Операционные системы

Введение (часть 2)

3. Основы компьютерной архитектуры
   - 3.1. Компьютер фон Неймана
   - 3.2. Аппарат прерываний
   - 3.3. Внешние устройства
   - 3.4. Аппаратная поддержка ОС и систем программирования

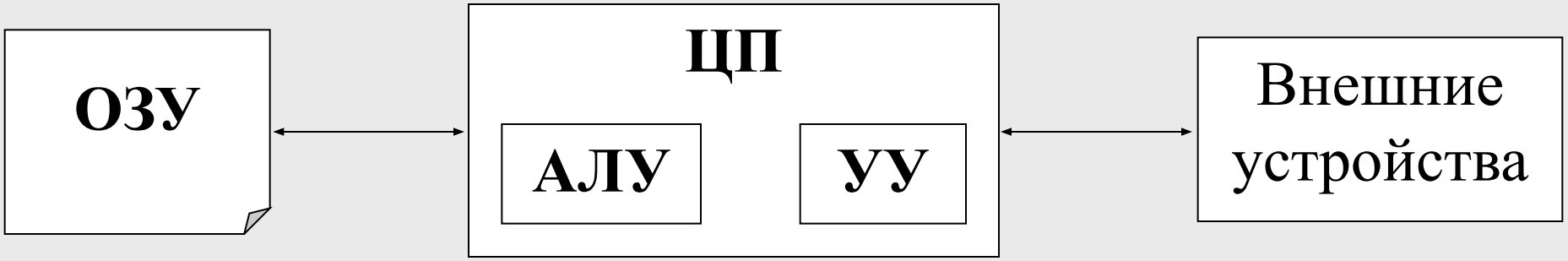
Компьютер фон Неймана

Историческая справка

- Джон фон Нейман (John Von Neumann)
- EDVAC (Electronic Discrete Variable Computer — Электронный Компьютер Дискретных Переменных)
- «Предварительный доклад о компьютере EDVAC» (A First Draft Report on the EDVAC)
- Джон Мочли (John Mauchly) и Джон Преспер Эккерт (John Presper Eckert)
- ENIAC (Electronic Numerical Integrator And Computer)

Компьютер фон Неймана

Структура, основные компоненты компьютера фон Неймана



Принципы построения компьютера фон Неймана

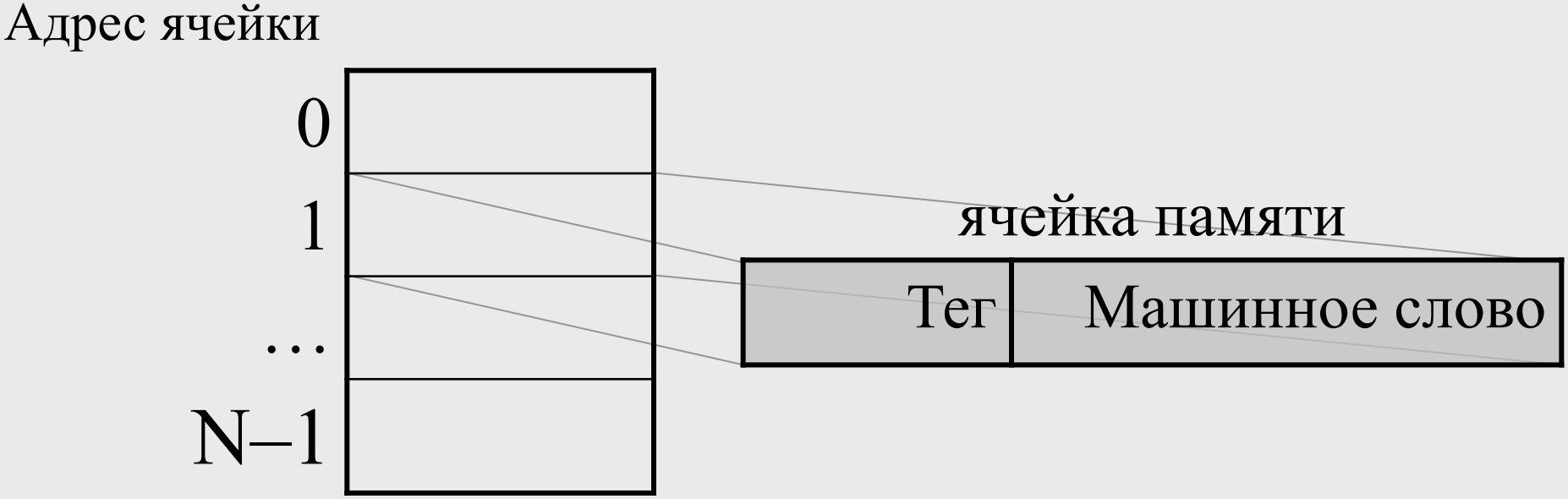
1. Принцип двоичного кодирования
2. Принцип программного управления
3. Принцип хранимой программы

Оперативное запоминающее устройство

ОЗУ предназначено для хранения программы, выполняющейся в компьютере.

Тег — поле служебной информации.

Машинное слово — поле программно изменяемой информации.

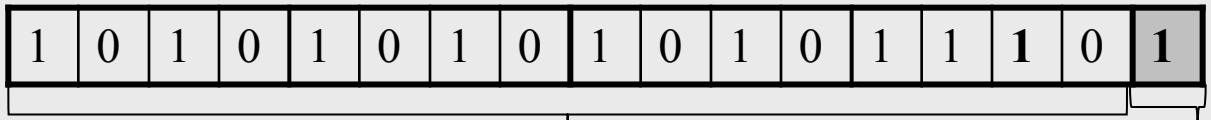


Оперативное запоминающее устройство

Использование содержимого поля служебной информации

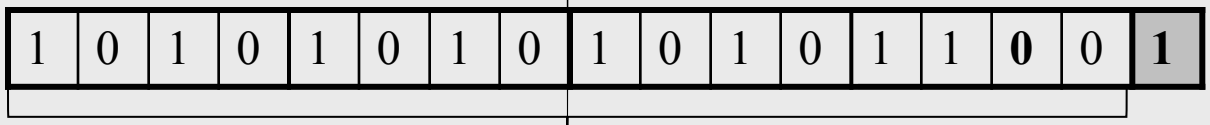
1. Контроль за целостностью данных

При записи слова в память контрольная сумма бит = 9 (1001b) ⇒ тег = **1**.



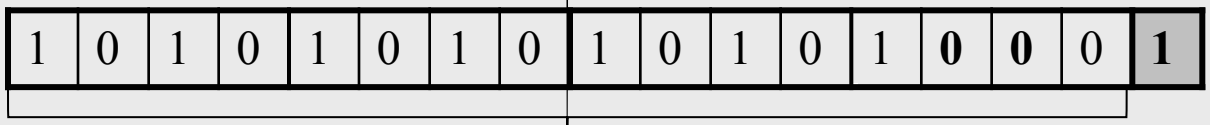
Ошибки нет

При чтении машинного слова 16 бит вычисляется сумма бит = 9 (1001b) и сверяется со значением тега.



Ошибка

При чтении машинного слова 16 бит вычисляется сумма бит = 8 (1000b), а тег = 1 ⟹ Сбой в работе ОЗУ



Ошибка не видна

При чтении машинного слова 16 бит вычисляется сумма бит = 7 (111b), и тег = 1 ⟹ Ошибка будет не выявлена

Пример контроля за целостностью данных по четности

Оперативное запоминающее устройство

Использование содержимого поля служебной информации

2. Контроль доступа к командам/данными

3. Контроль доступа к машинным типам данных

Оперативное запоминающее устройство

Производительность оперативной памяти — скорость доступа процессора к данным, размещенным в ОЗУ.

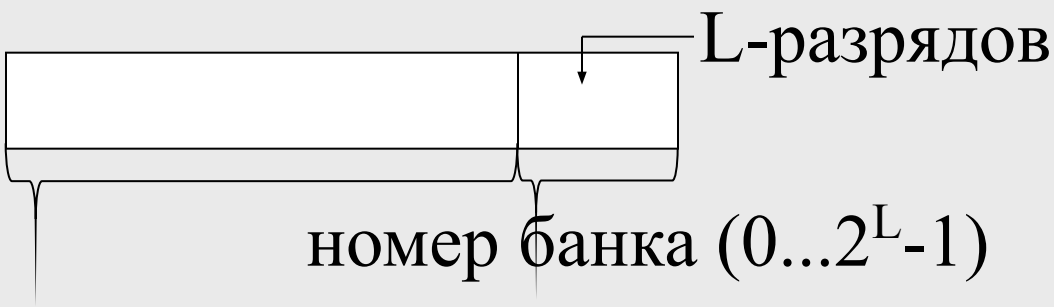
- **Время доступа (access time, t_{access})** — время между запросом на чтение слова из оперативной памяти и получением содержимого этого слова.
- **Длительность цикла памяти (cycle time, t_{cycle})** — минимальное время между началом текущего и последующего обращения к памяти.

$$(t_{cycle} > t_{access})$$

Оперативное запоминающее устройство

Расслоение памяти

К независимых банков памяти, где $К = 2^L$.



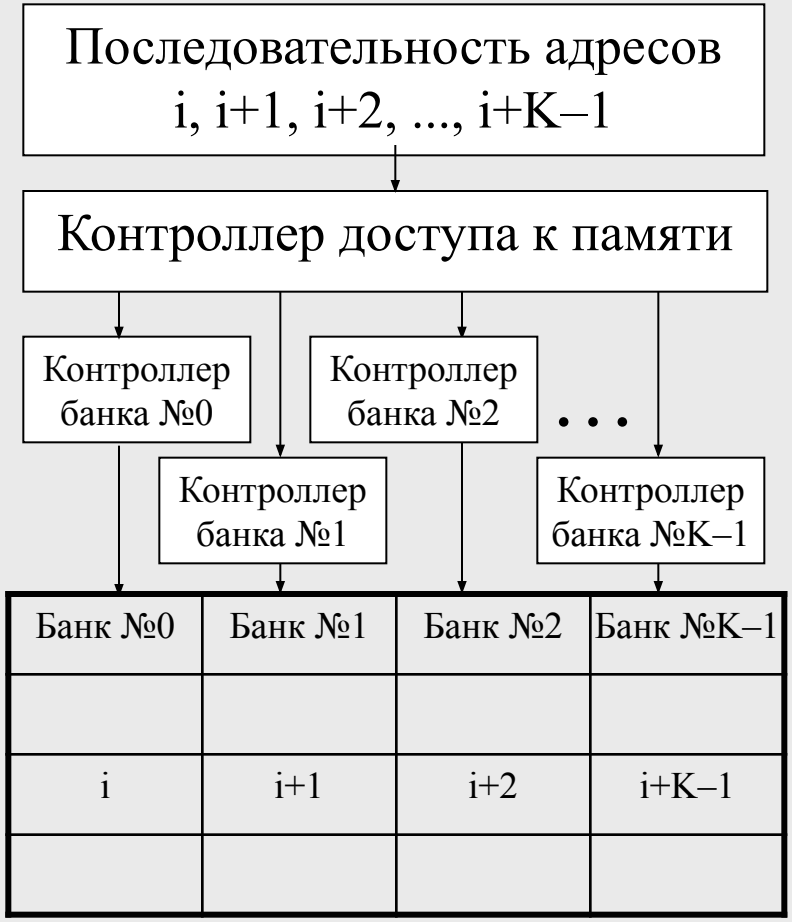
Оперативное запоминающее устройство

Расслоение памяти

Сравнение времени доступа

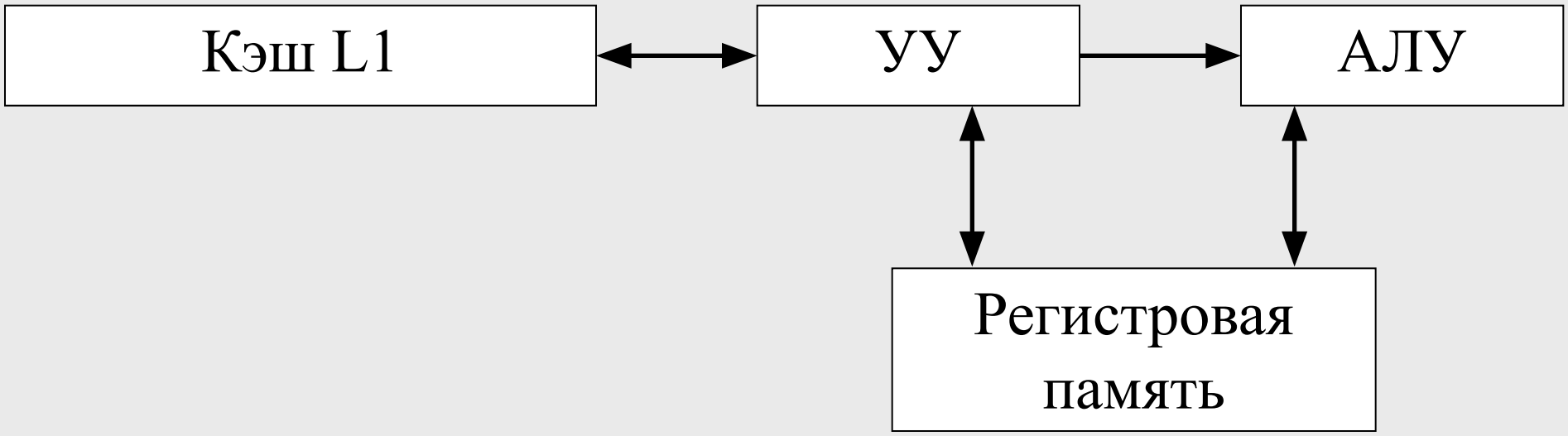
Последовательное чтение машинных слов по адресам:	ОЗУ без расслоения	ОЗУ с расслоением
A	$\sim t_{cycle}$	$\sim t_{access}$
A+1	$\sim t_{cycle}$	$\sim t_{access}$
A+2	$\sim t_{cycle}$	$\sim t_{access}$
Суммарное время:	$\sim 3*t_{cycle}$	$\sim 3*t_{access}$

Схема работы



Центральный процессор

Устройство управления (control unit) — координация выполнения команд.
Арифметико-логическое устройство (arithmetic/logic unit) —выполнение команд, арифметической или логической обработки операндов.
Регистровая память (register memory) — совокупность устройств памяти процессора - регистров.
Кэш-память (cache memory) — буферизация работы процессора с оперативной памятью.



Центральный процессор

Регистровая память

Регистровая память (регистровый файл)

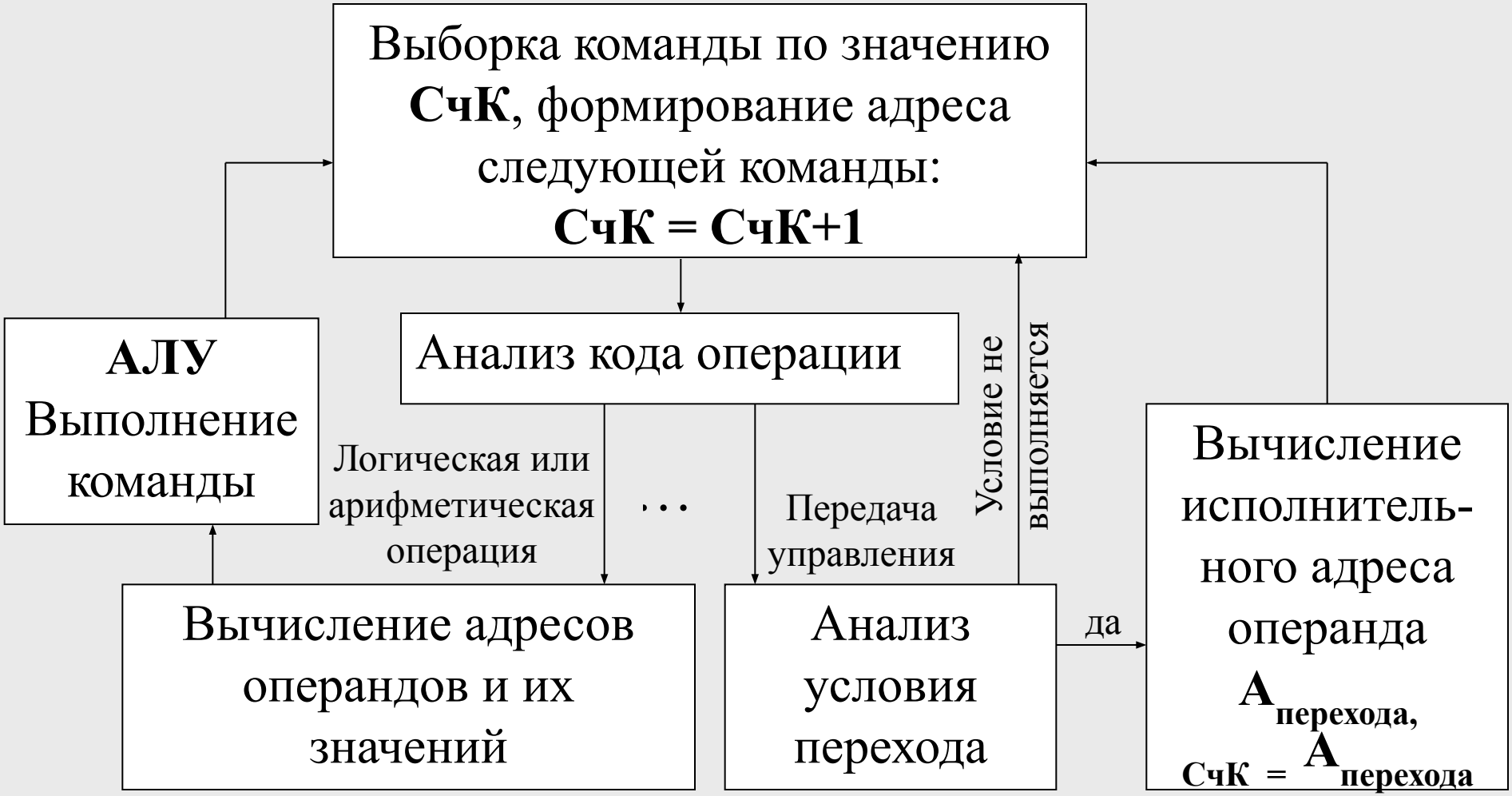
Регистры общего назначения (РОН)

Специальные регистры:

- **счетчик команд** (program counter)
- **указатель стека** (stack pointer)
- **слово состояние процессора** (processor status word)
- .....

Центральный процессор

Рабочий цикл процессора



Центральный процессор

Кэш-память (cache memory) первого уровня (L1)

1. Обмен данными между кэшем и оперативной памятью осуществляется **блоками** фиксированного размера
2. **Адресный тег** блока — содержит служебную информацию о блоке (соответствие области ОЗУ, свободен/занят блок, …)
3. Нахождение данных в кэше – **попадание** (**hit**). Если искомых данных нет в кэше, то фиксируется **промах (cache miss)**
4. При возникновении промаха происходит обновление содержимого кэша **— вытеснение**. Стратегии вытеснения:
   - случайный выбор блока
   - вытеснение наименее «популярного» блока (LRU — Least-Recently Used)
5. Вытеснение кэша данных:
   - **сквозное кэширование** (write-through caching)
   1. **кэширование с обратной связью** (write-back cache) — тег модификации (dirty bit)

Аппарат прерываний

Прерывание — событие в компьютере, при возникновении которого в процессоре происходит предопределенная последовательность действий.

Типы прерываний

- **Внутренние** — инициируются схемами контроля работы процессора
- **Внешние** — события, возникающие в компьютере в результате взаимодействия центрального процессора с внешними устройствами

Аппарат прерываний

Этап аппаратной обработки прерываний

Аппарат прерываний

Программный этап обработки прерываний

Внешние устройства

Внешние устройства

Внешние запоминающие устройства (ВЗУ)

Обмен данными:

- записями фиксированного размера — **блоками**
- записями произвольного размера

Доступ к данным:

- операции чтения и записи (жесткий диск, CD-RW)
- только операции чтения (CD-ROM, DVD-ROM, …)

Последовательного доступа:

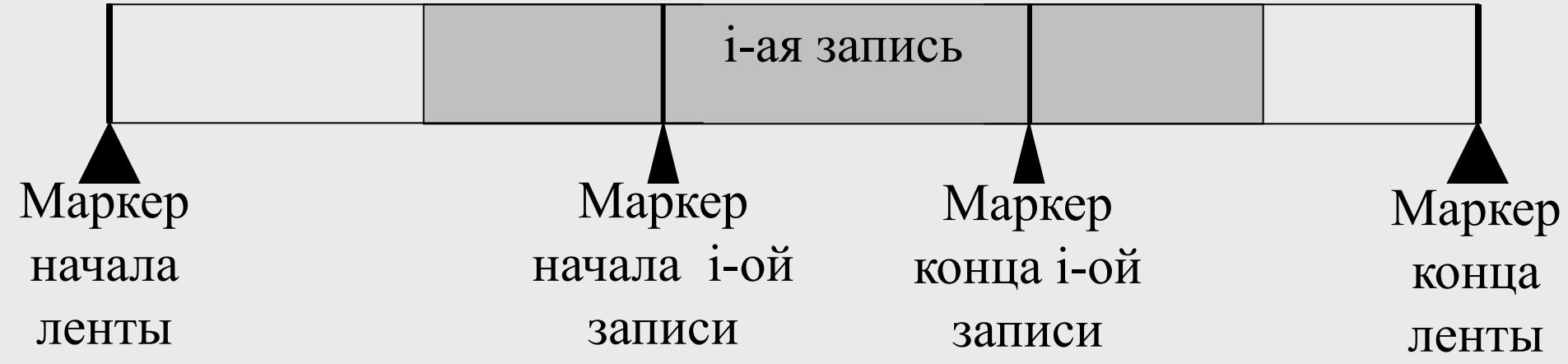
- Магнитная лента

Прямого доступа:

- Магнитные диски
- Магнитный барабан
- Магнито-электронные ВЗУ прямого доступа

Устройство последовательного доступа

Магнитная лента



Устрой ого доступа

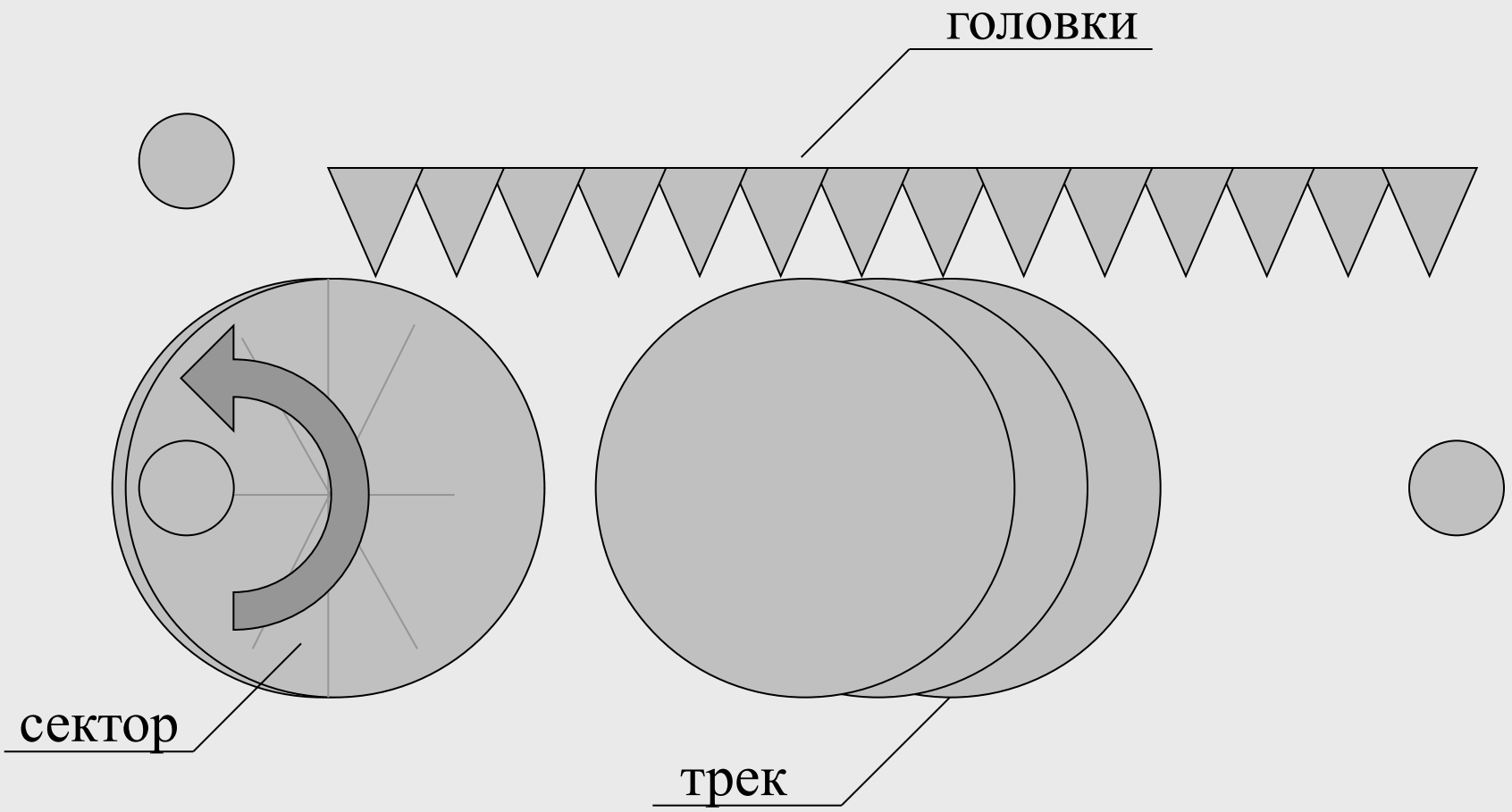
ки

Операции, необходимые для начала чтения (позиционирования)

1. Установка головки на требуемую дорожку
2. Поворот для совмещения головки с началом сектора

Устройство прямого доступа

Магнитный барабан

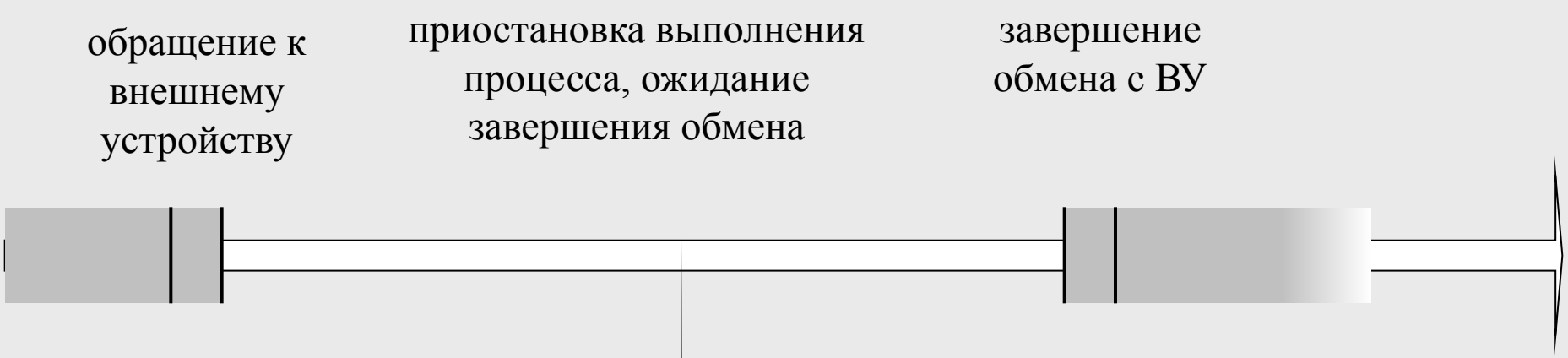


Операции, необходимые для начала чтения (позиционирования)

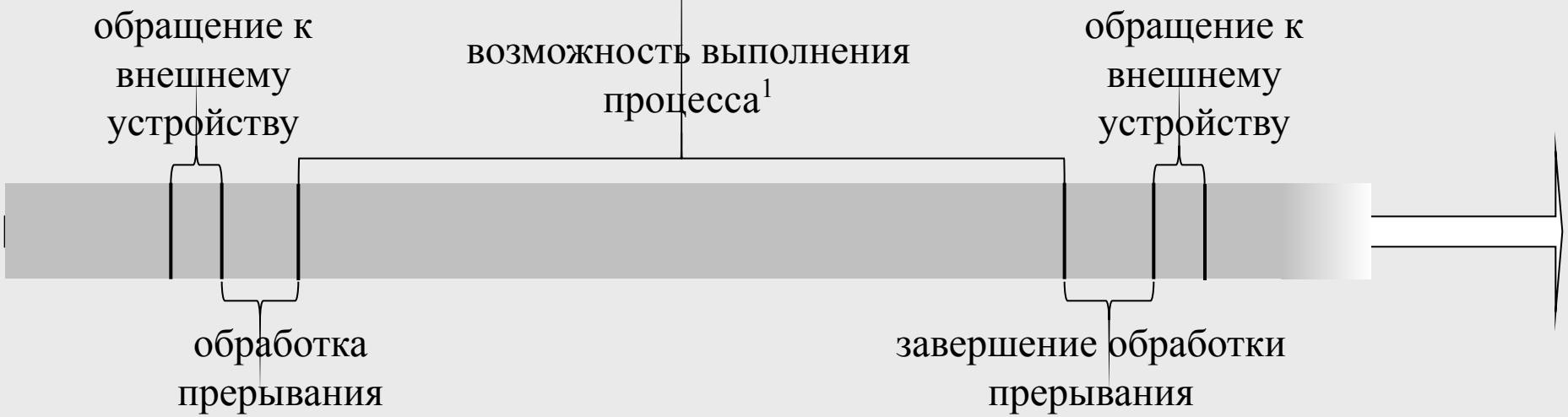
1. Поворот для совмещения головки с началом сектора

Модели синхронизации при обмене с внешними устройствами

Синхронная организация обмена

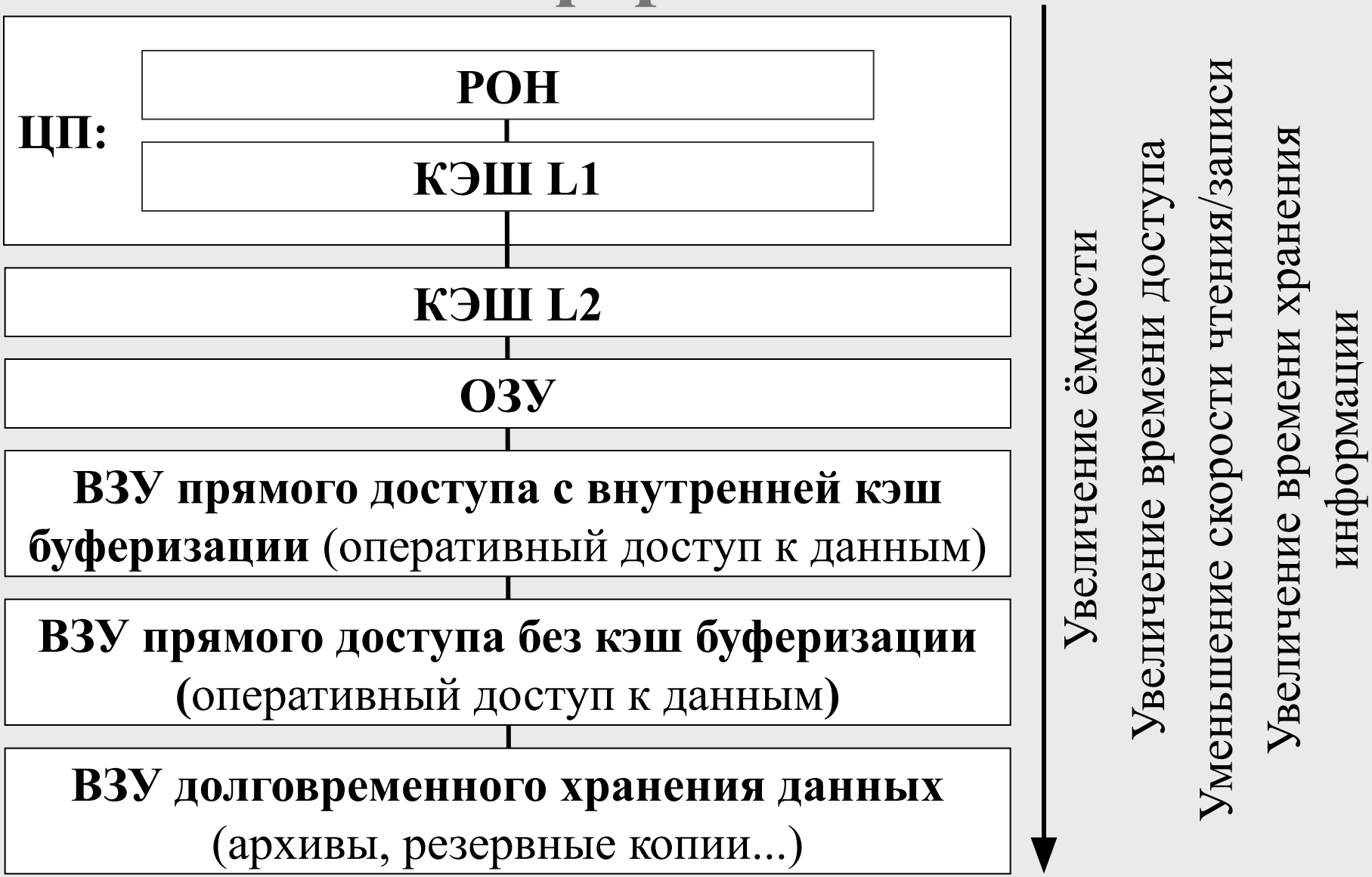


Асинхронная организация обмена



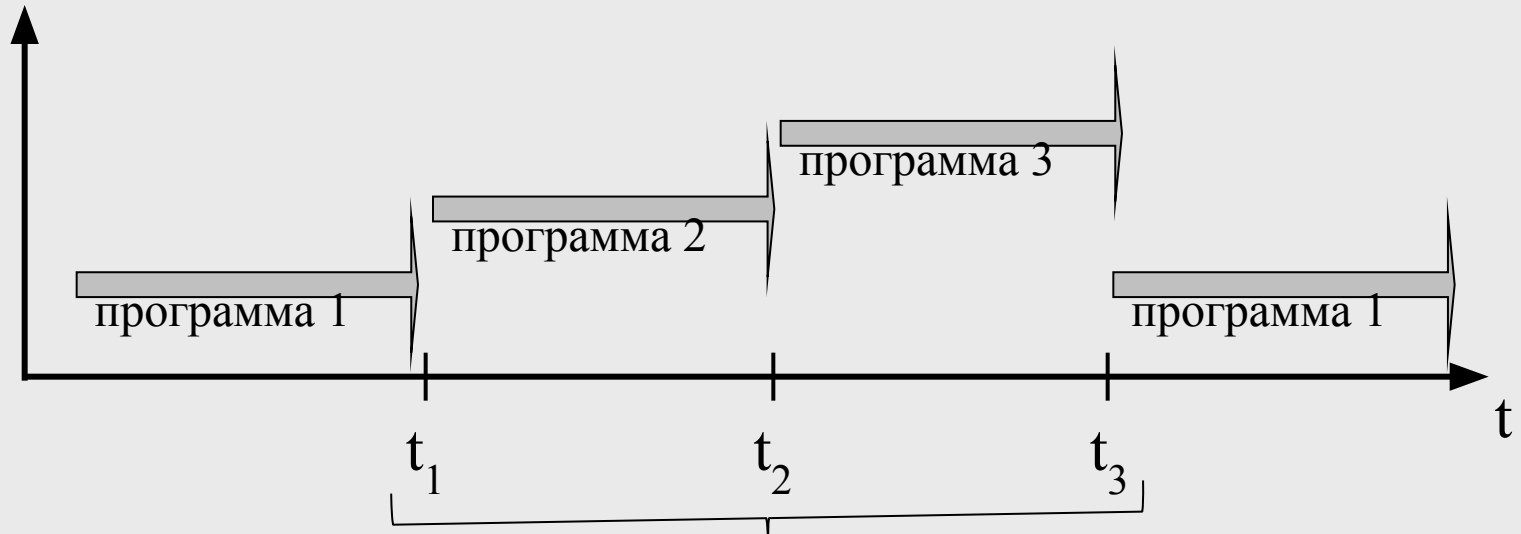
[1] Примечание: процесс выполняется до возникновения следующего прерывания

Иерархия устройств хранения информации



Аппаратная поддержка ОС и систем программирования

Мультипрограммный режим — режим, при котором возможна организация переключения выполнения с одной программы на другую.



Базовая аппаратная поддержка мультипрограммного режима

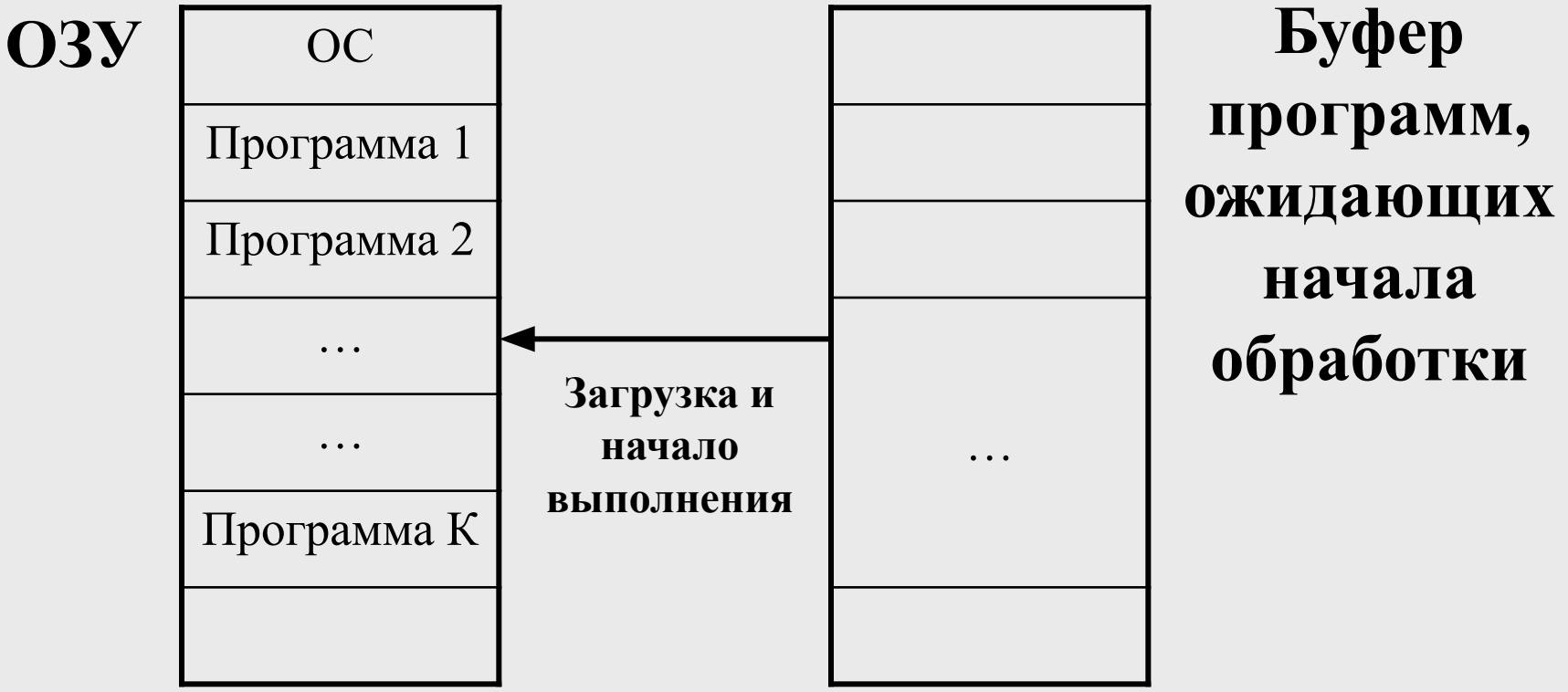
1. Аппарат **защиты памяти**
2. **Специальный режим операционной системы** (привилегированный режим или режим супервизора)
3. Аппарат прерываний (как минимум, **прерывание по таймеру**)

Аппаратная поддержка программных систем и мультипрограммного режима

Проблемы:

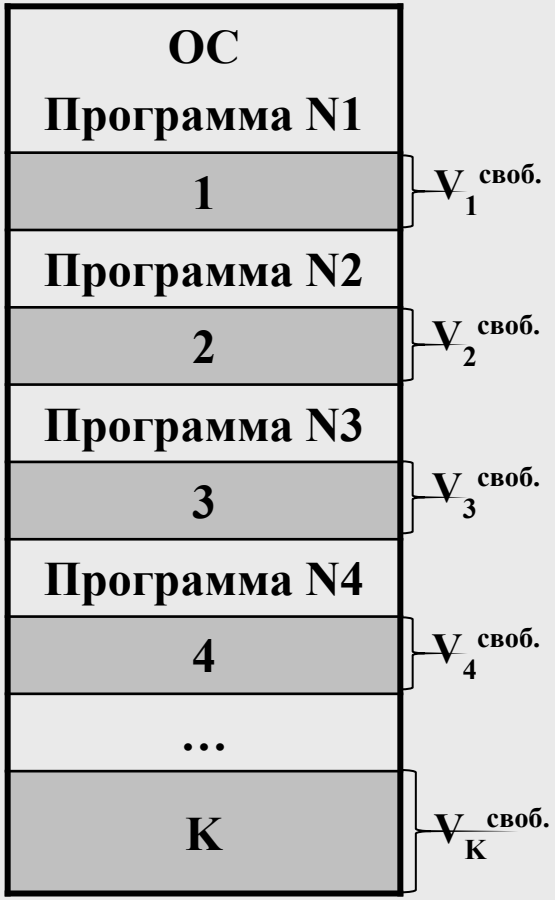
- Вложенные обращения к подпрограммам
- Накладные расходы при смене обрабатываемой программы
- Перемещаемость программы по ОЗУ
- Фрагментация памяти

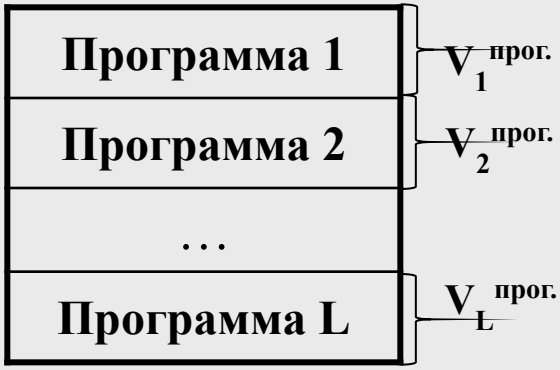
Перемещаемость программы по ОЗУ



Соответствие адресов, используемых в программе, области ОЗУ, в которой будет размещена данная программа

Фрагментация памяти



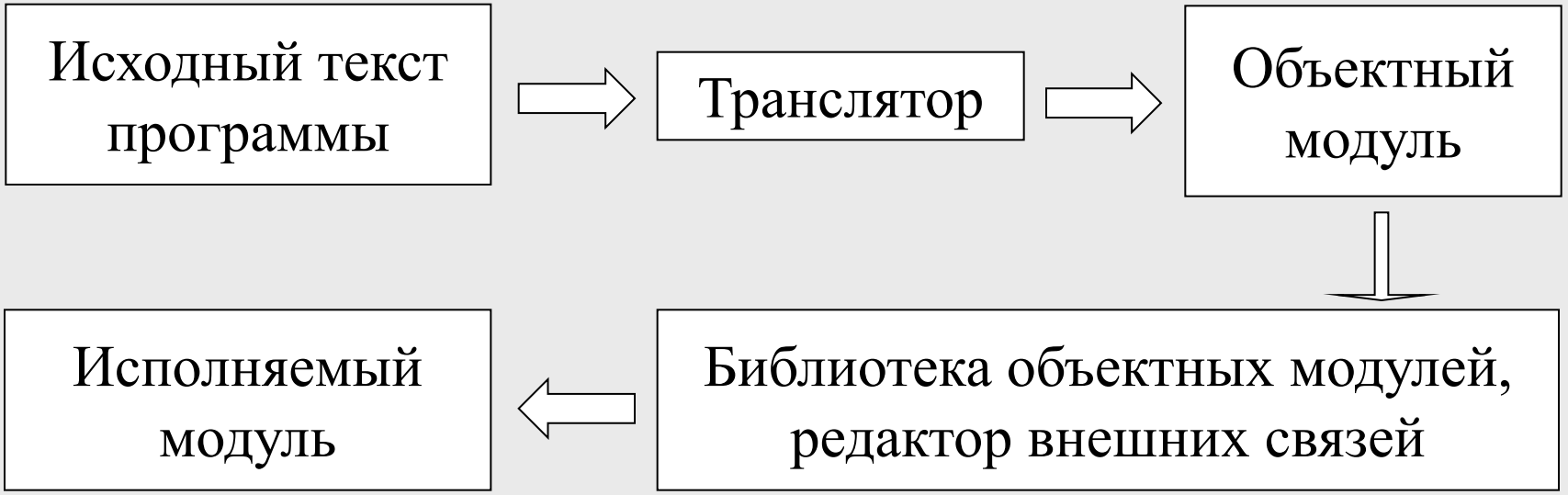


Буфер программ, ожидающих начала обработки

Проблема фрагментации: $V_i^{прог.} > V_j^{своб.}$ для $\forall$ i,j (несмотря на то, что $\exists \{i\}$: $\sum_{j=1}^{K} V_j^{своб.} > V_i^{прог.}$) => деградация системы

Несмотря на то, что имеется достаточное количество места в памяти, разместить ни одну из программ не удаётся.

Виртуальная память. Базирование



В исполняемом модуле используется программная (логическая или виртуальная) адресация

Проблема – установление соответствия между программной адресацией и физической памятью

Виртуальная память. Базирование

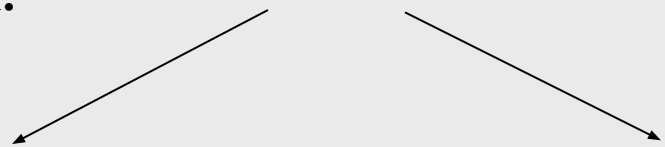
Аппарат виртуальной памяти — аппаратные средства компьютера, обеспечивающие преобразование (установление соответствия) программных адресов, используемых в программе с адресами физической памяти, в которой размещена программа во время выполнения.

Базирование адресов — реализация одной из моделей аппарата виртуальной памяти.

Виртуальная память. Базирование

Базирование адресов — решение проблемы **перемещаемости** программы по ОЗУ.

$$A_{исп.}^{прог.}$$



Абсолютный адрес

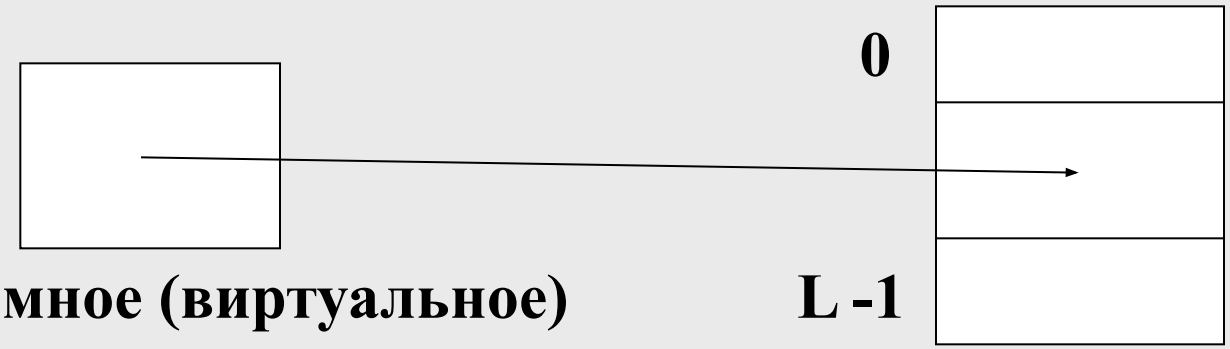
$$\Rightarrow A_{исп.}^{физ.} = A_{исп.}^{прог.}$$

Относительный (адрес относительно начала программы)

$$\Rightarrow A_{исп.}^{физ.} = A_{исп.}^{прог.} + \langle R_{базы} \rangle$$

Виртуальная память. Базирование

Базирование адресов — отображение виртуального адресного пространства программы в физическую память «один в один».



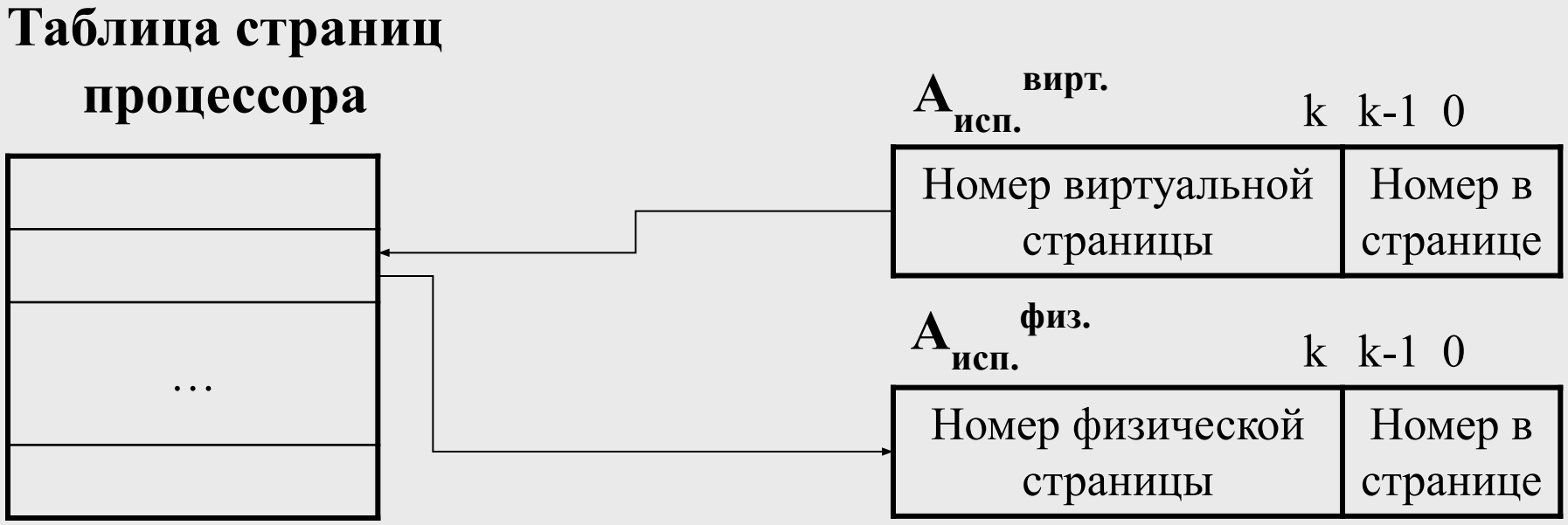
Виртуальная память. Страничная организация памяти

0-я страница
1-я страница
… **…**

Страницы - блоки фиксированного размера. Размер страницы — 2^k

Структура адреса

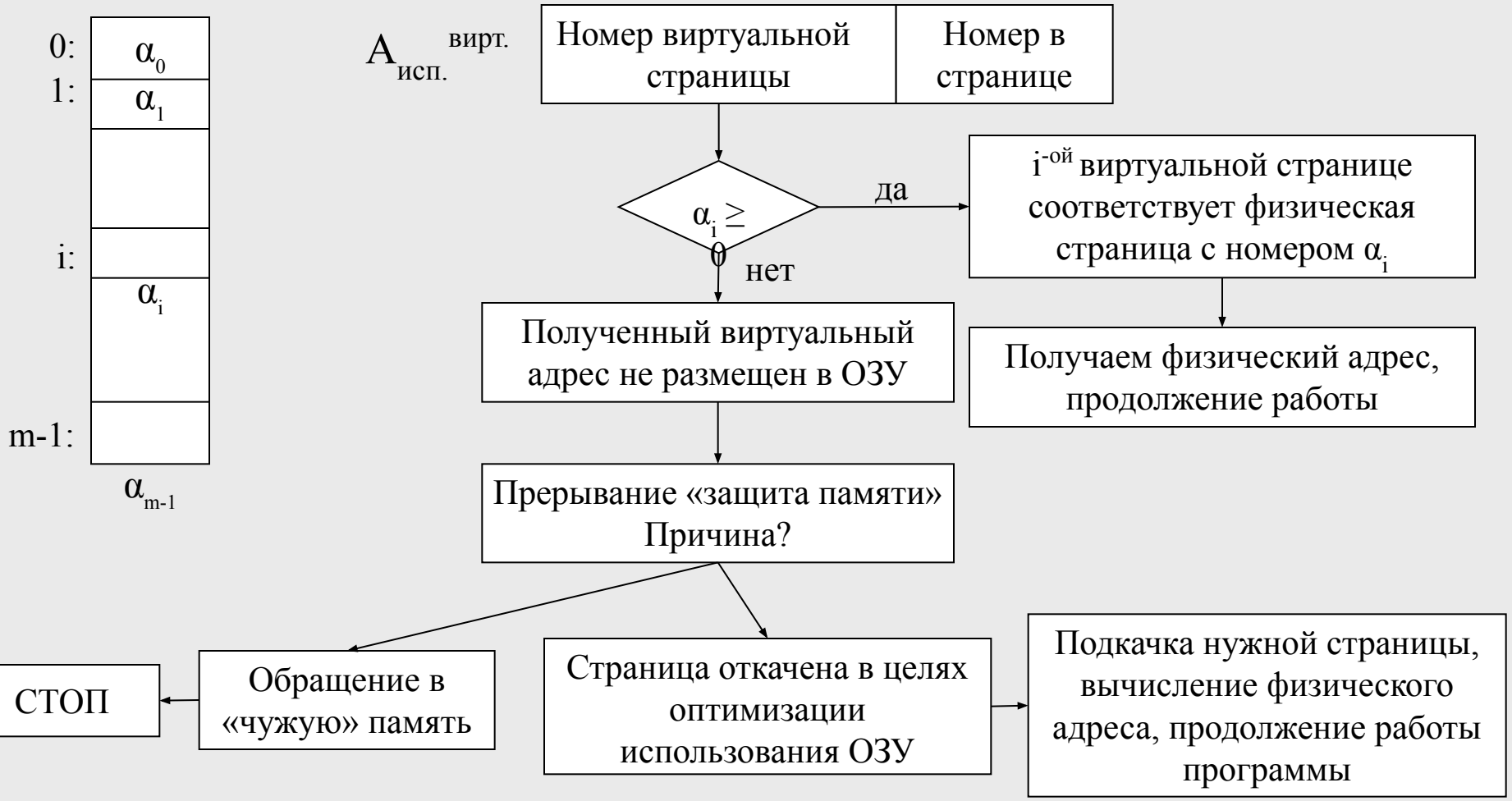
Виртуальная память. Страничная организация памяти



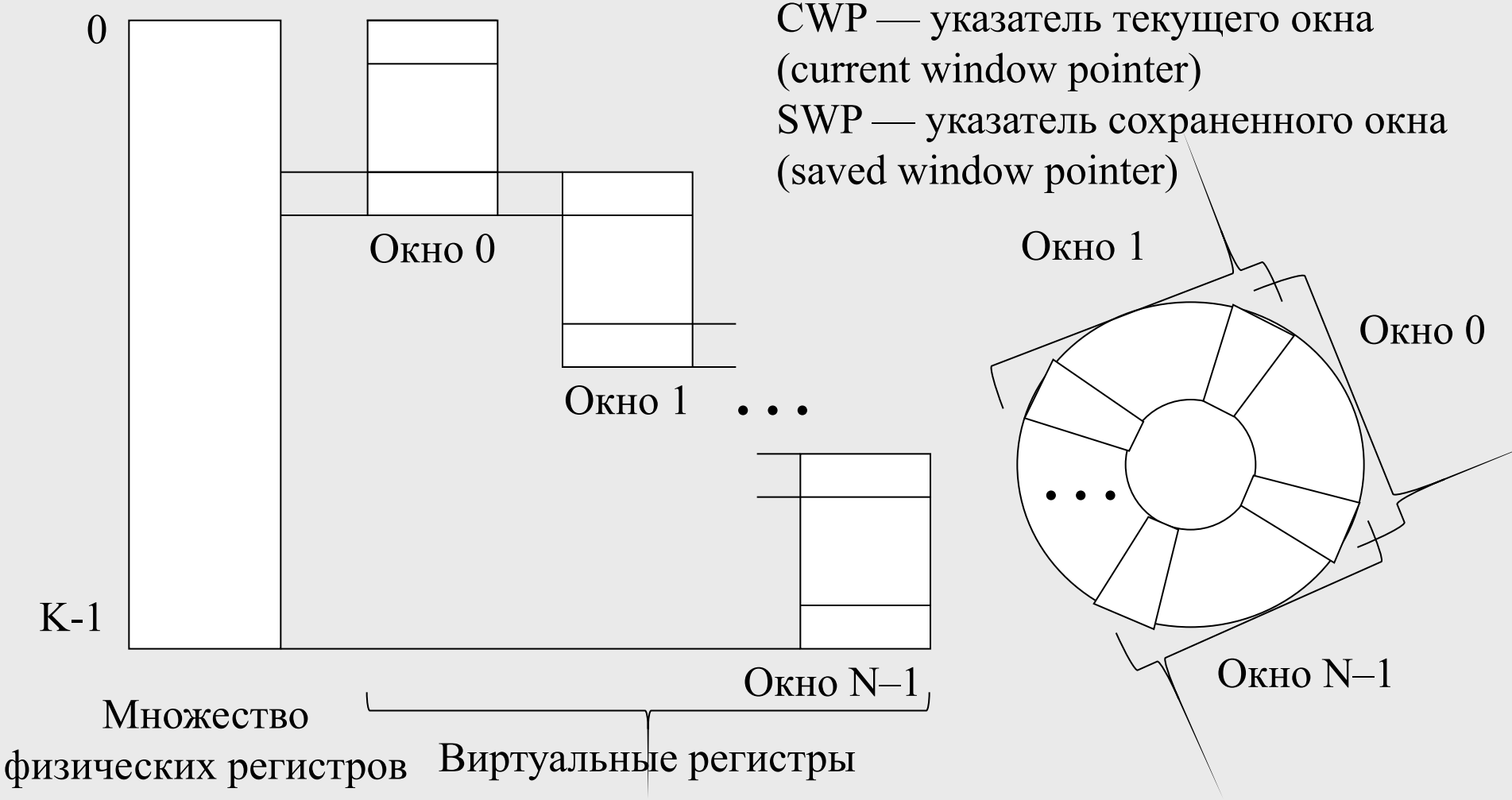
Преобразование виртуального адреса в физический — замена номера виртуальной страницы на соответствующий номер физической страницы

Виртуальная память. Страничная организация памяти

Модельный пример организации страничной виртуальной памяти



Регистровые окна (Register windows)



Аппаратная организация системного стека

Команды работы со стеком:
PUSH — добавить новый элемент
POP — удалить элемент из вершины стека

Использование системного стека может частично решать проблему минимизации накладных расходов при смене обрабатываемой программы и/или обработке прерываний.