Логические принципы организации ввода-вывода.

Структура системы вв, систематизация устройств и их интерфейс, функции базовой подсистемы вв.

Рассмотренные физические механизмы взаимодействия устройств ввода-вывода с вычислительной системой позволяют понять, почему разнообразные внешние устройства легко могут быть добавлены в существующие компьютеры. Все, что необходимо сделать пользователю при подключении нового устройства - это отобразить порты устройства в соответствующее адресное пространство, определить какой номер будет соответствовать прерыванию, генерируемому устройством, и, если нужно, закрепить за устройством некоторый канал DMA. Для нормального функционирования hardware этого будет достаточно.

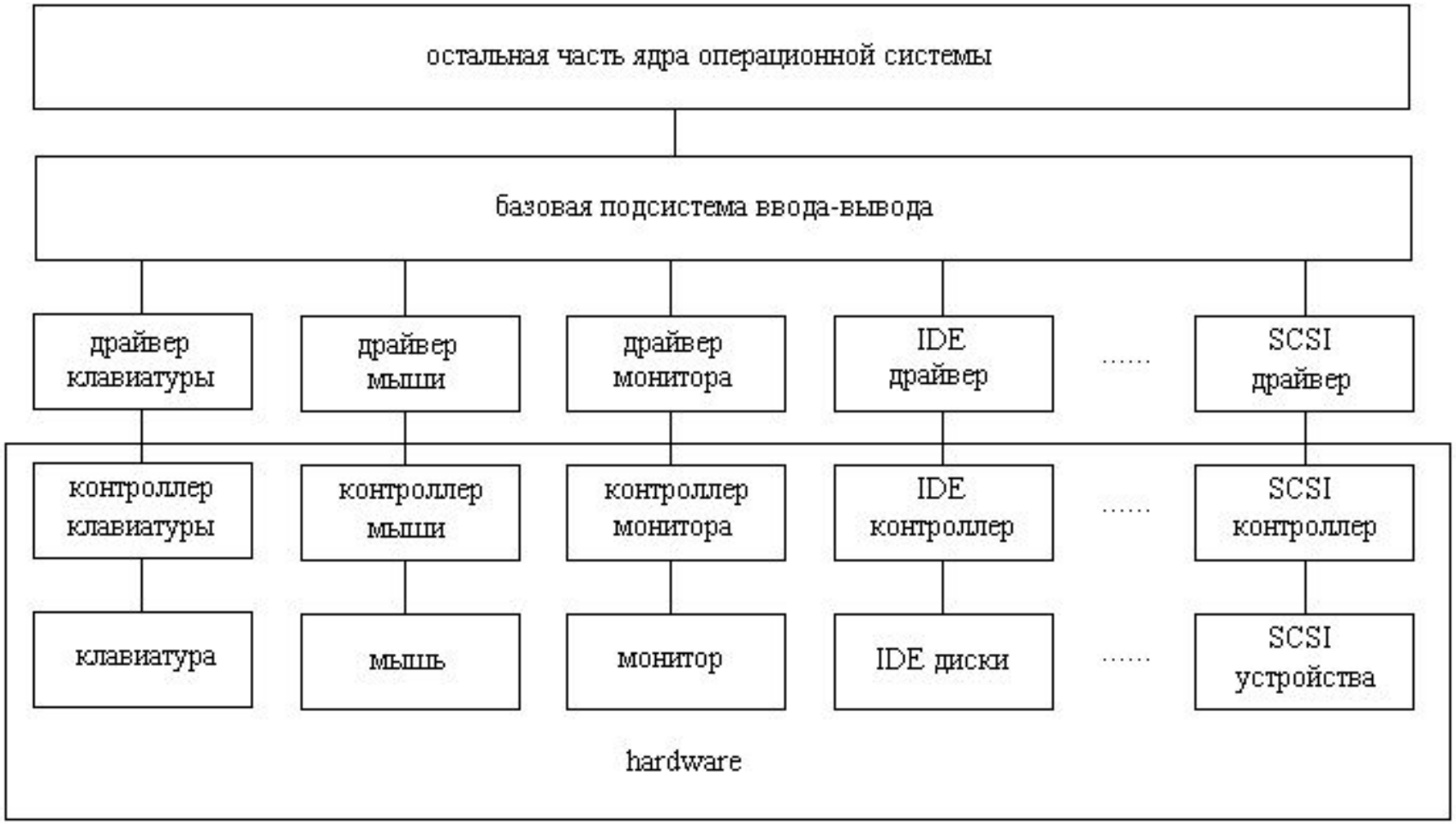
Однако мы с вами до сих пор ничего не говорили о том, как должна быть построена подсистема управления вводом-выводом в операционной системе для легкого и безболезненного добавления новых устройств, и какие функции вообще обычно на нее возлагаются.

Структура системы ввода-вывода.

Существует очень много разновидностей устройств, которые отличаются по выполняемым функциям и своим характеристикам, и кажется, что принципиально невозможно создать систему, которая без больших постоянных переделок позволяла бы охватывать все многообразие видов. Вот перечень лишь несколько направлений (далеко не полный), по которым различаются устройства:

1. Скорость обмена информацией может варьироваться в диапазоне от нескольких байт в секунду (клавиатура) до нескольких гигабайт в секунду (сетевые карты).
2. Некоторые устройства могут быть использованы параллельно несколькими процессами (являются разделяемыми), в то время как другие требуют монопольного захвата процессом.
3. 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.
4. Часть устройств умеет передавать данные только по одному байту последовательно (*символьные устройства*), а часть устройств умеет передавать блок байт как единое целое (*блочные устройства*).
5. Существуют устройства, предназначенные только для ввода информации, устройства, предназначенные только для вывода информации, и устройства, которые могут совершать и ввод, и вывод.

- Устройства ввода-вывода можно разделить на относительно небольшое число типов, отличающихся по набору операций, которые могут быть ими выполнены, считая все остальные различия несущественными.
- Затем можно будет специфицировать интерфейсы между ядром операционной системы, осуществляющим некоторую общую политику ввода-вывода, и программными частями, непосредственно управляющими устройствами, для каждого из таких типов. Более того, разработчики операционных систем получают возможность освободиться от написания и тестирования этих специфических программных частей, получивших название *драйверов*, передав эту деятельность производителям самих внешних устройств. Фактически мы приходим к использованию принципа уровневого или слоеного построения системы управления вводом-выводом для операционной системы



Систематизация внешних устройств и их интерфейс

Система типов устройств является далеко не полной и не строго выдержанной. Устройства обычно принято разделять по преобладающему типу интерфейса на следующие типы:

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

Такое деление является весьма условным. В некоторых операционных системах сетевые устройства могут не выделяться в отдельную группу, в некоторых – отдельные группы составляют звуковые устройства и видеоустройства и т.д. Некоторые группы в свою очередь могут разбиваться на подгруппы: подгруппа жестких дисков, подгруппа мышек, подгруппа принтеров.

- Мы рассмотрим только две группы устройств: символьные и блочные. Здесь символьные устройства – это устройства, которые умеют передавать данные только последовательно байт за байтом, а блочные устройства – это устройства, которые могут передавать блок байт как единое целое.
- К символьным устройствам обычно относятся устройства ввода информации, которые спонтанно, т.е. во времена непредсказуемые вычислительной системой, генерируют входные данные: клавиатура, мышь, модем, джойстик. К ним же относятся и устройства вывода информации, для которых характерно представление данных в виде линейного потока: принтеры, звуковые карты и т.д. По своей природе символьные устройства обычно умеют совершать две общих операции: ввести символ (байт) и вывести символ (байт) – ***get*** и ***put***.
- Для блочных устройств, таких как магнитные и оптические диски, ленты и т.п., естественными являются операции чтения и записи блока информации – ***read*** и ***write***, а также, для устройств прямого доступа, операция поиска требуемого блока информации – ***seek***.
- Драйвера символьных и блочных устройств должны предоставлять базовой подсистеме вв функции для осуществления описанных общих операций.

- Помимо общих операций некоторые устройства могут выполнять операции специфические, свойственные только им – например, звуковые карты умеют увеличивать или уменьшать среднюю громкость звучания, дисплеи умеют изменять свою разрешающую способность. Для выполнения таких специфических действий в интерфейс между драйвером и базовой подсистемой ввода-вывода обычно входит еще одна функция, позволяющая непосредственно передать драйверу устройства произвольную команду с произвольными параметрами, что позволяет задействовать любую возможность драйвера без изменения интерфейса. В операционной системе UNIX такая функция получила название ***ioctl*** (**i**nput-**o**utput **c**on**t**ro**l**).
- Помимо функций ***read***, ***write***, ***seek*** (для блочных устройств), ***get***, ***put*** (для символьных устройств) и ***ioctl*** в состав интерфейса обычно включают еще следующие функции:
  1. Функцию инициализации или повторной инициализации работы драйвера и устройства – ***open***.
  2. Функцию временного завершения работы с устройством (может, например, вызывать отключение устройства) – ***close***.
  3. Функцию опроса состояния устройства (если по каким-либо причинам работа с устройством производится методом опроса его состояния – например, в операционных системах Windows NT и Windows 9x так построена работа с принтерами через параллельный порт) – ***poll***.
  4. Функцию останова драйвера, которая вызывается при останове операционной системы или выгрузке драйвера из памяти, - ***halt***.

Функции базовой подсистемы ввода-вывода.

- Базовая подсистема ввода-вывода служит посредником между процессами вычислительной системы и набором драйверов. Системные вызовы для выполнения операций ввода-вывода трансформируются ею в вызовы функций необходимого драйвера устройства.
- Базовая подсистема предоставляет вычислительной системе такие услуги, как поддержка блокирующихся, не блокирующихся и асинхронных системных вызовов, буферизация и кэширование входных и выходных данных, осуществление spooling’а и монопольного захвата внешних устройств, обработку ошибок и прерываний, возникающих при операциях ввода-вывода, планирование последовательности запросов на выполнение этих операций.

Блокирующиеся, не блокирующиеся и асинхронные системные вызовы.

Системные вызовы, связанные с операций вв, можно разбить на три группы по способам реализации взаимодействия процесса и устройства вв :

- *блокирующиеся* системные вызовы, т.е. инициированный процесс переводится операционной системой из состояния ***исполнение*** в состояние ***ожидание***.

- *не блокирующиеся* системные вызовы, где процесс, применивший не блокирующийся вызов, не переводится в состояние ***ожидание*** вообще.

- *асинхронные* системные вызовы. роцесс, использовавший асинхронный системный вызов, никогда в нем не блокируется. Системный вызов инициирует выполнение необходимых операций ввода-вывода и немедленно возвращается, после чего процесс продолжает выполнять свою регулярную деятельность.

Буферизация и кэширование

Под *буфером* обычно понимается некоторая область памяти для запоминания информации при обмене данных между двумя устройствами, двумя процессами или процессом и устройством. Существуют три причины, приводящие к использованию буферов в базовой подсистеме ввода-вывода:

1) разные скорости приема и передачи информации, которыми обладают участники обмена.
2) разные объемы данных, которые могут быть приняты или получены участниками обмена единовременно
3) необходимостью копирования информации из приложений, осуществляющих ввод-вывод, в буфера ядра операционной системы и обратно. Допустим, что некоторый пользовательский процесс пожелал вывести информацию из своего адресного пространства на внешнее устройство.

Про всех остальных функции можно прочитать здесь.