

В идеальной многозадачной системе все процессы, выполняющиеся параллельно, независимы друг от друга (т.е. асинхронны). На практике такая ситуация маловероятна, поскольку рано или поздно возникают ситуации, когда параллельным процессам необходим доступ к некоторым общим ресурсам. Для этого необходимо введение на уровне ОС средств, предоставляющих такую возможность.

При выполнении параллельных процессов может возникать проблема, когда каждый процесс, обращающийся к разделяемым данным, исключает для всех других процессов возможность одновременного с ним обращения к этим данным - это называется **взаимоисключением (mutual exclusion)**.

Ресурс, который допускает обслуживание только одного пользователя за один раз, называется **критическим ресурсом**. Если несколько процессов хотят пользоваться критическим ресурсом в режиме разделения времени, им следует синхронизировать свои действия таким образом, чтобы этот ресурс всегда находился в распоряжении не более чем одного из них.

Участки процесса, в которых происходит обращение к критическим ресурсам, называются **критическими участками**.

Задача взаимного исключения

Критическая секция — часть программы, в которой есть обращение к совместно используемым данным. При нахождении в критической секции двух (или более) процессов, возникает состояние «гонки» («состязания»). Для избежания данной ситуации необходимо выполнение условий:

1. Два процесса не должны одновременно находиться в критических областях.
2. В программе не должно быть предположений о скорости или количестве процессоров.
3. Процесс, находящийся вне критической области, не может блокировать другие процессы.
4. Невозможна ситуация, в которой процесс вечно ждет попадания в критическую область.
5. критические области не должны иметь приоритета в отношении друг друга;
6. Решение о вхождении процессов в их критические области при одинаковом времени поступления запросов на такое вхождение и равноприоритетности процессов не откладывается на неопределенное время, а является конечным по времени;
7. относительные скорости развития процессов неизвестны и произвольны
8. любой процесс может переходить в любое состояние, отличное от активного, вне пределов своей критической области;
9. освобождение критического ресурса и выход из критической области должны быть произведены процессом, использующим критический ресурс, за конечное время.

Задача взаимного исключения является одной из ключевых проблем параллельного программирования.