


УПРАВЛЕНИЕ ВВОДОМ-ВЫВОДОМ

Физическая организация устройств ввода-вывода


Внешние устройства, выполняющие *операции* ввода-вывода, можно разделить на три группы:

- устройства, работающие с пользователем. Используются для связи пользователя с компьютером (принтеры, дисплеи, клавиатура, манипуляторы (мышь, джойстики) и т.п.);
- устройства, работающие с компьютером. Используются для связи с электронным оборудованием (дисковые устройства, контроллеры);
- коммуникации. Используются для связи с удаленными устройствами. К ним относятся модемы и адаптеры цифровых линий.




По другому признаку устройства ввода-вывода делятся на два типа: **блок-ориентированные** (блочные) устройства и **байт-ориентированные** (символьные) устройства.

Блок-ориентированные устройства хранят информацию в блоках фиксированного размера, каждый из которых имеет свой собственный адрес и каждый блок может быть прочитан независимо от остальных блоков. Самое распространенное блок-ориентированное устройство - диск.




Байт-ориентированные устройства не адресуемы и не позволяют производить операцию поиска, они генерируют или потребляют последовательность байтов, т.е. принимают или передают *поток* символов без какой-либо блочной структуры. Примерами являются терминалы, принтеры, сетевые адаптеры, мыши.

Некоторые из устройств не попадают ни в одну из этих категорий, например, часы, мониторы и др. И все же модель блочных и символьных устройств является настолько общей, что может использоваться в качестве основы для достижения независимости от устройств некоторого программного обеспечения операционных систем, имеющего дело с вводом-выводом.




Внешнее устройство обычно состоит из **механического** и **электронного** компонента. Электронный компонент называется контроллером устройства или адаптером. Механический компонент представляет собственно устройство.

В ПК контроллер принимает форму платы, вставляемой в слот расширения. Плата имеет разъем, к которому подключается кабель, ведущий к самому устройству. Многие контроллеры способны управлять несколькими устройствами. Интерфейс между контроллером и устройством является официальным стандартом (ANSI, IEEE или ISO), и различные компании могут выпускать отдельно контроллеры и устройства, удовлетворяющие данному интерфейсу.




Операционная система обычно имеет дело не с устройством, а с контроллером. Каждый контроллер имеет несколько регистров, которые используются для взаимодействия с центральным процессором. В некоторых компьютерах эти регистры являются частью физического адресного пространства. В других компьютерах адреса регистров ввода-вывода, называемых часто портами, образуют собственное адресное пространство за счет введения специальных операций ввода-вывода.

Каждый *контроллер* взаимодействует с драйвером системным программным модулем, предназначенным для управления данным устройством. Для работы с драйвером *контроллер* имеет несколько регистров, кроме того, он может иметь *буфер* данных, из которого *операционная система* может читать данные, а также записывать данные в него. Каждому управляющему регистру назначается *номер порта* ввода-вывода.



ОС выполняет ввод-вывод, записывая команды в регистры контроллера. Например, контроллер гибкого диска IBM PC принимает 15 команд, таких как READ, WRITE, FORMAT и т.д. Когда команда принята, процессор оставляет контроллер и занимается другой работой. При завершении команды контроллер организует прерывание для того, чтобы передать управление процессором операционной системе, которая должна проверить результаты операции. Процессор получает результаты и статус устройства, читая информацию из регистров контроллера.



Назначение и задачи подсистемы ввода-вывода
Обмен данными между пользователями, приложениями и периферийными устройствами компьютера выполняет специальная подсистема ОС – подсистема ввода-вывода.

Основными компонентами подсистемы ввода-вывода являются *драйверы, управляющие* внешними устройствами, и *файловая система*. В работе подсистемы ввода-вывода активно участвует *диспетчер прерываний*, который иногда считают частью подсистемы ввода-вывода.


Функции подсистемы ввода-вывода:

- организация параллельной работы устройств ввода-вывода и процессора;
- согласование скоростей обмена и кэширование данных;
- разделение устройств и данных между процессами (выполняющимися программами);
- обеспечение удобного логического интерфейса между устройствами и остальной частью системы;
- поддержка широкого спектра драйверов;
- динамическая загрузка и выгрузка драйверов без дополнительных действий с операционной системой;
- поддержка нескольких различных файловых систем;
- поддержка синхронных и асинхронных операций ввода-вывода.


Организация программного обеспечения ввода-вывода

Основная идея организации программного обеспечения ввода-вывода состоит в разбиении его на несколько уровней, причем нижние уровни обеспечивают экранирование особенностей аппаратуры от верхних, а те, в свою очередь, обеспечивают удобный интерфейс для пользователей.

Ключевым принципом является независимость от устройств. Вид программы не должен зависеть от того, читает ли она данные с гибкого диска или с жесткого диска.




Другим важным вопросом для программного обеспечения ввода-вывода является **обработка ошибок**. Вообще говоря, ошибки следует обрабатывать как можно ближе к аппаратуре. Если контроллер обнаруживает ошибку чтения, то он должен попытаться ее скорректировать. Если же это ему не удастся, то исправлением ошибок должен заняться драйвер устройства. И только если нижний уровень не может справиться с ошибкой, он сообщает об ошибке верхнему уровню.




Еще один ключевой вопрос - это использование блокирующих (синхронных) и неблокирующих (асинхронных) передач. Большинство операций физического ввода-вывода выполняется асинхронно - процессор начинает передачу и переходит на другую работу, пока не наступает прерывание. ОС выполняет операции ввода-вывода асинхронно, но представляет их для пользовательских программ в синхронной форме.

Последняя проблема состоит в том, что одни устройства являются разделяемыми, а другие - выделенными. Диски - это разделяемые устройства, так как одновременный доступ нескольких пользователей к диску не представляет собой проблему.




Принтеры - это выделенные устройства, потому что нельзя смешивать строчки, печатаемые различными пользователями. Наличие выделенных устройств создает для операционной системы некоторые проблемы.

В случае совместного использования ОС должна оптимизировать последовательность операций ввода-вывода для различных процессов в целях повышения общей производительности. При разделении устройства между процессами может возникнуть необходимость в разграничении данных процессов друг от друга. Обычно такая потребность появляется при совместном использовании последовательных устройств, которые, в отличие от устройств прямого доступа, не адресуются.



Типичный представитель такого устройства – принтер. Для таких устройств организуется *очередь заданий* на *вывод*, при этом каждое задание представляет собой порцию данных, которую нельзя разрывать, например, документ для печати.

Для хранения очереди заданий используется спул-*файл*, который согласует скорость работы принтера и оперативной памяти и позволяет организовать *разбиение* данных на логические порции. Процессы могут одновременно выполнять *вывод* на принтер, помещая данные в свой раздел спул-файла.




Программное обеспечение ввода-вывода состоит из четырех слоев (Рисунок 1):


- обработка прерываний,
- драйверы устройств,
- независимый от устройств слой операционной системы,
- пользовательский слой программного обеспечения.



Рисунок 1 - Многоуровневая организация подсистемы ввода-вывода




Обработка прерываний. Прерывания должны быть скрыты как можно глубже в недрах операционной системы, чтобы как можно меньшая часть ОС имела с ними дело. Наилучший способ состоит в разрешении процессу, инициировавшему операцию ввода-вывода, заблокировать себя до завершения операции и наступления прерывания. При наступлении прерывания процедура обработки прерывания выполняет разблокирование процесса, инициировавшего операцию ввода-вывода или посылая процессу сообщение.



Прерываниями обычно называют сигналы, которые генерируются программным или аппаратным обеспечением, когда конкретное событие или процесс требует немедленного внимания.

Каждое устройство в системе получает свой номер прерывания IRQ (Interrupt request) .

Суть прерываний в следующем. Процессор обрабатывает какой-то код в обычном режиме. В это время с оборудованием происходит какое-то событие, например вы передвинули мышку. Процессор получает сигнал о том что нужно прервать свою текущую работу и обработать запрос от оборудования.



Этот запрос вызывает функцию, которая называется обработчиком прерывания (ISR). И процессор начинает обрабатывать этот ISR. И только после обработки ISR, он возвращается к обработке той задачи, на которой был прерван.

Прерывания могут быть:


Асинхронные – это прерывания исходящие от внешних устройств. Они могут произойти в любой момент.

Синхронные – это события в самом процессоре, которые могут произойти при различных условиях. Например, при делении на ноль, при переполнение стека, при обращении к недопустимым адресам памяти.

Прерывания позволяют операционной системе использовать процессор максимально эффективно, совмещая основную работу с операциями ввода / вывода.

Разное оборудование имеет разные приоритеты прерываний, которые называются IRQL (Interrupt Request Level). Суть в том, чтобы более приоритетные прерывания могли прервать менее приоритетные, но не наоборот.


Для разных архитектур используются разные IRQL. Архитектура x64 для оборудования выделяет уровни от 3 до 12. Выше находятся, например IPI прерывания (межпроцессорные), когда один процессор может прервать работу другого. А ниже программные прерывания. А когда процессор не обслуживает прерывания, то он работает на нулевом уровне.



Драйверы устройств. Весь зависимый от устройства код помещается в драйвер устройства. Каждый драйвер управляет устройствами одного типа или, может быть, одного класса.


В операционной системе только драйвер устройства знает о конкретных особенностях какого-либо устройства. Например, только драйвер диска имеет дело с дорожками, секторами, цилиндрами, временем установления головки и другими факторами, обеспечивающими правильную работу диска.

Драйвер устройства принимает запрос от устройств программного слоя и решает, как его выполнить.



Если драйвер был свободен во время поступления запроса, то он начинает выполнять запрос немедленно. Если же он был занят обслуживанием другого запроса, то вновь поступивший запрос присоединяется к очереди уже имеющихся запросов, и он будет выполнен, когда наступит его очередь.

Независимый от устройств слой операционной системы. Большая часть программного обеспечения ввода-вывода является независимой от устройств.




Типичными функциями для независимого от устройств слоя являются:

- обеспечение общего интерфейса к драйверам устройств,
- именованное устройств,
- защита устройств,
- обеспечение независимого размера блока,
- буферизация,
- распределение памяти на блок-ориентированных устройствах,
- распределение и освобождение выделенных устройств,
- уведомление об ошибках.

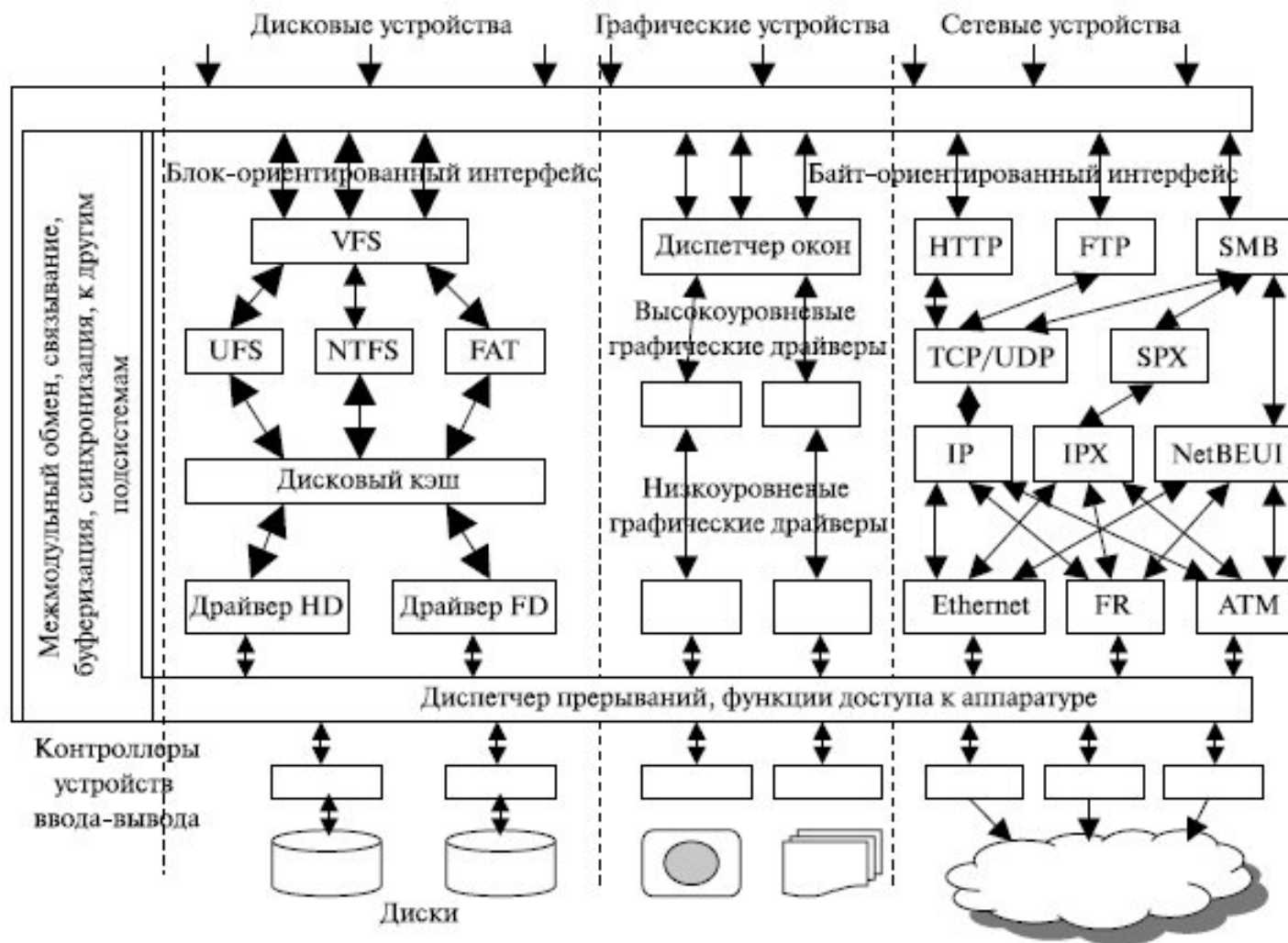
Пользовательский слой программного обеспечения. Хотя большая часть программного обеспечения ввода-вывода находится внутри ОС, некоторая его часть содержится в библиотеках, связываемых с пользовательскими программами. Системные вызовы, включающие вызовы ввода-вывода, обычно делаются библиотечными процедурами. Стандартная библиотека ввода-вывода содержит большое число процедур, которые выполняют ввод-вывод и работают как часть пользовательской программы.


Другой категорией программного обеспечения ввода-вывода является **подсистема спулинга (spooling)**.

Спулинг - это способ работы с выделенными устройствами в мультипрограммной системе. Например, типичное устройство, требующее спулинга - принтер. Пользовательский процесс может монополизировать принтер на произвольное время. Поэтому создается специальный процесс - монитор, который получает исключительные права на использование этого устройства. Также создается специальный каталог, называемый каталогом спулинга. Для того, чтобы напечатать файл, пользовательский процесс помещает выводимую информацию в этот файл и помещает его в каталог спулинга. Процесс-монитор по очереди распечатывает все файлы, содержащиеся в каталоге спулинга.



Программное обеспечение подсистемы ввода-вывода делится не только на горизонтальные слои, но и на вертикальные. В данном случае в качестве примера приведены три вертикальные подсистемы управления дисками, графическими устройствами и сетевыми адаптерами. Естественно, таких подсистем может быть больше. Например, сюда можно добавить подсистему управления текстовыми терминалами или подсистему управления специализированными устройствами, такими как аналого-цифровые и цифро-аналоговые преобразователи.





В каждой вертикальной подсистеме – несколько слоев модулей. Нижний слой образует аппаратные драйверы, *управляющие* аппаратурой внешних устройств. Как правило, этот слой программного обеспечения не имеет дела с вопросами логической организации данных, например, с файлами или сложными графическими объектами. Функции вышележащих слоев в значительной степени зависят от типа вертикальной подсистемы. В подсистеме ввода-вывода имеются модули универсального назначения. Эти модули организуют работу всех остальных компонентов подсистемы ввода-вывода, взаимодействие с пользовательскими процессами и другими подсистемами ОС. Эти организующие функции распределяются по всем уровням, образуя оболочку, называемую менеджером ввода-вывода.

Вопросы к лекции

- 1 Охарактеризуйте физическую организацию устройств ввода-вывода.
- 2 Охарактеризуйте организацию программного обеспечения ввода-вывода.
- 3 На какие слои делится программное обеспечение ввода-вывода?
- 4 Охарактеризуйте каждый слой программного обеспечения ввода-вывода.