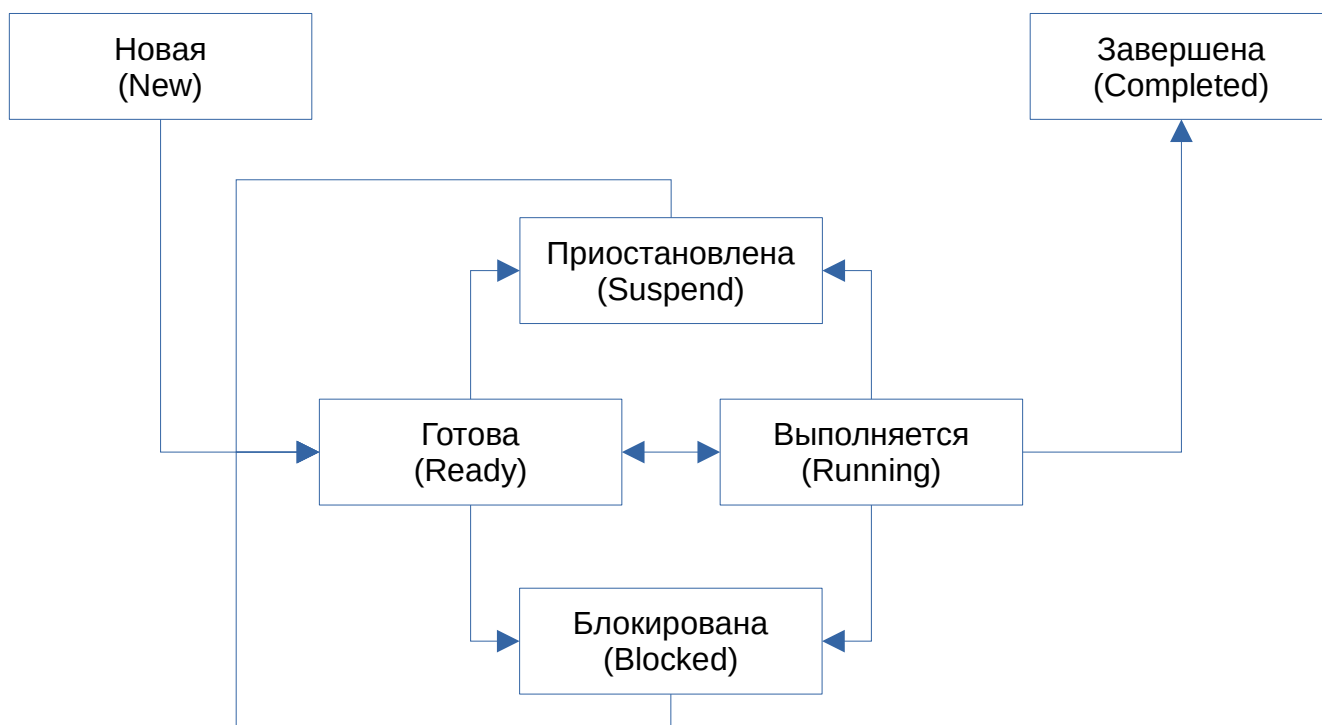
ЯДРО, ПЛАНИРОВЩИК И МНОГОЗАДАЧНОСТЬ

Квант времени

- время, когда процессор выполняет инструкции задачи
- это время строго определено работой таймера тактирования планировщика (обычно, 1 мсек)

По завершении кванта (по прерыванию таймера) работает планировщик — переключает контекст. Время работы планировщика ненормировано (не квантуется). Соответственно, чем быстрее будет выполнено переключение контекста, тем быстрее начнет выполняться полезная работа (следующий набор инструкций).

Состояние задачи



При грубом приближении, задача может находиться в следующих состояниях:

- создана (новая) / работает / завершена

Задача с состоянием «Создана» или «Завершена» не получает процессорного времени (не квантуется / полезная работа не выполняется).

Во время работы задача может иметь несколько различных состояний:

Выполняется

- задача выполняется процессором в данный момент времени

Готова

- задача не выполняется процессором в данный момент времени
- очередность запуска готовых задач определяется их приоритетом; при наличии задач с одинаковым приоритетом, очередность определяется временем, сколько задача находится в состоянии готовности (задача с большим временем запускается первой)

ЯДРО, ПЛАНИРОВЩИК И МНОГОЗАДАЧНОСТЬ

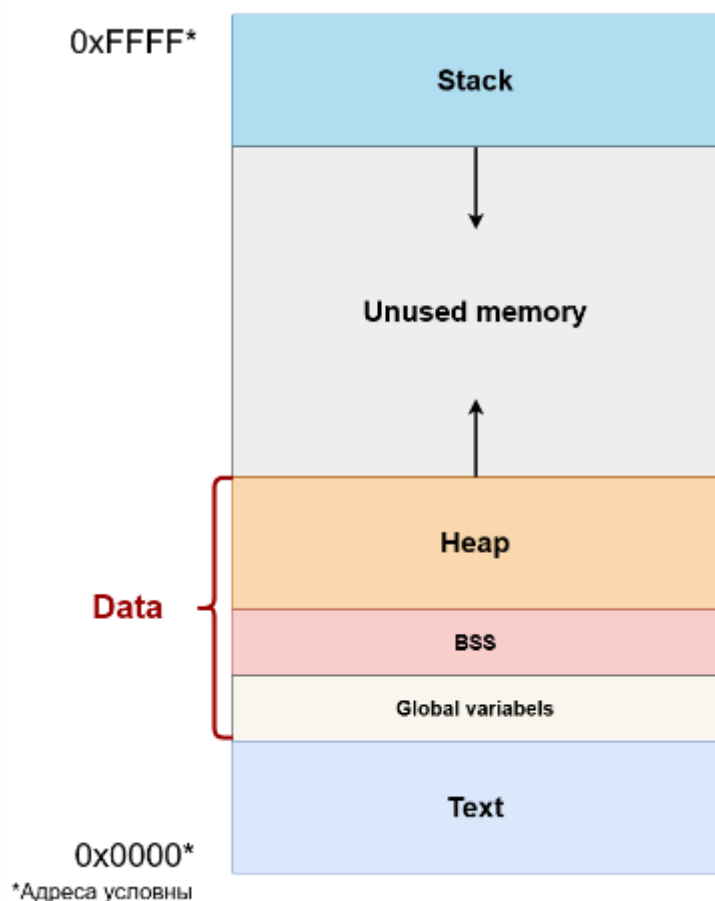
Блокирована

- задача не выполняется процессором в данный момент времени
- ожидается наступление события, связанного с другим объектом ядра ОС (обычно это элементы межпроцессного взаимодействия, которые «защищают» какие-то общие ресурсы — например, когда одна задача работает с сетевым интерфейсом, то остальные претендующие задачи должны ждать — блокированы):
 - ожидание данных в очереди
 - ожидание освобождения семафора, мьютекса
 - ожидание завершения задержки (Delay)
- ожидаемое событие переводит блокируемую задачу в состояние готовности
- нахождение задачи в данном состоянии предотвращает «подвисание» процессора

Приостановлена

- задача не выполняется процессором в данный момент времени
- аналогично блокированному состоянию, но задача переводится в это состояние и выводится из него «вручную» - явным вызовом специальных функций (suspend / resume)

Сегменты памяти



Stack

- для хранения контекста
- от старших адресов к младшим

Unused memory

- свободная память
- с одной стороны заполняется стеком
- с другой стороны заполняется кучей

Heap

- динамически выделяемая

BSS

- глобальные переменные (не инициализированные)

Global variables

- глобальные переменные (инициализированные)

Text

- инструкции (код)

ПРОЦЕСС

Программа

- всего лишь пассивная последовательность инструкций

Процесс (Process)

- совокупность инструкций и данных (например, значений локальных переменных), что находятся в оперативной памяти и обрабатываются процессором; т. е. данный исполнимый код полностью владеет всеми вычислительными ресурсами вычислительного устройства

Вся информация о процессе содержится в специальной структуре данных, поддерживаемой ОС для каждого процесса

- PCB (Process Control Block) / TCB (Task Control Block)

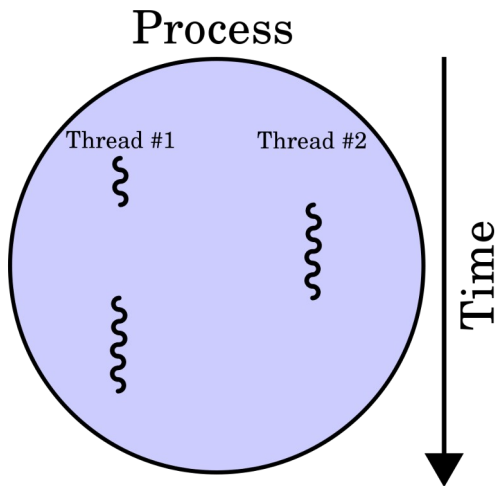
Process ID	- уникальный идентификатор
State	- текущее состояние
Pointer	- указатель на родительский процесс (для потоков)
Priority	- приоритет
Program counter	- адрес следующей выполняемой инструкции
CPU registers	- список используемых регистров процессора
I/O information	- список используемых ресурсов В/В
Accounting information	- уровень нагрузки на процессор, статистика
etc....	- прочая информация

ПОТОК

Каждый процесс может делиться на потоки (Thread), которые обеспечивают функциональный параллелизм (многозадачность) на уровне одной программы.

Реализация потоков выполнения и процессов в разных ОС отличается друг от друга, но в большинстве случаев поток выполнения находится внутри процесса.

Взаимодействие между потоками процесса реализуется средствами межпроцессного взаимодействия.



КЛАССИФИКАЦИЯ

- **По типу**
 - общего назначения
 - реального времени (ОСРВ)
 - сетевые
 - мобильные
- **По типу устройства ядра (архитектуре)**
 - монолитные
 - модульные (гибридные)
 - микро-, нано-, экзо-, уни-

ОПЕРАЦИОННАЯ СИСТЕМА ОБЩЕГО НАЗНАЧЕНИЯ

General Purpose Operating System
GPOS

Основанная задача

- оптимально распределить ресурсы вычислительного устройства между программами
- без жестких требований ко времени реакции системы на внешние события

На что ориентирована

- на пользователя (одно- или многопользовательский режим)
- защита системы, программного обеспечения и данных

Как позиционируется

- набор готовых программ (приложений), готовых к использованию
- устанавливается / располагается на диске внешней энергонезависимой памяти

Кому предназначена

- пользователь средней квалификации

Аппаратные требования

- память (ОЗУ и программ)
 - от десяткой мегабайт до гигабайт
- процессор
 - производительный (желательно многоядерный)
- видеоконтроллер
 - для подключения монитора
- каналы В/В
 - для манипуляторов человеко-машинного интерфейса (мышь, клавиатура)
- сетевые интерфейсы

Windows, Unix-подобные, Linux

ОПЕРАЦИОННАЯ СИСТЕМА РЕАЛЬНОГО ВРЕМЕНИ

Real-Time Operating System
RTOS

Основная задача

- обеспечить многозадачность, масштабируемость
- наиболее быстро успеть среагировать на внешние события (прерывания)
- гарантия надежности и отказоустойчивости (например, уровня SIL3)

На что ориентирована

- на обработку внешних событий (прерываний)

Как позиционируется

- набор программных средств (инструмент) для создания аппаратно-программного комплекса реального времени
- как встраиваемая (для микроконтроллеров)
 - встраивается в основной программный код и компилируется вместе как одно целое
 - располагается во встроенной или внешней энергонезависимой памяти
- как общего назначения с микроядром реального времени
 - компонуется / собирается из готовых программных компонентов (образ системы)
 - располагается во внешней энергонезависимой памяти

Кому предназначена

- квалификационный разработчик встраиваемых систем

Аппаратные требования

- память (ОЗУ и программ)
 - от десяткой килобайт до мегабайт
- процессор
 - микроконтроллер, система на кристалле

Встраиваемые:

- FreeRTOS
- uOS
- Zephyr
- EmBox

Linux-подобные + микроядро OSCPВ:

- QNX
- Xenomai

Windows-подобные:

- Windows CE

ОПЕРАЦИОННАЯ СИСТЕМА СЕТЕВАЯ

Network Operating System

Основная задача

- обеспечить многозадачность, масштабируемость
- обеспечить функции и службы для работы сетевого устройства (маршрутизатор, коммутатор или межсетевой экран)
- без жестких требований ко времени реакции системы на внешние события

На что ориентирована

- на пользователя (одногопользовательский режим)
- защита системы, программного обеспечения и данных
- на работу с сетями передачи данных (корпоративными и промышленными, в том числе высокоскоростными и высконагруженными)

Как позиционируется

- набор программных средств (инструмент) для создания аппаратно-программного комплекса, реализующего функции сетевого устройства
- как встраиваемая (для микроконтроллеров)
 - встраивается в основной программный код и компилируется вместе как одно целое
 - располагается во встроенной или внешней энергонезависимой памяти
- как общего назначения
 - компонуется / собирается из готовых программных компонентов (образ системы)
 - располагается во внешней энергонезависимой памяти

Кому предназначена

- квалификационный разработчик встраиваемых систем

Аппаратные требования

- память (ОЗУ и программ)
 - от десятков килобайт до мегабайт
- процессор
 - микроконтроллер, система на кристалле
- сетевые интерфейсы

DD-WRT, OpenWRT, ZyNOS, JUNOS

ОПЕРАЦИОННАЯ СИСТЕМА МОБИЛЬНАЯ

Mobile Operating System

Основная задача

- обеспечить многозадачность, масштабируемость
- обеспечить функции и службы для работы мобильных устройств (смартфон, планшет, КПК, нетбук)
- без жестких требований ко времени реакции системы на внешние события

На что ориентирована

- на пользователя (одно- или многопользовательский режим)
- защита системы, программного обеспечения и данных
- на работу с сетями передачи данных (сотовые операторы, радиосвязь и т.п.)

Как позиционируется

- набор программных средств (инструмент) для создания аппаратно-программного комплекса, реализующего функции мобильного устройства
- как встраиваемая (для микроконтроллеров)
 - встраивается в основной программный код и компилируется вместе как одно целое
 - располагается во встроенной или внешней энергонезависимой памяти
- как общего назначения
 - компонуется / собирается из готовых программных компонентов (образ системы)
 - располагается во внешней энергонезависимой памяти

Кому предназначена

- квалификационный разработчик встраиваемых систем

Аппаратные требования

- память (ОЗУ и программ)
 - от десятков мегабайт до гигабайт
- процессор
 - производительный (желательно многоядерный)
- видеоконтроллер
- сетевые интерфейсы

Android, iOS