

# Операционные системы

Управление памятью

1	0	1		
0	0	1	1	
1	0	0	1	1
1	1	0	0	
1	0	0	1	1
0	0	1	1	
0	1	0	0	
0			0	

# Управление памятью



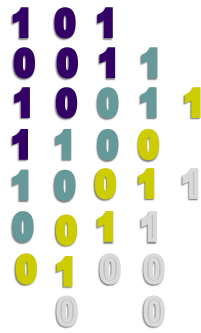
- Одна из основных задач ОС
- Память — относится к основным ресурсам вычислительной системы
- Менеджер памяти
  - Модуль ОС, который управляет памятью

# Сегментация памяти



- Одна из концепций управления памятью
- Изначально — участки памяти, хранящие информацию, которую система отображает в память нескольких процессов
- Двумерный адрес
  - Номер сегмента + смещение в сегменте
- Разные сегменты — разные типы данных
  - Атрибуты сегмента: права доступа, тип операций
- Поддержка оборудованием

# Принципы управления памятью



- Разделение памяти на физическую и логическую
  - Виртуальные адреса
- Иерархия памяти
  - Для оптимизации использования быстрых видов памяти
  - Регистры процессора → кэш процессора → оперативная память → внешняя память на жёстких дисках

# Функции ОС по управлению памятью



- Отображение логических адресов на физическую память
- Распределение памяти между конкурирующими процессами
- Защита адресных пространств процессов
- Выгрузка процессов на диск, когда в оперативной памяти
- Учёт свободной и занятой памяти

# Связывание адресов



- Этап компиляции (Compile time)
  - Когда на стадии компиляции известно точное место размещения процесса в памяти, тогда генерируются абсолютные адреса
  - Если стартовый адрес программы меняется, необходимо перекомпилировать код

# Связывание адресов



- Этап загрузки (Load time)
  - Если не известно где процесс будет размещён, компилятор генерирует перемещаемый код
  - Связывание откладывается до момента загрузки
  - Если стартовый адрес меняется, нужно перезагрузить код с учётом новой величины
- Этап выполнения (Execution time)
  - Если процесс может быть перемещён во время выполнения из одного сегмента памяти в другой, связывание откладывается до времени выполнения

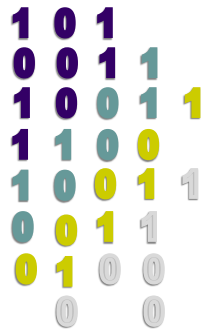
# Схемы управления памятью



- Один процесс на всю память
  - Простейший своппинг
- Фиксированные разделы
  - Процессы по очереди полностью размещаются в раздел соответствующего размера
  - Разделы создаются во время загрузки ОС
  - Являются логическими конструкциями



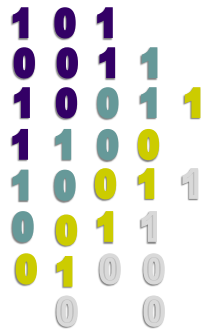




# Фиксированные разделы

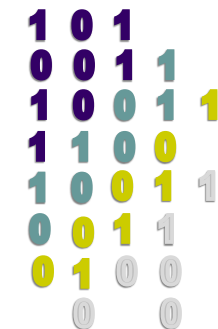
- Аналогичный подход применяется и при размещении файлов на диске
- Недостатки
  - Количество процессов ограничено количеством разделов
  - Существенная фрагментация памяти и свободного места

# Другие подходы



- Один процесс в памяти
- Расположение процесса относительно ОС
  - Сверху, снизу, посередине
  - Связано с расположением вектора прерываний
- Оверлейная структура
  - В системах с малым объёмом логического адресного пространства (MSDOS – 1Mb)
  - Удерживать в памяти только необходимые в данный момент инструкции
  - Алгоритмы перемещения и связывания, драйвер оверлеев, файлы оверлеев на ФС

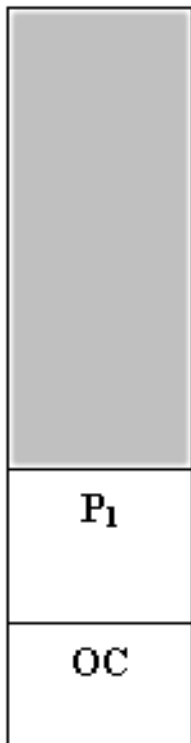
# СВОПИНГ



- В системах с разделением времени
  - Перенос памяти процесса полностью на жёсткий диск
  - Восстановление может быть в ту же область памяти или в другую
    - В зависимости от типа связывания
  - Квант времени >> время загрузки из свопа
  - Выгружается занятая память, неработающих процессов
  - Выгрузка в специальную область на жёстком диске в обход файловых систем

# Переменные разделы

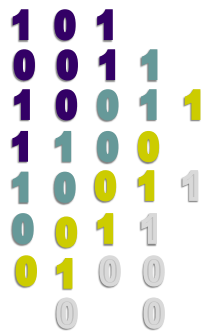
1	0	1		
0	0	1	1	
1	0	0	1	1
1	1	0	0	
1	0	0	1	1
0	0	1	1	
0	1	0	0	
				0





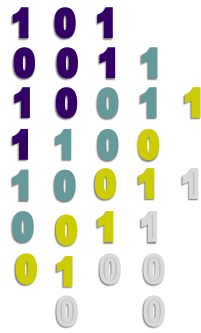
# Борьба с фрагментацией

- Не требовать непрерывности адресного пространства процесса
  - Страничная память
- Дефрагментация (сжатие)
  - Перемещение занятых(свободных) участков для объединения их в одну область
  - Схема с перемещаемыми разделами
  - Накладные расходы
  - Приводит к перебазированию кода процессов



# Проблема больших программ

- Программа размером больше свободной оперативной памяти
  - Вариант: оверлейная структура (перекрывание)
  - Предполагает участие программиста, усложнение разработки программ
  - Успехи в разработке оборудования позволил перенести эту работу на компьютер
  - 1959г — виртуальная память



# Виртуальная память

- Адресация объёма памяти больше чем существующая физическая память
- Защита памяти процессов и ОС
- Гибкое распределение процессов, запуск нескольких процессов
  - Размещение только части программы в памяти
  - Программа никак не связана с объёмом оперативной памяти компьютера
  - Меньше I/O чем при полном свопинге



# Виртуальная память



- Логические адреса программы — виртуальные адреса
  - Формируют виртуальное адресное пространство
  - Без поддержки оборудования непосредственно отображается на физическое пространство памяти
  - Аппаратная поддержка
    - Виртуальный адрес интерпретируется оборудованием, как указатель на то, где находится физическая ячейка

# Модели виртуальной памяти

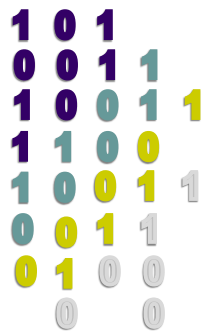


- Страничная модель
- Сегментная модель
- Сегментно-страничная модель

# Аппаратная поддержка виртуальной памяти



- Виртуальное адресное пространство любого процесса для 32 битных систем:  
 $2^{32}$  байт = 4Гб
  - Может быть больше физической памяти
- Механизм преобразования адресов предусматривает ведение таблиц соответствия адресов
  - Адреса отображаются не побайтно, а блоками
  - Программа получает память блоками



# Страничная память

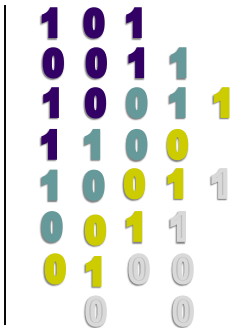
- Виртуальная и физическая память представляются состоящими из наборов блоков (страниц) одинакового размера
  - Виртуальные блоки — страницы
  - Физические блоки — страничные кадры
  - Свопинг осуществляется всегда только страницами
  - Страница фиксированного размера кратного 2, определяется особенностью архитектуры



# Страничная память

- Из виртуального адреса определяется номер страницы и смещение ячейки в этой странице
- Для отображения используется таблица страниц и каталог таблиц
- Для ссылки на таблицу страниц используется специальный регистр процессора
- В таблице хранятся атрибуты страницы, в том числе и бит присутствия
  - Если страницы нет в памяти, то она будет автоматически загружена с жёсткого диска

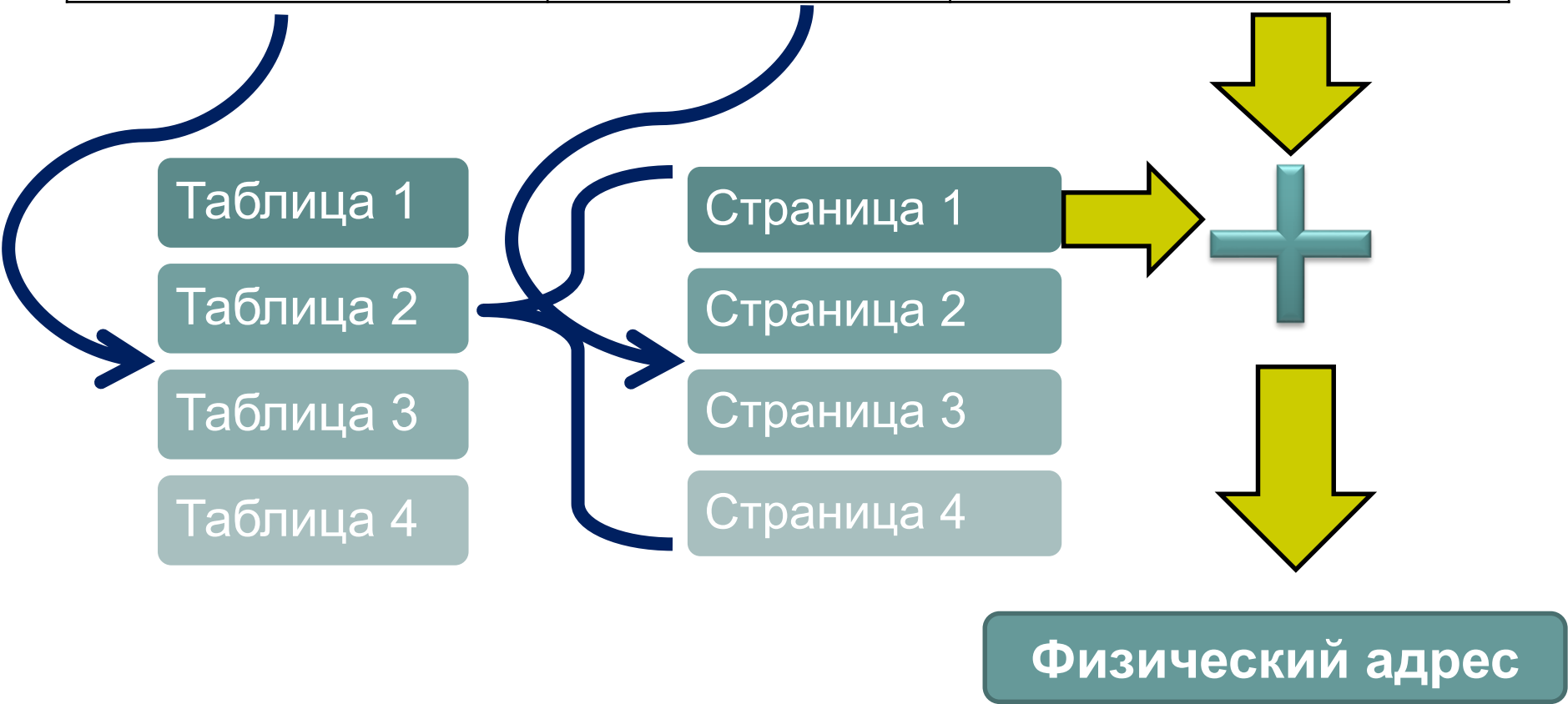
# Страничная память



31	22	21	12	11	0
Таблица		Страница			Смещение в странице

- Старшие десять бит — индекс в каталоге таблиц страниц
- Следующие десять бит используются для выбора страницы
- Таблица страниц может описывать до 1024 страниц размером 4096 байт

A 10x10 grid of numbers 0-9 in various colors and sizes, with some numbers having motion blur effects.



# Особенности реализации



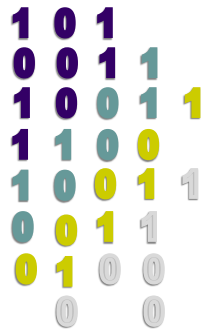
- При отсутствии страницы в памяти возникает исключение (page fault)
  - Вызывается обработчик ОС, который загружает страницу, предварительно освобождая место
- При нехватке страничных кадров редко используемые страницы сбрасываются на ЖД
  - Нужно следить за «редкостью» их использования
- Атрибуты страниц могут быть использованы для контроля доступа и др.



# Фрагментация



- Для ОС
  - Фрагментация отсутствует
  - Вся память гранулирована
- Для процесса
  - Адресное пространство непрерывно
  - Занимает целое количество страниц
  - ~ 0.5 страницы на процесс простаивают
- Переключение контекста приводит к обновлению регистра каталога таблиц страниц и сбросу кеша



# Защита памяти процессов

- Такой механизм управления отображением адресов памяти исключает возможность доступа процесса к памяти другого процесса
  - В его каталоге страниц нет страниц памяти другого процесса
  - Доступ за пределами каталога страниц невозможен
  - Только ОС управляет каталогом страниц
  - Каталог страниц кешируется

# Сегментная и сегментно-страничная организации памяти



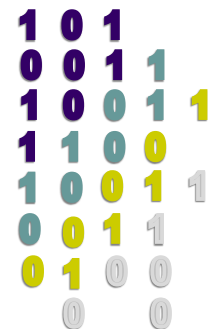
- mov ax, 6124h:0002h

- П
- у
- Е
- д
- 

Аппаратная поддержка сегментов  
относительно слабо  
распространена

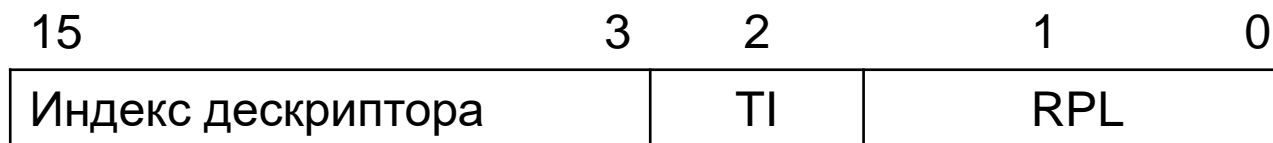
В основном только лишь на  
процессорах архитектуры Intel x86

- LDT — таблица локальных дескрипторов — по одной на каждую задачу



# Селекторы дескрипторов

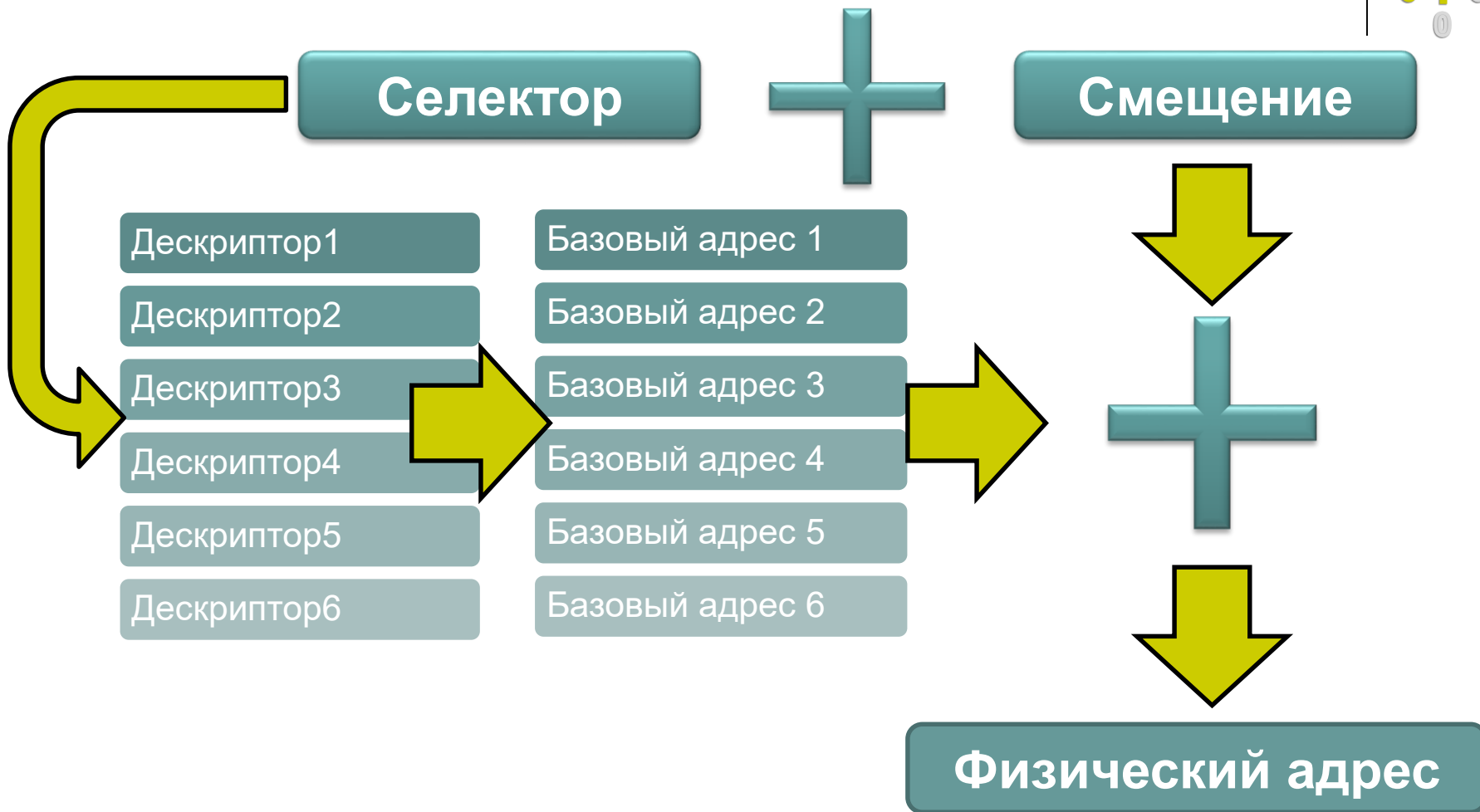
- Сегментные регистры CS, DS, SS и ES хранят не сами базовые адреса сегментов, а селекторы, по которым из таблицы извлекаются дескрипторы сегментов



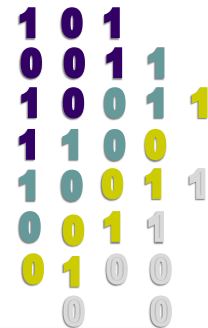
- Ti — Table Indicator
  - 0 — глобальная таблица дескрипторов GDT
  - 1 — локальная LDT
- RPL — уровень привилегий запроса

# Сегментная адресация памяти

1	0	1	
0	0	1	1
1	0	0	1
1	1	0	0
1	0	0	1
0	0	1	1
0	1	0	0
0			



# Дескрипторы сегментов



Базовый адрес (линейный 32-битный адрес начала сегмента)				Лимит (20-битное число)	
Бит гранулярнос ти	Бит присутствия	Бит разрядности	Тип дескриптора	Тип сегмента	Доступ DPL

- бит разрядности (0/1 — 16-битный/32-битный сегмент)
- бит гранулярности (0 — лимит в байтах, 1 — лимит в 4-килобайтных единицах)
- Бит присутствия указывает, что сегмент реально есть в памяти
- Бит разрешения чтения/записи
- Тип сегмента
  - 0 — сегмент данных, 1 — сегмент кода
  - И т.д.

# Особенности реализации таблицы страниц

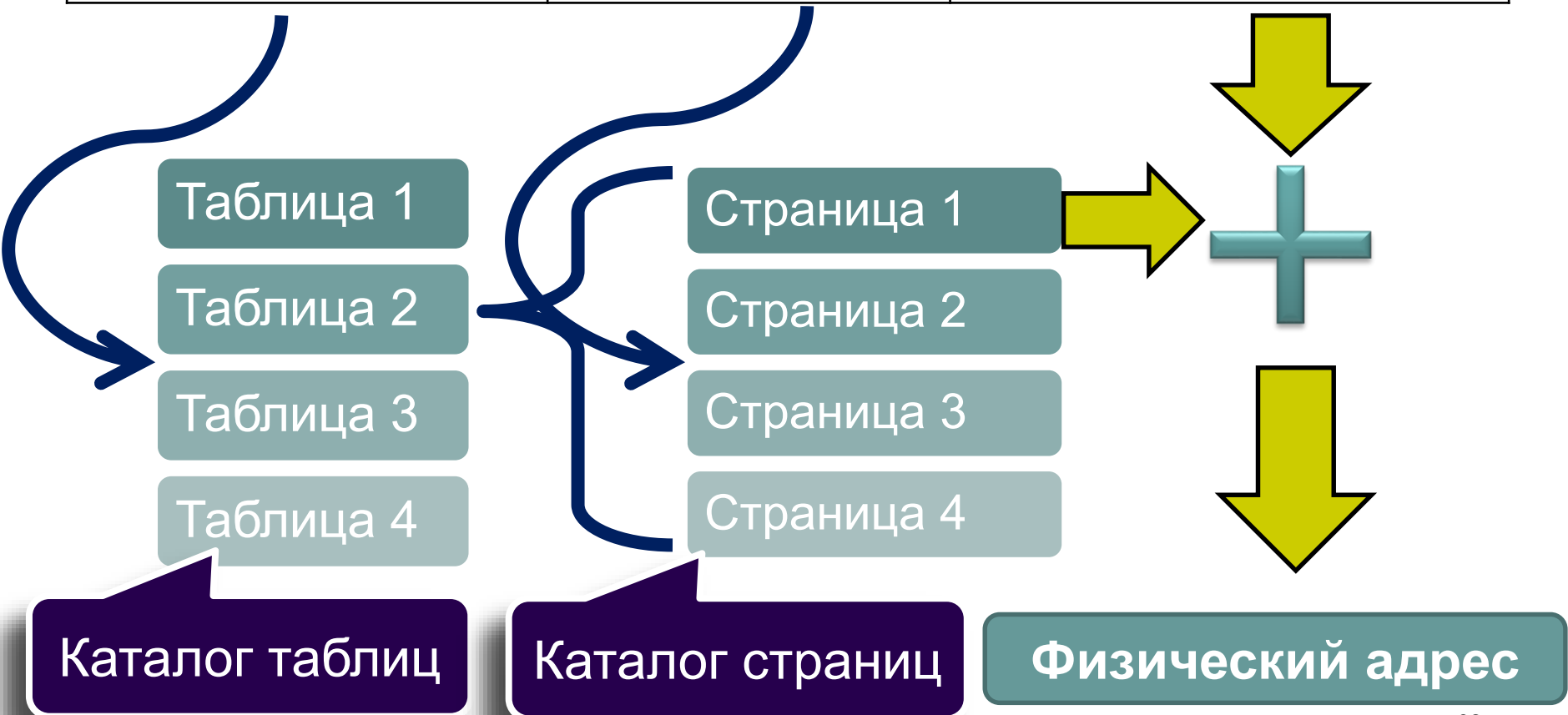


- Для x86 возможно 1М страниц по n байт
  - У каждого процесса своя таблица
  - В сумме они занимают ощутимую часть памяти
  - Работа с таким набором может замедляться
  - **Отображение должно быть быстрым!**
- Используется многоуровневые таблицы
  - Таблица верхнего уровня — каталог таблиц
  - Таблица второго уровня — каталог страниц
    - Можно не держать в памяти и кеше все 1М страниц

# Страничная память

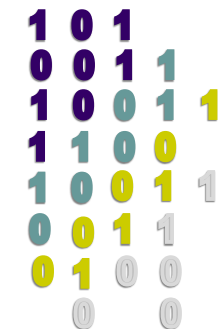
1 0 1  
0 0 1 1  
1 0 0 1 1  
1 1 0 0  
1 0 0 1 1  
0 0 1 1  
0 1 0 0  
0 0 0

31	22	21	12	11	0
Таблица			Страница		Смещение в странице





# Многоуровневые таблицы



- Производительность?
  - Каждый уровень — отдельная таблица в памяти
  - Преобразование адреса — несколько обращений к памяти
- Количество уровней зависит от архитектуры
  - DEC PDP-11 — 1
  - Intel, DEC VAX — 2
  - Sun SPARC, DEC Alpha — 3
  - Motorola — изменяемый уровень пейджинга
  - Zero level paging — MIPS R2000

# Ассоциативная память



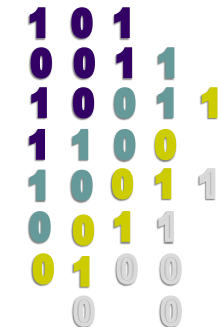
- Дорогой вид памяти
  - Translation lookaside buffer (TLB)
- Кеширует часть записей из таблицы страниц
  - Число записей в TLB от 8 до 2048
  - Одновременный поиск во всех записях
    - Поиск во всем наборе по номеру за один такт
  - Hit ratio — процент попадай в кеш таблиц
  - При 90% — скорость доступа на 10% медленнее кеша
  - При переключении контекста — сбрасывается для ограничения доступа к страницам



# Размер страницы

- Чем больше страница тем меньше каталоги
  - Легче управлять
  - Но тем больше теряется памяти
    - $\frac{1}{2}$  страницы
- Исторически размер страницы растёт
  - DEC PDP-11 — 8 Кбайт
  - Motorola 68030 — может быть задан программно

# page fault



- Исключение при
  - Отсутствии страницы в памяти
  - При попытке чтения несуществующей страницы
  - Запись в «read-only» страницы
- Время доступа к странице в свопе
  - Время обслуживания исключения
  - Время подкачки страницы из вторичной памяти
    - Замещение другой странице при нехватке памяти
  - Время рестарта процесса

# Стратегии управления страничной памятью



- Стратегия выборки (fetch policy)
  - В какой момент следует переписать страницу из вторичной памяти в первичную если её нет в памяти
  - По запросу
  - С упреждением
    - Чтение с упреждением загружает кроме самой страницы, также несколько страниц, окружающих её

# Стратегии управления страничной памятью



- Стратегия размещения (placement policy)
  - В какое место первичной памяти поместить страницу
  - В системах со страничной организацией в любой свободный страничный кадр
  - В системах с сегментной организацией
    - Нужна стратегия, аналогичная стратегии с переменными разделами

# Стратегия замещения (replacement policy)



- Какую страницу нужно выгрузить во внешнюю память, чтобы освободить место
  - Нужно оптимизировать хранение самой нужной информации
  - При вытеснении передаётся 2 страницы!
    - Бит модификации + read-only страницы
  - Простейший вариант — выгрузить любую физическую страницу
    - Одни процессы хаотично и сильно влияют на другие

# Стратегия замещения

## Алгоритмы



- FIFO
  - Временные метки страницы, список страниц
- Оптимальный алгоритм
  - Теоретический — замещать ту страницу, которая будет использована позже всего
- LRU (The Least Recently Used)
  - Выгружается страница не использовавшаяся дольше
  - Эффективен без специального оборудования
  - При обновлении списка вручную замедляется в 10 раз



# NFU (Not Frequently Used)



- Упрощённый аналог LRU
- По таймеру ОС ищет страницы в памяти с флагом доступа и для каждой
  - Увеличивает счётчик присутствия
  - Сбрасывает флаг
  - Запоминает самые редко используемые
  - Они и будут замещены
- Алгоритм ничего не забывает
  - Эффект неравномерного использования
  - Принудительно уменьшать счётчик

# Thrashing



- Пробуксовка, трешинг
- Если больше времени тратится на подкачку страниц, нежели на выполнение
  - Если процессу не хватает страничных кадров он генерирует page fault
  - Глобальный алгоритм замещения выгружает страницы других процессов
  - Они также генерируют page fault
  - Вся система «буксует»
  - Локальные алгоритмы уменьшают эффект

# Особенности функционирования менеджера памяти



- Аппаратные и архитектурные особенности
  - Например, фиксация страниц в памяти
    - Асинхронный ввод-вывод и выгрузка страницы
    - Если ввод-вывод реализован через отдельный процессор (DMA transfer) — хаос!
  - Локализованная страница не подлежит замещению
    - Системные страницы, ядро, драйвера, подкачанная, но не использованная страница и т.д.
  - Реализация «копирования при записи»

# Вопросы?

1	0	1		
0	0	1	1	
1	0	0	1	1
1	1	0	0	
1	0	0	1	1
0	0	1	1	
0	1	0	0	
	0		0	