

# HelIOS — учебная операционная система и создание курса на её основе

Александр Андреев, Виктор Яковлев

Московский физико-технический институт

15 июня, 2021

# Курс «Низкоуровневое программирование 2020»

- ▶ Цель – дать более полное представление о том, как работают ОС
- ▶ Аудитория – Зий курс МФТИ

# Структура курса

- ▶ <https://github.com/carzil/mipt-llp-2020>
- ▶ На семинарах читалась теоретическая информация на примере HelIOS
- ▶ Домашние задания в виде лабораторных работ
- ▶ Экзамен отсутствовал – результат складывался из оценок домашних заданий

## x86 vs ARM

- ▶ Две самые распространённые архитектуры
- ▶ x86 занимает почти весь server-side и desktop
- ▶ ARM – смартфоны, микроконтроллеры, но постепенно захватывает традиционные ниши x86
- ▶ CISC vs RISC
- ▶ Имеет смысл изучать обе архитектуры: ARM более перспективен, x86 нескоро выйдет из моды

## Структура курса: изучаемые темы

- ▶ Процесс загрузки и Multiboot-совместимые загрузчики
- ▶ Работа с контроллерами прерываний, обработка прерываний
- ▶ Работа с памятью (virtual memory, 4-page addressing, etc)
- ▶ Планировщики процессов
- ▶ Работа с АТА-совместимыми устройствами (включая работу DMA)
- ▶ Базовые принципы работы файловых систем на примере ext2
- ▶ Изоляция пользовательских процессов

# Зачем нужна базовая ОС для обучения?

- ▶ Проще осознавать информацию на примере
- ▶ Базовый «каркас» для лабораторных работ
- ▶ Есть студенты, которым курс неинтересен по разным причинам, они не хотят писать всё сами с нуля
- ▶ Единообразии структуры кода в ДЗ  $\Rightarrow$  проверка становится легче

- ▶ Написана и использовалась в MIT для курса Operating Systems Engineering
- ▶ Сопровождается книгой, которая описывает основные дизайн решения
- ▶ Есть отдельный порт под ARM
- ▶ **Ужасный** кодстайл
- ▶ Чтение с дисков реализовано в polling-режиме, отсутствует DMA
- ▶ Не поддерживает 64 битные процессоры
- ▶ Не поддерживается ARM
- ▶ Если пытаться улучшить xv6, книга про неё становится бесполезна

- ▶ Аналогично xv6, использовалась в MIT для курса Operating Systems Engineering
- ▶ Чтение с дисков реализовано в polling-режиме, отсутствует DMA
- ▶ Не поддерживает 64 битные процессоры
- ▶ Не поддерживается ARM



# MINIX

- ▶ Изначально написана в качестве примера для книги Э. Танненбаума
- ▶ Реализована в виде микроядра — нетрадиционная архитектура для современных ОС
- ▶ Уже давно не учебная  $\Rightarrow$  слишком высокий порог для входа
- ▶ Реализовано всё то, что хочется оставить на лабораторные работы

## Почему важен выбор языка для ОС?

- ▶ Инструмент для студентов, а не тема курса
- ▶ Должен быть актуальным и востребованным

# Про Rust

- ▶ Современный язык, предоставляющий много compile-time гарантий
- ▶ Реализована хорошая библиотека (crate) для x86
- ▶ Слишком большой порог для входа
- ▶ Курс слишком короткий, чтобы давать новый язык студентам

# Про C++

- ▶ Современный C++ скрывает большинство операций с памятью – недопустимо для ОС
- ▶ Большинство абстракций бесполезны (стримы, треды, fs) или требуют своих аллокаторов памяти (контейнеры, smart pointers)
- ▶ Классы и деструкторы – не хватает в C

# Про С

- ▶ Традиционно используется для создания ОС
- ▶ Колоссальное количество библиотек и программ до сих пор написано на С
- ▶ Огромное сообщество
- ▶ Уже изучен студентами в рамках курса АКООС

# HelloS

- ▶ <https://github.com/carzil/hellos>
- ▶ Open source, лицензия MIT

## Helios: ключевые особенности

- ▶ Монолитное higher-half ядро
- ▶ Для управления прерываниями используется современный APIC
- ▶ 32-битная адресация памяти
- ▶ Чтение с ATA-совместимых дисков при помощи DMA/PIO
- ▶ Примитивный планировщик процессов
- ▶ Поддерживает запуск в QEMU, на «сыром» железе не тестировалась

# Лабораторные работы в курсе

1. Подготовка окружения (настройка кросс-компилятора, QEMU, GDB, make etc)
2. Калибровка APIC таймера и разработка драйвера PS/2-совместимой клавиатуры
3. Менеджер виртуальной памяти и физической памяти
4. Реализация механизма переключения контекста и изоляции процессов



## Текущие планы

- ▶ Реализация HelIOS одновременно для ARM и для x86 (переиспользуя высокоуровневый код)
- ▶ Красиво оформленные семинарские презентации и соответствующие им ридинги
- ▶ Cheat sheet для работы с QEMU, GDB и системами сборки
- ▶ Более проработанные лабораторные работы с заданными критериями оценки и советами по реализации
- ▶ Планируется добавить материал про внутреннее устройство Linux

Спасибо за внимание!  
Вопросы?