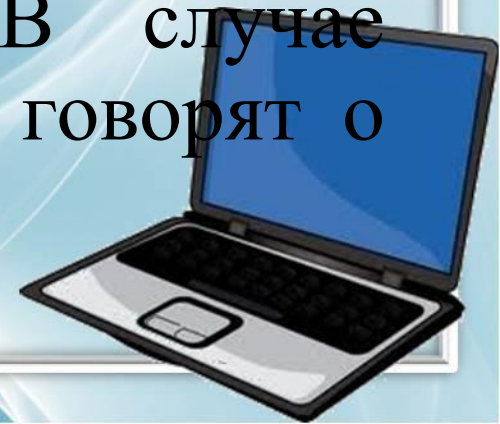


Основные виды ресурсов и возможности их разделения



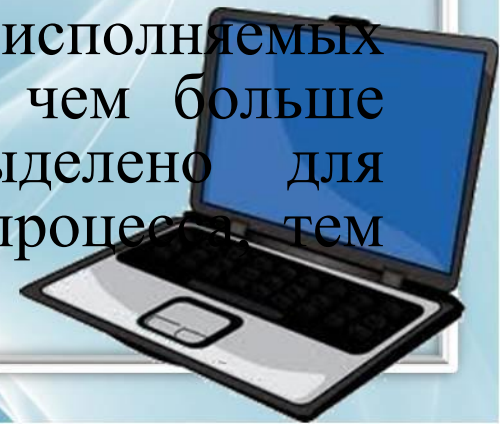
Процессорное время

- Прежде всего, одним из важнейших ресурсов является сам **процессор**, точнее — **процессорное время**. Процессорное время делится попеременно (параллельно). Имеется множество методов разделения этого ресурса. Разговор о процессоре как об одном из ресурсов более характерен для мультипроцессорных систем. В случае однопроцессорных систем чаще говорят о процессорном времени.



Оперативная память

- Вторым видом ресурсов вычислительной системы можно считать память. **Оперативная память** может делиться и **одновременно** (то есть в памяти одновременно может располагаться несколько задач или, по крайней мере, текущих фрагментов, участвующих в вычислениях), и **попеременно** (в разные моменты оперативная память может предоставляться для разных вычислительных процессов). Память — очень интересный вид ресурса. Дело в том, что в каждый конкретный момент времени процессор при выполнении вычислений обращается к очень ограниченному числу ячеек оперативной памяти. С этой точки зрения желательно память выделять для возможно большего числа параллельно исполняемых задач. С другой стороны, как правило, чем больше оперативной памяти может быть выделено для конкретного текущего вычислительного процесса, тем лучше будут условия его выполнения.



Поэтому проблема эффективного разделения оперативной памяти между параллельно выполняемыми вычислительными процессами является одной из самых актуальных. **Внешняя память** тоже является ресурсом, который часто необходим для выполнения вычислений. Когда говорят о внешней памяти (например, памяти на магнитных дисках), то собственно память и доступ к ней считаются разными видами ресурса. Каждый из этих ресурсов может предоставляться независимо от другого. Но для полноценной работы с внешней памятью необходимо иметь оба этих ресурса. Собственно внешняя память может разделяться и одновременно, а вот доступ к ней всегда разделяется попеременно.



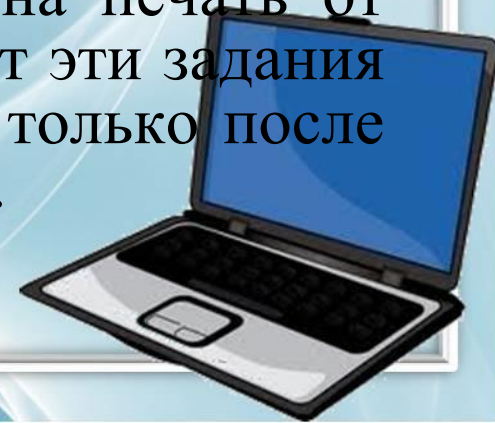
внешние устройства

- Если говорить о **внешних устройствах**, то они, как правило, могут разделяться **параллельно**, если используются механизмы прямого доступа. Если же устройство работает с **последовательным доступом**, то оно не может считаться **разделяемым ресурсом**. Простыми и наглядными примерами внешних устройств, которые не могут быть разделяемыми, являются **принтер и накопитель на магнитной ленте**. Действительно, если допустить, что принтер можно разделять между двумя процессами, которые смогут его использовать (управлять его работой) попеременно, то результаты печати, скорее всего, окажутся негодными — фрагменты выведенного текста могут перемешаться таким образом, что будет не понятно, что есть что.



Примеры разделяемых и неразделяемых ресурсов

- Аналогично и для накопителя на магнитной ленте. Если один процесс начнет что-то читать или писать, а второй при этом запросит перемотку ленты на ее начало, то оба вычислительных процесса не смогут выполнить свои вычисления. Здесь следует заметить, что при работе с устройствами печати мы, тем не менее, явно наблюдаем возможность печатать из разных программ, выполняющихся параллельно. Однако необходимо знать, что это реализуется за счет того, что каждый вычислительный процесс получает свой **виртуальный принтер**, который он ни с кем не разделяет. А операционная система, получив задания на печать от выполняющихся задач, сама упорядочивает эти задания и передает очередное задание на принтер только после полного завершения предыдущего задания.



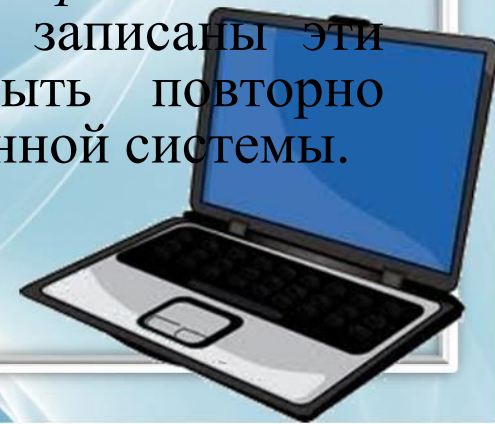
программные модули

- Очень важным видом ресурсов являются *программные модули*. Прежде всего, мы будем рассматривать *системные программные модули*, поскольку именно они обычно считаются программными ресурсами и поэтому в принципе могут распределяться между выполняющимися процессами.



Однократно используемые программные модули

- Программные модули могут быть однократно используемыми и многократно (или повторно) используемыми. *Однократно используемыми называют такие программные модули, которые могут быть правильно выполнены только один раз, то есть в процессе своего выполнения они могут испортить себя: либо повреждается часть кода, либо исходные данные, от которых зависит ход вычислений. Очевидно, что однократно используемые программные модули являются неделимым ресурсом. Более того, их, как правило, вообще не распределяют как ресурс системы. Системные однократно используемые программные модули, как правило, задействуются только на этапе загрузки операционной системы. При этом следует иметь в виду тот очевидный факт, что собственно двоичные файлы, которые обычно хранятся на системном диске и в которых и записаны эти модули, не портятся, а потому могут быть повторно использованы при следующем запуске операционной системы.*



Повторно используемые программные модули

- *Повторно используемые программные модули, в свою очередь, могут быть непривилегированными, привилегированными и реентерабельными. Все они допускают корректное повторное выполнение программного кода при обращении к нему из другой программы.*



Привилегированные программные модули

- *Привилегированные программные модули работают в так называемом привилегированном режиме, то есть при отключенной системе прерываний (часто говорят, что прерывания закрыты), когда никакие внешние события не могут нарушить естественный порядок вычислений. Как результат, программный модуль выполняется до своего конца, после чего он может быть вновь вызван на исполнение из другой залами (другого вычислительного процесса). С позиций стороннего наблю-*



Непривилегированные программные модули

- Непривилегированные программные модули — это обычные программные модули, которые могут быть прерваны во время своей работы. Следовательно, такие модули в общем случае нельзя считать разделяемыми, потому что если после прерывания выполнения такого модуля, исполняемого в рамках одного вычислительного процесса, запустить его еще раз по требованию другого вычислительного процесса, то промежуточные результаты для прерванных вычислений могут быть потеряны.*



реентерабельные программные модули

В противоположность этому, реентерабельные программные модули (допускают повторное многократное прерывание своего исполнения и повторный их запуск по обращению из других задач (вычислительных процессов). Для этого реентерабельные программные модули должны быть созданы таким образом, чтобы было обеспечено сохранение промежуточных результатов для прерываемых вычислений и возврат к ним, когда вычислительный процесс возобновляется с прерванной ранее точки. Это может быть реализовано двумя способами: с помощью статических и динамических методов выделения памяти под сохраняемые значения.*

*** Реентерабельный — допускающий повторные прерывания (дословный переводсанглийского слова •re-enterable*)**

