

О МНОГОПОТОЧНОЙ ИЛИ МНОГОНИТЕВОЙ ОБРАБОТКЕ ИНФОРМАЦИИ

Свойства языка Java в среде NetBeansIDE предусматривают возможность для поддержки многопоточного (многонитиевого) программирования. Следует отметить, что многонитиевое или многопоточное приложение включает две части, которые выполняются параллельно. Каждую такую часть программы (приложения) будем называть потоком (нитью). Каждый поток определяет путь для выполнения команд. Такая возможность многопоточности (многонитиевости) будем относить к многозадачности.

Целью доклада является особенности многопоточной обработки информации на языке Java в среде NetBeansIDE.

В многозадачности следует выделить две разновидности различаемых на основе процессов и потоков. Рассмотрим их отдельно и выясним, в чем заключается их различия.

Под процессом понимаем – исполняемое приложение (программа), которое при многозадачности является средством, при котором могут выполняться параллельно два и более приложения (программы) на компьютере. Обращаем внимание на то, что используя процессы, при многозадачности, имеется возможность совместно выполнять приложения электронных таблиц, текстового редактора обращения в Интернет. Также дополнительно, следует отметить, что при координации примеров (задач) на основе процессов можно использовать планировщика – при многозадачности.

Под потоком понимаем – координируемую единицу исполняемого кода (поток исполнения). При наявности потоков, каждый процесс имеет один або больше потоков при многозадачности. Это расширяет возможности потоков, что позволяет в приложении (программе) совместно выполнять несколько заданий (задач) два и более. Такое положение дел, позволяет в одной программе решать несколько задач. Следует оговориться, обратить внимание на то, что на языке Java добавки (программы) в среде NetBeansIDE поддерживают многозначность, используя процессы. Следует указать, что управлять процессами в самих программах (дodatках) на языке Java в среде NetBeansIDE – нельзя этого делать. На указанном языке управлять можно только потоками.

Остановимся на вопросе в чем преимущества многопоточной обработки информации. Оно разрешает, писать приложения (программы) работающие эффективно на учете времени простоя программ при выполнении. Для большей ясности, остановимся на устройствах ввода-вывода, которые работают на много медленнее центрального процессора, скажем это устройства, подключенные к сетевым портам, накопители на дисках или клавиатура, которые работают намного медленнее. В этом плане, добавку (программе) приходится простаивать ожидать доставки данных на устройство ввода-вывода или приема информации из него. Во время простоя при многопоточной обработке добавка решает другую задачу во время необходимого ожидания. Что особенно ценно, например, при этом: *одна часть программы* выполняет – буферизацию блока отправляемых данных, *другая* – чтение текстовой информации, вводимой с клавиатуры; третья – отправляет файл через соединение с Интернетом.

Остановимся на многопроцессорных или многоядерных вычислительных системах, хотя они и получили признание, но отсутствует их внедрение, это настораживает. В настоящее время в основном используются однопроцессорные системы. Следует отметить, что языковые средства организации многопоточной обработки в Java пригодны для обеих разновидностей вычислительных систем. Многопоточная обработка практически не используются, студентов знакомят только на старших курсах, учить надо с первых курсов.

Как уже указывали, что в многопроцессорных или многоядерных системах два потока или более могут выполняться параллельно. Что позволяет повысить производительность программ и скорость выполнения отдельных операций, мы пришли к чему стремились к повышению быстродействия и увеличению производительности.

Рассмотрим четыре состояния потока:

1. *выполняющимся; готовым к выполнению*, при получении времени и ресурсов;
2. *приостановленным*, т.е. возобновленным в дальнейшем;
3. *заблокированным* в ожидании ресурсов для своего выполнения;
4. *завершенным*, выполнение окончено и не может возобновлено.

Выводы. В докладе подробно уделено внимание особенностям многопоточной обработки информации на языке Java в среде NetBeansIDE. Указываются в Java языковые средства для поддержки потоков. Решены вопросы синхронизации координации потоков.