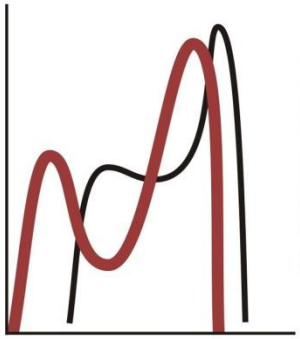


ИНСТИТУТ
МАТЕМАТИКИ
МЕХАНИКИ
КОМПЬЮТЕРНЫХ
НАУК

имени И.И. Воровича —

Архитектура компьютера и операционные системы

Лекция 20. Подсистема ввода-вывода.

Андреева Евгения Михайловна
доцент кафедры информатики и вычислительного эксперимента

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Виды деятельности вычислительной системы

- Обработка информации
- Операции ввода-вывода

С точки зрения программиста:

Обработка информации – выполнение команд процессора над данными, находящимися в памяти, независимо от уровня иерархии

Ввод-вывод – обмен данными между памятью и устройствами, внешними по отношению к ней и процессору

С точки зрения ОС:

Обработка информации – выполнение команд процессора над данными, лежащими в памяти на уровнях не ниже основной памяти

Ввод-вывод – все остальное

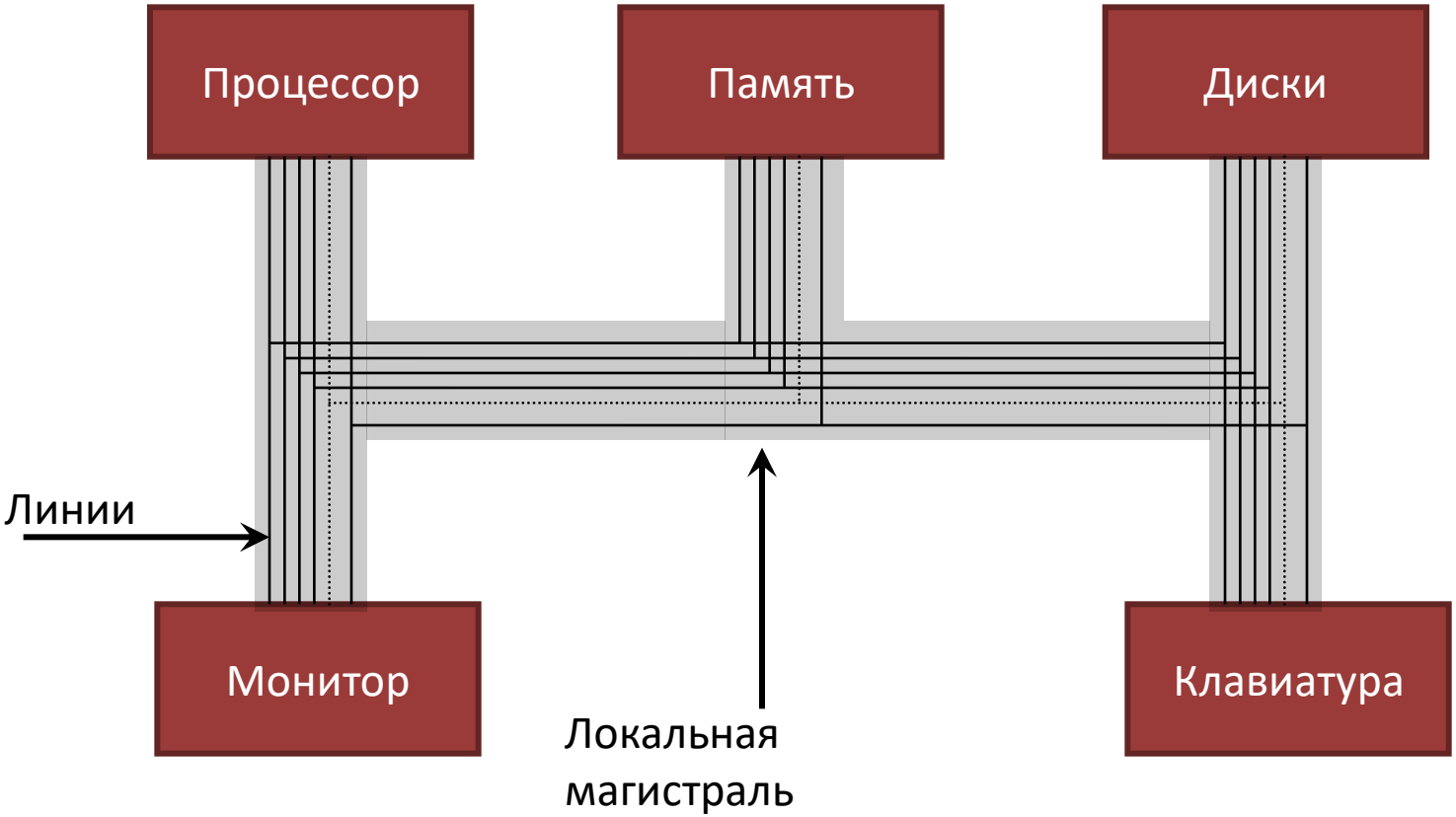
ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Виды деятельности вычислительной системы

- Обработка информации
  - Что делается?
  - Как делается?
- Операции ввода-вывода
  - Что делается?
  - Как делается?

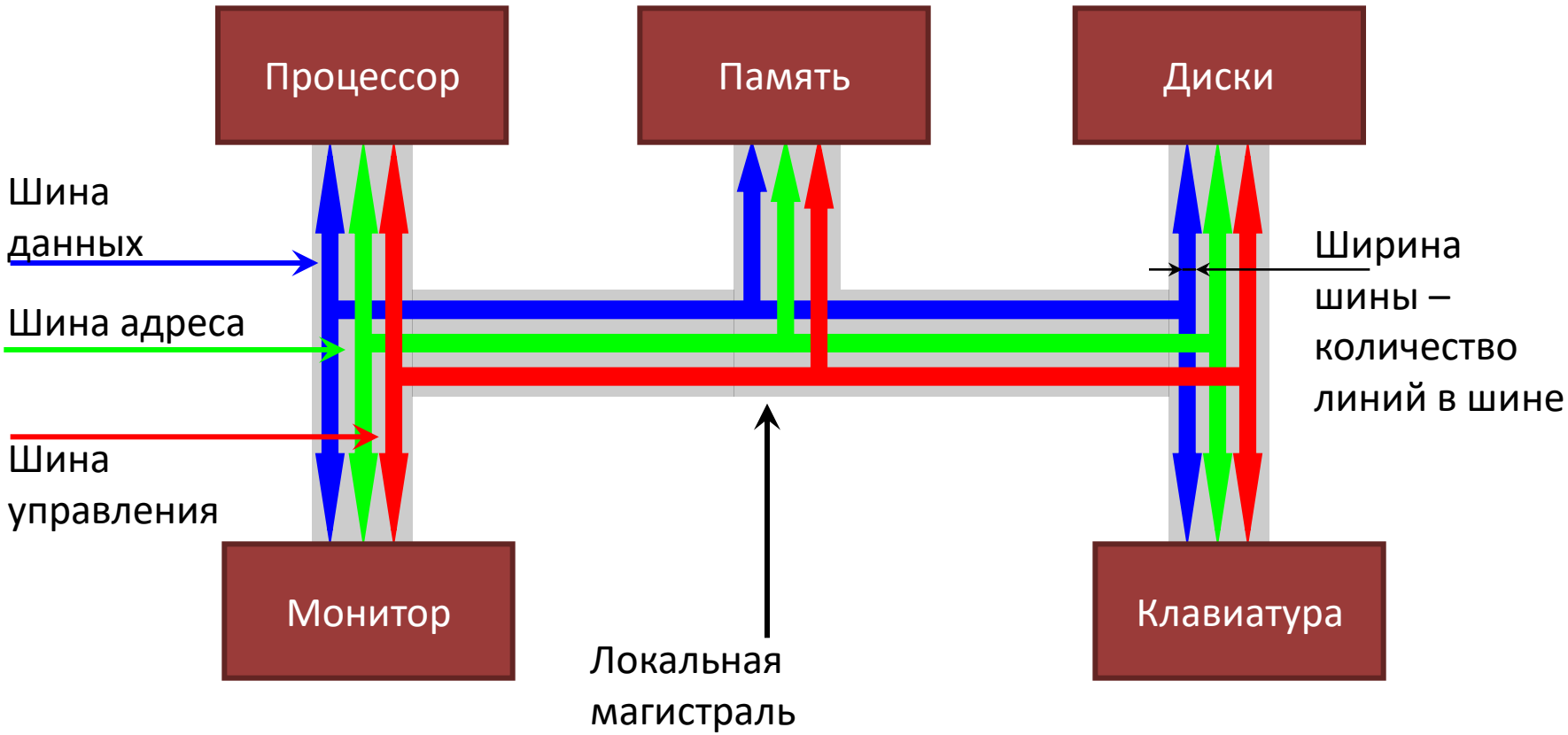
ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК
имени И.И. Воровича

Общие сведения об архитектуре компьютера



ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Общие сведения об архитектуре компьютера



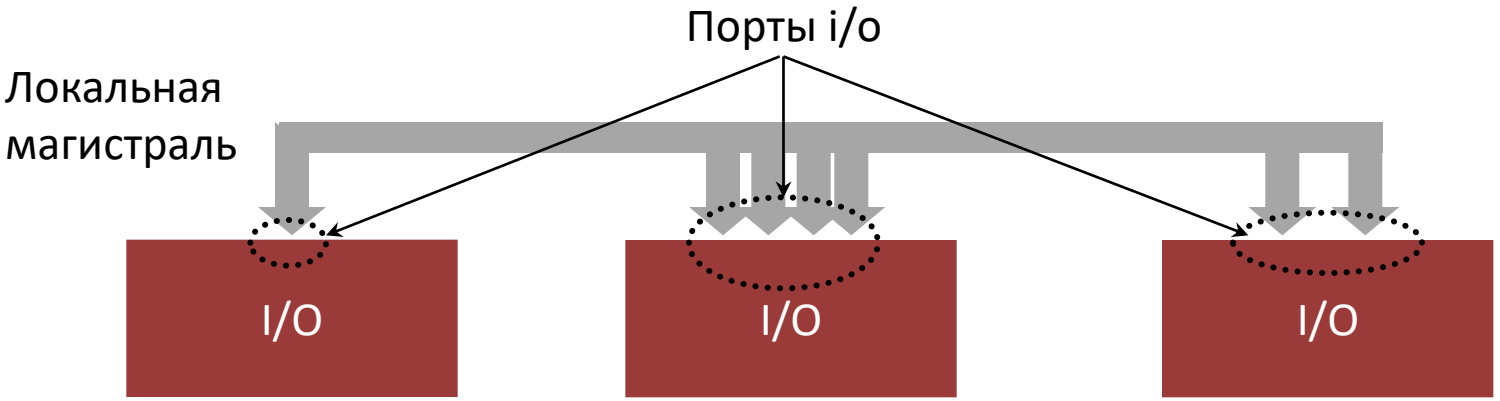
Передача информации из процессора в память

1. На адресной шине выставить сигналы для адреса памяти
2. На шине данных выставить сигналы для данных
3. На шине управления выставить сигналы работы с памятью и операции записи

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Память и устройства I/O

- Память:
  - Локализована в пространстве
  - Ячейки взаимно однозначно отображаются на линейное адресное пространство памяти.
- Устройства I/O:
  - Пространственно разнесены и подключаются к локальной магистрали через порты ввода-вывода.



ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Устройства I/O

- Пространственно разнесены и подключаются к локальной магистрали через порты ввода-вывода.
- Порты ввода-вывода взаимно однозначно отображаются на линейное адресное пространство ввода-вывода (иногда на линейное адресное пространство памяти)

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Передача информации из процессора в порт

Порт отображен в адресное пространство ввода-вывода

1. На адресной шине выставить сигналы для адреса **порта**
2. На шине данных выставить сигналы для данных
3. На шине управления выставить сигналы работы с **устройствами ввода-вывода** и операции записи

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Память и устройства I/O

- Занесение информации в память завершает операцию записи
- Занесение информации в порт часто инициализирует реальное совершение ввода-вывода

Что делать после получения информации через порт и как предоставить информацию для чтения из порта определяют контроллеры устройств

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Общие принципы

- Устройства ввода-вывода подключаются к локальной магистрали через порты
- Могут существовать два адресных пространства: пространство памяти и пространство ввода-вывода
- Порты обычно отображаются в адресное пространство ввода-вывода и иногда – в адресное пространство памяти
- Какое адресное пространство использовать определяется типом команды или типом операндов
- Управлением устройством ввода-вывода, приемом и передачей данных через порты и выставлением сигналов на магистрали занимаются контроллеры

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Структура контроллера устройства

Регистр состояния
(read only со стороны процессора)

- Бит занятости
- Бит готовности данных
- Бит ошибки

Регистр управления
(write only со стороны процессора)

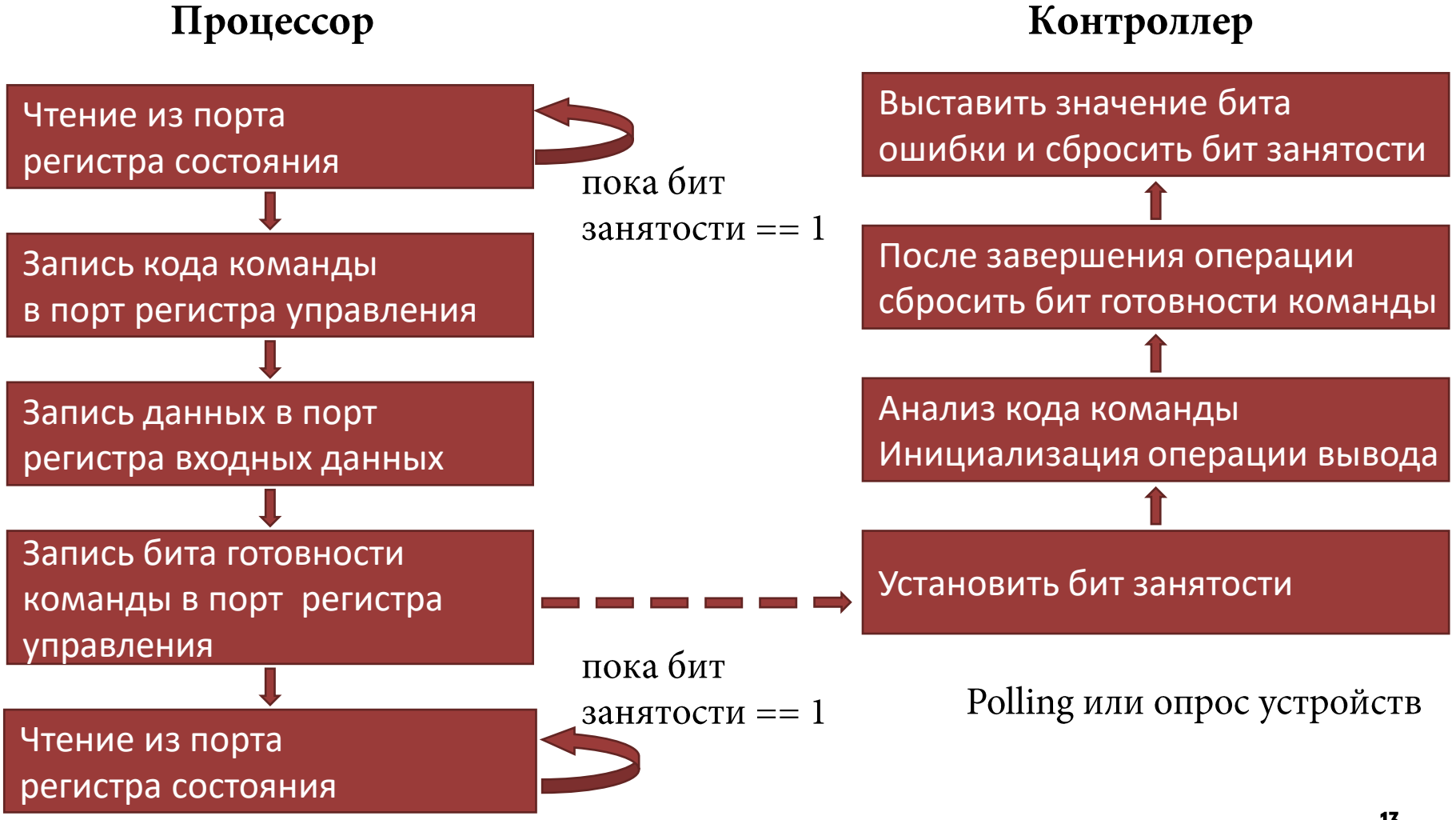
- Биты кода команды
- Биты режима работы
- Бит готовности команды

Регистр выходных данных
(read only со стороны процессора)

Регистр входных данных
(write only со стороны процессора)

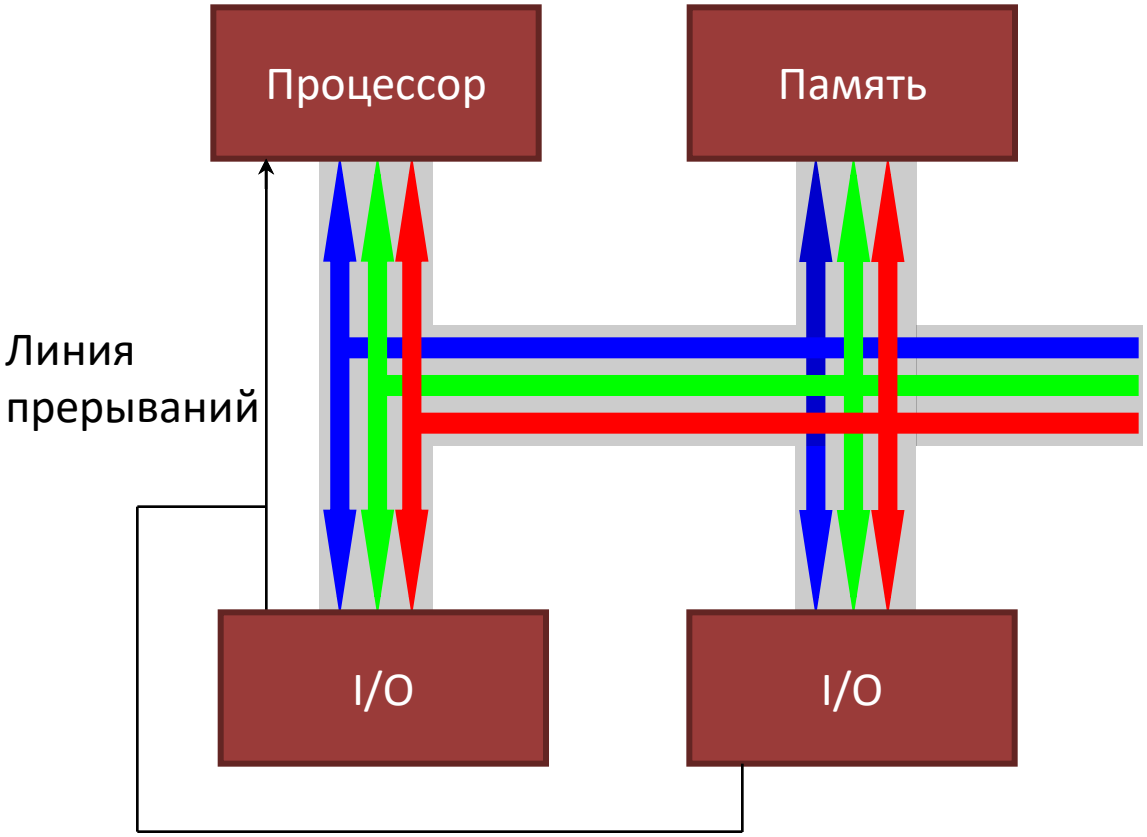
ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Вывод данных на внешнее устройство



ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

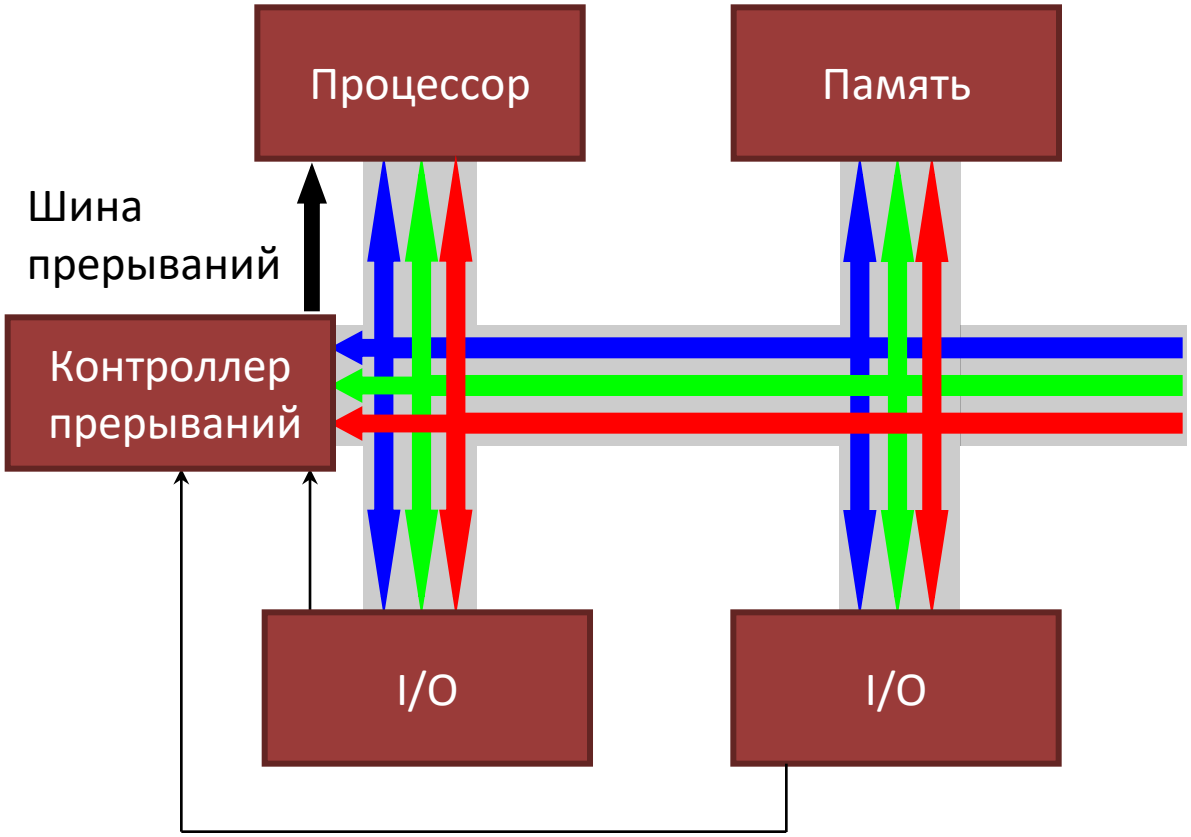
Вывод данных на внешнее устройство



1. После выполнения команды процессор обнаруживает сигнал на линии прерываний
2. Сохраняет часть регистров
3. Передает управление по заранее определенному адресу
4. Обрабатывает прерывание
5. Восстанавливает контекст

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Вывод данных на внешнее устройство



1. После выполнения команды процессор обнаруживает сигнал на линии прерываний
2. Сохраняет часть регистров
3. Передает управление по заранее определенному адресу
4. Обрабатывает прерывание
5. Восстанавливает контекст

Внешние прерывания, исключительные ситуации и программные прерывания

Внешние прерывания

- Обнаруживаются процессором ***между*** выполнением команд
- Сохраняется часть контекста перед выполнением ***следующей*** команды
- ***Не связаны*** с работой процессора и ***непредсказуемы***

Исключительные ситуации

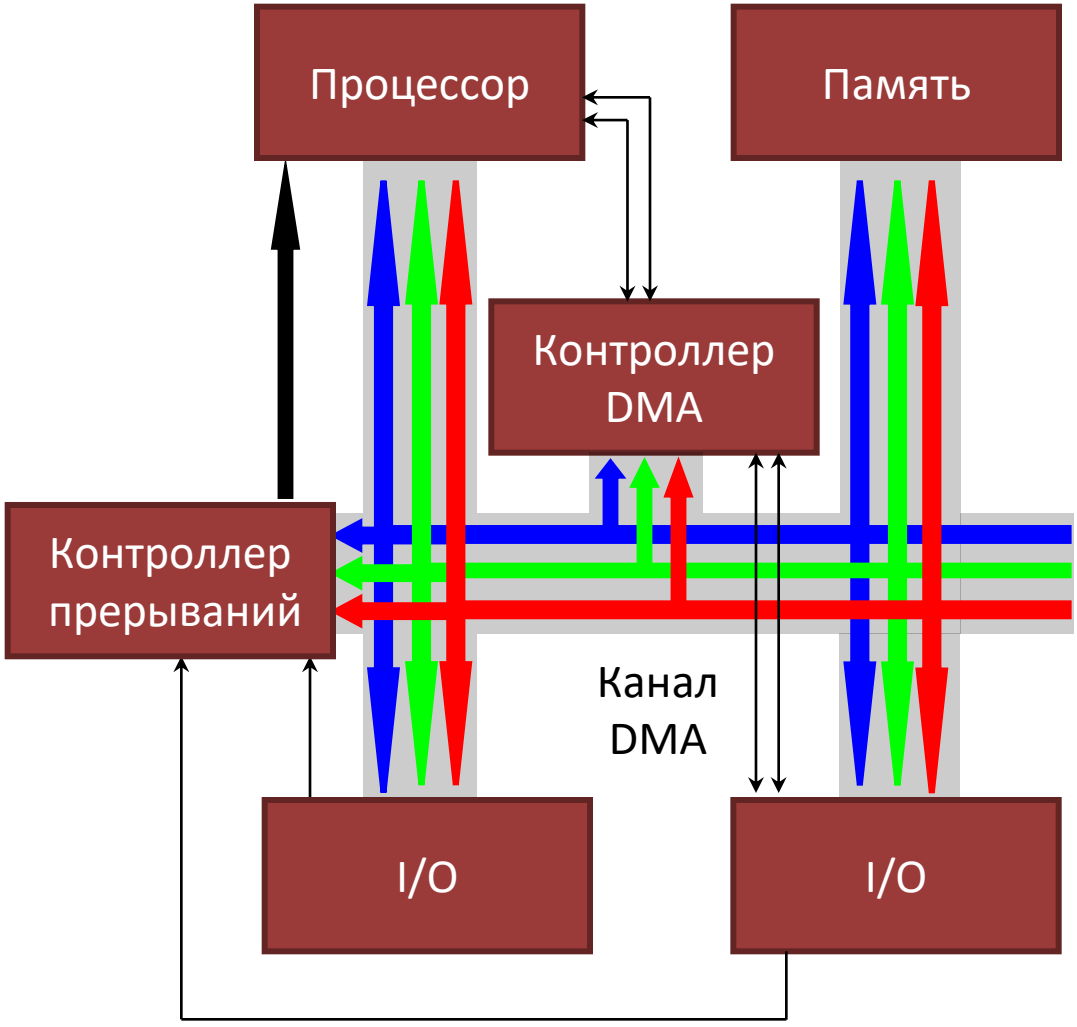
- Обнаруживаются ***во время*** выполнения команды
- Сохраняется часть контекста перед выполнением ***текущей*** команды
- ***Связаны*** с работой процессора, но ***непредсказуемы***

Программные прерывания

- Происходят ***в результате*** выполнения команды
- Сохраняется часть контекста перед выполнением ***следующей*** команды
- ***Связаны*** с работой процессора и ***предсказуемы***

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Прямой доступ к памяти
Direct memory access – DMA



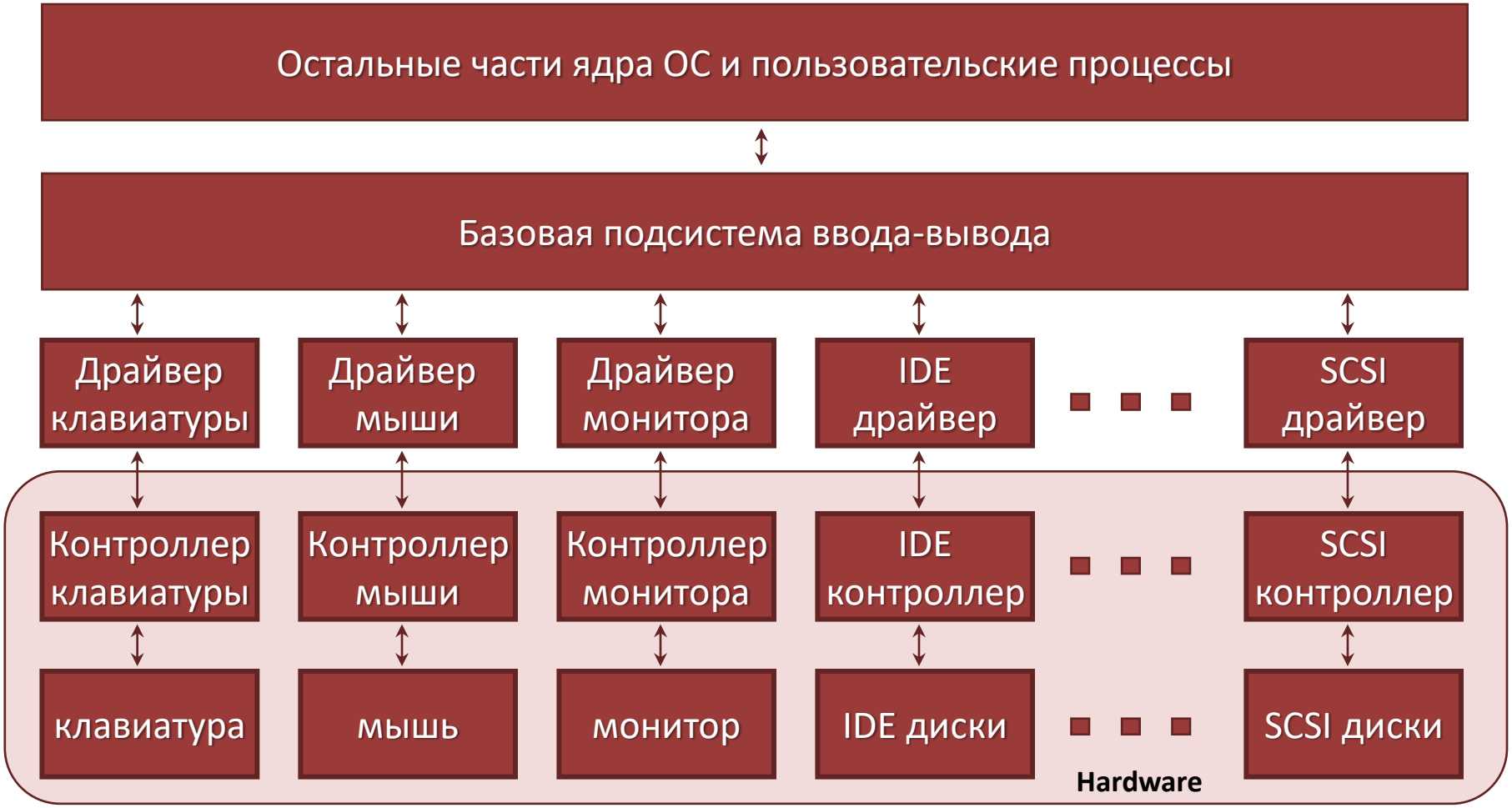
1. Контроллер DMA программируется
2. После получения сигнала от устройства I/O запрашивает у процессора управление магистралью
3. Получив управление, выставляет адрес и извещает устройство I/O
4. Используя шины данных и управления совместно с устройством I/O передает информацию
5. Возвращает управление магистралью

Основные направления различия устройств ввода-вывода

- Скорость обмена информацией (от нескольких байтов до нескольких Гигабайтов в секунду)
- Возможность использования несколькими процессами параллельно
- Запоминание выведенной информации для последующего ввода
- Символьные и блочные
- Только для ввода информации, только для вывода информации и read-write устройства

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Структура системы ввода-вывода



Систематизация внешних устройств

- Символьные устройства (клавиатура, модем, терминал и т.д.)
- Блочные устройства (магнитные и оптические диски и ленты и т.д.)
- Сетевые устройства (сетевые карты)
- Все остальные (таймеры, графические дисплеи, видеокамеры и т.д.)

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Интерфейс между базовой подсистемой ввода-вывода и драйверами

Символьные устройства

Ввести символ – get

Вывести символ – put

Блочные устройства

Прочитать блок – read

Записать блок – write

Найти блок – seek

- Выполнить произвольную команду – ioctl
- (Ре)инициализировать драйвер и устройство – open
- Временно завершить работу с устройством – close
- Остановить работу драйвера – stop
- Опросить состояние устройства – poll

Функции базовой подсистемы ввода-вывода

- Поддержка блокирующихся, неблокирующихся и асинхронных вызовов
- Буферизация и кэширование входных и выходных данных
- Осуществление spooling'а и монопольного захвата внешних устройств
- Обработка ошибок и прерываний
- Планирование последовательности запросов на выполнение операций ввода-вывода

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Блокирующиеся, неблокирующиеся и асинхронные вызовы

- При блокирующемся системном вызове процесс переходит из состояния исполнение в состояние ожидание. После выполнения операций ввода-вывода в полном объеме он разблокируется.
- При неблокирующемся системном вызове операции ввода-вывода могут быть выполнены неполностью. Процесс либо не блокируется совсем, либо блокируется не более чем на определенное время.
- При асинхронном системном вызове процесс никогда не блокируется. Операции ввода-вывода выполняются в полном объеме.

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Причины буферизации

- Разные скорости приема и передачи информации участников обмена
- Разные объемы данных, которые могут быть приняты или переданы участниками обмена единовременно
- Необходимость копирования данных из приложения в ядро ОС и обратно

Буфер – область памяти для запоминания информации при обмене данными между устройствами, процессами или между устройством и процессом

Разница между кэшем и буфером

- Буфер служит для согласования параметров участников обмена информацией и для ее промежуточного хранения. Кэш применяется для ускорения доступа к данным.
- Кэш всегда содержит копию данных, существующих где-либо еще. Буфер часто содержит единственный экземпляр данных в системе.

Кэш (cache) – область быстрой памяти, содержащая копию данных, расположенных где-либо в более медленной памяти, предназначенная для ускорения работы вычислительной системы

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Spooling и захват устройств

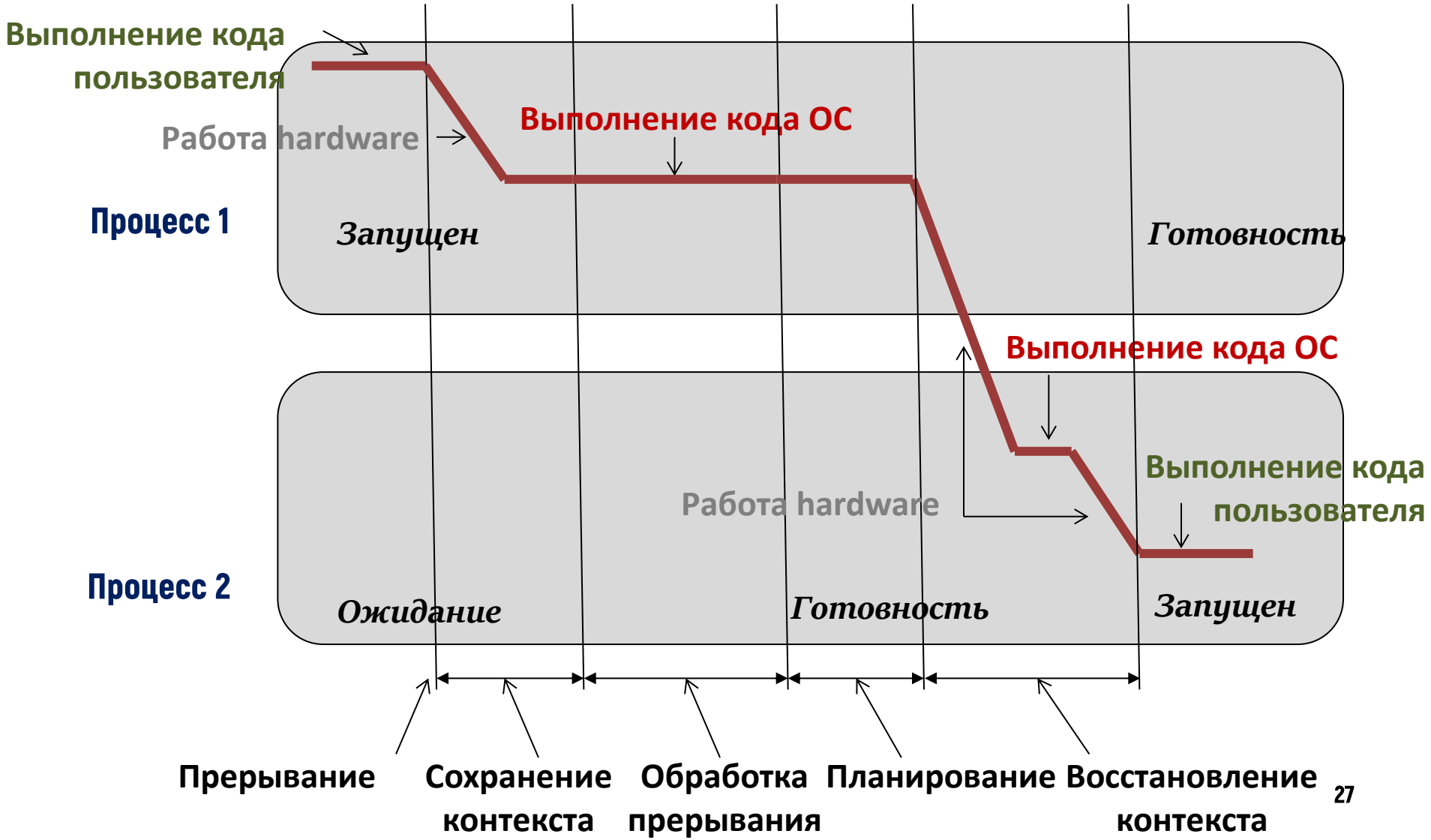
Способы использования неразделяемых устройств

- Монопольный захват устройства.
- Spooling.

Spool – буфер, содержащий входные или выходные данные для устройства, на котором следует избегать чередования его использования различными процессами

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Обработка прерываний и ошибок



ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Обработка прерываний и ошибок

Действия операционной системы

- Определение устройства, выдавшего прерывание.
- Взаимодействие с устройством.
- Проверка успешности выполнения операции.
- Попытка устранения возможных ошибок.
- Определение процесса, ожидающего этого прерывания. Перевод его из состояния *ожидание* в состояние *готовность*.
- Если есть еще процессы с неудовлетворенными запросами к этому устройству – инициализация нового запроса.

Действия по обработке прерывания и компенсации ошибок могут быть частично делегированы драйверу устройства – функция intr в интерфейсе драйвера

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Планирование запросов

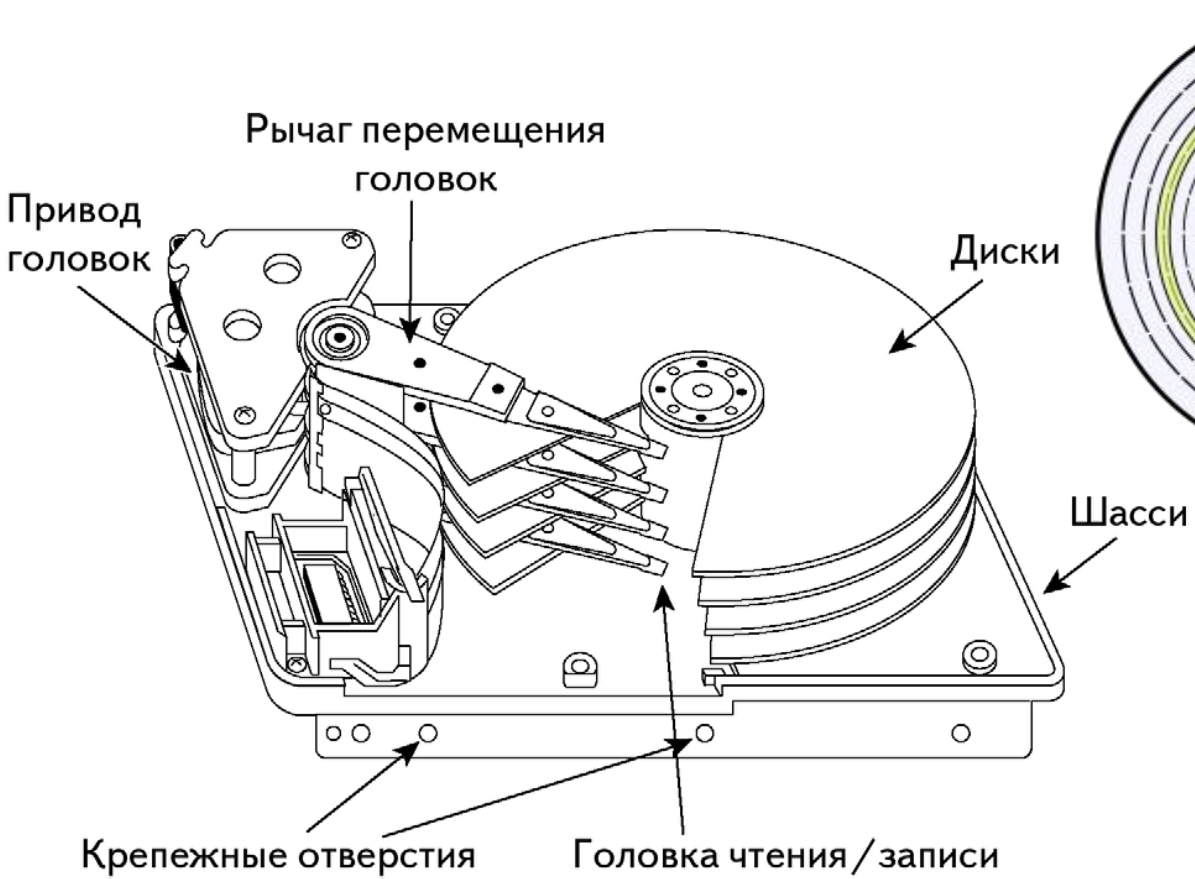
Для блокирующихся и асинхронных системных вызовов

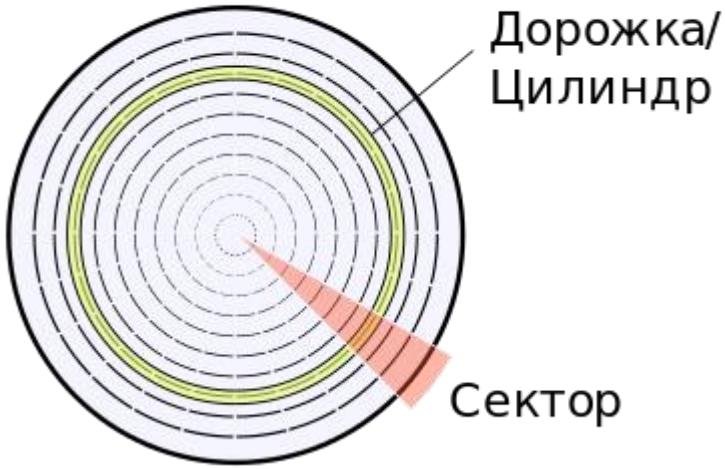
- При занятости устройства запрос ставится в очередь к данному устройству.
- После освобождения устройства необходимо принять решение: какой из запросов в очереди инициировать следующим – планирование запросов.

Действия по планированию запросов могут быть частично или полностью делегированы драйверу устройства – функция strategy в интерфейсе драйвера

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК
имени И.И. Воровича

Строение жесткого диска





ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Параметры планирования

- Запрос полностью характеризуется:
  - типом операции
  - номером цилиндра
  - номером дорожки
  - номером сектора
- Параметр планирование – время, необходимое для выполнения запроса.
  - Время выполнения запроса = transfer time + positioning time
  - Positioning time = seek time + positioning latency

Единственным параметром запроса остается seek time – время пропорциональное разнице между номером цилиндра в запросе и номером текущего цилиндра

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Алгоритмы планирования запросов к жесткому диску

- Пусть
  - Диск имеет 100 цилиндров (от 0 до 99)
  - Очередь запросов: **23, 67, 55, 14, 31, 7, 84, 10**
  - Текущий цилиндр – **63**

- Алгоритм FCFS (First Come First Served)

  63 -> 23 -> 67 -> 55 -> 14 -> 31 -> 07 -> 84 -> 10

  Всего перемещение на 329 цилиндров

- Алгоритм SSTF (Short Seek Time First)

  63 -> 67 -> 55 -> 31 -> 23 -> 14 -> 10 -> 07 -> 84

  Всего перемещение на 141 цилиндр

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Алгоритмы планирования запросов к жесткому диску

- Пусть
  - Диск имеет 100 цилиндров (от 0 до 99)
  - Очередь запросов: **23, 67, 55, 14, 31, 7, 84, 10**
  - Текущий цилиндр – **63**

- Алгоритм SCAN

  63 -> 55 -> 31 -> 23 -> 14 -> 10 -> 07 -> 0 -> 67 -> 84

  Всего перемещение на 147 цилиндров
- Алгоритм LOOK

  63 -> 55 -> 31 -> 23 -> 14 -> 10 -> 07 -> 67 -> 84

  Всего перемещение на 133 цилиндра
- Алгоритм C-SCAN

  63 -> 55 -> 31 -> 23 -> 14 -> 10 -> 07 -> 0 -> 99 -> 84 -> 67
- Алгоритм C-LOOK

  63 -> 55 -> 31 -> 23 -> 14 -> 10 -> 07 -> 84 ->67

ИНСТИТУТ МАТЕМАТИКИ МЕХАНИКИ КОМПЬЮТЕРНЫХ НАУК имени И.И. Воровича

Домашнее задание

- Читать книгу Таненбаум Э., Бос Х. Современные операционные системы, стр. 380-480.