Память процесса
Виртуальное адресное пространство

Память программы это массив с последовательной нумерацией ячеек. Индекс байта в этом огромном массиве и называется его **адресом**, а весь массив целиком - **памятью программы**.

Память программы не тождественна объему оперативной памяти. Программа работает исключительно с так называемой "виртуальной памятью". Виртуальная память - это имитация реальной памяти. Она позволяет каждой программе: считать, что:

1) установлено максимальное теоретически возможное количество оперативной памяти;

2) считать, что она является единственной программой, запущенной на машине.

- Для 32 бит систем 2^32 = 4 Гб
- Для 64 бит систем 2^64 = 16 Эб

4 Гб или 16 Эб есть **у каждой программы**. Т.е. каждой программе отводится своё личное закрытое адресное пространство. Такая изолированность означает, что программа А в своем адресном пространстве может хранить какую-то запись данных по адресу $12345678, и одновременно у программы В по тому же адресу $12345678 (но уже в его адресном пространстве) может находиться совершенно иная запись данных.

Если программы выделяют в их адресных пространствах больше памяти, чем есть в системе физической памяти, то часть памяти из ОЗУ переносится на диск ("винчестер") - в т.н. файл подкачки (его ещё называют страничным файлом, page file, SWAP-файлом или "свопом"). Когда программа обращается к своим данным, которые были выгружены на диск, то операционная система автоматически загрузит данные из файла подкачки в ОЗУ. И всё это происходит под капотом - т.е. совершенно незаметно для программы. С точки зрения программы, ей кажется, что она работает с 4 Гб или 16 Эб RAM.

Адресное пространство, хотя действительно однородно и непрерывно более чем на 99%, но в нём есть несколько специальных областей недоступных пользовательским процессам.

Гранулярность выделения памяти равна именно **64 Кб**

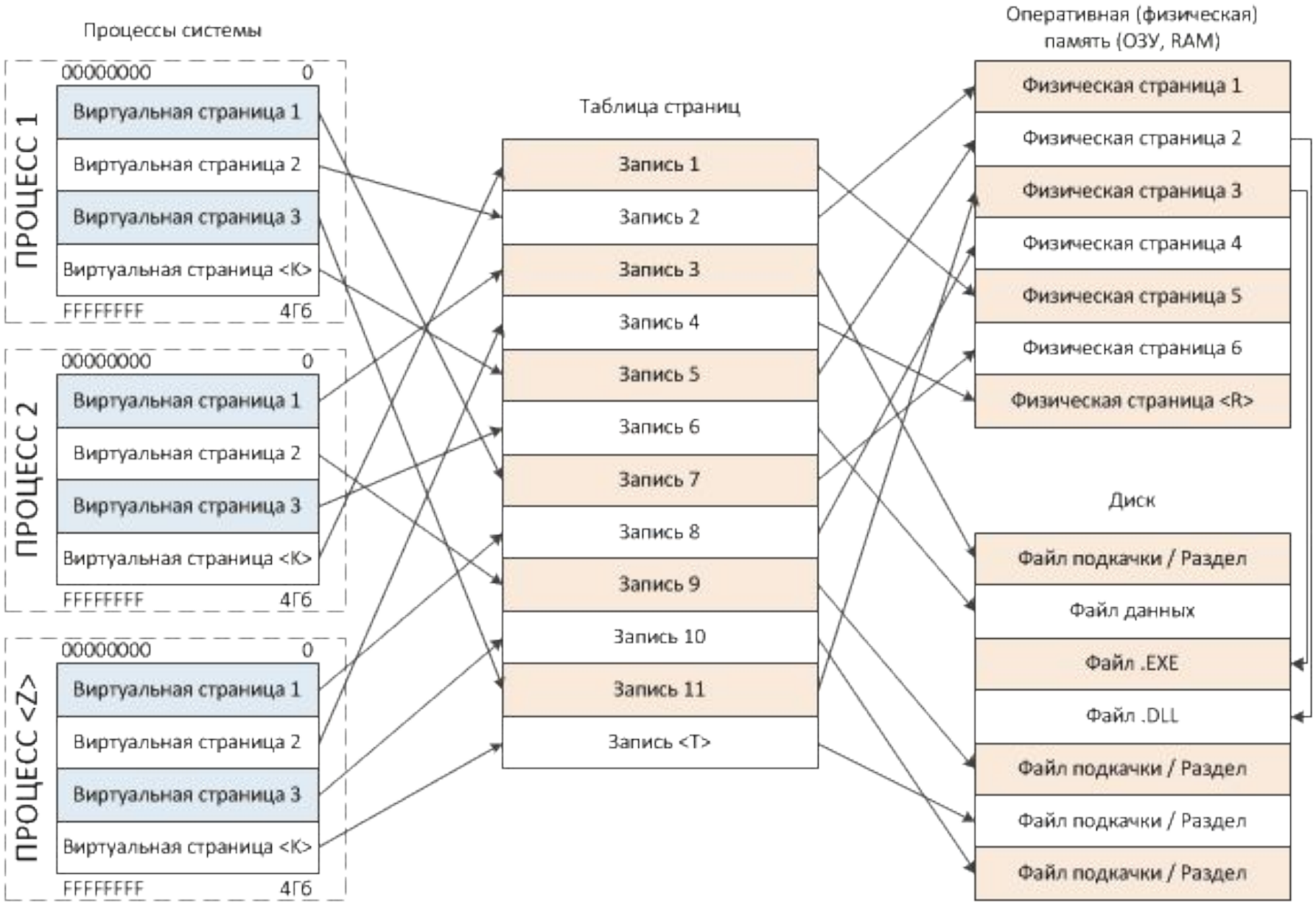
В виртуальном адресном пространстве каждой программы сосуществуют сама программа и операционная система. Та часть, где работает ваша программа (и о котором мы говорили всё это время выше), называется разделом для кода и данных пользовательского режима (user mode). Та часть, где работает операционная система, называется разделом для кода и данных режима ядра (kernel mode). Обе эти части находятся в едином адресном пространстве программы.

То есть, код и данные, которые в данный момент обрабатываются/исполняются, физически располагаются в ОЗУ. Довольно важный момент для осознания того, как же работает ОС с памятью.

Сопоставлением (отображением) виртуальных адресов на физические адреса ОЗУ или файла подкачки занимается так называемый "**диспетчер виртуальной памяти**"

Диспетчер виртуальной памяти - код уровня ядра, который служит для организации подсистемы виртуальной памяти, создания таблицы адресов для процессов, организации общего доступа к памяти, осуществления защиты на уровне страниц, поддержки возможность отображения файлов на память, распределения физической памяти между процессами, организации выгрузки/загрузки страниц между физической памятью и файлом подкачки, обеспечения всех процессом достаточным для функционирования объемом физической памяти.

- Виртуальная память делится на блоки одинакового размера – виртуальные страницы. В Windows страницы бывают большие (x86 – 4 МБ, x64 – 2 МБ) и малые (4 КБ). Физическая память (ОЗУ) также делится на страницы точно такого же размера, как и виртуальная память. Общее количество малых виртуальных страниц процесса в 32 разрядных системах равно 1 048 576 (4 ГБ / 4 КБ = 1 048 576).


Процессы системы
ПРОЦЕСС 1
00000000
0
Виртуальная страница 1
Виртуальная страница 2
Виртуальная страница 3
Виртуальная страница <K>
FFFFFFFF
4ГБ
ПРОЦЕСС 2
00000000
0
Виртуальная страница 1
Виртуальная страница 2
Виртуальная страница 3
Виртуальная страница <K>
FFFFFFFF
4ГБ
ПРОЦЕСС <Z>
00000000
0
Виртуальная страница 1
Виртуальная страница 2
Виртуальная страница 3
Виртуальная страница <K>
FFFFFFFF
4ГБ
Таблица страниц
Запись 1
Запись 2
Запись 3
Запись 4
Запись 5
Запись 6
Запись 7
Запись 8
Запись 9
Запись 10
Запись 11
Запись <T>
Оперативная (физическая) память (ОЗУ, RAM)
Физическая страница 1
Физическая страница 2
Физическая страница 3
Физическая страница 4
Физическая страница 5
Физическая страница 6
Физическая страница <R>
Диск
Файл подкачки / Раздел
Файл данных
Файл .EXE
Файл .DLL
Файл подкачки / Раздел
Файл подкачки / Раздел
Файл подкачки / Раздел

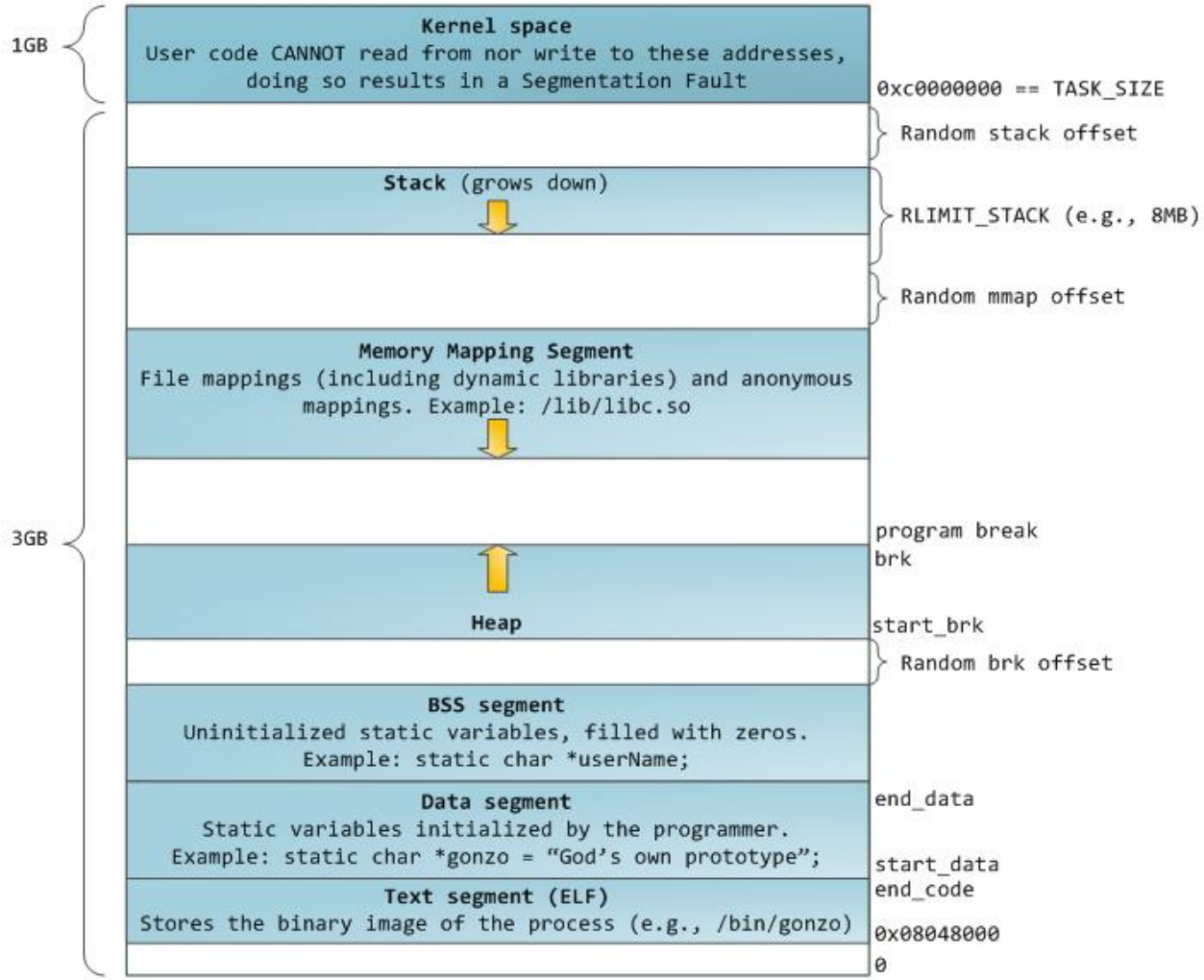
Синхронизация между виртуальными и физическими страницами памяти поддерживается аппаратно на уровне процессора, и называется трансляцией адресов. Данная технология "на лету" преобразует виртуальный адрес в физический и обратно. Отсюда следует немаловажный вывод:

Некоторые механизмы, которые использует диспетчер виртуальной памяти в операционной системе, имеют аппаратную поддержку на уровне процессора.

Основные концепции виртуальной памяти таковы:

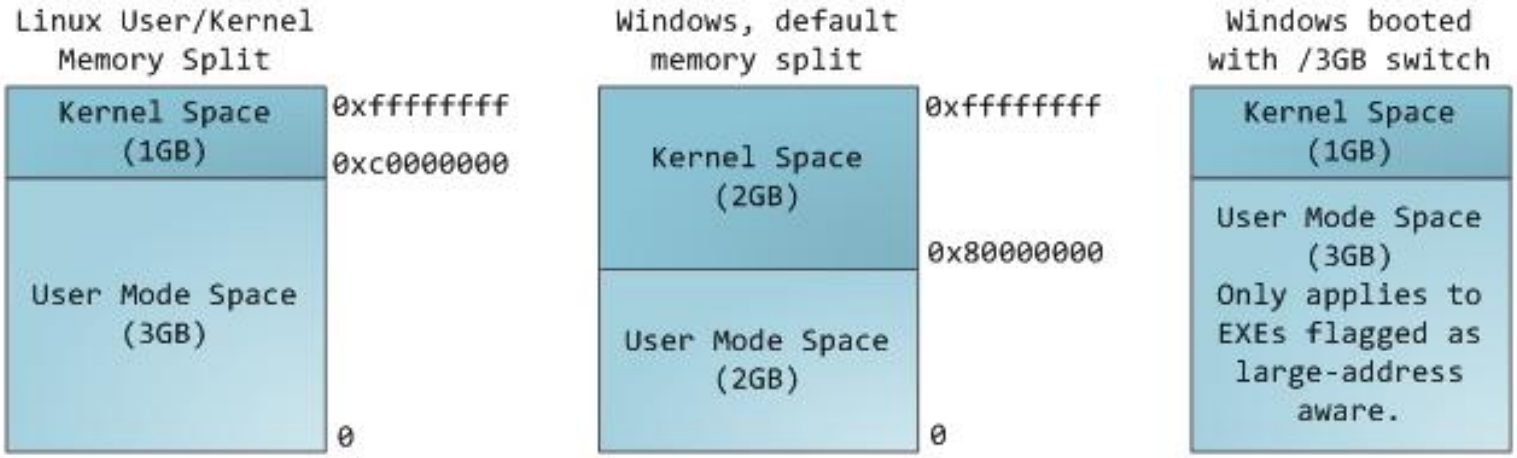
- Память, доступная программе, никак не связана с физической памятью.
- Каждая программа работает в своей виртуальном адресном пространстве. Размер этого пространства может быть больше размера установленной в машине оперативной памяти.
- Адресное пространство процесса (программы) изолировано от других подобных адресных пространств.

Начало	Конец	Размер	Описание
00000000	0000FFFF	64Кб	Область нулевых указателей. Зарезервировано. Данная область всегда маркируется как свободная (Free). Область применяется для выявления некорректных, нулевых указателей в коде.. Данная особенность позволяет программистам обнаруживать баги в коде.
00010000	7FFEFFFF	2Гб (3Гб)	Пользовательский режим (User mode). Пользовательская часть кода и данных. В это пространство загружается пользовательское приложение, с разбивкой по секциям. Отображаются все проецируемые в память файлы, доступные данному процессу. В этом пространстве создаются пользовательская часть стеков потоков приложения. Тут присутствуют основные системные библиотеки ntdll.dll, kernel.dll, user32.dll, gdi32.dll.
7FFF0000	7FFFFFFF	64Кб	Область некорректных указателей. Зарезервировано. Данная область всегда маркируется как свободная (Free). Область применяется для выявления некорректных, вышедших за пределы пользовательской памяти, указателей в коде. Попытка чтения/записи по


1GB
3GB
Kernel space
User code CANNOT read from nor write to these addresses, doing so results in a Segmentation Fault
0xc0000000 == TASK_SIZE
Random stack offset
Stack (grows down)
RLIMIT_STACK (e.g., 8MB)
Random mmap offset
Memory Mapping Segment
File mappings (including dynamic libraries) and anonymous mappings. Example: /lib/libc.so
program break
brk
Heap
start_brk
Random brk offset
BSS segment
Uninitialized static variables, filled with zeros.
Example: static char *userName;
Data segment
Static variables initialized by the programmer.
Example: static char *gonzo = "God's own prototype";
end_data
start_data
Text segment (ELF)
Stores the binary image of the process (e.g., /bin/gonzo)
end_code
0x08048000
0


Пользовательское ВАП
0000 0000
Исполняемый образ
Библиотеки DLL
Куча процесса
Стеки потоков
7FFF FFFF
Системное ВАП
8000 0000
Ядро
Исполнительная система
HAL
Загрузочные драйверы
C000 0000
Таблицы страниц
C040 0000
C080 0000
Системный кэш
Пул подкачиваемой памяти
Пул неподкачиваемой памяти
FFC0 0000
FFFF FFFF

- В 32 разрядных Windows пользовательском ВАП располагаются исполняемый образ процесса, динамически подключаемые библиотеки (DLL, dynamic-link library), куча процесса и стеки потоков.
- При запуске программы создается, при этом в память загружаются код и данные программы (исполняемый образ, executable image), а также необходимые программе динамически подключаемые библиотеки (DLL). Формируется куча (heap) – область, в которой процесс может выделять память динамическим структурам данных (т. е. структурам, размер которых заранее неизвестен, а определяется в ходе выполнения программы). По умолчанию размер кучи составляет 1 МБ, но при компиляции приложения или в ходе выполнения процесса может быть изменен. Кроме того, каждому потоку предоставляется стек (stack) для хранения локальных переменных и параметров функций, также по умолчанию размером 1 МБ.


Linux User/Kernel
Memory Split
Kernel Space
(1GB)
User Mode Space
(3GB)
0xffffffff
0xc0000000
0
Windows, default
memory split
Kernel Space
(2GB)
User Mode Space
(2GB)
0xffffffff
0x80000000
0
Windows booted
with /3GB switch
Kernel Space
(1GB)
User Mode Space
(3GB)
Only applies to
EXEs flagged as
large-address
aware.

- **Physical Address Extension** (PAE) — режим работы встроенного блока управления памятью (PAE) — режим работы встроенного блока управления памятью x86-совместимых процессоров, в котором используются 64-битные элементы таблиц страниц (из которых для адресации используются только 36 бит), с помощью которых процессор может адресовать 64 ГБ физической памяти (вместо 4 ГБ, адресуемых при использовании 32-разрядных таблиц), хотя каждая задача (программа) всё равно может адресовать максимум до 4 ГБ виртуальной памяти

- В 32-битных клиентах Microsoft Windows (начиная с Windows XP SP2) использование 36-битного PAE включается ключом /PAE в файле boot.iniВ 32-битных клиентах Microsoft Windows (начиная с Windows XP SP2) использование 36-битного PAE включается ключом /PAE в файле boot.ini, однако, максимальный физический адрес доступного операционной системе ОЗУ искусственно ограничен на уровне ядра по маркетинговым соображениям[8]. В Windows XP это ограничение составляет 4 ГБ, 32-разрядный Windows Server 2003 Enterprise Edition поддерживает до 64 ГБ. Существуют программы, позволяющие обойти ограничение на доступную память[9], но их использование является нарушением лицензионного соглашения Microsoft[10]. В случае 4 ГБ ОЗУ, память можно использовать в Windows XP почти полностью, переместив системные области адресного пространства выше 4 ГБ, такую функцию поддерживают некоторые версии BIOS. Тем не менее, согласно заявлениям Microsoft, введение 4 ГБ ограничения адресного пространства связано с отсутствующей или плохой поддержкой 36-битного адресного пространства некоторыми драйверами устройств[11].

- **Address Windowing Extensions** (AWE) — программный интерфейс (AWE) — программный интерфейс в ОС (AWE) — программный интерфейс в ОС семейства Microsoft Windows, позволяющий 32-битному приложению получить доступ к оперативной памяти, размер которой превышает размеры доступного приложению виртуального адресного пространства (2-3[1] гигабайта). Процесс отображения дополнительной памяти в адресное пространство приложения при помощи AWE называется «windowing» (оконный доступ) и схож с концепцией «overlay гигабайта). Процесс отображения дополнительной памяти в адресное пространство приложения при помощи AWE называется «windowing» (оконный доступ) и схож с концепцией «overlay» используемой, например, в DOS гигабайта). Процесс отображения дополнительной памяти в адресное пространство приложения при помощи AWE называется «windowing» (оконный доступ) и схож с концепцией «overlay» используемой, например, в DOS. AWE подходит для приложений, обрабатывающих большие объёмы данных, например СУБД, научных и инженерных приложений.
- Приложение выделяет регион памяти, называемый окно (*window*) в виртуальном адресном пространстве и запрашивает при помощи интерфейса выделение одного или нескольких регионов физической памяти. Позже интерфейс позволяет приложению отобразить любой выделенный регион физической памяти на окно в адресном пространстве. Допустимо создание нескольких окон, таким образом, что суммарный размер окон не превышает размер доступного виртуального адресного пространства. Размеры окон и регионов физической памяти могут быть произвольными (кратными размеру