

Система управления ВВОДОМ-ВЫВОДОМ

Виды работ вычислительной системы

- Обработка информации
- Операции по осуществлению ввода-вывода информации

Общие сведения об архитектуре компьютера. Шины

- Шины данных
- Адресные шины
- Шины управления

Действия, которые должны быть выполнены для передачи информации из процессора в память

- *На адресной шине процессор должен выставить сигналы, соответствующие адресу ячейки памяти, в которую будет осуществляться передача информации.*
- *На шину данных процессор должен выставить сигналы, соответствующие информации, которая должна быть записана в память.*
- *После выполнения действий 1 и 2 на шину управления выставляются сигналы, соответствующие операции записи и работе с памятью, что приведет к занесению необходимой информации по нужному адресу.*

Передача данных в порт

- *На адресной шине процессор должен выставить сигналы, соответствующие адресу порта, в который будет осуществляться передача информации, в адресном пространстве ввода-вывода.*
- *На шину данных процессор должен выставить сигналы, соответствующие информации, которая должна быть передана в порт.*
- *После выполнения действий 1 и 2 на шину управления выставляются сигналы, соответствующие операции записи и работе с устройствами ввода-вывода (переключение адресных пространств!), что приведет к передаче необходимой информации в нужный порт.*

Современная архитектура компьютеров

- Устройства ввода-вывода подключаются к системе через порты.
- Могут существовать два адресных пространства: пространство памяти и пространство ввода-вывода.
- Порты, как правило, отображаются в адресное пространство ввода-вывода и иногда – непосредственно в адресное пространство памяти.
- Использование того или иного адресного пространства определяется типом команды, выполняемой процессором, или типом ее операндов.
- Физическим управлением устройством ввода-вывода, передачей информации через порт и выставлением некоторых сигналов на магистрали занимается контроллер устройства.

Структура контроллера устройства

- каждый контроллер имеет по крайней мере четыре внутренних регистра
 - регистр состояния
 - регистр управления
 - регистр входных данных
 - регистр выходных данных

Регистр состояния

- Регистр состояния содержит биты, значение которых определяется состоянием устройства ввода-вывода и которые доступны только для чтения вычислительной системой:
 - бит занятости
 - бит готовности данных
 - бит ошибки

Регистр управления

- Регистр управления получает данные, которые записываются вычислительной системой для инициализации устройства ввода-вывода или выполнения очередной команды, а также изменения режима работы устройства.

Регистр выходных данных

- Регистр выходных данных служит для помещения в него данных для чтения вычислительной системой, а регистр входных данных предназначен для помещения в него информации, которая должна быть выведена на устройство.

Опрос устройств и прерывания. Исключительные
ситуации и системные вызовы

Вывод информации, помещающейся в регистр входных данных

- Процессор в цикле читает информацию из порта регистра состояний и проверяет значение **бита занятости**. Если **бит занятости** установлен, то это означает, что устройство еще не завершило предыдущую операцию, и процессор уходит на новую итерацию цикла. Если **бит занятости** сброшен, то устройство готово к выполнению новой операции, и процессор переходит на следующий шаг.
- Процессор записывает код команды вывода в порт регистра управления.
- Процессор записывает данные в порт регистра входных данных.
- Процессор устанавливает **бит готовности команды**. В следующих шагах процессор не задействован.
- Когда контроллер замечает, что **бит готовности команды** установлен, он устанавливает **бит занятости**.
- Контроллер анализирует код команды в регистре управления и обнаруживает, что это команда вывода. Он берет данные из регистра входных данных и инициирует выполнение команды.
- После завершения операции контроллер обнуляет **бит готовности команды**.
- При успешном завершении операции контроллер обнуляет **бит ошибки** в регистре состояния, при неудачном завершении команды – устанавливает его.
- Контроллер сбрасывает **бит занятости**.

Внешние прерывания

- Внешнее прерывание обнаруживается процессором между выполнением команд (или между итерациями в случае выполнения цепочечных команд).
- Процессор при переходе на обработку прерывания сохраняет часть своего состояния перед выполнением **следующей** команды.
- Прерывания происходят **асинхронно** с работой процессора и **непредсказуемо**, программист никоим образом не может предугадать, в каком именно месте работы программы произойдет прерывание.

Исключительные ситуации

- Исключительные ситуации обнаруживаются процессором во время выполнения команд.
- Процессор при переходе на выполнение обработки исключительной ситуации сохраняет часть своего состояния перед выполнением текущей команды.
- Исключительные ситуации возникают **синхронно** с работой процессора, но **непредсказуемо** для программиста, если только тот специально не заставил процессор делить некоторое число на ноль.

Программные прерывания

- Программное прерывание происходит в результате выполнения специальной команды.
- Процессор при выполнении программного прерывания сохраняет свое состояние перед выполнением **следующей** команды.
- Программные прерывания, естественно, возникают синхронно с работой процессора и **абсолютно предсказуемы** программистом.

Прямой доступ к памяти (Direct Memory Access – DMA)

- Контроллер прямого доступа к памяти имеет несколько спаренных линий – каналов DMA, которые могут подключаться к различным устройствам
- При прямом доступе к памяти процессор и контроллер DMA по очереди управляют локальной магистралью
- При подключении к системе нового устройства, которое умеет использовать прямой доступ к памяти, обычно необходимо программно или аппаратно задать номер канала DMA

Логические принципы организации ВВОДА-ВЫВОДА

- Пользователю при подключении нового устройства достаточно отобразить порты устройства в соответствующее адресное пространство, определить, какой номер будет соответствовать прерыванию, генерируемому устройством, и, если нужно, закрепить за устройством некоторый канал DMA.

Структура системы ввода-вывода

- Скорость обмена информацией может варьироваться в диапазоне от нескольких байтов в секунду до нескольких гигабайтов в секунду.
- Одни устройства могут использоваться несколькими процессами параллельно, в то время как другие требуют монопольного захвата процессом.
- Устройства могут запоминать выведенную информацию для ее последующего ввода или не обладать этой функцией. Устройства, запоминающие информацию, в свою очередь, могут дифференцироваться по формам доступа к сохраненной информации: обеспечивать к ней последовательный доступ в жестко заданном порядке или уметь находить и передавать только необходимую порцию данных.
- Часть устройств умеет передавать данные только по одному байту последовательно а часть устройств умеет передавать блок байтов как единое целое.
- Существуют устройства, предназначенные только для ввода информации, устройства, предназначенные только для вывода информации, и устройства, которые могут выполнять и ввод, и вывод.

Структура системы ввода-вывода



Систематизация внешних устройств и интерфейс между базовой подсистемой ввода-вывода и драйверами

- символьные (клавиатура, модем, терминал и т. п.);
- блочные (магнитные и оптические диски и ленты, и т. д.);
- сетевые (сетевые карты);
- все остальные (таймеры, графические дисплеи, телевизионные устройства, видеокамеры и т. п.);

Функции

- read, write, seek (для блочных устройств)
- get, put (для символьных устройств)
- open - функция инициализации или повторной инициализации работы драйвера и устройства
- close - функция временного завершения работы с устройством
- poll - функция опроса состояния
- halt - функция остановки драйвера, которая вызывается при остановке операционной системы или выгрузке драйвера из памяти

Функции базовой подсистемы ВВОДА-ВЫВОДА

- базовая подсистема ввода-вывода служит посредником между процессами вычислительной системы и набором драйверов.
- поддержка блокирующихся, неблокирующихся и асинхронных СИСТЕМНЫХ ВЫЗОВОВ
- буферизация и кэширование входных и выходных данных
- spooling и монопольного захвата внешних устройств
- обработка ошибок и прерываний, возникающих при операциях ввода-вывода,
- планирование последовательности запросов на выполнение этих операций

Блокирующиеся, неблокирующиеся и асинхронные системные вызовы

- блокирующиеся системные вызовы
- неблокирующиеся системные вызовы
- асинхронные системные вызовы

Буферизация и кэширование

- Первая причина буферизации – это разные скорости приема и передачи информации, которыми обладают участники обмена.
- Вторая причина буферизации – это разные объемы данных, которые могут быть приняты или получены участниками обмена одновременно.
- Третья причина буферизации связана с необходимостью копирования информации из приложений, осуществляющих ввод-вывод, в буфер ядра операционной системы и обратно.

Spooling и захват устройств

- Буфер, содержащий входные или выходные данные для устройства, на котором следует избегать чередования его использования различными процессами.
- В некоторых операционных системах вместо использования spooling применяется механизм монопольного захвата устройств процессами.

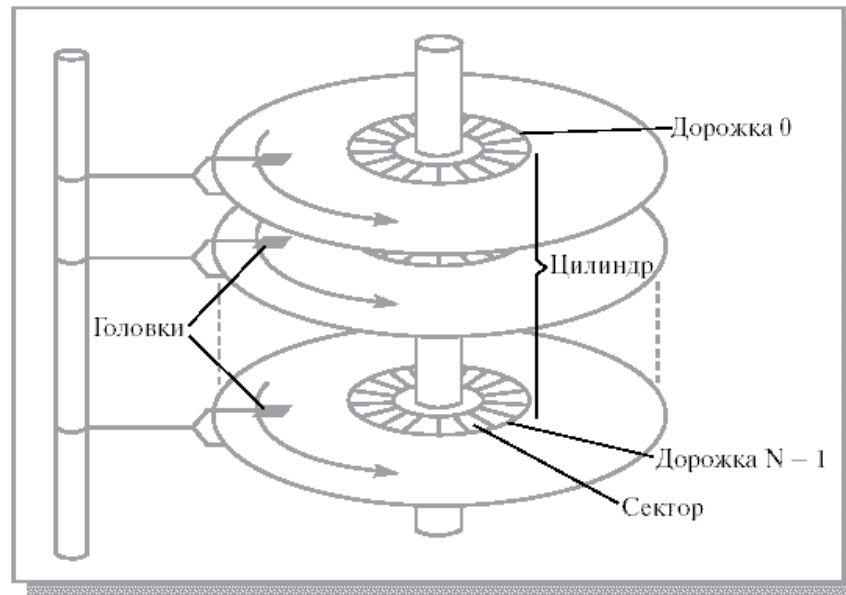
Обработка прерываний и ошибок

- определении того, какое именно устройство выдало прерывание
- зная устройство, можно выявить процесс, который инициировал выполнение соответствующей операции
- определить успешность завершения операции, проверив значение **бита ошибки** в регистре состояния устройства
- компенсацию возникшей ошибки

Планирование запросов

Строение жесткого диска и параметры планирования

- Алгоритм First Come First Served (FCFS)
- Алгоритм Short Seek Time First (SSTF)



Алгоритмы сканирования

- SCAN
- C-SCAN
- LOOK
- C-LOOK

Вопросы?