

Системы ввода/вывода

Лекция 11.

Операционные системы. Linux

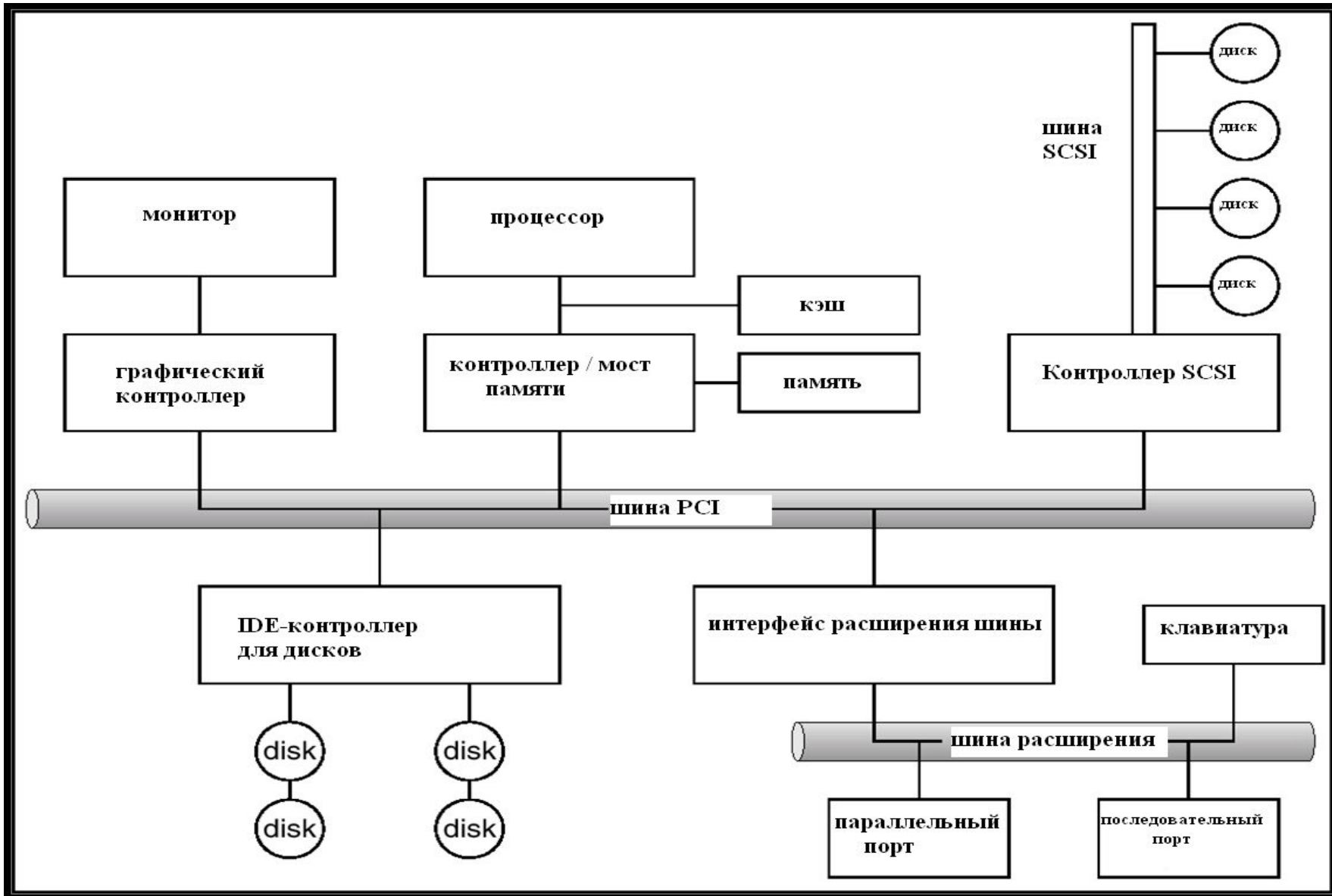
СИСТЕМЫ ВВОДА-ВЫВОДА

- **Аппаратура ввода-вывода**
- **Интерфейс приложений для ввода-вывода**
- **Подсистема ввода-вывода в ядре ОС**
- **Преобразование запросов на ввод-вывод в аппаратные операции**
- **Производительность систем ввода-вывода**

Аппаратура для ввода-вывода

- **Многообразие устройств ввода-вывода**
(HDD, streamers, flash, ZIP drives, JAZ drives, optical drives, DVD, Blu-Ray, etc.)
- **Общие концепции**
 - Порт
 - Шина (bus) - цепочка устройств прямого доступа
 - Контроллер (host adapter)
- **Устройства управления командами ввода-вывода**
- **Устройства имеют адреса, используемые:**
 - Командами непосредственного ввода-вывода
 - Командами ввода-вывода, отображаемого в память

Типовая структура шины ПК



Расположение портов для устройств на ПК (частично)

диапазон адресов устройств ввода-вывода (шестнадцатиричных)	устройство
000-00F	DMA-контроллер
020-021	контроллер прерываний
040-043	таймер
200-20F	игровой контроллер
2F8-2FF	последовательный порт (вторичный)
320-32F	контроллер жесткого диска
378-37F	параллельный порт
3D0-3DF	графический контроллер
3F0-3F7	контроллер гибких дисков (дискет)
3F8-3FF	последовательный порт (первичный)

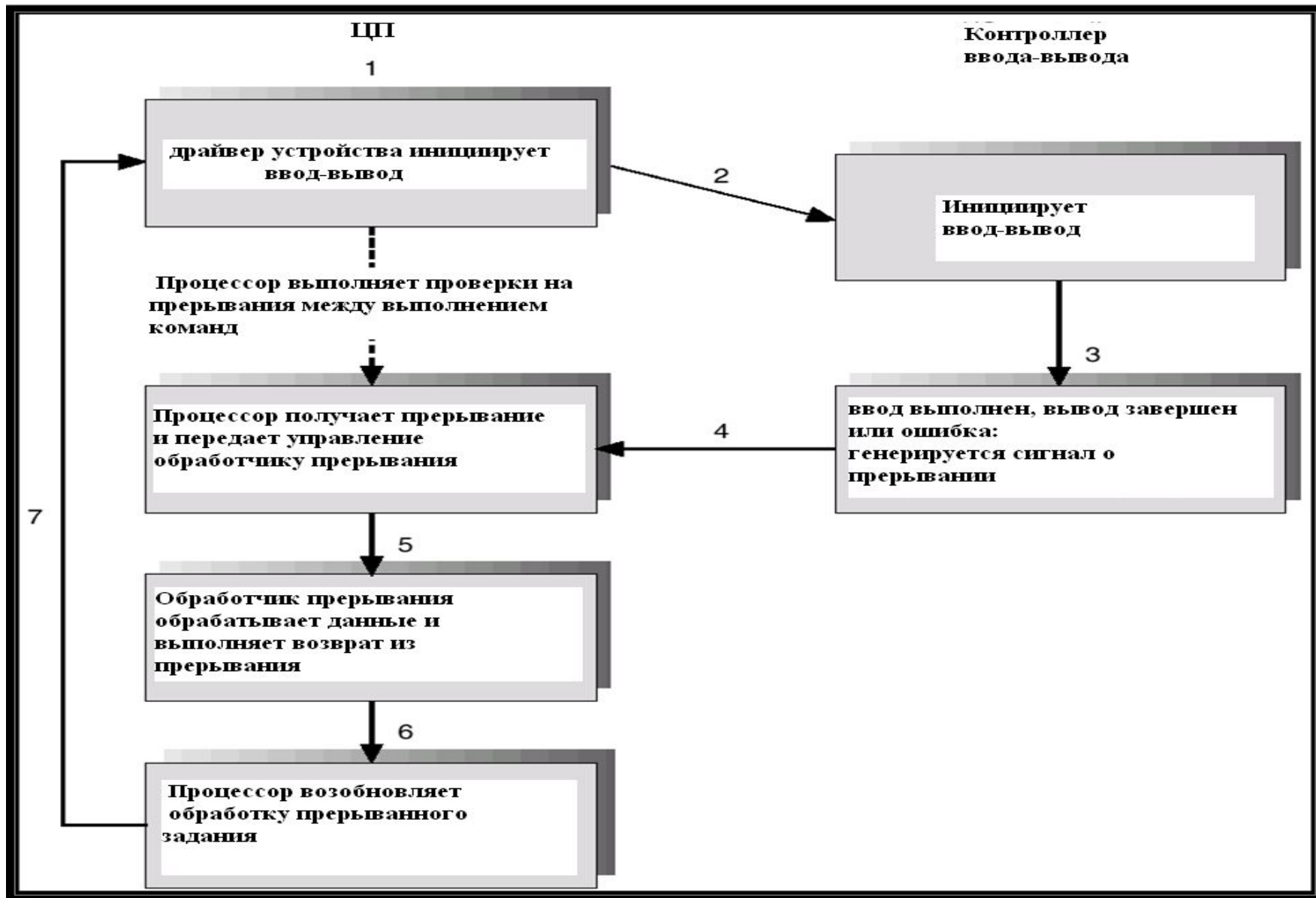
Опрос устройств (polling)

- Определяет состояние устройства
 - `command-ready`
 - `busy`
 - `error`
- Цикл `busy-wait` ожидания ввода-вывода с устройством

Прерывания

- Линия запросов на прерывания переключается устройством ввода-вывода
- Обработчик прерываний получает сигнал о прерывании
- Сигнал может быть замаскирован (**maskable**), чтобы игнорировать или задержать прерывание
- Вектор прерываний – используется с целью переадресовки прерывания для обработки соответствующим обработчиком (**handler**)
 - Основан на приоритетах
 - Не маскируемый

Цикл ввода-вывода, управляемого прерываниями



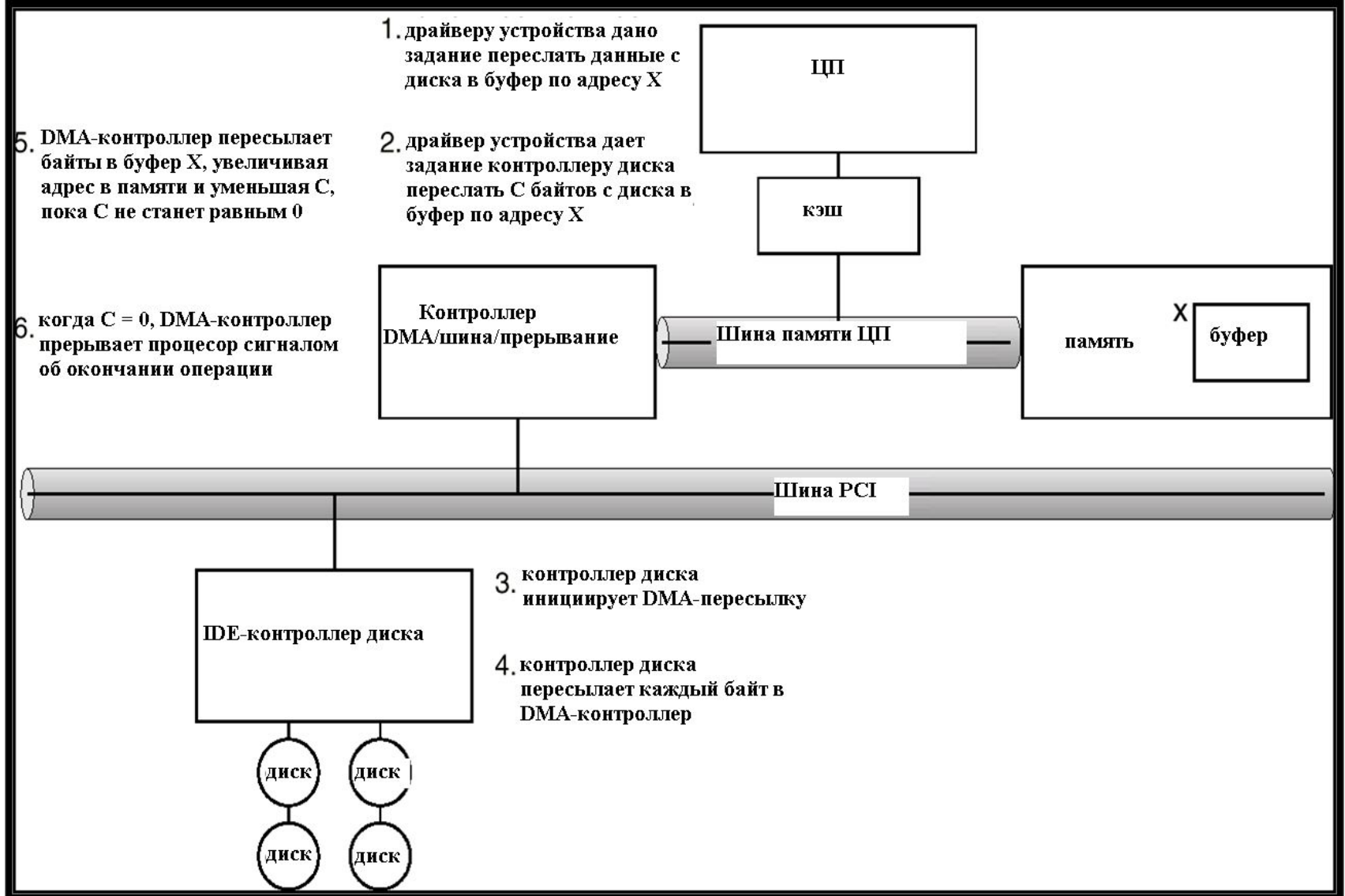
Вектор прерываний (событий) в процессоре Intel Pentium

номер прерывания	описание
0	ошибка при делении
1	исключение при отладке
2	прерывание по <code>pull</code>
3	точка остановки
4	прерывание, обнаруженное <code>INTO</code>
5	исключение по выходу за границы
6	неверный код операции
7	устройство недоступно
8	двойное прерывание
9	переполнение сегмента сопроцессора
10	неверный сегмент состояния задачи
11	сегмент отсутствует
12	ошибка стека
13	общее прерывание по защите
14	отказ страницы
15	(зарезервировано Intel, не использовать)
16	ошибка в операции с плавающей точкой
17	контроль выравнивания
18	контроль аппаратуры
19 - 31	зарезервировано Intel, не использовать
32 - 255	маскируемые прерывания

Прямой доступ к памяти (Direct Memory Access)

- Используется с целью избежать программируемого ввода-вывода для больших пересылок данных
- Требуется специальная аппаратура – DMA-контроллера
- Позволяет избежать участия процессора в пересылках больших объемов данных непосредственно между устройством ввода-вывода и памятью

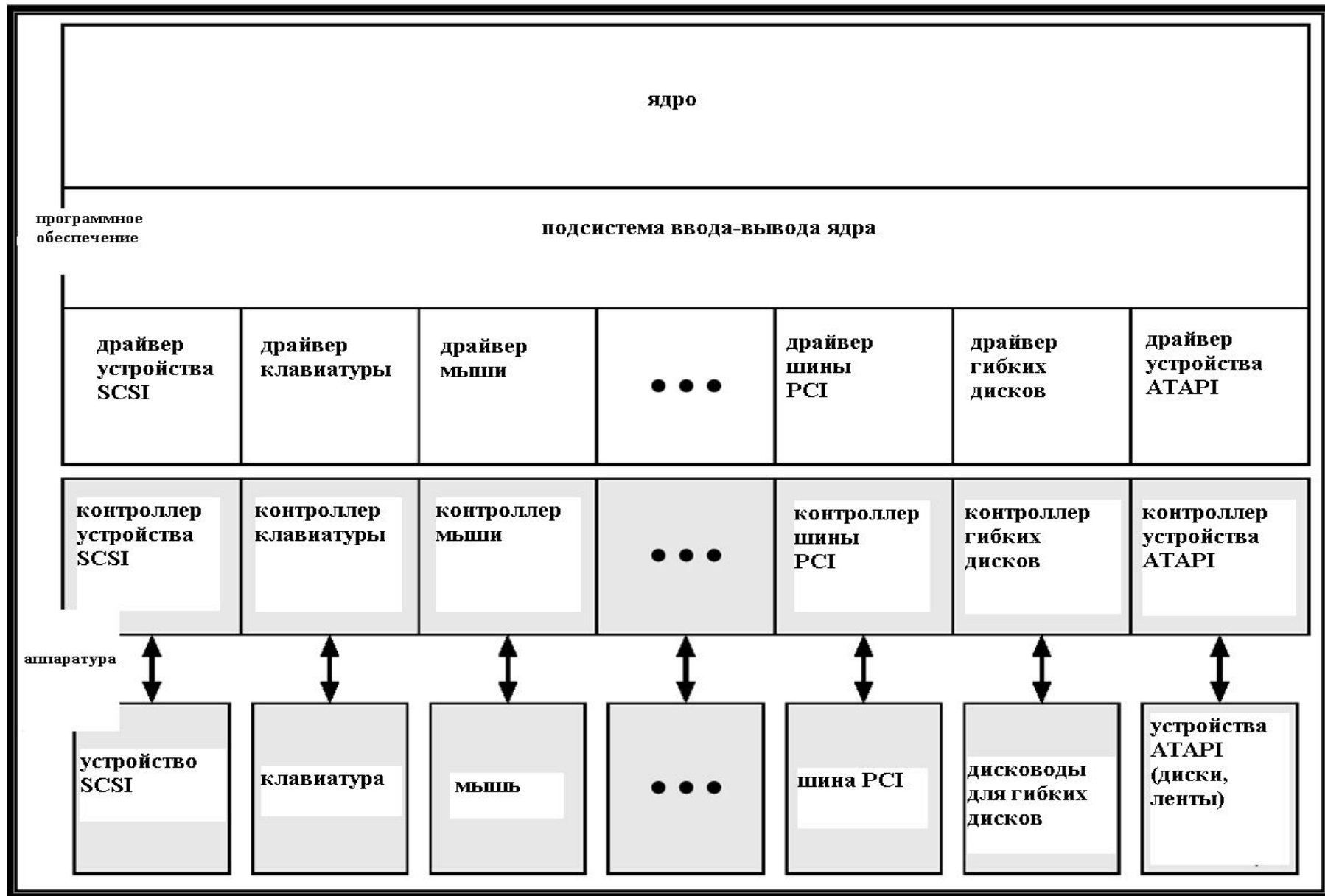
Процесс выполнения DMA (Direct Memory Access)



Программный интерфейс ввода-вывода

- Системные вызовы для ввода-вывода инкапсулируют поведение устройств ввода-вывода в общих (generic) классах
- Уровень драйвера устройства скрывает различия между контроллерами ввода-вывода от ядра ОС
- **Устройства различаются по многим параметрам:**
 - Работа с потоками символов или с блоками
 - Последовательный или прямой доступ
 - Разделяемое или специализированное
 - По скорости выполнения операций
 - Чтение/запись, или только чтение, или только запись

Структура модулей ввода-вывода в ядре



Характеристики устройств ввода-вывода

классификация	варианты	примеры
режим передачи данных	символьный блочный	терминал диск
метод доступа	последовательный произвольный	модем CD-ROM
метод передачи данных	синхронный асинхронный	лента клавиатура
возможность совместного доступа	монопольный общий	лента клавиатура
скорость устройства	латентность время поиска скорость передачи задержка между операциями	
направленность ввода-вывода	только чтение только запись чтение-запись	CD-ROM графический контроллер диск

Блочные и символьные устройства

- **Блочные устройства – устройства управления дисками**
 - Команды: read, write, seek
 - Чистый (raw) ввод-вывод или доступ к файловой системе
 - Возможен доступ к memory-mapped - файлу
- **Символьные устройства – клавиатуры, мыши, последовательные порты**
 - Команды: `get`, `put`
 - Библиотеки верхнего уровня допускают построчное редактирование

Сетевые устройства

- Существенно отличаются от блочных и символьных; имеют свой собственный интерфейс
- Unix и Windows NT/9i/2000 включают сокетный интерфейс
 - Отделяет сетевой протокол от сетевой операции
 - Включает функцию **select**
- Различны по подходам к реализации (конвейеры - pipes, FIFO, потоки, очереди, почтовые ящики)

Часы и таймеры

- Поддерживают информацию о текущем времени, прошедшем (**elapsed**) отрезке времени, таймере
- Если программируемые интервалы времени используются для работы с таймером – периодические прерывания
- **ioctl** (UNIX) – поддерживает работу с часами и таймером

Блокируемый (синхронный) и не блокируемый (асинхронный) ввод-вывод

ВЫВОД

- **Блокируемый** – процесс задерживается, пока ввод-вывод не закончится
 - Прост для использования и понимания
 - Недостаточен для некоторых применений
- **Не блокируемый** – возврат из системного вызова для ввода-вывода происходит по мере доступности информации
 - Пользовательский интерфейс для копирования данных (буферизация)
 - Реализация с помощью многопоточности (multi-threading): ввод-вывод выделяется в отдельный поток
 - Быстрый возврат с числом байтов, фактически прочитанным или записанным
- **Асинхронный** – процесс выполняется одновременно с выполнением ввода-вывода
 - Сложен в использовании
 - Подсистема ввода-вывода генерирует сигнал в

Подсистема ввода-вывода в ядре ОС

- **Планирование**

- Упорядочение запросов на ввод-вывод в очередях к каждому устройству

- **Буферизация** – запись данных в память в процессе передачи между устройствами

- Для балансировки устройств с разными скоростями

- Чтобы справиться с несоответствием размера данных для работы с устройством

- Для поддержки “семантики копирования”

Подсистема ввода/вывода в ядре ОС

- **Кэширование** – быстрая память, в которой хранится копия данных
 - Всегда только копия
 - Ключ к повышению производительности
- **Буферизация ввода-вывода (spooling)** – задержка вывода на устройство
 - Если устройство может обработать только одну операцию в каждый момент
 - Например, печать
- **Резервирование устройства** – обеспечение монопольного доступа к устройству
 - Системные вызовы для размещения и освобождения
 - Контроль отсутствия тупиков (deadlocks)

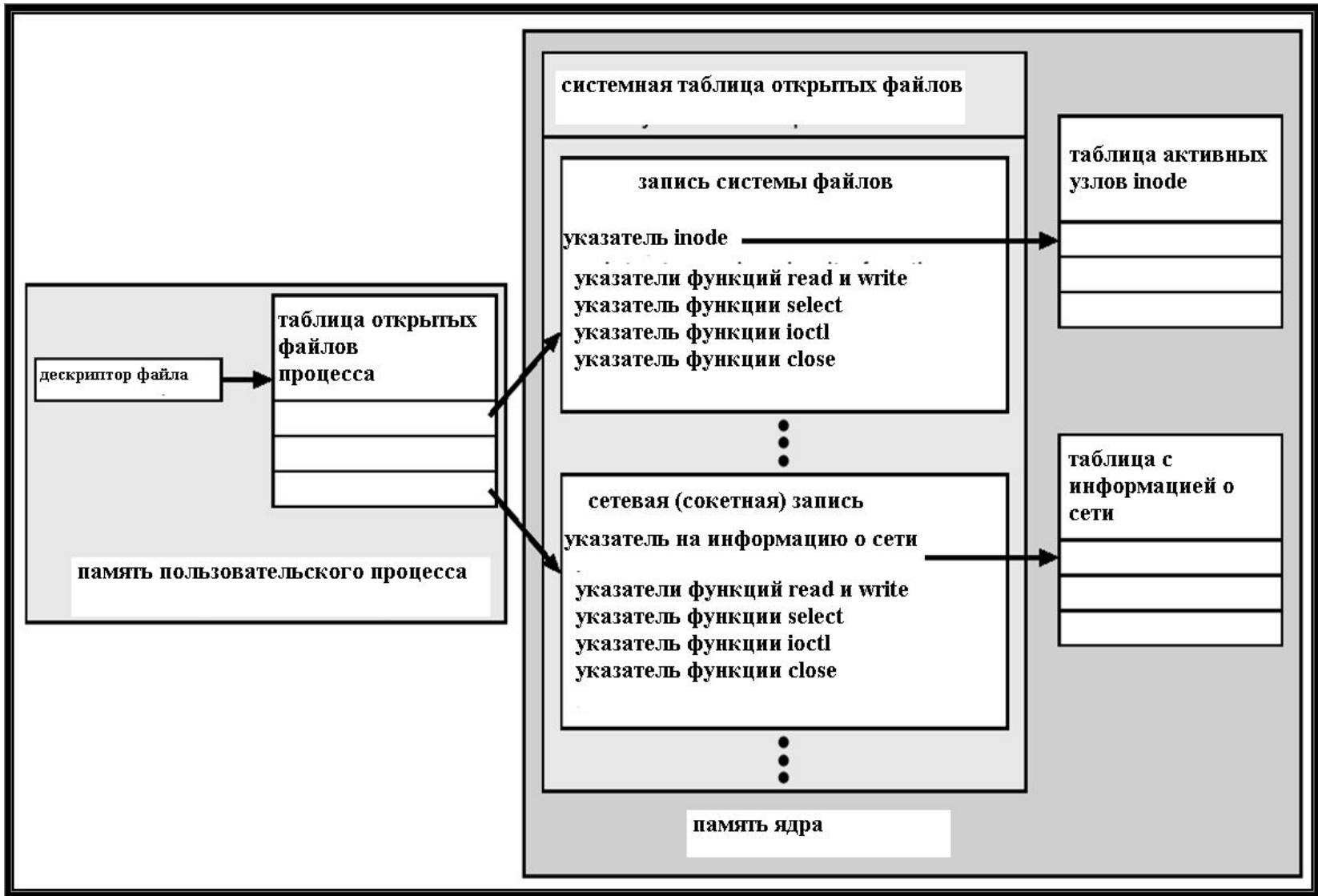
Обработка ошибок

- ОС поддерживает восстановление информации после чтения с диска, недоступности устройства, временных сбоев при записи
- В большинстве случаев возвращается номер (код) ошибки, в случае, если запрос на ввод-вывод завершается неудачно
- В системных журналах хранится информация об обнаруженных проблемах

Структуры данных в ядре ОС

- В ядре хранится информация о состоянии для компонент ввода-вывода, включая таблицы открытых файлов, сетевых соединений, состояние символьных устройств
- Очень большое число сложных структур данных для контроля буферов, распределения памяти и др.
- Многие из них используют объектно-ориентированные методы и передачу сообщений для реализации ввода-вывода

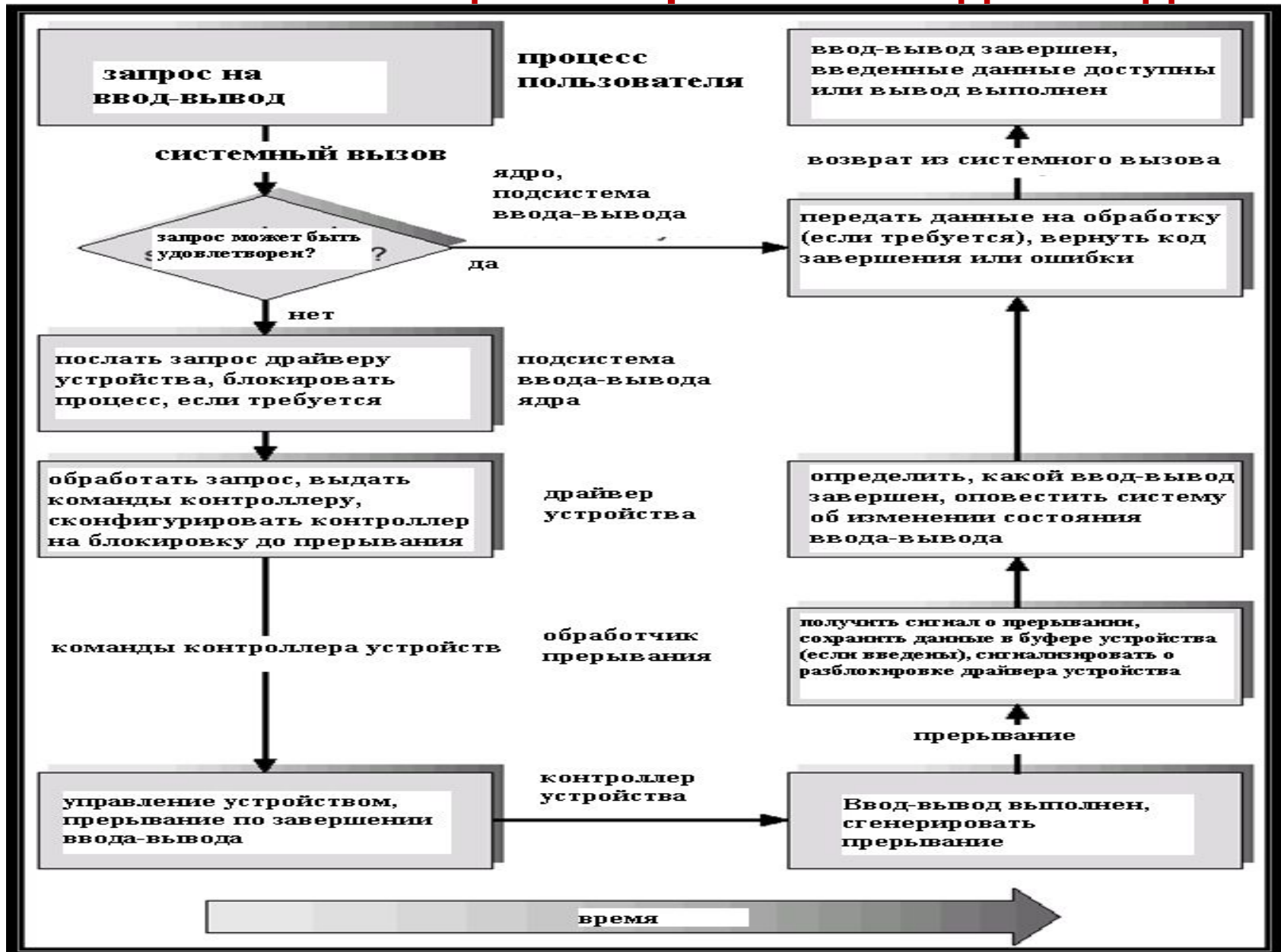
Структура модулей ввода-вывода в ядре UNIX



Запросы ввода-вывода на аппаратные операции

- Рассмотрим процесс чтения из дискового файла:
 - Определение устройства, на котором хранится файл
 - Трансляция имени в представление устройства
 - Физически считанные данные с диска в буфере
 - Данные становятся доступными для запросившего их процесса
 - Управление возвращается процессу

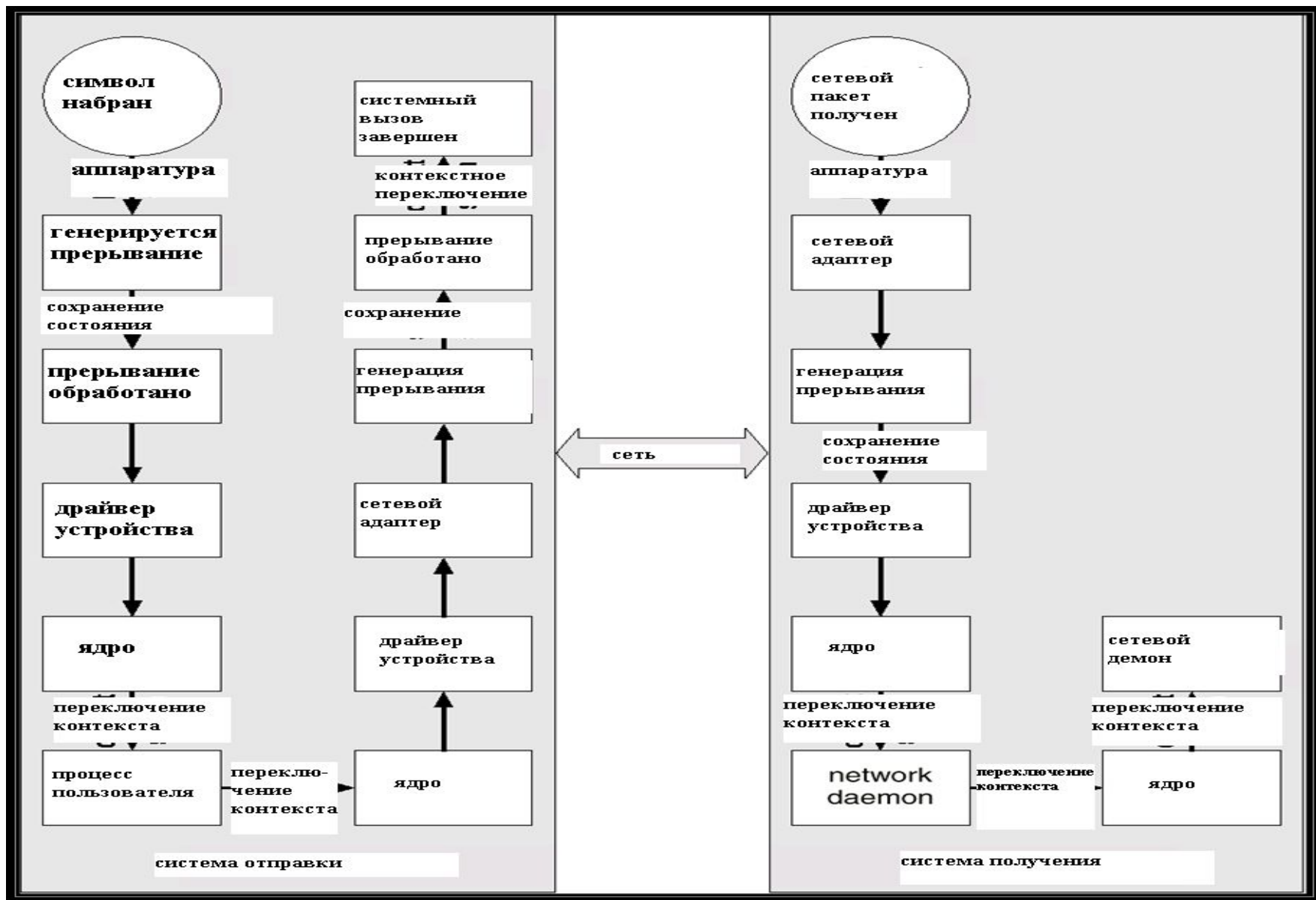
Жизненный цикл запроса на ввод-вывод



Производительность

- **Ввод-вывод** – важный фактор в производительности системы:
 - Требуется от процессора исполнения драйвера устройства - кода уровня ядра ОС
 - Контекстные переключения, связанные с прерываниями
 - Копирование данных
 - Особенно напряженным является сетевой трафик

Взаимодействие между компьютерами



Улучшение производительности

- Сократить число контекстных переключений
- Сократить объем копирования данных
- Сократить число прерываний, используя большие переходы, интеллектуальные контроллеры, опрос устройств
- Использовать DMA (Direct Memory Access)
- Сбалансировать нагрузку на процессор, память и шину и производительность ввода-вывода с целью повышения суммарной производительности