# Лекция №3

*Управление памятью в ОС Windows*

# Основные понятия

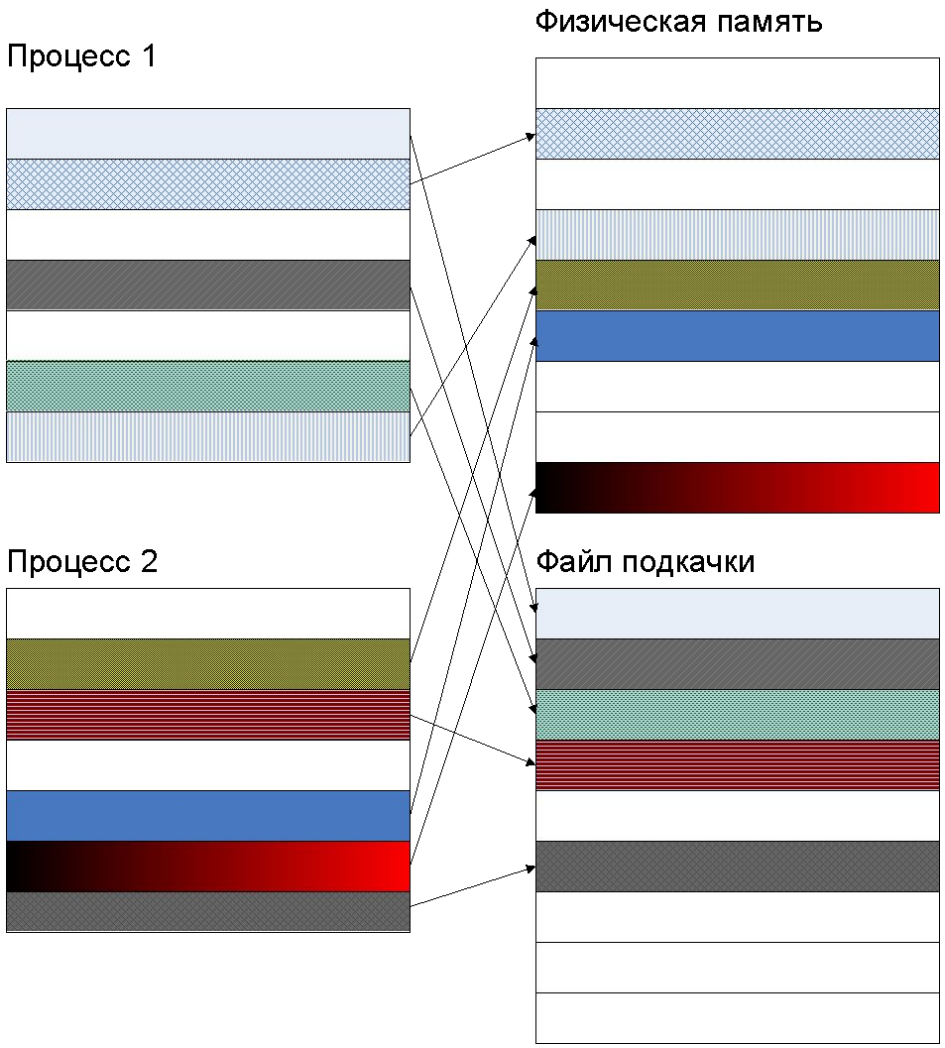
- ***Физическая память*** представляет собой упорядоченное множество ячеек и все они пронумерованы, то есть с каждой из них можно обратиться, указав ее порядковый номер (адрес). Количество ячеек физической памяти ограничено и фиксировано
- ***Виртуальная память*** создает иллюзию того, что каждый процесс имеет доступ к 4Гб непрерывного адресного пространства
- ***Виртуальное адресное пространство процесса*** является набором адресов, доступным всем потокам этого процесса
- ОС распределяет адресное пространство физической и виртуальной памяти **страницами** (pages) – блоками по 4Кб

Механизмы управления памятью решают две главные задачи:

- **Трансляция**, или проецирование, виртуального адресного пространства процесса на физическую память. Это позволяет ссылаться на корректные адреса физической памяти, когда нити, выполняемые в контексте процесса, читают и записывают его в виртуальном адресном пространстве
- **Подкачка части содержимого памяти на диск**, когда нити или системный код пытаются задействовать больший объем физической памяти, чем тот, который имеется в наличии, и загрузка страниц обратно в физическую память по мере необходимости

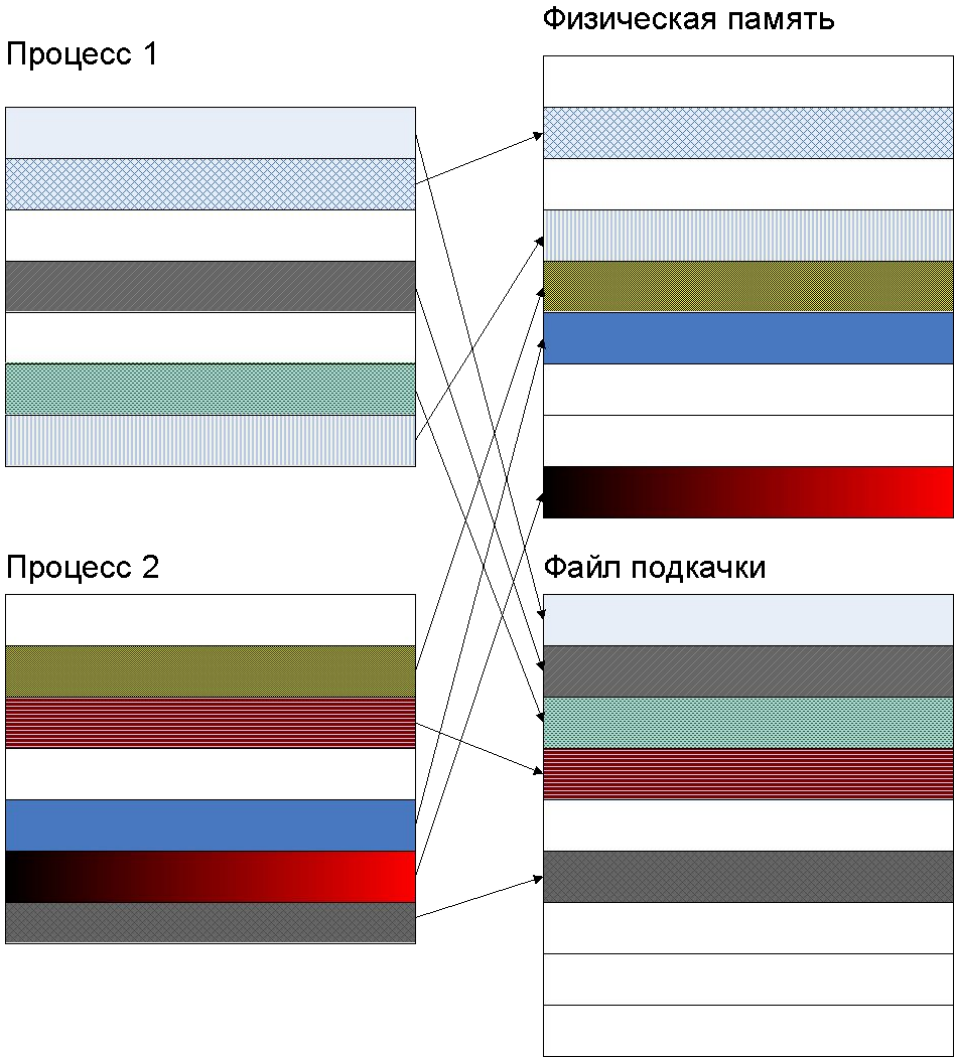
*Физическое подмножество виртуального адресного пространства процесса называется* ***рабочим набором (working set)***

# Организация виртуальной памяти



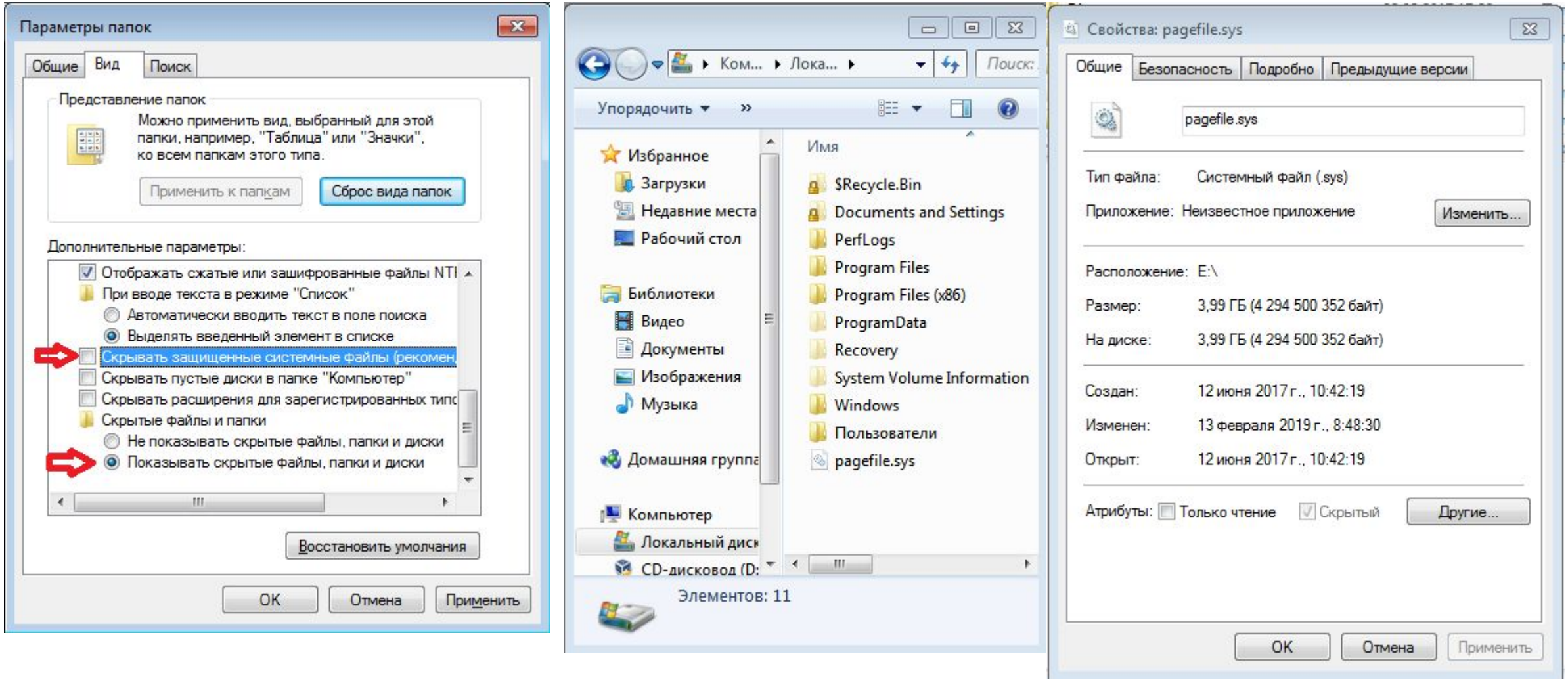

- В период выполнения *диспетчер памяти (в Ntoskrnl.exe)*, транслирует, или проецирует (maps) виртуальные адреса на физические, по которым хранятся данные
- Подкачка данных на диск освобождает физическую память для других процессов или самой ОС
- Диспетчер памяти опирается на аппаратную поддержку механизма подкачки
- В процессе работы система виртуальной памяти использует один или несколько файлов подкачки, расположенных на жестком диске (pagefile.sys)

# Страницы виртуальной памяти имеют три состояния:



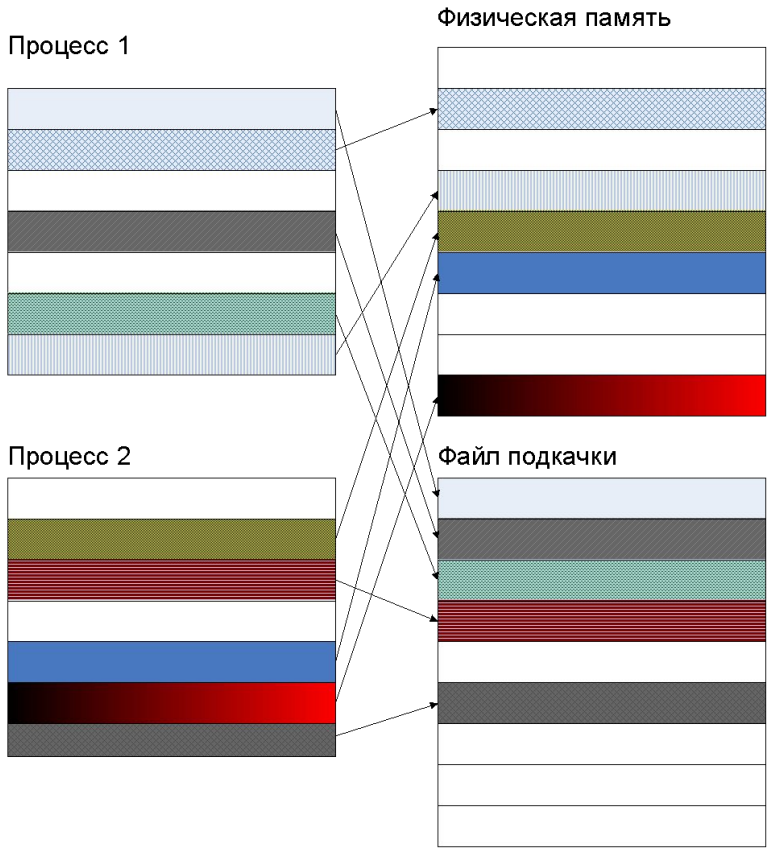

- Большинство страниц пусто, поскольку процесс их не использует, они никуда не отображаются
- *Используемые страницы* отображаются с помощью невидимого для процесса указателя в область физической оперативной памяти (ОЗУ)
- Некоторые страницы, *к которым не было обращений в течение определенного времени*, отображаются с помощью невидимого для процесса указателя в 4Кб раздел файла подкачки (pagefile.sys)

# Файл подкачки




**pagefile.sys** может быть создан на каждом логическом диске системы, находится в корне дерева каталогов и является *скрытым* и *системным* файлом

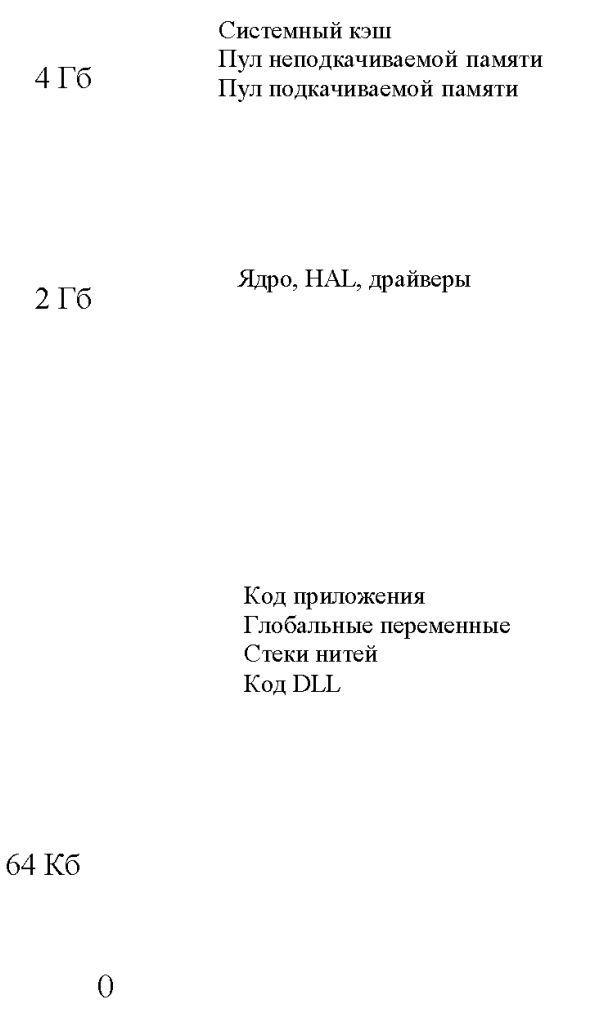
# Процесс управления местоположением страниц – в ОЗУ или в страничном файле называется ***подкачкой страниц по запросу***




- Приложение делает попытку сохранить данные в памяти
- Диспетчер виртуальной памяти перехватывает запрос и определяет, сколько страниц необходимо для его выполнения. После этого он отображает неиспользуемую физическую память на нужные незанятые области виртуального пространства процесса. При этом диспетчер виртуальной памяти скрывает от приложения (процесса) способ организации физической памяти. Когда приложение обратится к конкретному виртуальному адресу, он будет транслирован в уникальный физический адрес, не конфликтующий с другими процессами
- Если в системе не хватает физической памяти, диспетчер виртуальной памяти выполняет поиск страниц ОЗУ, не использовавшихся в течение определенного времени, и копирует эти страницы в страничный файл (pagefile.sys), находящийся на жестком диске. Освободившаяся область ОЗУ отображается на виртуальное адресное пространство запросившего память процесса

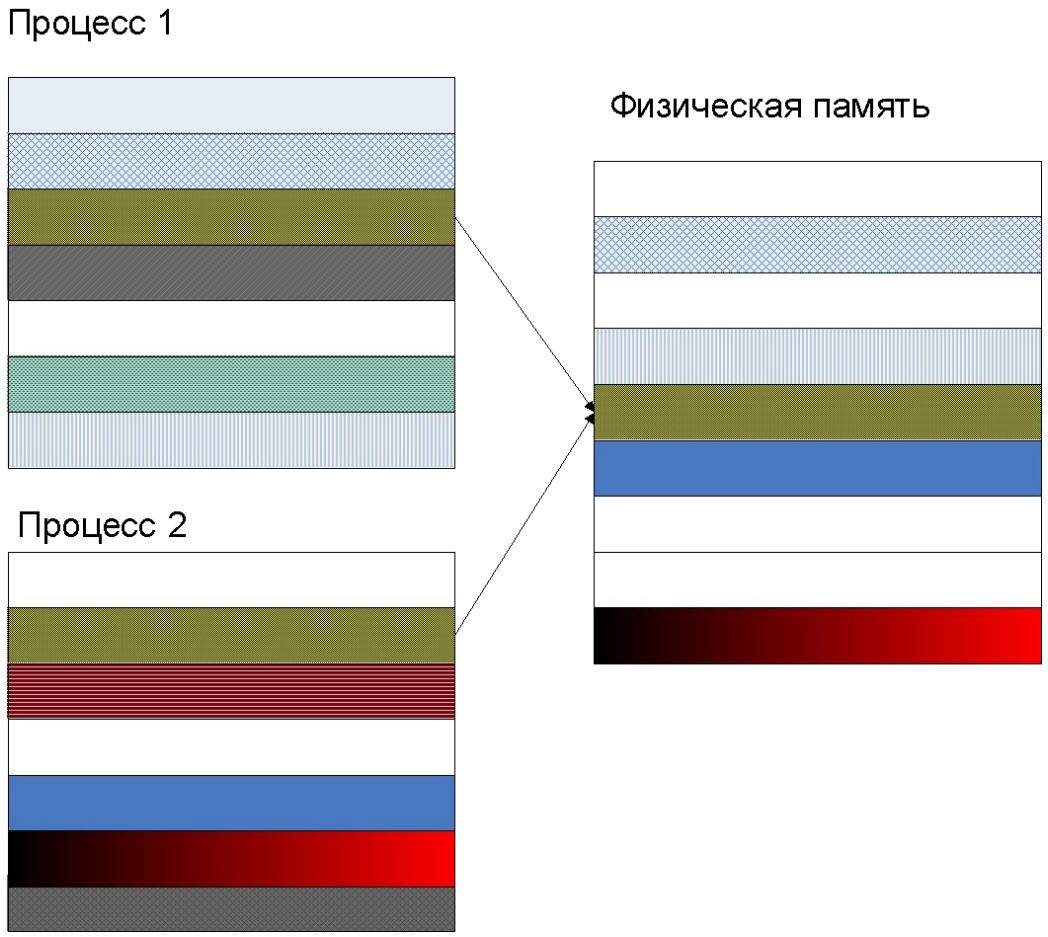
# Структура адресного пространства пользовательского процесса

## Системная память



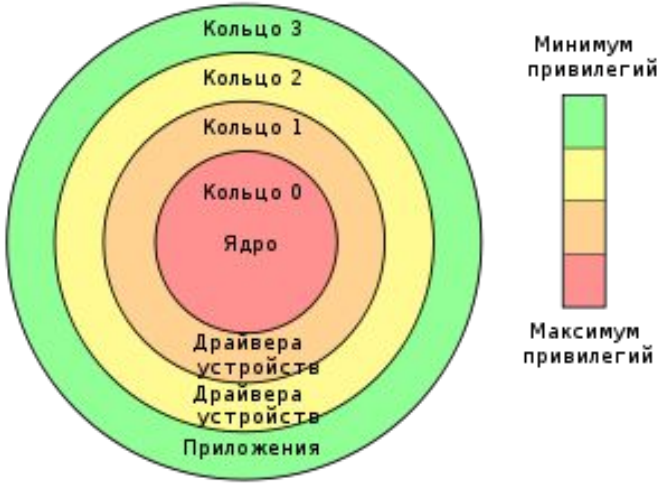

- По умолчанию каждый пользовательский процесс в Windows 32-bit получает закрытое адресное пространство размером до 2 Гб, а остальные 2 Гб занимает ОС
- Код ОС  отображается в верхние 2Гб виртуального адресного пространства процесса
- При инициализации системы диспетчер памяти создает два типа динамических пулов памяти, используемых компонентами режима ядра для выделения системной памяти:
- **Пул неподкачиваемой памяти** (nonpaged pool). Состоит из диапазонов системных виртуальных адресов, которые всегда присутствуют в физической памяти и доступны в любой момент.
- **Пул подкачиваемой памяти** (paged pool). Регион виртуальной памяти в системном пространстве, содержимое которого система может выгружать в страничный файл и загружать из него

# Разделение памяти процессами. Разделяемая память




- *Разделяемой* (shared memory) называется память, видимая более, чем одному процессу или присутствующая в виртуальном пространстве более, чем одного процесса.
- Например, если два процесса используют одну и ту же DLL, имеет смысл загрузить ее код в физическую память один раз и сделать ее доступной всем процессам, в виртуальной памяти которых присутствует эта DLL.
- Каждый процесс поддерживает закрытые области памяти для хранения собственных данных, но программный код и страницы немодифицируемых данных в принципе

# Режим ядра и пользовательский режим




Кольца привилегий архитектуры x86 в защищённом режиме

- Windows и Linux используют два режима доступа к процессору:
  - пользовательский (user mode – кольцо 3)
  - ядра (kernel mode - кольцо 0)
- Код приложений работает в пользовательском режиме, тогда как код ОС (например, системные сервисы и драйверы устройств) – в режиме ядра (режим супервизора)
- В режиме ядра предоставляется доступ ко всей системной памяти и разрешается выполнять любые машинные команды процессора
- Хотя каждый Win32-процесс имеет свою (закрытую) память, код ОС и драйверы устройств, работающие в режиме ядра, делят единое виртуальное адресное пространство
- Каждая страница в виртуальной памяти помечается тэгом, определяющим, в каком режиме должен работать процессор для чтения и/или записи данной страницы
- Страницы в системном пространстве доступны лишь в режиме ядра, а все страницы в пользовательском адресном пространстве – в пользовательском режиме. Страницы только для чтения (например, содержащие лишь исполняемый код) ни в каком режиме для записи недоступны

# Режим ядра и пользовательский режим

- Windows не предусматривает никакой защиты системной памяти от компонентов, работающих в режиме ядра. Иначе говоря, код ОС и драйверов устройств в режиме ядра получает полный доступ к системной памяти и может обходить средства защиты Windows для обращения к любым объектам
- Надо быть осторожным при загрузке драйвера устройства от стороннего поставщика: перейдя в режим ядра, он получит полный доступ ко всем данным ОС. Такая уязвимость стала одной из причин, по которой в Windows 2000 был введен механизм проверки цифровых подписей драйверов, предупреждающий пользователя о попытке установки неавторизированного (неподписанного) драйвера
- Режим гипервизора (Hypervisor mode), который называют Ring -1, реализуется с целью поддержки технологий виртуализации на уровне аппаратного обеспечения. Это позволяет достигнуть *одновременного выполнения нескольких операционных систем на одном процессоре* без существенных потерь производительности. При выполнении привилегированных операций операционными системами в режиме *супервизора* управление передается специальной программе — *гипервизору*.
- Гипервизор осуществляет *арбитраж* использования имеющихся аппаратных ресурсов несколькими операционными системами аналогично тому как сами операционные системы осуществляют *распределение ресурсов между несколькими задачами*. По сути, гипервизор обычно является *небольшим ядром*, которое управляет распределением ресурсов между несколькими операционными системами и работает уровнем ниже, чем сами операционные системы.