

Санкт-Петербургский политехнический университет Петра Великого

# REACTOS

Курс: **Проектирование ОС и компонентов**

---

Студент: **Д.В. Круминьш**

Группа: **13541/3**



Преподаватель: **Е.В. Душутина**



**ReactOS** — это свободная и открытая операционная система(лицензия **GNU General Public License**). Система была написана с нуля и имеет своей целью повторение архитектуры Windows-NT 5.2 SP1(Windows 2003 SP и Windows XP 64-bit Edition), созданной Microsoft, от аппаратного до прикладного уровня.

Проекту более 20 лет, так и не вышел из альфа версии.

Основная причина - сопротивление монополии Windows, путем создания открытой системы с сохранением функциональности.

- Позволяет студентам и молодым разработчикам принять участие в крупном проекте, получить опыт при разработке приложения/ядра/драйверов.
- Позволяет компаниям не платить за лицензию Windows.
- Позволяет низко-уровневым разработчикам понять как работает Windows.

## Активная разработка

- x86. В первую очередь разработка идет для данной архитектуры.
- AMD64.

## Прочие архитектуры

- ARM. Успешный запуск ядра системы, но не более.
- Xbox (i386);
- PowerPC (ppc).

В планах и другие архитектуры, но не хватает программистов.

Минимальные требования для установки:

- **RAM** - минимум 64 MB, рекомендуется 256 MB.
- **Процессор** - архитектура x86 или x64.
- **HDD** - IDE/SATA минимально 450 MB
  - Файловые системы FAT16/FAT32.
- **Video** - VGA совместимая видеокарта

Имеется ограниченная поддержка аппаратных устройств, список протестированных устройств весьма невелик.

Имеется около 30 постоянных разработчиков. Около 170 разработчиков, которые в какое-то время что-то написали.

Труд ниодного из разработчиков не оплачивается.[1]

Имеется контрактная разработка.

Для сравнения, над **NT 5.2** работали 2000 разработчиков, 2400 тестировщиков, а размер кода составляет около 50 миллионов строк.[2]

Репозиторий - <https://github.com/reactos/reactos>

На 14.03.2018 в репозитории, в ветке мастер совершено **71 273** коммита, размер исходного кода составляет **383 МВ**, проект имеет **35** веток.

**9,000,000+** строчек кода.

Язык программирования	Процент кода
C	88.6%
C++	9.1%
Objective-C	0.6%
Ассемблер	0.5%
CMake	0.4%
Yacc	0.2%
Other	0.6%

Для сборки системы, необходимо использовать официальную среду сборки - **RosBE**.

Некоторые функции:

- **CHARCH** - Изменение архитектуры, для которой будет производиться сборка. (x86, amd64)
- **UPDATE** - Обновляет все файлы до самых последних версий.
- **RADDR2LINE** - Переводит адреса программ в имена файлов и номера строк для помощи разработчикам в поиске особых ошибок в ReactOS.

```
<\SystemRoot\System32\NTOSKRNL.EXE: 85fa8 >  
  
raddr2line ntoskrnl.exe 85fa8  
  
C:\Users\Ged\MyFiles\ReactOS\clean_source\output-i386\ntoskrnl\  
    ↪ ntoskrnl.exe  
obj-i386\ntoskrnl\ex\zw.S:253 (ZwClearEvent)
```

ReactOS можно собрать в самой ReactOS.

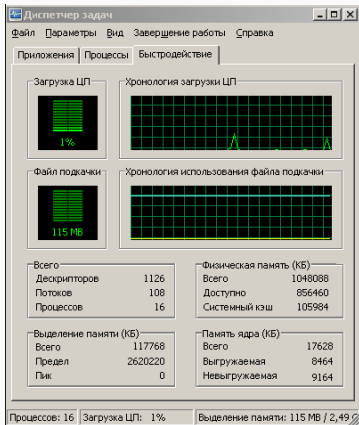


Разработка(клонирование) происходит на основе:

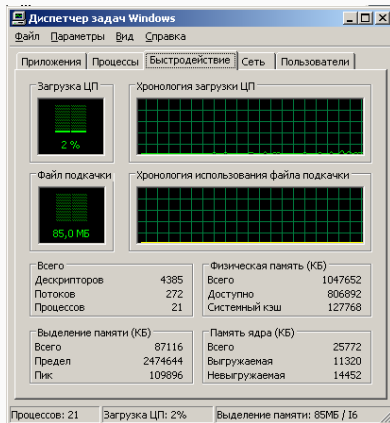
- Документации из msdn;
- Информации из открытых источников;
- Прослеживанием вызовов через WinDbg;
- Реверс инжиниринг(!!!).

Риверс-инжинеринг легален в большинстве стран, но имеются патенты на алгоритмы.

# Диспетчер задач



ReactOS



Windows 2003

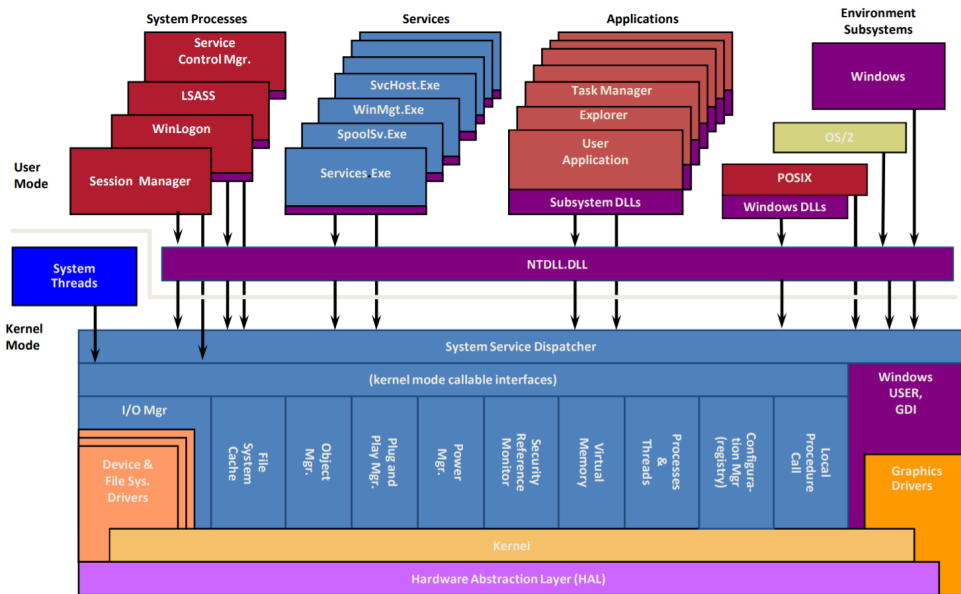
# История

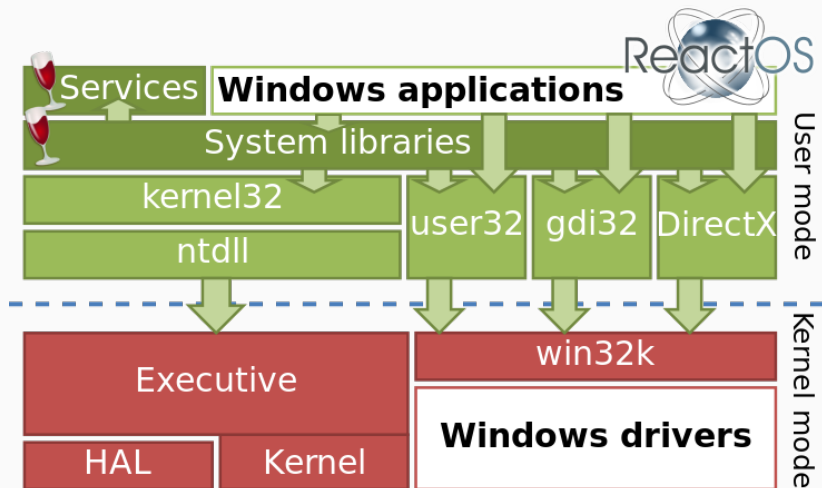


АРХИТЕКТУРА

---

# Архитектура Windows NT





ReactOS имеет много общего с Wine, благодаря своей открытости проекты поддерживают друг друга. **Общие черты:**

- Установка приложений Windows;
- Запуск приложений Windows.

**Отличия:**

- Работа на линуксе против работы на реальном железе;
- Поддержка драйверов Windows.

**HAL**(Hardware Abstraction Layer - Слой аппаратных абстракций) - позволяет запускать ОС на различных платах x86 без именений в ядре. Находится между аппаратным и програмными уровнями.

**win32k** - интерфейс для вывода информации(на монитор, принтер, другие устройства).

**Kernel**(NTOSKRNL) - микроядро, дизайн которого поддерживает различные архитектуры(x86-64, MIPS, Alpha...) реализует асинхронные вызовы процедур ядра (APC), вызовы отложенной процедуры (DPC), процессы, потоки, мьютексы, семафоры, блокировки...

Наиболее важные компоненты:

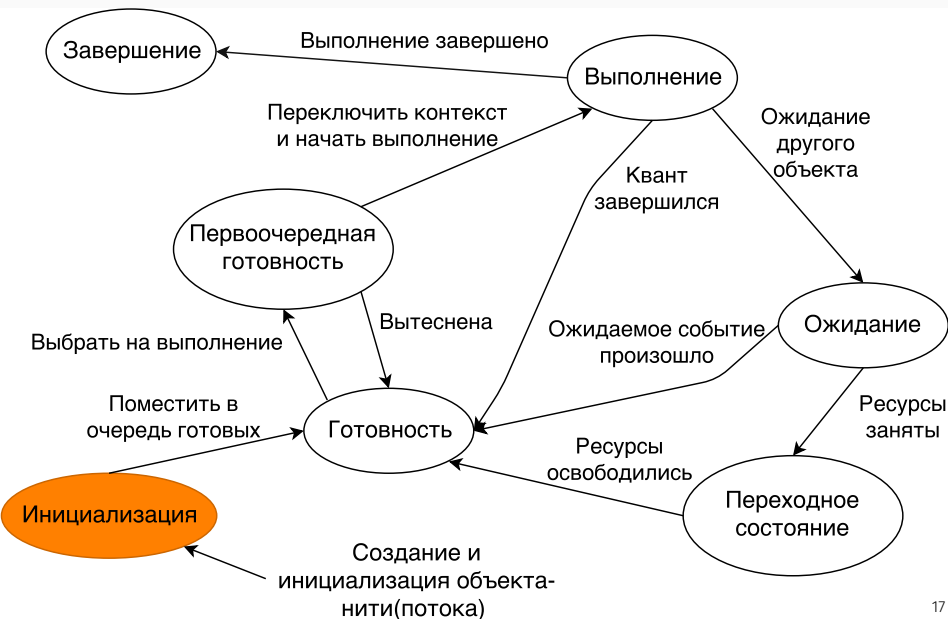
1. Планировщик;
2. Диспетчер;
3. Executive(исполнительная подсистема).



Планировщик ядра NT - это планировщик на основе потоков, что означает, что он планирует потоки, а не процессы.

Реализована вытесняющая многозадачность, при которой операционная система не ждет, когда поток сам захочет освободить процессор, а принудительно снимает его с выполнения после того, как тот израсходует отведенное ему время (квант), или если в очереди готовых появился поток с более высоким приоритетом.

Поддерживается 32 уровня приоритетов, разделенных на два класса - класс **реального времени** и класс **переменных приоритетов**. Потоки реального времени, приоритеты которых находятся в диапазоне от 16 до 31, являются более приоритетными процессами и используются для выполнения задач, критичных ко времени.



Диспетчер - компонент ядра, который берёт на себя обработку объектов диспетчера, которые используются для синхронизации и оповещения.

- **Событие** - Действие, начатое, как правило, вне рамок программы.
- **Семафор** - Классический способ ограничения доступа к общим ресурсам.
- **Поток** - Самая малая функциональная единица кода, выполняющегося на центральном процессоре.
- **Таймер** - Счетчики, которые либо увеличиваются либо уменьшаются на определённой фиксированной частоте.
- **Мьютекс** - Объект, позволяющий нескольким потокам получать синхронный доступ к ресурсам и избегать одновременного использования общих ресурсов.

Executive(исполнительная подсистема) - подсистема, обеспечивающая драйверам высокоуровневый доступ к объектам диспетчера ядра, а также поддержку высокоуровневой функциональности процессов и потоков, такой, как маркеры (управляемые подсистемой защиты), объекты, дескрипторы, квоты, WIN32K, задания, порты (управляемые подсистемой вызова локальных процедур (LPC)) и т.д.

Предоставляет несколько своих объектов, например быстрый мьютекс (fast mutex), обратные вызовы (callbacks), блокировки с заталкиванием указателя (pushlocks).

Также, она предоставляет высокоуровневый доступ к некоторым структурам управления памятью и функциям, например, выделению пула.

## Менеджер объектов(Object Manager)

- Создает, удаляет и управляет объектами NT executive - абстрактными типами данных, используемых для представления ресурсов системы.

## Менеджер кэша

- Реализация кэширования диска.

## Монитор безопасности

- Устанавливает правила защиты на локальном компьютере. Охраняет ресурсы операционной системы, выполняет защиту и регистрацию исполняемых объектов.

## Менеджер процессов

- Создает и завершает, приостанавливает и возобновляет процессы и нити, а также хранит о них информацию.

## Хорошо

- Планировщик, HAL, менеджер процессов и потоков.

## Еще требуется доработка

- I/O подсистема, менеджер конфигураций, менеджер безопасности.

## Плохо

- Менеджер памяти, менеджер кэшей, библиотека поддержки файловой системы. Все три компонента так и не были корректно реализованы с самого начала проекта.

## Не существует

- Менеджер питания.

- Совместимость с драйверами для NT 6.x (Vista/7/8/10)
- Завершение поддержки печати
- Безотказная работа ПО
- Поддержка DirectX
- Завершение реализации Wi-Fi
- Исправление графических ошибок
- Разметка дисковых разделов с использованием NTFS, exFAT,
- Работа с SSD, RAID и составными томами напрямую.

**В далеком будущем:** выход из альфа версии, поддержка прочих архитектур.

- Антивирусы активно использовали дыры(недокументированные API) в windows в следствии чего в reactOs для корректной работы их нужно специально эмулировать.
- Когда в reactOs происходят сбои, их анализируют с помощью **WinDBG**, которая спрашивает хотите ли вы отправить данные в Microsoft, чтобы их проанализировали.
- Пользователи по сути являются тестировщиками.
- Данный проект позволяет заявить о себе.
- Один из разработчиков **ReactOs** читал лекции **Windows** программистам о том как работает **Windows**.
- Выдающиеся программисты ушли работать в **Microsoft** и другие крупные компании.
- Также имеется "синий экран смерти".



**ReactOS** это:

- не Windows, это 100% свободная и открытая реимплементация Windows;
- не Linux, драйвера и программы работают нативно;
- не Wine, от Wine используется лишь небольшая часть;
- уникальный проект, которому требуется еще много работы.

## Идеи для работы:

- Дополнить ReactOS(в ветку на гитхабе) новой функциональностью(много рисков).
  - Высокая сложность, по большей части остался kernel mode.
  - Время на принятие дополнений в основную ветку.
  - Код ревью.
  - Риск, что кто-то другой это реализует.
- Изменить поведение системы в той или иной мере.
- В Jira проекта имеется список багов для исправлений. Взять один из багов, проанализировать причину его возникновения и попытаться исправить.

1. ReactOS Tech talk in Google Montreal by Alex Ionescu in 2013. - URL: <https://www.youtube.com/watch?v=HNPoCz1lBoQ>
2. Maraia, Vincent. (2005). The Build Master: Microsoft's Software Configuration Management Best Practices.
3. ReactOS Kernel. URL: <https://www.reactos.org/wiki/Kernel>
4. ReactOS Ntoskrnl. URL: <https://reactos.org/wiki/Ntoskrnl.exe>
5. Build Environment. URL: [https://winehq.org.ru/Build\\_Environment](https://winehq.org.ru/Build_Environment)
6. ReactOS GitHub. URL: <https://github.com/reactos/reactos>
7. ReactOS ports. URL: [https://www.reactos.org/wiki/ReactOS\\_ports](https://www.reactos.org/wiki/ReactOS_ports)
8. Концепции Windows NT. URL: [http://citforum.ru/operating\\_systems/sos/glava\\_39.shtml](http://citforum.ru/operating_systems/sos/glava_39.shtml)
9. Концепции Windows NT. URL: [http://rgata.ru/sites/mpoevs/uploads/citforum/operating\\_systems/winntadm/winntadm\\_01.html](http://rgata.ru/sites/mpoevs/uploads/citforum/operating_systems/winntadm/winntadm_01.html)