Операционные системы

Курс лекций для гр. 4057/2

Лекция №9

Вопросы к лекции 8

1. В каких ситуациях и для каких страниц стратегия OPT достижима?
2. Решите задачи и докажите утверждение в Приложении (след. слайд).
3. Что будет, если период между сбросами битов в ноль очень мал ? Очень велик ?
4. Почему страница с R=0, M=1 приоритетнее для вытеснения, чем с R=1, M=0 ?
5. Пусть время доступа к активной странице = 200 нс, время подкачки отсутствующей страницы = 20 мс (TLB отсутствует). Насколько мала должна быть относительная частота страничных прерываний f, чтобы эффективное (т.е, среднее) время доступа к памяти не превышало 220 нс, т.е. было только на 10% больше ? (f оценивайте отношением числа обращений, приводящих к страничным прерываниям, к общему числу обращений к памяти.)
6. Почему для многопроцессорной системы NUR - не рационально?
7. Пусть несколько приложений выполняются под Windows некоторое время, а потом все они одновременно закрываются и перезапускаются снова. Почему в первое время после перезапуска общая производительность системы обычно ниже, чем до него?
8. Если запустить интерактивное мультимедиа-приложение (напр., игру), не закрыв другие приложения, то в первое время производительность его будет низка (игра будет «тормозить», особенно, если объем графических данных - большой). Объясните причину этого эффекта и предложите программный способ преодоления этого недостатка под Windows.

Содержание

Разнообразие устройств вв/вы

- Центральные устройства компьютера – ЦП + память (оперативная)
- У. вв/вы = периферийные (т.е. , внешние) устройства двух видов:
  - соединяющие компьютер с внешним миром – с преобразованием представления информации (мышь, микрофон, …) или без него (сеть, …)
  - внешняя память (магн. диски и сменные носители – флэш, CD-ROM, …)

Устройство	Скорость
Клавиатура	10 байт/с
Мышь	100 байт/с
Принтер	100 Кб/с
Шина USB	1,5 Мб/с
Монитор	60 Мб/с
Диск	100 Мб/с
Лок. сеть 1 Гбит	128 Мб/с

Символьные устройства передают данные по одному байту: клавиатура, мышь, звуковая карта, принтер, …

Блочные устройства передают блок байтов как единое целое: диск, CD-ROM, …

Остальные: графический дисплей, сеть, таймер, …

Односторонние устройства - только вв или вы: мышь, монитор, …

Двусторонние: диск, сеть, терминал, …

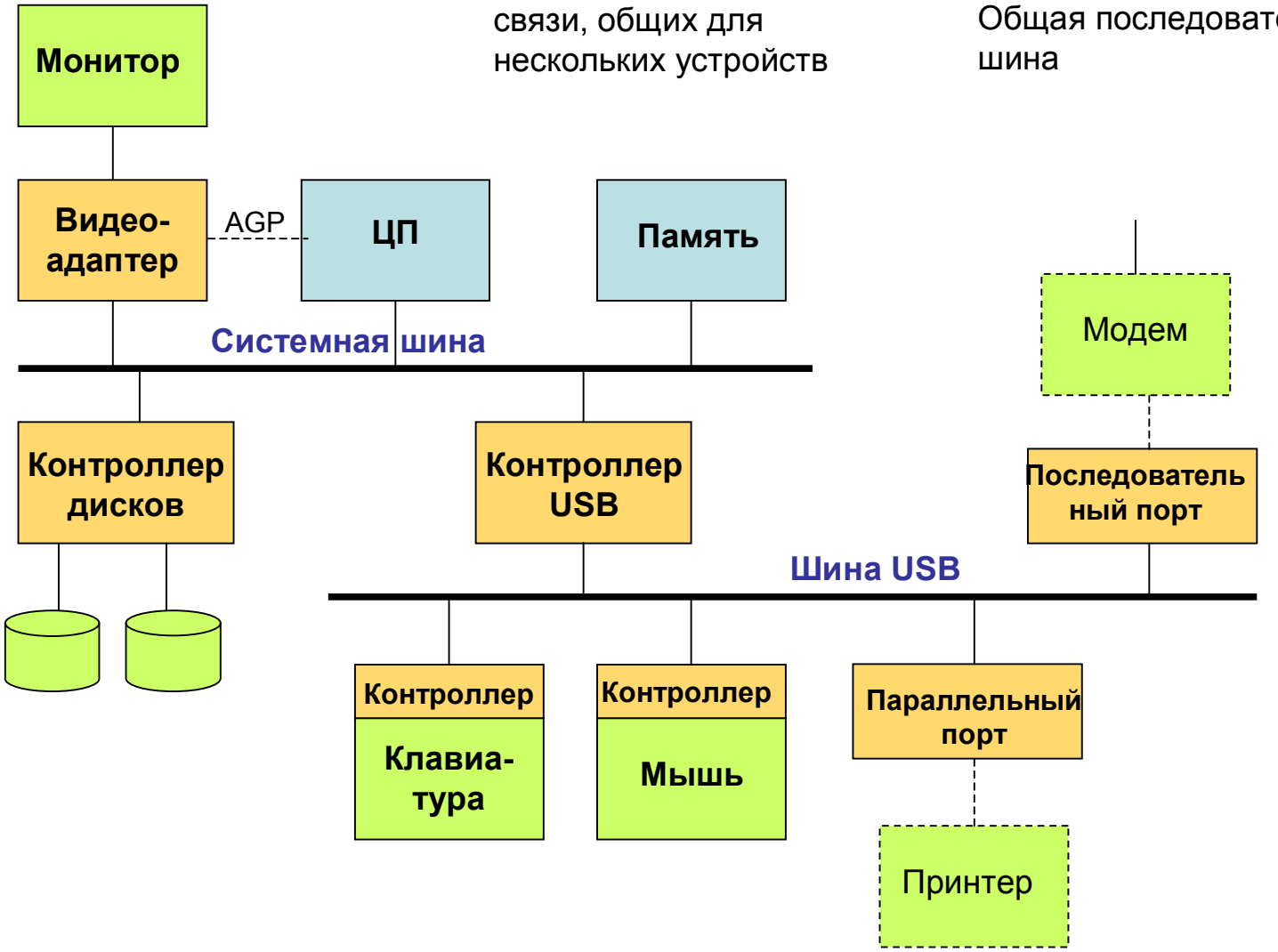
Разделяемые (shared) устройства: диск, сеть, …

Неразделяемые устройства: принтер, CD-ROM, …

Упрощенная структура PC

Шина (Bus) - набор линий связи, общих для нескольких устройств

USB – Universal Serial Bus – Общая последовательная шина



Основные понятия

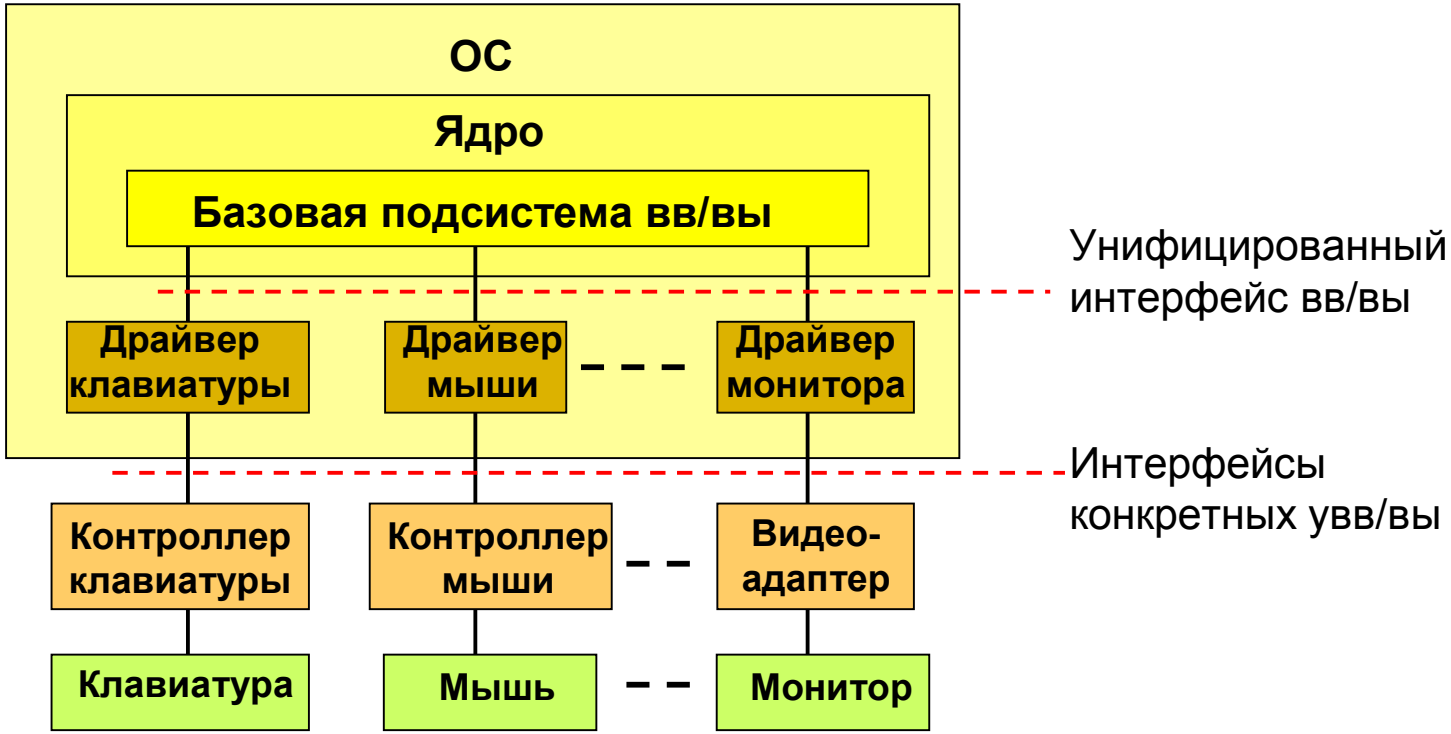
- Порт - точка соединения увв/вы с компьютером (физическая или логическая)
- Контроллер (адаптер) - устройство управления увв/вы, портом или шиной
  - м. б. одной или несколькими микросхемами или отдельной платой (card)
  - м.б. встроен в устройство или в компьютер
- Физический порт - это один или несколько регистров в контроллере для данных и сигналов управления - кодов, которые читает и пишет процессор

  Обычно порт состоит из 4-х регистров контроллера
  - **Состояние:** {свободно, занято, ошибка} - читается ЦП
  - **Управление:** команда устройства {читать, писать, завершить, …} - пишется ЦП
  - **Данные вв:** читаются в ЦП
  - **Данные вы:** пишутся из ЦП

  Два способа адресации регистров контроллера
  - Специальные команды вв/вы манипулируют адресами портов вв/вы
  - Регистры устройства отображаются на адреса памяти

«Слоеная» структура программного управления вв/вы

- Непосредственное управление операцией вв/вы выполняет программа-драйвер, специфичная для типа устройства
- Контроллер интерпретирует команды устройства, сгенерированные драйвером и посланные им в регистр управления



Уровни автономной работы контроллера

1. Программный вв/вы с опросом состояния: процессор выполняет всю работу
   Пример: мышь
2. Вы/вы с прерываниями: процессор не тратит времени на ожидание, пока не выполнится операция вв-вы
   Модем, символьный принтер
3. Контроллер имеет прямой доступ к памяти (DMA)
   Диск
4. Контроллер содержит процессор вв/вы, выполняющий программу, которая находится в основной памяти или в памяти контроллера
   Видеоадаптер с графическим ускорителем

Взаимодействие ЦП с увв-вы (1)

1. Программный вв-вы с опросом состояния (polling)

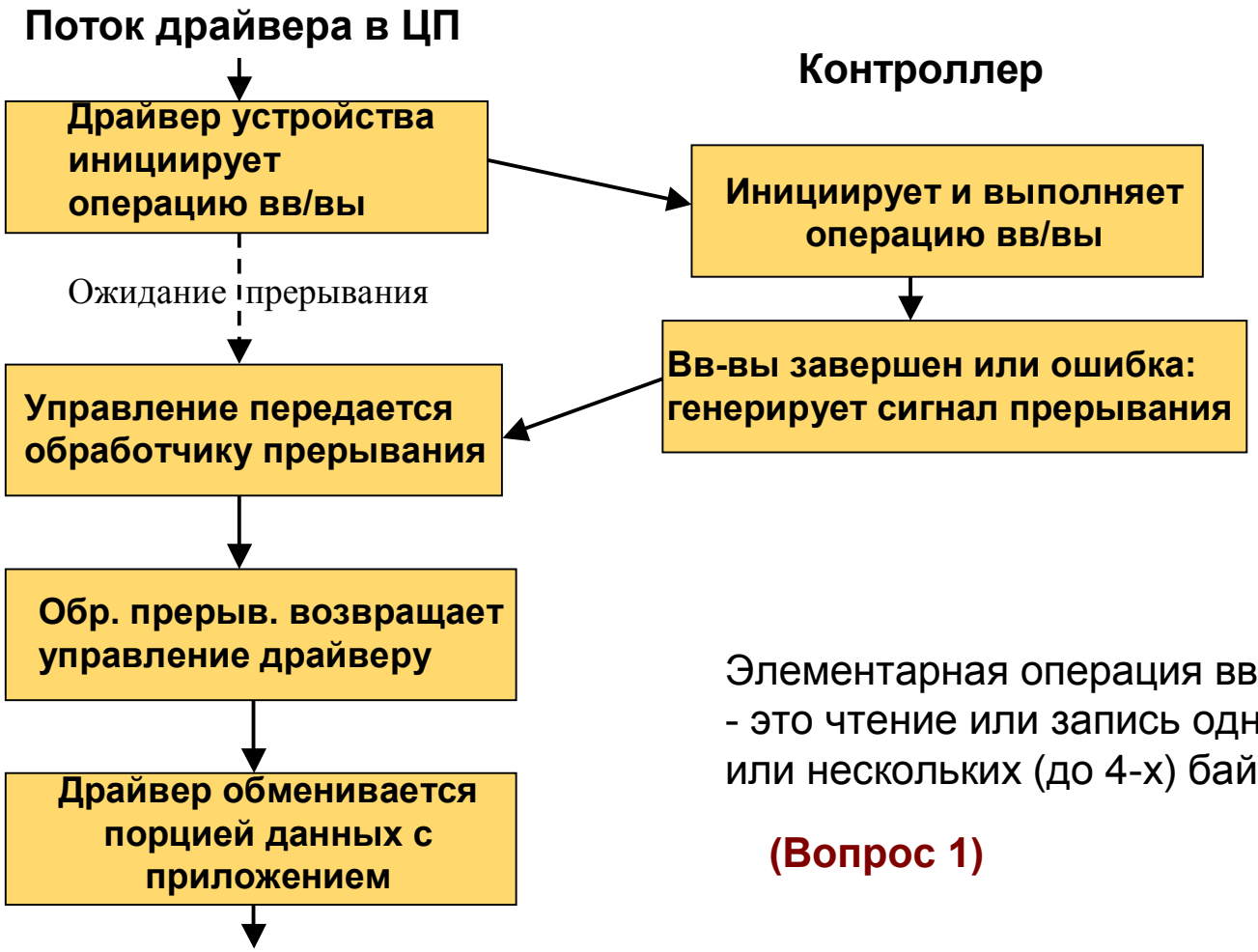
1. ЦП циклически читает регистр состояния, пока оно не станет "свободно"
2. ЦП пишет команду “Read" или “Write" в регистр управления
3. Контроллер устанавливает состояние "занято", читает регистр управления и затем
   - читает регистр данных вв и записывает в память
   - или пишет в регистр данных вы
4. Контроллер устанавливает состояние "свободно”

Пример - драйвер мыши с частотой 125 гц (т.е., каждые 8 мс):

- опрашивает контроллер мыши
- получает данные датчика перемещения мыши и так накапливает данные о ее текущих координатах
- если обнаруживает сигнал «Нажата кнопка», то генерирует событие для приложения

Взаимодействие ЦП с увв-вы (2)

2. Прерывания по завершении элементарной операции вв/вы

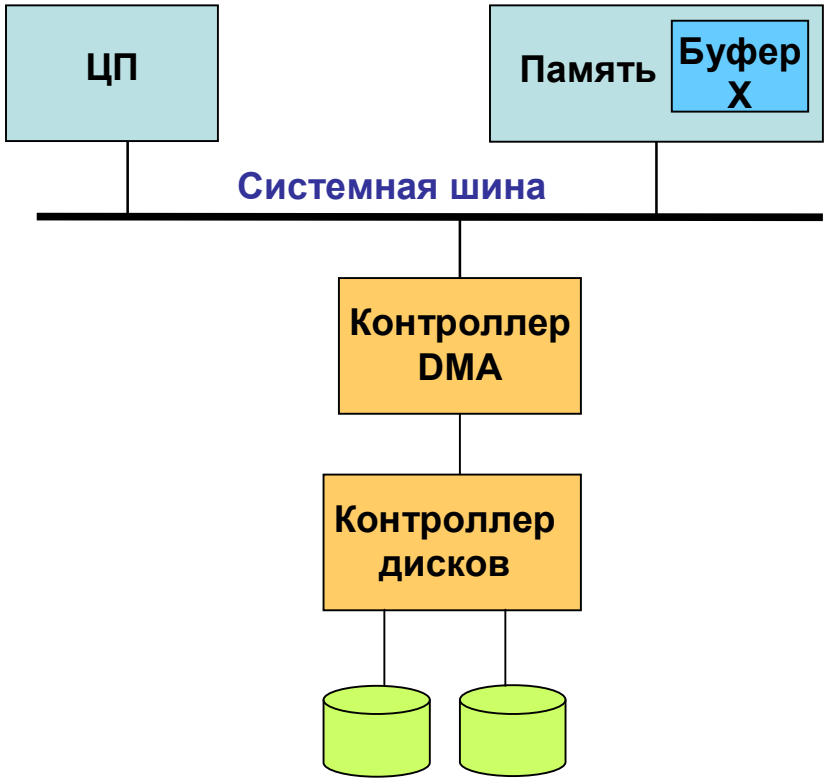


Элементарная операция вв/вы - это чтение или запись одного или нескольких (до 4-х) байтов.

(Вопрос 1)

Взаимодействие ЦП с увв-вы (3)

3. Прямой доступ к памяти (DMA)



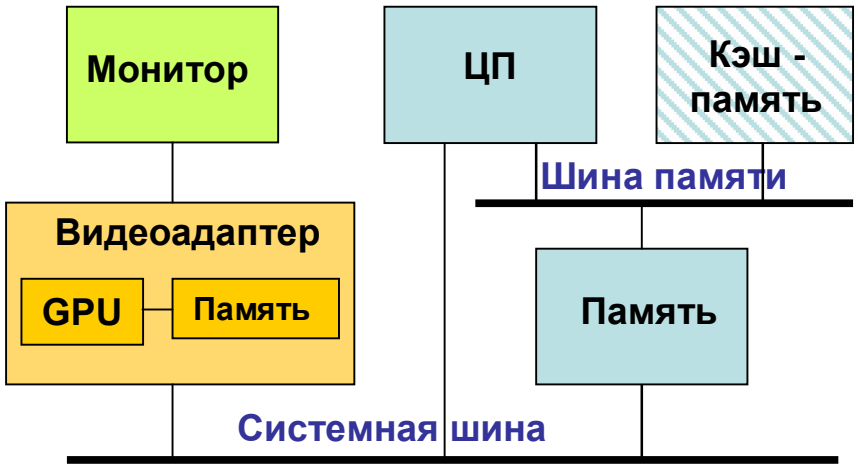
(Вопросы 2, 3)

Последовательность действий при вводе:

1. ЦП приказывает драйверу устройства передать C байтов с диска в буфер X
2. Драйвер приказывает контроллеру диска передать C байтов с диска в буфер X
3. Контроллер диска инициирует обмен через DMA
4. Контроллер диска пересылает байт за байтом в контроллер DMA
5. Контроллер DMA пересылает байт за байтом в буфер X, увеличивая каждый раз адрес и уменьшая C на единицу
6. Когда C=0, DMA генерирует прерывание ЦП, сообщая о завершении операции вв

Взаимодействие ЦП с увв-вы (4)

4. Процессор вв/вы: GPU – Graphics Processor Unit



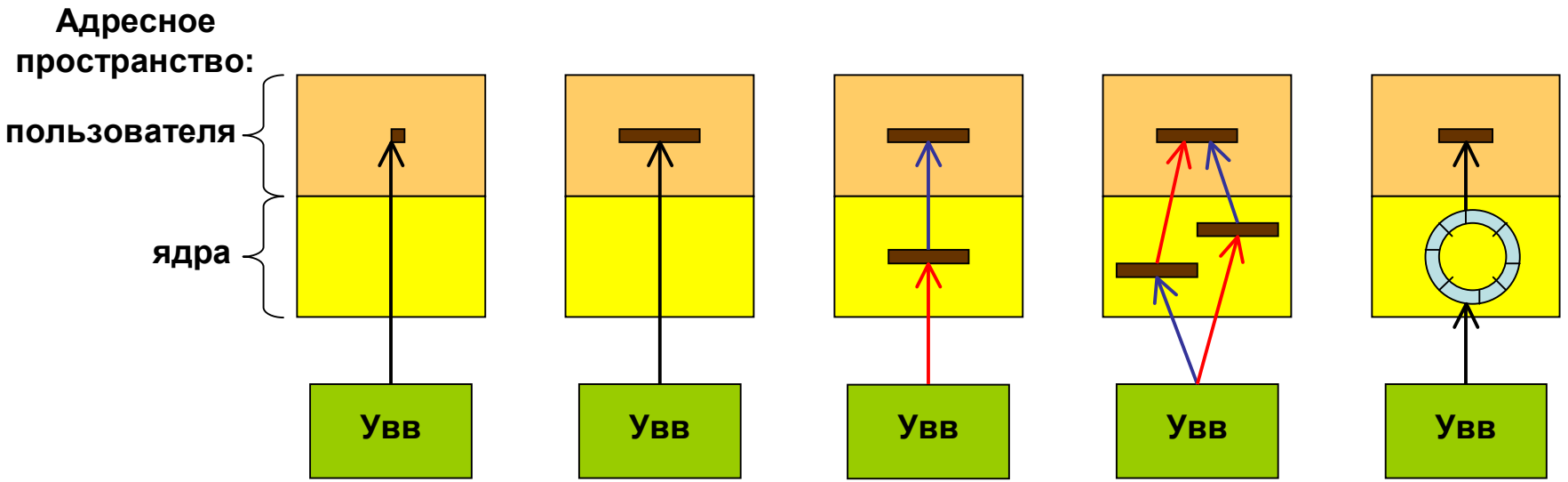
- ЦП формирует программу вывода графической информации (shader) и пересылает ее в память видеоадаптера
- GPU выполняет программу shader автономно
- HLSL (High Level Shader Language) - специальный язык программирования GPU

Функции подсистемы вв/вы

- Назначение и распределение устройств
  - Отображение логических устройств (каналов) на физические увв/вы – часть контекста сеанса пользователя или его *профиля* в многопользовательской ОС **(Вопрос 4)**
  - Выделение неразделяемого устройства процессу – задача взаимного исключения
- Буферизация

  Буфер – область памяти для промежуточного хранения вв/вы данных. Причины:
  - разница в размерах порций данных (единиц передачи) разных увв/вы (напр., модем → диск)
  - разница в скоростях работы ЦП и увв/вы
  - неравномерная скорость работы увв/вы
- Диспетчеризация
  - управление очередями к увв/вы

Буферизация потока байтов вв.



Нет буфера

Побайтовый ввод

Буфер пользователя

Нужна фиксация страницы в памяти (запрет выгрузки)

Буфер в ядре с копированием пользователю

Двойная буферизация: поочередный ввод и копирование

Циклический буфер: схема производитель-потребитель

Согласование размеров единиц передачи устройств: накопление большей порции из нескольких меньших

Сглаживание разницы в скоростях ЦП и у.вв/вы: накопление данных на стороне ЦП, как более быстрого

NB: копирование – одна операция

Сглаживание неравномерной скорости передачи, напр., при сетевом вв/вы

Интерфейс между подсистемой вв/вы и драйверами

Для символьных устройств: get, put

Для блочных устройств: read, write, seek

Общие функции:

- инициализация работы (open)
- завершение работы (close)
- управление (ioctl в Unix)

Для сетевых устройств интерфейс *сокета* (socket - розетка) (в Win2k – Winsock API) с вызовами:

- создать сокет
- подключить его к удаленному адресу (порту)
- ждать соединения
- послать/получить пакет

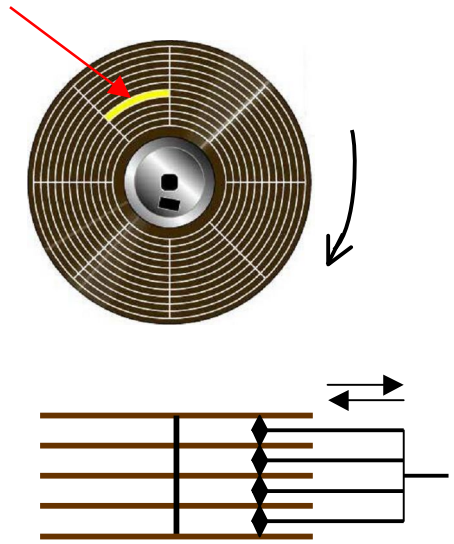
Режимы вв/вы

Виды системных вызовов вв/вы

- Синхронный, или блокирующий (blocking): процесс ждет, когда устройство выполнит обмен данными и вернет код статуса по завершении вв/вы, после чего он продолжает работу и использует полученные данные
- Асинхронный – более производительный: операция вв/вы совмещается с продолжением процесса. Для вв/вы порождается поток, сигнализирующий позже о завершении операции
  - в Unix порождение вручную – дочерний поток, в Win2k - автоматически
- Неблокирующий, или быстрый – для коротких операций обмена: процесс не переводится в состояние ожидания; системный вызов быстро возвращает результат
  - пример: периодический опрос клавиатуры приложением

Характеристики магнитных дисков

Сектор = блок = 512 байт



Скорость вращения: 7-15 тыс. об/мин

Скорость обмена: 50 – 200 Мбайт/с

Емкость: 1 Тб

Ср. время задержки доступа: 10 мс:

- поиск цилиндра: 7 мс
- ожидание сектора: 3 мс

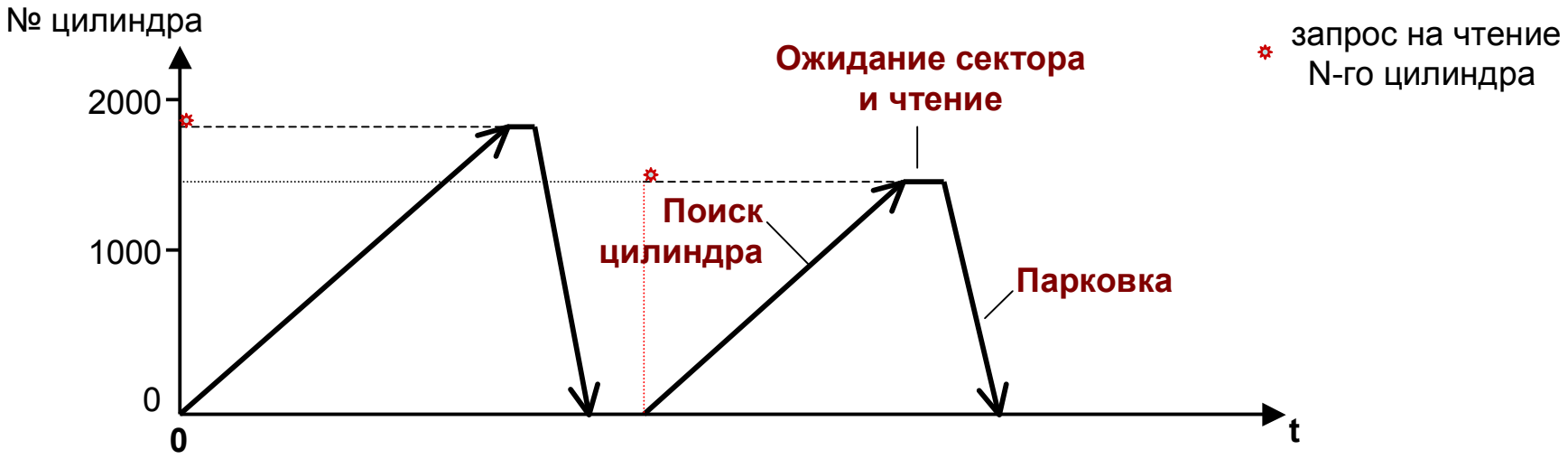
NB: это время достигло нижнего предела; 20 лет назад оно было = 20 мс

(Вопрос 5)

Время поиска соседнего цилиндра: 1 мс

Пример: диск ATA 8 Гб: 16383 цилиндра, 16 головок, 63 сектора

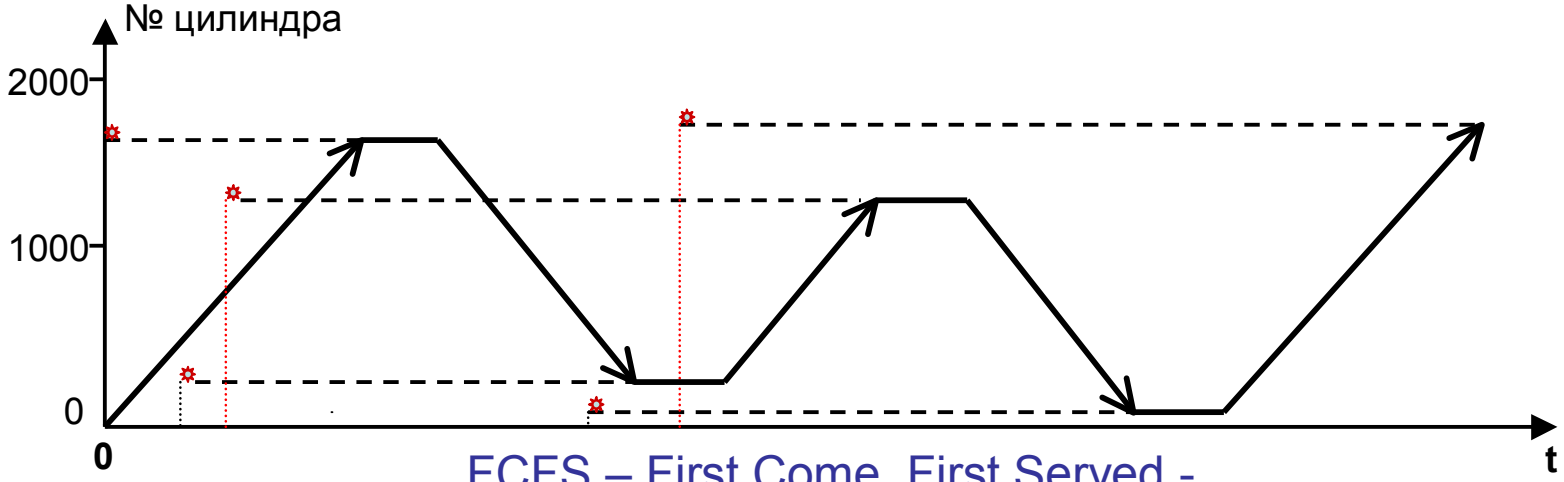
График движения головок при редких одиночных запросах чтения/записи



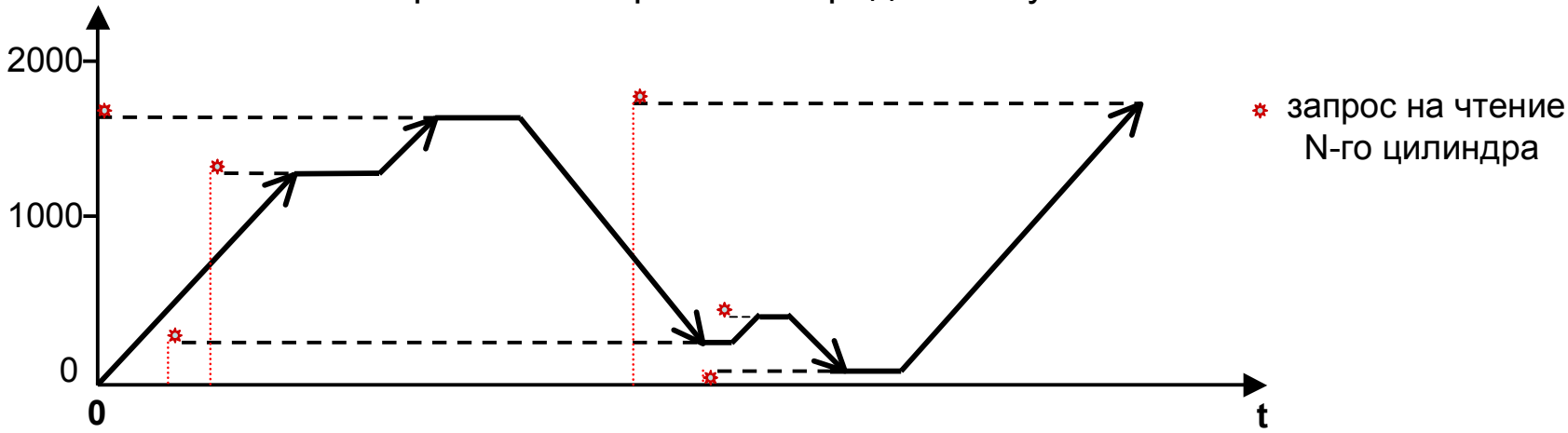
При парковке скорость движения головок больше, чем при движении с позиционированием на определенном цилиндре

На самом деле частота запросов на чтение/запись – до 10^3 -10^4 в сек → образуются очереди запросов

Управление очередью к диску



FCFS – First Come, First Served -
обработка запросов в порядке поступления



SSTF - Smaller Seek Time First –
первым обслуживается запрос к ближайшему цилиндру –
полная аналогия с дисциплиной SJF при диспетчеризации процессов

Характеристики дисциплин обслуживания очереди к диску

- FCFS: справедливый, однако большое среднее время обслуживания
- SSTF: минимальное суммарное (и значит, среднее) время ожидания; минус - опасность голодания запросов к периферийным цилиндрам
- SCAN («элеваторный» алгоритм): движение в одном направлении, пока все запросы на нем не будут обслужены, затем инверсия направления. Минус: большая дисперсия времени ожидания: его верхняя граница = 2T, где T – время обхода всех цилиндров в одном направлении (T ~= 20 мс)
- C-SCAN: верхняя граница времени ожидания ~= T
  - вдвое меньше дисперсия времени ожидания, чем у SCAN (Вопрос 6)

В современных контроллерах дисков эти алгоритмы реализуются аппаратно, с улучшениями:

- учет расположения нужных секторов на дорожках
- упреждающее чтение нескольких секторов дорожки в буфер («кэш») диска: 2 - 4 Мбайт и более

Подсистема вв/вы Win2k

- Нацелена на быструю разработку драйверов новых устройств
- Сильно развит графический вв/вы (прежде всего для GUI) – более 1000 вызовов API
- Диспетчер вв/вы создает пакет запросов вв/вы (IRP – I/O request packet – управляющую структуру данных запроса
- Драйвер получает IRP, выполняет операцию и возвращает IRP диспетчеру, чтобы тот либо завершил эту операцию (и уничтожил IRP), либо передал пакет другому драйверу для дальнейшей обработки
- Диспетчер вв/вы содержит общий для различных драйверов код (около ста функций), благодаря чему драйверы становятся проще и компактнее
- Диспетчер управляет очередями запросов вв/вы и таймаутами драйверов

Управление буферами вв/вы

Три способа управления буферами:

- Буферизованный вв/вы (buffered I/O): диспетчер выделяет в пуле невыгружаемых кадров (в т.н. неподкачиваемой памяти) буфер, в/из которого копируется содержимое области памяти приложения
- Прямой вв/вы (direct I/O): создавая IRP, диспетчер вв/вы фиксирует пользовательский буфер (делает его неподкачиваемым)
- Без управления (neither I/O) – используется драйверами файловой системы: диспетчер вв/вы не участвует в управлении буферами, это делает драйвер

- Стандартный размер буфера – 1 страница

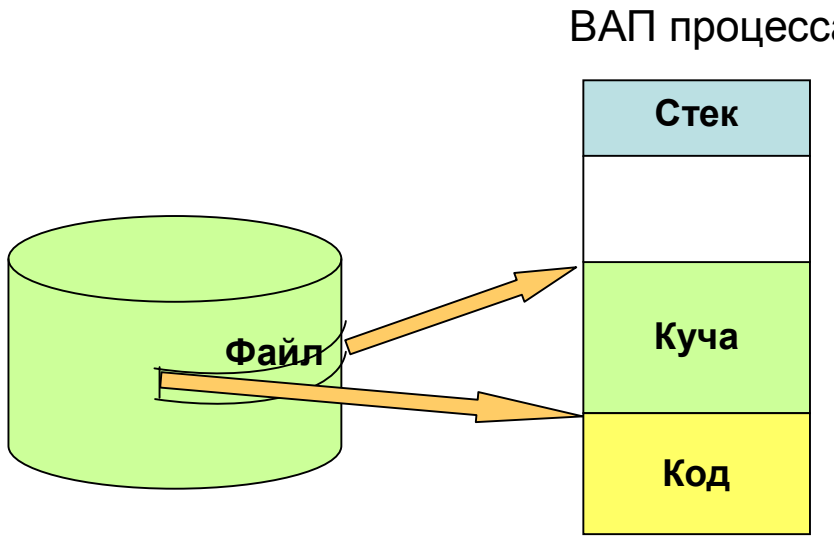
Когда вызывающие потоки передают порции данных менее 1 страницы (4 Кб), драйверы обычно используют буферизованный вв/вы, иначе – прямой

Режимы вв/вы

- Синхронный - применяется по умолчанию в большинстве приложений
- Асинхронный. Чтобы продолжать выполнение после выдачи запроса вв/вы, поток должен указать параметр FILE_FLAG_OVERLAPPED при вызове CreatFile. О завершении вв/вы приложению просигнализирует сигнализирующий объект (событие, порт завершения вв/вы или сам объект «файл»).
- Быстрый вв/вы (fast I/O): IRP не генерируются; подсистема вв/вы напрямую обращается к драйверу
- Вв/вы в проецируемые файлы (mapped file I/O)

Вв/вы в проецируемые файлы

- Дисковый файл интерпретируется как часть виртуальной памяти процесса
- Программа может обращаться к такому файлу как к большому массиву, что гораздо быстрее, чем дисковый вв/вы
  - при этом диспетчер памяти использует свой механизм подкачки отсутствующих страниц и записи измененных
- В самой ОС этот режим используется при загрузке и запуске исполняемых программ и при кэшировании файлового вв/вы



Подсистема вв/вы Unix/Linux

- Все увв/вы включены в файловую систему в виде специальных файлов в каталоге /dev; доступ - как к файлам: read, write, seek …

  Пример: `cp file_name /dev/lp`

- Блочный специальный файл состоит из последовательности нумерованных блоков; возможен доступ к блоку по его номеру
- Символьный специальный файл – поток символов
- С каждым специальным файлом связан драйвер соответствующего увв /вы
- В Unix драйверы статически компонуются вместе с ядром
- В Linux драйверы динамически подгружаются во время работы ОС
- Бедный графический вв/вы; GUI – в надстройке X/Window

Заключение

- Большое разнообразие увв/вы – сотни видов, тысячи марок
- Разработка драйверов нестандартных устройств – отдельная специализация; для облегчения работы – SDK для Windows
- Синхронный и асинхронный вв/вы аналогичен тому же при коммуникации
- Unix и Linux имеют менее богатые возможности графики на мониторе, чем Windows (OpenGL vs. DirectX)
- Тенденция: процессор графического вывода – GPU – становится настолько мощным и богатым функционально, что его используют для решения задач численных методов; это направление называется GP GPU – General Purpose Graphics Processor Unit

Вопросы к лекции 9

1. Каковы достоинства и недостатки протоколов 1 и 2? Какой из них для каких устройств применяют?
2. Несмотря на то, что процессор не занят обработкой прерываний на каждый пересылаемый элемент данных во время операции обмена блока через DMA, его производительность все же несколько снижается. Почему?
3. Доступ DMA к памяти обычно выполняется с более высоким приоритетом, чем доступ процессора. Почему?
4. Приведите примеры переназначения устройств вв/вы командами оболочки Unix.
5. Почему быстродействие дисковой памяти оказалось «замороженным», хотя быстродействие электронных компонентов и емкость дисков удваивались, по закону Мура, каждые два года? Что ограничивает его?
6. Предложите идею алгоритма C-SCAN