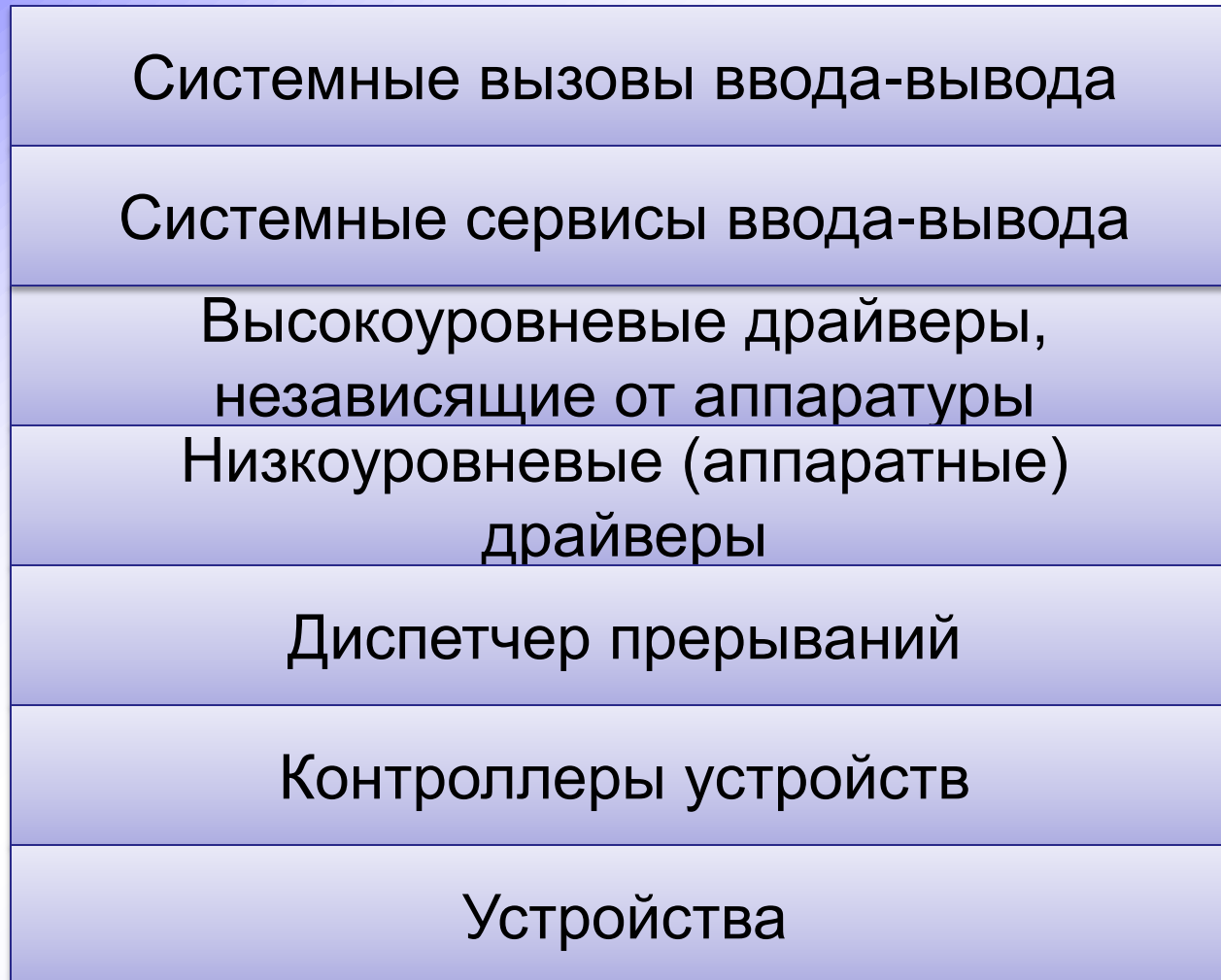


# Управление устройствами

(часть 2)

# Структура подсистемы ввода-вывода

# Общая схема управления устройствами ввода-вывода



# Слои подсистемы управления устройствами ввода-вывода

- Верхний слой – системные вызовы ввода-вывода.

Задачи: переадресация пользовательских запросов драйверам соответствующих устройств ввода-вывода и возвращение результатов.

- Нижний слой – взаимодействие с контроллерами внешних устройств.

Задача: экранирование драйверов от особенностей аппаратной платформы. Может включать диспетчер прерываний.

# Слои подсистемы управления устройствами ввода-вывода

- Слой системных сервисов ввода-вывода – диспетчер ввода-вывода, диспетчер Plug and Play, диспетчер электропитания.

Задачи: создание стандартного внутреннего интерфейса, облегчающего взаимодействие компонентов подсистемы друг с другом, а также организация взаимодействия с модулями других подсистем.

# Информация, необходимая для управления устройствами

- Состав устройств ввода-вывода и способы их подключения.
- Аппаратные ресурсы, закреплённые за имеющимися в системе устройствами ввода-вывода.
- Логические (символьные) имена устройств, используя которые вычислительные процессы могут запрашивать те или иные операции ввода-вывода.

# Информация, необходимая для управления устройствами

- Адреса размещения драйверов устройств ввода-вывода и области памяти для хранения текущих значений переменных, определяющих работу с этими устройствами.
- Области памяти для хранения информации о текущем состоянии устройства и параметрах, определяющих режимы работы устройства.

# Информация, необходимая для управления устройствами

- Данные о текущем процессе, который работает с данным устройством.
- Адреса тех областей памяти, которые содержат данные, участвующие в операциях ввода-вывода (получаемые при операциях ввода данных и выводимые на устройство при операциях вывода данных).



# Используемые информационные структуры

- Таблица оборудования, каждая запись которой соответствует устройству ввода-вывода.
- Таблица виртуальных устройств, устанавливающая соответствие виртуальных устройств реальным устройствам (записям таблицы оборудования).
- Таблица векторов прерываний.

# Содержимое таблицы оборудования

- Тип устройства, его конкретная модель, символическое имя и характеристики устройства.
- Способ подключения устройства (через какой интерфейс, к какому разъёму, какие порты и линия запроса прерывания используются и т. д.).
- Состояние устройства.

# Содержимое таблицы оборудования

- Номер и адрес канала, если он используется для управления устройством.
- Информация о том, используется или нет буферизация при обмене данными с устройством, «имя» (или просто адрес) буфера, если такой выделяется из системной области памяти.

# Содержимое таблицы оборудования

- Информация о драйвере, который должен управлять этим устройством; адреса, необходимые для работы с драйвером.
- Установка тайм-аута и ячейки для счётчика тайм-аута.
- Поле указателя для связи задач, ожидающих устройство.

# Диспетчер ввода-вывода

Определяет модель доставки запросов на ввод-вывод драйверам устройств:

- создаёт IRP (пакет запроса ввода-вывода);
- передаёт указатель на IRP соответствующему драйверу;
- удаляет пакет по завершении операции ввода-вывода.

Позволяет драйверу вызывать другие драйверы, управляет буферами запросов ввода-вывода и т.д.

# Структура IRP

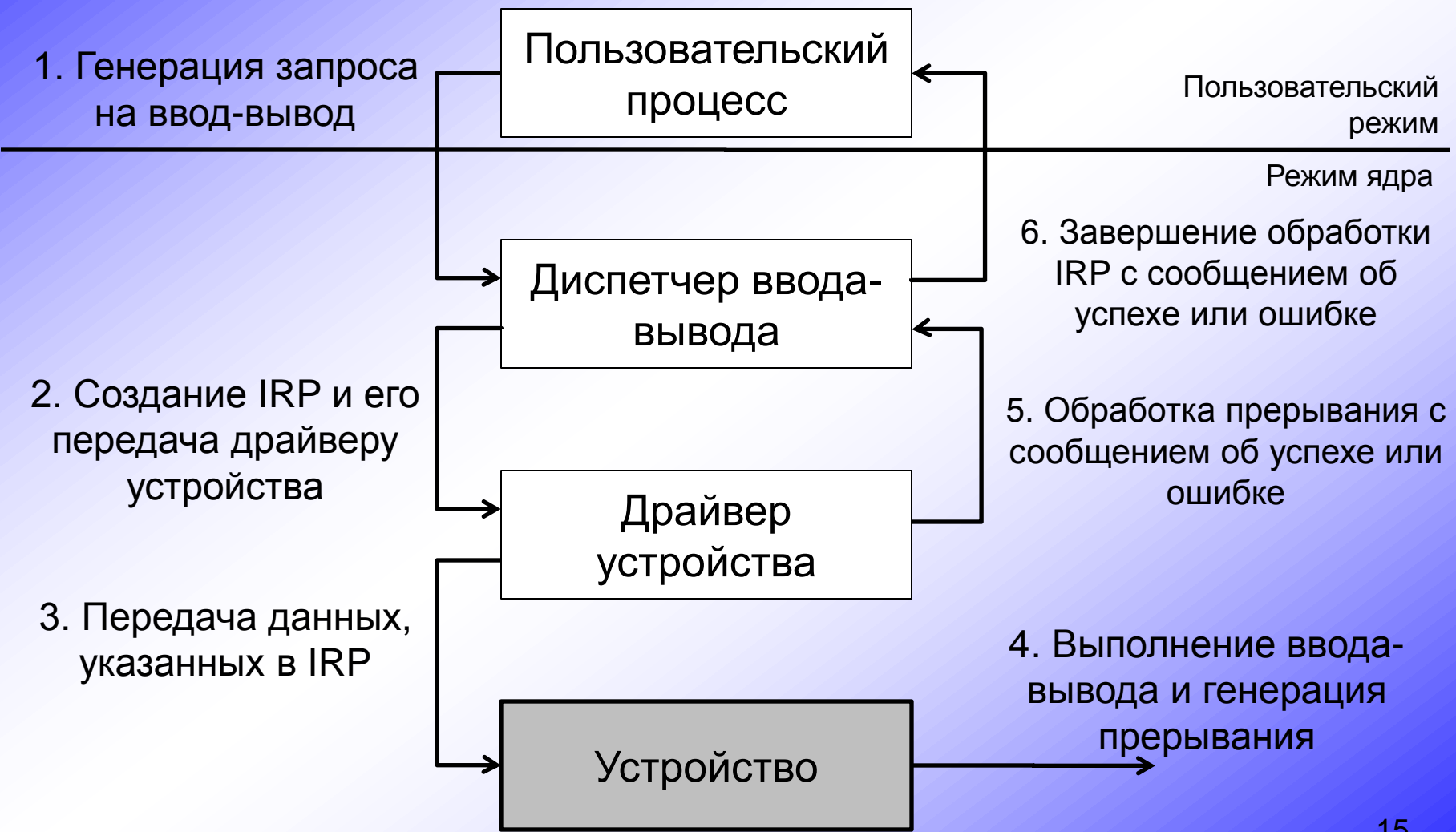
## Заголовок:

- тип и размер запроса;
- указатель на буфер, если ввод-вывод с использованием буфера;
- данные о состоянии, изменяющиеся по мере обработки запроса;
- сведения о синхронности или асинхронности запроса.

## Блок стека:

- номер функции;
- параметры, специфичные для функции;
- указатель на представление в памяти разделяемых ресурсов вызывающего потока.

# Запрос ввода-вывода

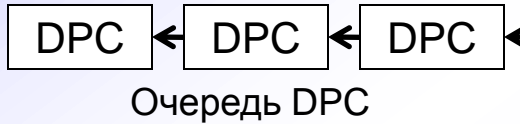


# Обслуживание прерывания от устройства

Драйвер устройства



3. Обработчик перехватывает прерывание, выполняет функции, критичные ко времени выполнения, и помещает DPC в очередь.



2. Диспетчер прерываний передаёт управление обработчику прерывания этого устройства.



Уровни IRQL

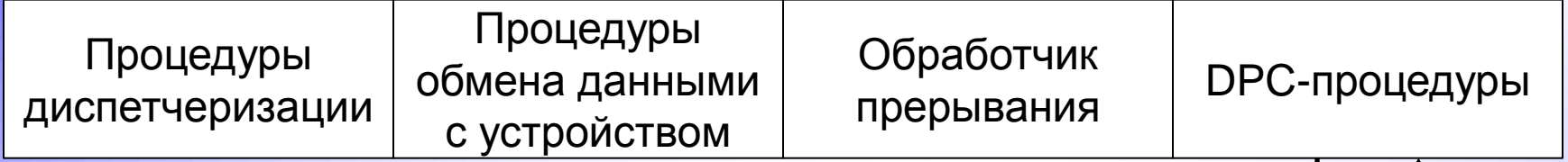
1. Устройство генерирует прерывание



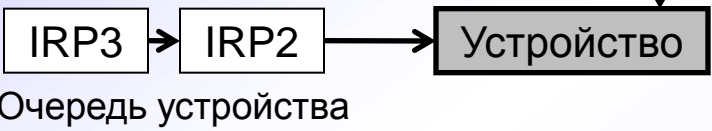


# Обслуживание прерывания от устройства

Драйвер устройства



3. DPC-процедура начинает обработку запросов из очереди устройства, после чего заканчивает обслуживание прерывания.



1. IRQL снижается и начинается обработка DPC.



Уровни IRQL

2. Диспетчер прерываний передаёт управление DPC-процедуре драйвера.

# Завершение обработки запроса на ввод-вывод

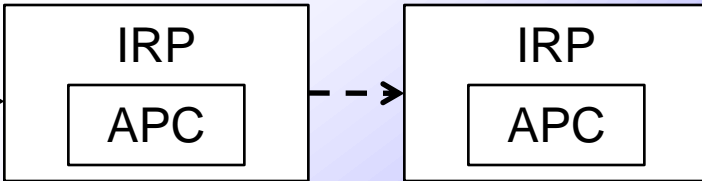
Диспетчер ввода-вывода



2. Диспетчер ввода-вывода ставит APC в очередь для завершения обработки IRP в контексте вызывающего потока.

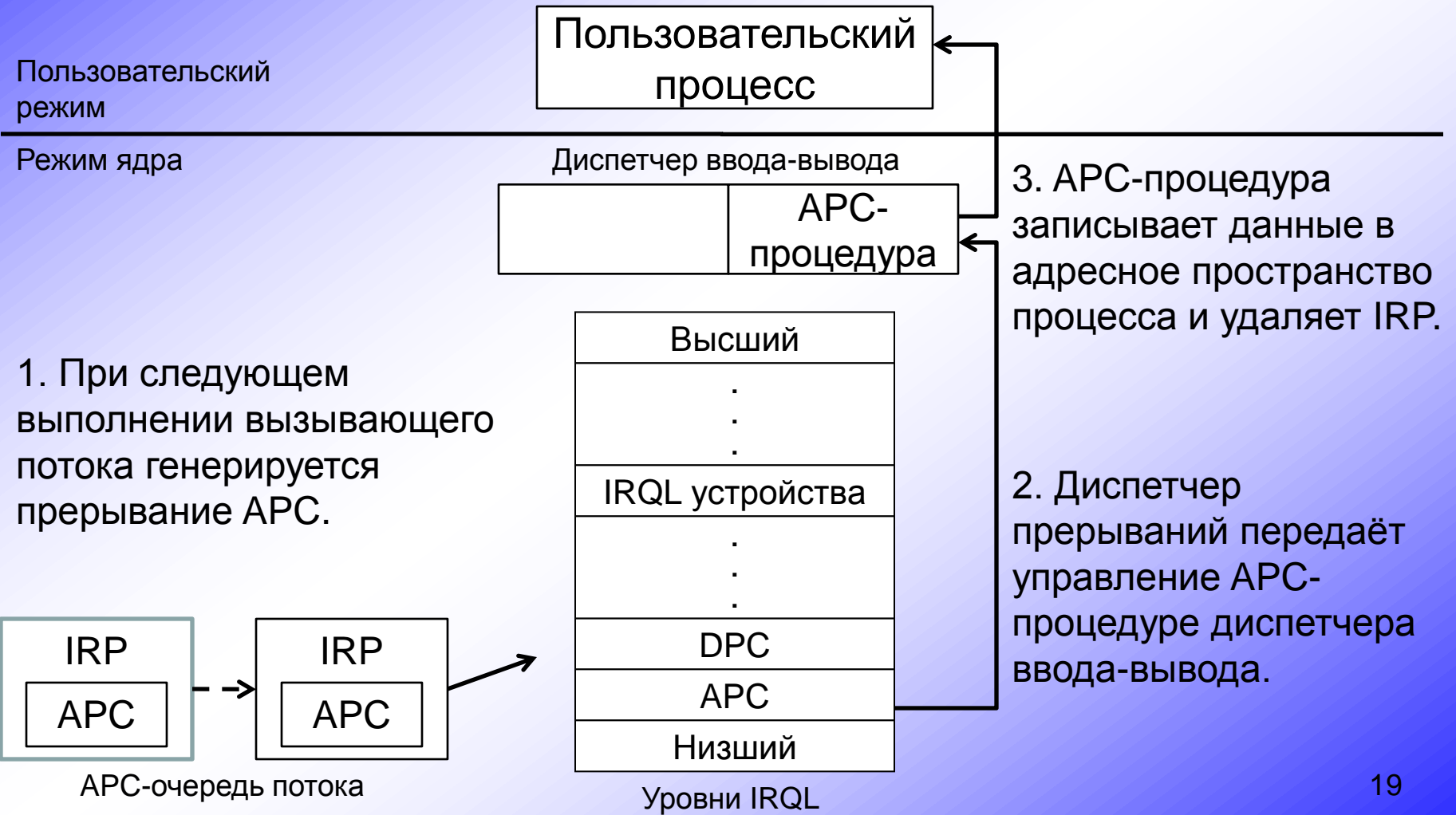
1. DPC-процедура вызывает диспетчер ввода-вывода для обработки исходного запроса (IRP).

Драйвер устройства



APC-очередь потока

# Завершение обработки запроса на ввод-вывод



# Функции диспетчера Plug and Play (PnP)

- Автоматическое распознавание установленных устройств (перечисление при загрузке и обнаружение или удаление во время работы системы).
- Выделение аппаратных ресурсов и их перераспределение (прерывания, диапазоны адресов ввода-вывода и т.д.).
- Загрузка драйверов соответствующих устройств.
- Позволяет приложениям и драйверам обнаруживать изменения в аппаратной конфигурации.

# Запросы диспетчера Plug and Play драйверам устройств

- Запрос описания устройств, обнаруженных на шинах, соответствующим драйверам шин.
- Запрос процедуры добавления устройства драйверам устройства.
- Запрос на запуск устройства с указанием ресурсов, назначенным устройству.
- Запрос на проверку возможности отключения устройства.
- Запрос на отключение устройства.

# Запросы диспетчера Plug and Play драйверам устройств

- Запрос на проверку возможности временной приостановки работы устройства при перераспределении ресурсов.
- Запрос на временную приостановку устройства.
- Запрос на немедленное прекращение работы драйвера при неожиданном удалении устройства.

# Переходы Plug and Play-состояний устройств



# Диспетчер электропитания

Состояние	Энергопотребление	Возобновление работы ПО	Аппаратная задержка
S0 (полностью активное)	Максимальное	–	Нет
S1 (состояние ожидания)	Менее S0, но более S2	Система возобновляет работу с той точки, где она была прервана (возвращение в S0)	Менее 2 секунд
S2 (состояние ожидания)	Менее S1, но более S3	Система возобновляет работу с той точки, где она была прервана (возвращение в S0)	От 2 секунд и более



# Диспетчер электропитания

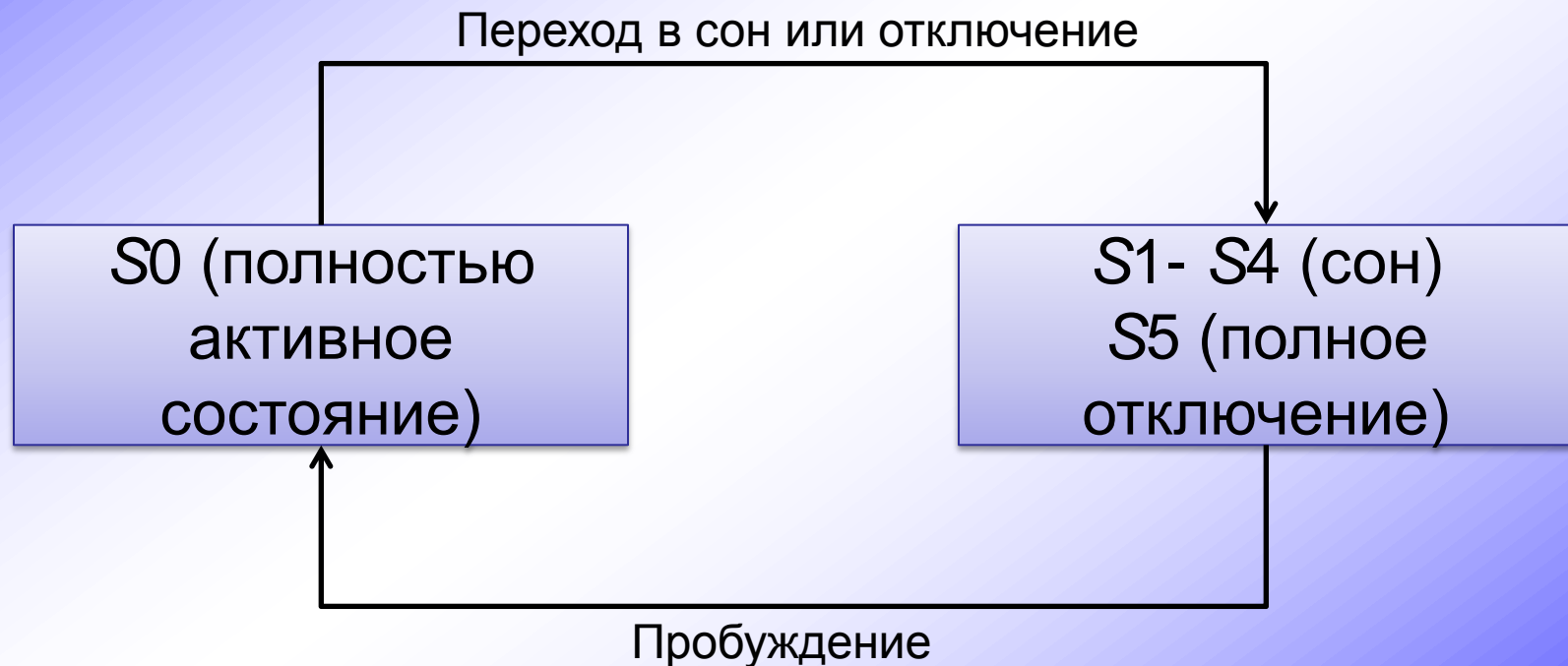
Состояние	Энергопотребление	Возобновление работы ПО	Аппаратная задержка
S3 (состояние ожидания)	Менее S2, процессор отключен	Система возобновляет работу с той точки, где она была прервана (возвращение в S0)	От 2 секунд и более
S4 (спящий режим)	Ток подаётся на кнопку включения электропитания и контур пробуждения	Система возобновляет работу с той точки, где она была прервана (возвращение в S0)	Длительная, неопределённая
S5 (полное отключение)	Ток подаётся на кнопку включения электропитания	Система загружается заново	Длительная, неопределённая

# Диспетчер электропитания

Решение о переходе из одного состояния в другое принимается на основе:

- уровня активности системы;
- уровня заряда аккумуляторов;
- наличия запросов приложений на выключение компьютера или перехода в ждущий/спящий режим;
- действий пользователя;
- заданных параметров электропитания.

# Переходы между состояниями энергопотребления



# Состояния энергопотребления устройств

- $D0$  – устройство полностью включено.
- $D1$ ,  $D2$  – промежуточные состояния.
- $D3$  – устройство полностью выключено.

При переходе системы из одного состояния в другое драйвер получает команду от диспетчера электропитания изменить состояние своего устройства.

Драйвер может самостоятельно понизить уровень энергопотребления устройства, например, при его долгом простое.

# Драйверы внешних устройств

# Особенности драйверов

- Входят в состав ядра ОС, работая в привилегированном режиме.
- Непосредственно управляют внешним устройством, взаимодействуя с его контроллером при помощи команд ввода-вывода.
- Обрабатывают прерывания от контроллера устройства.

# Особенности драйверов

- Предоставляют программисту удобный логический интерфейс для работы с устройством, экранируя низкоуровневые детали управления и организации данных.
- Взаимодействуют с другими модулями ядра ОС при помощи стандартизированного интерфейса, описывающего формат передаваемых данных, структуру буферов, способы включения драйвера в состав ОС, способы вызова драйвера и т.д.

# Блок- и байт-ориентированные драйверы

- Блок-ориентированные драйверы управляют устройствами, хранящими информацию в блоках фиксированного размера, каждый из которых имеет свой адрес (параллельно используемые). Возможно кэширование данных и поиск заданного блока.
- Байт-ориентированные драйверы управляют устройствами, генерирующими или потребляющими последовательности байт (последовательно используемые).



# Структура драйвера

- Секция запуска – запускается для включения устройства ввода-вывода или просто для инициализации очередной операции ввода-вывода.
- Одна или несколько секций продолжения – основной обработчик прерывания.
- Секция завершения – выключает устройство ввода-вывода или завершает операцию.

# Набор процедур секции запуска

- Инициализирующая – глобальная инициализация драйвера и регистрация остальных процедур.
- Добавления устройства – создание объекта «устройство» при получении уведомления обнаружения нового устройства от диспетчера PnP.

# Набор процедур секции запуска

- Процедуры диспетчеризации – реализация открытия, закрытия, чтения, записи и др. возможностей устройства.
- Инициации ввода-вывода – передача данных с устройства и на него в случае поддержки очередей входящих запросов.

# Секция продолжения

- Секций может быть несколько, если требуется несколько прерываний для выполнения одной логической операции.
- Если секция продолжения одна, то она сама передает управление в ту или иную часть кода подпрограммы обработки прерывания.
- Если секций продолжения несколько, то передача управления от одной секции к другой производится диспетчером прерываний.

# Набор процедур секции продолжения

- Обслуживания прерываний – получает управление при генерации прерывания устройством.
- DPC-процедура обработки прерываний – выполняет основную часть обработки прерываний, инициирует завершение текущей и выполнение следующей операции из DPC очереди на данном устройстве.

# Набор процедур секции завершения

- Завершения ввода-вывода – уведомляет об удачном или неудачном завершении (или отмене) операции и позволяет освободить ресурсы.
- Отмены ввода-вывода.
- Выгрузки – освобождает все системные ресурсы, задействованные драйвером.
- Уведомления о завершении работы системы – очистка при завершении работы системы.
- Регистрации ошибок.

# Многоуровневые драйверы

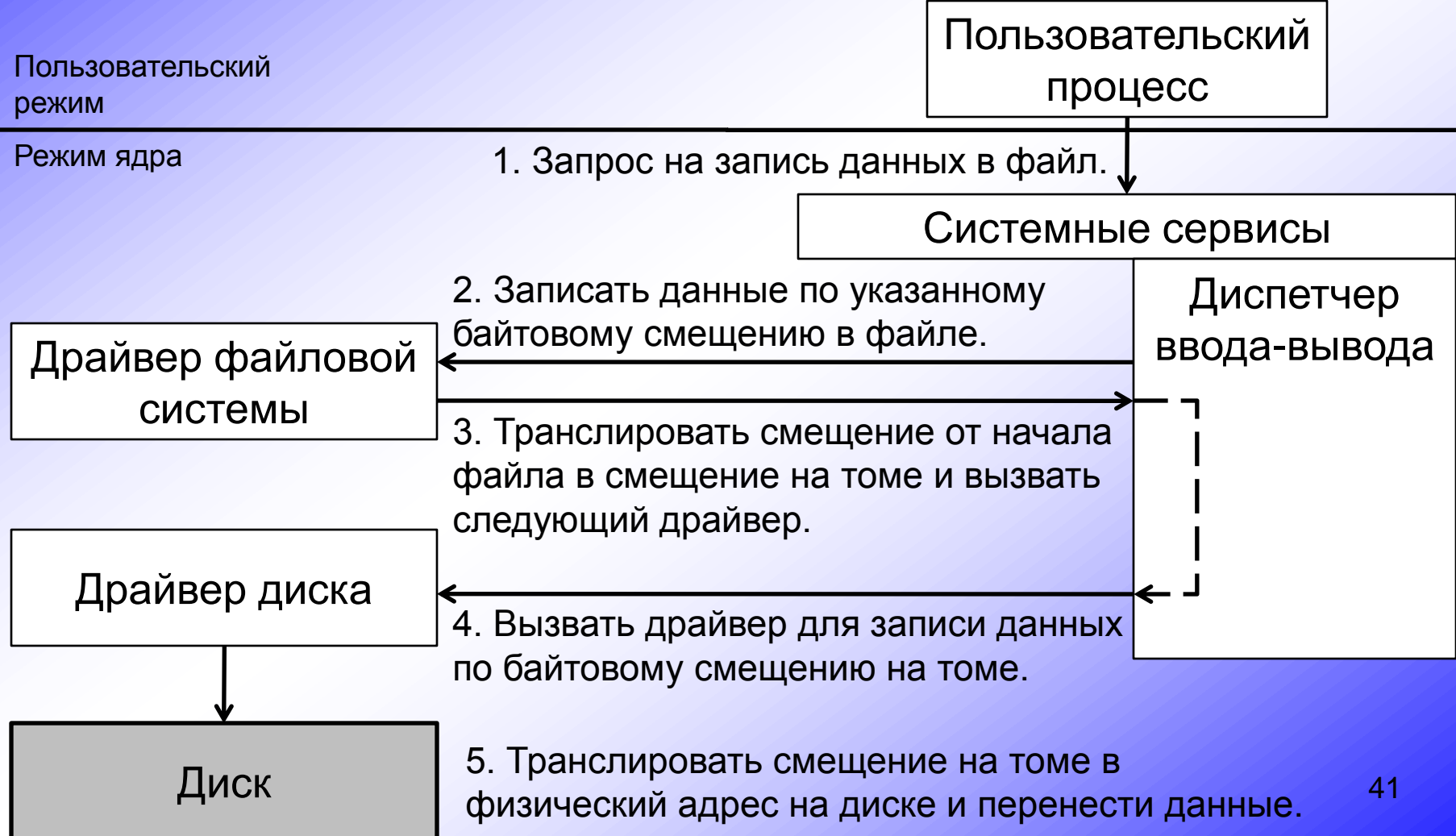
- Низкоуровневые драйверы – вызываются по прерываниям; непосредственно управляют аппаратурой.
- Высокоуровневые драйверы – взаимодействуют с устройством через низкоуровневые драйверы; работают с более абстрактными управляющими командами и представлением данных.
- Несколько разноуровневых драйверов, управляющих одним устройством, можно рассматривать как набор отдельных драйверов или как один многоуровневый драйвер.

# Преимущества многоуровневого представления драйверов

- Поддержка нескольких наборов функций (например, файловых систем) и выбор необходимой, вместо жёсткого набора функций монолитного драйвера.
- Простое включение новых драйверов в систему в связи с однотипной работой с различными драйверами одного типа устройств.



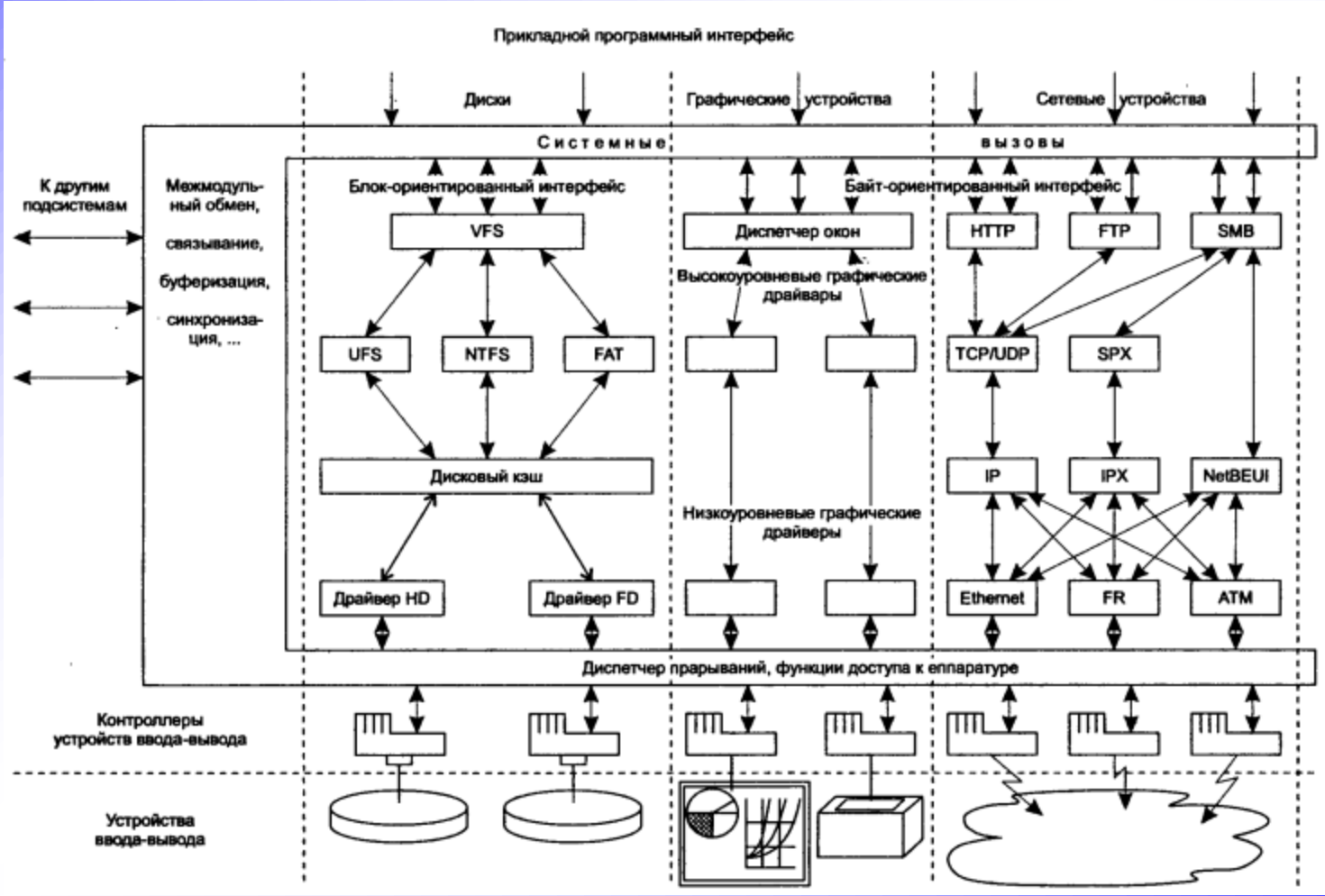
# Работа с многоуровневыми драйверами



# Работа с многоуровневыми драйверами



# Примеры многоуровневых драйверов



# Рассмотренные вопросы

- Структура подсистемы ввода-вывода.
- Функции диспетчера ввода-вывода, диспетчера Plug and Play, диспетчера электропитания.
- Структура и функции драйверов.
- Многоуровневая структура драйверов.

**Всем спасибо –  
все свободны,  
если нет вопросов**