Лекция 4. Основные компоненты операционной системы

Архитектура ОС Windows

9 октября 2014 г.

Обзор подсистемы Windows

Имя файла	Назначение
`Csrss.exe`	Процесс подсистемы окружения (Client/Server Runtime Sub-System).

`Basesrv.dll`
`Winsrv.dll`
`Csrsrv.dll`
	Библиотеки процесса окружения.

`Win32k.sys`	Драйвер режима ядра.
`Conhost.exe`	Процесс диспетчера консоли.

`Kernel32.dll`
`User32.dll`
`Gdi32.dll`
`Advapi32.dll`
	Библиотеки подсистемы.

—	Графические драйверы для дисплея, принтера и видео

Таблица 1: состав подсистемы Windows

Вызов функции библиотеки подсистемы

Возможные сценарии вызовов функций DLL подсистемы

1. Функция полностью реализована в DLL (`GetCurrentProcess()`, `GetCurrentProcessId()`).
2. Требуется вызов (вызовы) исполнительной системы (`ReadFile()`, `WriteFile()`).
3. Требуется работа в процессе подсистемы окружения (клиент-серверный вызов, DLL ждёт).
4. Одновременно (2) и (3) (`CreateProcess()`, `CreateThread()`).

Функции процесса подсистемы окружения

Функции процесса `Csrss.exe`

- Создание /завершение процессов и потоков.
- Назначение букв сетевым дискам (`WNetAddConnection2()`, `WNetAddConnection3()`).
- Получение имён временных файлов (`GetTempFileName()`).

Функции компонентов подсистемы

Функции библиотек процесса подсистемы окружения (`Basesrv.dll`, ...)

- Создание/удаление процессов и потоков.
- Части реализации виртуальной машины DOS (Windows 32).
- Поддержка сборок SxS и манифеста.
- Различные функции (`GetTempFileName()`, `ExitWindowsEx()` ...)

Функции драйвера `Win32k.sys`

- Диспетчер окон.
- GDI.
- Обёртки над DirectX, реализованной в другом драйвере.

Функции библиотеки поддержки системы

Функции библиотеки поддержки системы (`Ntdll.dll`)

- Вызов функций исполнительной подсистемы (`Ntoskrnl.exe`) (`NtCreateFile()`, ...)
- Собственные функции:
  - Загрузка образа (PE — Portable Executable).
  - Работа с кучами.
  - Взаимодействие с подсистемой Windows (`Csrss.exe`).
  - Общие функции поддержки выполнения (часть C runtime library).
  - Отладка в режиме пользователя.
  - Диспетчер APC (Asynchronous Procedure Call, аснихронный вызов процедур) и исключений режима пользователя.
  - Механизм журналирования событий ПО (ETW — Event Tracing for Windows).

Функции исполнительной системы

Виды функций исполнительной системы (`Ntoskrnl.exe`)

- Системные сервисы: экспортируемые, доступные из режима пользователя.
- Функции драйверов устройств (`DeviceIoControl()`).
- Экспортируемые, доступные из режима ядра, документированные в WDK (Windows Driver Kit).
- Экспортируемые, доступные из режима ядра, не документированные в WDK (видеодрайвером при загрузке, ...)
- Не экспортируемые.

Функции исполнительной системы (продолжение)

Компоненты исполнительной системы (`Ntoskrnl.exe`)

- Диспетчер конфигурации.
- Диспетчер процессов.
- Монитор состояния защиты (SRM — Security Reference Monitor).
- Диспетчер ввода/вывода.
- Диспетчер PnP (Plug and Play).
- Диспетчер питания.
- Подпрограммы инструментирования управления Windows (WMI — WDM Windows Management Instrumentation).
- Диспетчер кеша.
- Диспетчер памяти.
- Средство логической предвыборки.
- Функции поддержки вышеперечисленных компонент.

Функции исполнительной системы (продолжение)

Функции поддержки компонент исполнительной системы (`Ntoskrnl.exe`)

- Диспетчер объектов.
- Передовой локальный вызов процедур (Advanced Local Procedure Call, ALPC).
- Общие функции библиотеки времени выполнения.

Функции исполнительной системы (окончание)

Другие функции исполнительной системы (`Ntoskrnl.exe`)

- Библиотека отладки ядра.
- Каркас отладки режима пользователя.
- Механизм транзакций ядра.
- Библиотека гипервизора (поддержка системы Hyper-V)
- Диспетчер ошибок (errata).
- Система проверки драйверов (Driver Verifier).
- Трассировка событий для Windows.
- Инфраструктура диагностики Windows.
- Архитектура аппаратных ошибок Windows.
- Библиотека времени исполнения файловых систем.

Функции ядра

Функции ядра (`Ntoskrnl.exe`)

- Объекты ядра:
  - Объекты управления.
  - Объекты диспетчеризации.
- Область ядра, относящаяся к управлению процессором (Kernel Processor Control Region — KPCR), $\supset$ блок управления (Kernel Processor Control Block — KPRCB).
- Поддержка оборудования.

Функции HAL

Функции (Hal.dll)

- Интерфейсы ввода/вывода;
- Контроллеры прерываний;
- Механизмы межпроцессорного взаимодействия;
- ...

HalAllocateCommonBuffer()	HalGetBusDataByOffset()
HalAllocateHardwareCounters()	HalGetDmaAlignmentRequirement()
HalAssignSlotResources()	HalGetInterruptVector()
HalExamineMBR()	HalReadDmaCounter()
HalFreeCommonBuffer()	HalReturnToFirmware()
HalFreeHardwareCounters()	HalSetBusData()
HalGetAdapter()	HalSetBusDataByOffset()
HalGetBusData()	HalTranslateBusAddress()

Таблица 2: функции, экспортируемые HAL (Hal.dll)

Драйверы устройств

Работа драйверов устройств

- В контексте потока, инициировавшего запрос ввода/вывода.
- В контексте системного потока режима ядра.
- В результате прерывания.

Виды драйверов устройств

- Аппаратных устройств.
- Файловой системы.
- Фильтра файловой системы.
- Сетевые перенаправители и серверы.
- Драйверы протоколов.
- Потоковых фильтров ядра.

Модель WDM

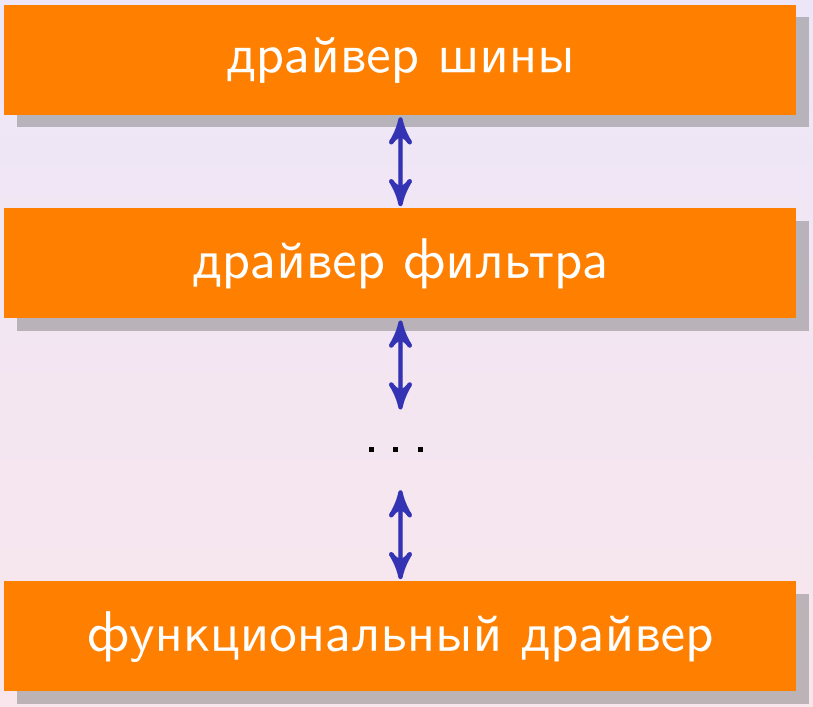

Рис. 1: схема модели драйверов WDM (Windows Driver Model)

Каркас WDF

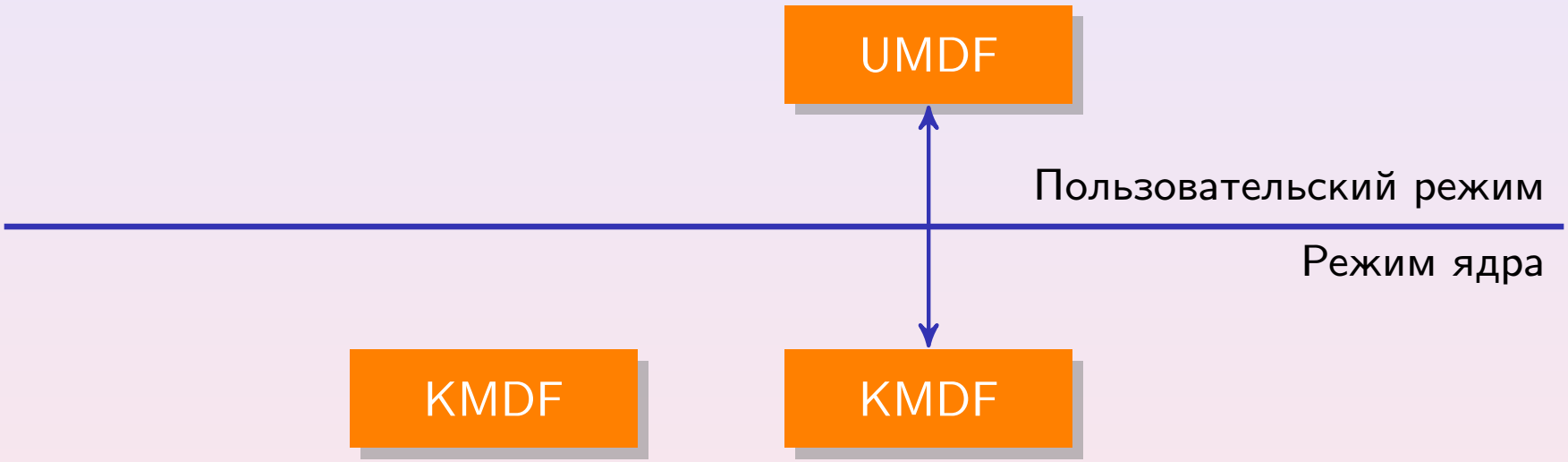

Рис. 2: схема WDF (Windows Driver Foundation)

Обзор системных процессов

Имя файла	Назначение
—	Процесс простоя (Idle)
—	Системный процесс (System)
`smss.exe`	Диспетчер сеансов.
`lsm.exe`	Диспетчер локальных сеансов.
`Csrss.exe`	Подсистема Windows.
`WinInit.exe`	Процесс инициализации.
`WinLogon.exe`	Процесс обработки входа в систему.
`Services.exe`	Диспетчер управления службами (+ дочерние службы, ∋ `Svchost.exe`).
`lsass.exe`	Сервер проверки подлинности локальной системы безопасности.

Таблица 3: системные процессы

Взаимоотношения между системными процессами

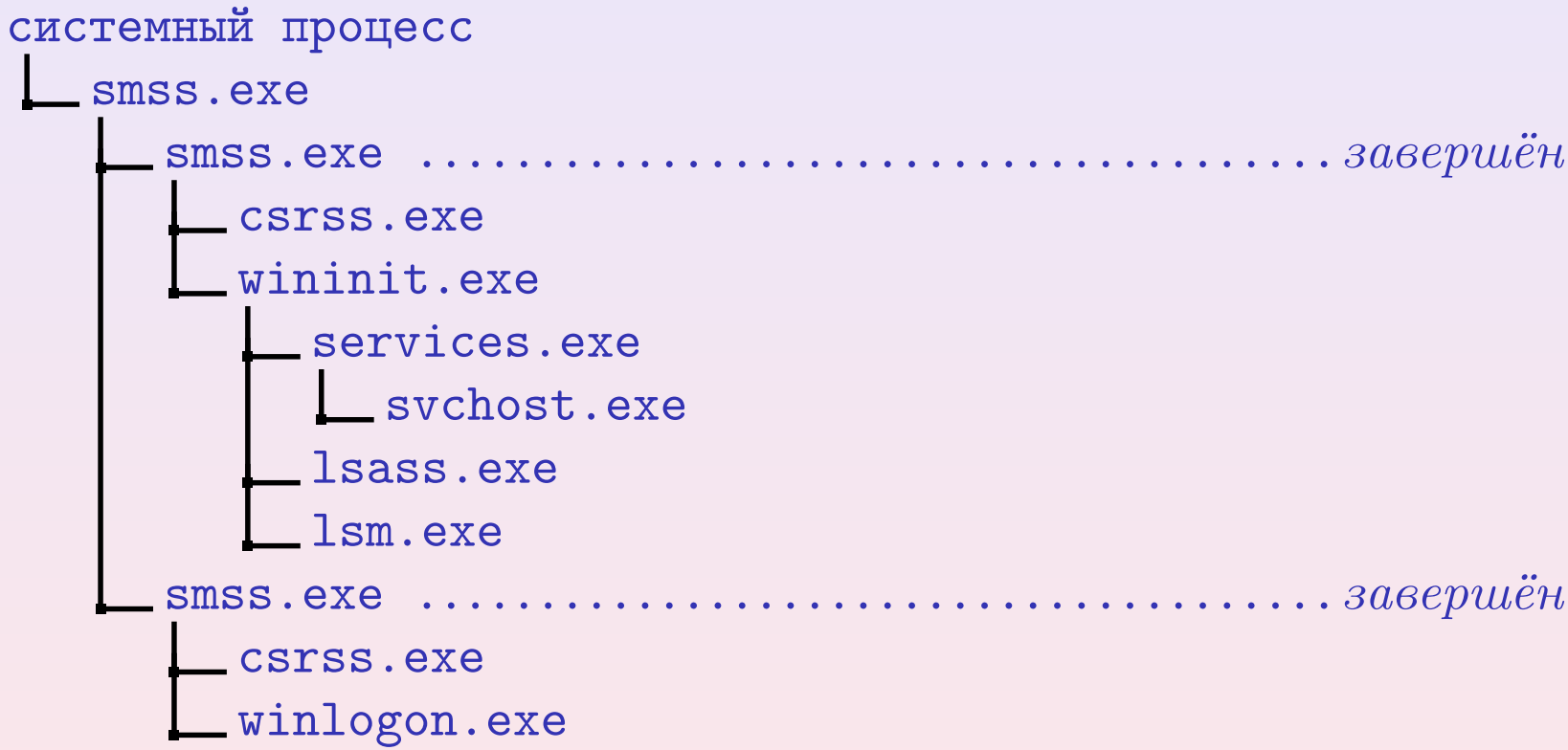

Рис. 3: родительски-дочерние отношения между системными процессами

Системные потоки

Особенности работы системных потоков

- Создаются с помощью функции `PsCreateSystemThread()` (`Ntoskrnl.exe`) системой и драйверами.
- Работают в общем адресном пространстве ядра.
- Должны выделять память из куч системы (подкачиваемый, неподкачиваемый набор).
- По умолчанию принадлежат системному процессу.

Диспетчер сеансов

Запуск диспетчера сеансов (`smss.exe`)

- Создаётся системным потоком, выполняющим последнюю стадию инициализации исполнительной среды и ядра.
- Проверяет, запущен ли первым (по параметрам командной строки).
- Создаётся максимум 4 сеанса + 1 на каждый дополнительный процессор во время загрузки и создания терминальных сеансов.
- По завершении инициализации сеанса завершается.

Работа диспетчера сеансов

Работа первого экземпляра диспетчера сеансов (smss.exe)

1. Помечает процесс и исходный поток как критический
2. Увеличивает базовый приоритет процесса до 11.
3. Создаёт именованные каналы и почтовые ящики для взаимодействия с Smss, Csrss, Lsm.
4. Создаёт порт ALPC для приёма команд.
5. Создаёт глобальные переменные окружения из реестра.
6. Создаёт символические ссылки на устройства DOS (NUL, PRN, ...)
7. Создаёт глобальный каталог \Sessions в пространстве имён диспетчера объектов.
8. Запускает программы в ветке ...\BootExecute (Autochk.exe).
9. Выполняет отложенные переименования файлов.

Работа диспетчера сеансов (продолжение)

Работа первого экземпляра диспетчера сеансов (окончание)

10. Инициализирует файлы подкачки.
11. Инициализирует оставшуюся часть реестра.
12. Запускает программы в ветке `...\SetupExecute`.
13. Открывает известные библиотеки `...\KnownDLLs` и отображает их в постоянные области.
14. Создаёт поток, отвечающий на запросы создания сеансов.
15. Запускает `Smss.exe` для инициализации неинтерактивного сеанса 0.
16. Запускает `Smss.exe` для инициализации интерактивного сеанса 1.
17. Ждёт завершения `Csrss.exe` сеанса 0.

Работа диспетчера сеансов (окончание)

Работа экземпляра диспетчера сеансов, создающего сеанс

1. Вызывает функцию `NtSetSystemInformation()` для установки структур данных сеанса на уровне ядра. Это приводит к вызову `MmSessionCreate()`, настраивающему виртуальное адресное пространство сеанса, ∋ подкачиваемые наборы, структуры сеанса внутри `Win32k.sys` и других драйверов пространства сеанса.
2. Создаёт процессы сеанса (По умолчанию, `Csrss.exe`).
3. Создаёт `Wininit.exe` для сеанса 0 или `Winlogon.exe` для интерактивных.

Оконная станция и рабочий стол

Определения

Оконная станция: (Window Station) — ∋

- буфер обмена;
- таблицу атомов;
- один или больше рабочих столов.

Рабочий стол: (Desktop) — ∋

- логическую поверхность экрана;
- элементы пользовательского интерфейса: окна, меню, средства захвата.

Работа процесса инициализации

Работа процесса инициализации (`Wininit.exe`)

1. Помечает свой процесс как критический.
2. Инициализирует инфраструктуру планирования режима пользователя (волокна — fibres, потоки UMS).
3. Создаёт глобальный каталог для временных файлов.
4. Создаёт оконную станцию (`Winsta0`) и два рабочих стола (`Winlogon`, `Default`) для работы процессов в сеансе 0.
5. Запускает диспетчер управления службами (`Services.exe`).
6. Запускает сервер подсистемы аутентификации локальной безопасности (`Lsass.exe`).
7. Запускает диспетчер локальных сеансов (`Lsm.exe`).
8. Ждёт завершения работы системы.

Диспетчер локальных сеансов

Функции диспетчера локальных сеансов (`Lsm.exe`)

Управление сеансами терминального сервера на локальной системе:

- Запросы к `Smss.exe` для запуска новых сеансов (с созданием процессов, например, `Csrss.exe` и `Winlogon.exe`) через порт ALPC.
- Оповещает `Csrss.exe` через локальный RPC о событиях:
  - подключение/отключение;
  - завершение;
  - широковещательная рассылка системного сообщения.
- Получает оповещения от `Winlogon.exe` через RPC:
  - вход/выход пользователя;
  - запуск/завершение оболочки;
  - подключение/отключение от сеанса;
  - блокировка/разблокировка рабочего стола.

Работа процесса входа

Активация процесса входа (`Winlogon.exe`)

- Запрос пользователя на вход.
- Нажатие комбинации SAS (Security Attention Sequence).

Работа процесса входа

1. Запускает `LogonUI.exe`, инициализирующий поставщиков удостоверений.
2. Дополнительно может загружать библиотеки сетевых поставщиков, выполняющих вторичную аутентификацию.
3. Отправляет удостоверение `Lsass.exe`, который при успехе создаёт маркер доступа для пользователя (второй при UAC).
4. С полученным маркером создаёт процессы пользователя (по умолчанию, `Userinit.exe`, выполняющий инициализацию среды и запуск оболочки, по умолчанию, `Explorer.exe`).