

Виды параллелизма:

1. Естественный параллелизм независимых задач. Он заключается в том, что в систему поступает непрерывный поток не связанных между собой задач, т.е. решение любой задачи не зависит от результатов решения других задач. В этом случае использование нескольких процессоров однозначно повышает производительность.

2. Параллелизм независимых ветвей. Имеет очень широкое реальное распространение. Заключается в том, что при решении большой задачи могут быть выделены отдельные независимые части - ветви программы, которые при наличии нескольких процессоров могут выполняться параллельно и независимо друг от друга.

Двумя независимыми ветвями будем считать части, для которых выполняются следующие условия:

- ни одна из входных величин для ветви не является выходной для другой ветви (отсутствие функциональных связей);
- обе ветви не производят запись в одни и те же ячейки памяти (отсутствие связи по использованию ОП);
- условия выполнения одной ветви не зависят от результатов или признаков, полученных при выполнении другой ветви (независимость по управлению);
- обе ветви должны выполняться по разным блокам программы (программная независимость).

На самом деле задача обнаружения и реализации параллелизма независимых ветвей - очень сложная и объёмная. Особенная сложность - неизвестность длительности исполнения каждой из множества ветвей. На практике невозможно избежать простоев в работе процессоров, мы лишь минимизируем их.

3. Параллелизм объектов или данных. Имеет место тогда, когда по одной и той же программе обрабатывается некая совокупность данных, поступающая в систему одновременно. Примеры: данные от РЛС - все сигналы обрабатываются по одной и той же программе, задачи векторной алгебры, в которых выполняются одинаковые операции над парами чисел двух аналогичных объектов.

Основным признаком многозадачной ОС является способность совмещать выполнение нескольких прикладных программ. Большое значение при этом имеет способ совмещения, то есть на каком уровне или как конкретно реализовано совмещение.

Если однопроцессорная, но многозадачная, система выделяет каждой прикладной программе определенный квант времени, спустя который она переключается на выполнение следующей программы, то это **система с разделением времени** (time-sharing system).

Первые ОС, реализованные на персональных компьютерах, сильно уступали в концептуальном плане и по своим реальным возможностям системам с разделением времени, давно реализованным в mainframe- компьютерах. Win 16: обработав очередное сообщение, приложение передает управление ОС, которая может передать управление другому приложению. Такой вид многозадачности, при котором ОС передает управление от одного приложения другому не в любой момент времени, а только когда текущее приложение отдает управление системе, получил, как было упомянуто, название **кооперативной многозадачности** (cooperative multi-tasking).

Начиная с Windows 95 реализован принципиально другой вид многозадачности, в котором ОС действительно контролирует и управляет процессами, потоками и их переключением. Способность ОС прервать выполняемый поток практически в любой момент времени и передать управление другому ожидающему потоку определяется термином preemptive multitasking — **преимущественная, или вытесняющая, многозадачность**: все существующие в данный момент потоки, часть из которых может принадлежать одному и тому же процессу, претендуют на процессорное время и, с точки зрения пользователя должны выполняться одновременно. Для создания этой иллюзии система через определенные промежутки времени забирает управление, анализирует свою

очередь сообщений, распределяет сообщения по другим очередям в пространстве процессов и, если считает нужным, переключает потоки => устойчивость к зависаниям

Вытесняющая и невытесняющая многозадачность. Важнейшим разделяемым ресурсом является процессорное время. Способ распределения процессорного времени между несколькими одновременно существующими в системе вычислительными процессами во многом определяет особенность ОС. Среди множества существующих способов реализации многозадачности можно выделить две группы алгоритмов:

- невытесняющая многозадачность (NetWare, Windows 3.x); (планирование целиком на ОС)
- вытесняющая многозадачность (Windows NT, OS/2, Unix). (планирование распределено между ОС и прикладными программами)

В зависимости от областей использования **многозадачные** ОС подразделяются на три типа:

- системы пакетной обработки (например, ОС ЕС);
- системы с разделением времени (Unix, VMS);
- системы реального времени (QNX, RT/11).