

# Table of Contents

<a href="#">Общее описание курса</a>	1.1
--------------------------------------	-----

## Введение

<a href="#">Вводная часть. Термины и понятия</a>	2.1
--	-----

## Лекция. Теория. Устройство карты OpenStreetMap (OSM)

<a href="#">Лекция. Теория. Устройство карты OpenStreetMap (OSM)</a>	3.1
--	-----

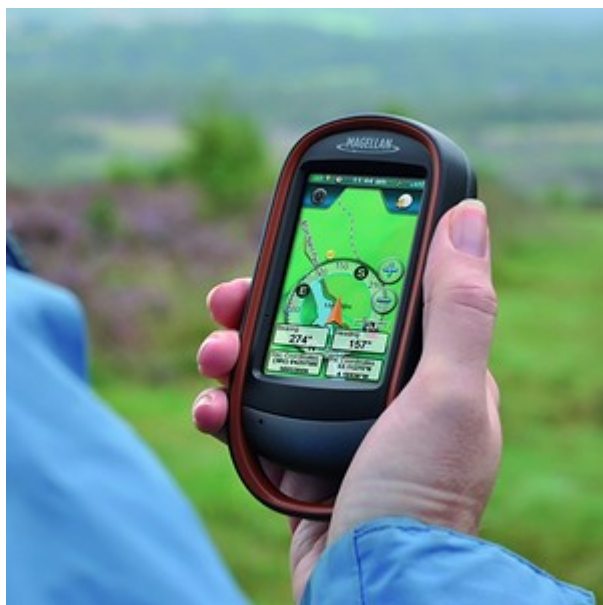
# Общее описание курса

fffffffffffffffffffffg vghhhhgsgggggggggggg  
hhkkhgfjfhgfhgf аргртни

планшет

# Вводная часть. Термины и понятия

Абзац 1



Если вы читали предыдущие статьи моего блога, то могли заметить, что некоторое время назад я начал увлекаться низкоуровневым программированием. Я написал несколько статей о программировании под x86\_64 в Linux. В то же время я начал "погружаться" в исходный код Linux. Я испытываю большой интерес к пониманию того, как работают низкоуровневые вещи, как запускаются программы на моем компьютере, как они расположены в памяти, как ядро управляет процессами и памятью, как работает сетевой стек на низком уровне и многие другие вещи. Поэтому я решил написать ещё одну серию статей о ядре Linux для x86\_64.

Замечу, что я не профессиональный хакер ядра и не пишу под него код на работе. Это просто хобби. Мне просто нравятся низкоуровневые вещи и мне интересно наблюдать за тем, как они работают. Так что, если вас что-то будет смущать или у вас появятся вопросы или замечания, пишите мне в твиттер 0xAX, присылайте письма на email или просто создавайте issue. Я ценю это.

Все статьи также будут доступны в репозитории Github, и, если вы обнаружите какую-нибудь ошибку в содержимом статьи, не стесняйтесь присылать pull request.

Заметьте, что это не официальная документация, а просто материал для обучения и обмена знаниями.

Требуемые знания

+Понимание кода на языке C

+Понимание кода на языке ассемблера (AT&T синтаксис)

В любом случае, если вы только начинаете изучать подобные инструменты, я постараюсь объяснить некоторые детали в этой и последующих частях. Ладно, простое введение закончилось, и теперь можно начать "погружение" в ядро и низкоуровневые вещи.

Я начал писать эту книгу, когда актуальной версией ядра Linux была 3.18, и с этого момента многое могло измениться. При возникновении изменений я буду соответствующим образом обновлять статьи.

Магическая кнопка включения, что происходит дальше?

Несмотря на то, что это серия статей о ядре Linux, мы не будем начинать непосредственно с его исходного кода (по крайней мере в этом параграфе). Как только вы нажмёте магическую кнопку включения на вашем ноутбуке или настольном компьютере, он начинает работать. Материнская плата посылает сигнал к источнику питания. После получения сигнала, источник питания обеспечивает компьютер надлежащим количеством электричества. После того как материнская плата получает сигнал "питание в норме" (Power good signal), она пытается запустить CPU. CPU сбрасывает все остаточные данные в регистрах и записывает предустановленные значения каждого из них.

CPU серии Intel 80386 и старше после перезапуска компьютера заполняют регистры следующими предустановленными значениями:

# **Лекция. Теория. Устройство карты OpenStreetMap (OSM)**

## **Подзаголовок 1**

## **Подзаголовок 2**

Текст-текст-текст