

1. Что означают аббревиатуры SISD, SIMD, MISD, MIMD?

SISD (Single Instruction Single Data) - это архитектура ЭВМ в классификации параллельных вычислительных систем, согласно которой процессор выполняет одну инструкцию за другой последовательно, работая при этом с одним потоком данных. Не являются параллельными ЭВМ.

SIMD (Single Instruction Multiple Data) - это архитектура ЭВМ в классификации параллельных вычислительных систем, согласно которой процессор(ы) выполняет(ют) одну инструкцию за другой, работая при этом с несколькими потоками данных.

MISD (Multiple Instruction Single Data) - это архитектура ЭВМ в классификации параллельных вычислительных систем, согласно которой процессоры выполняют несколько инструкций параллельно, оперируя над одним потоком данных.

MIMD (Multiple Instruction Multiple Data) - это архитектура ЭВМ в классификации параллельных вычислительных систем, согласно которой процессоры выполняют параллельно несколько инструкций, оперируя над несколькими наборами данных.

2. Какие типы многозадачности существуют и в чем их различия?

Существует 2 вида многозадачности: вытесняющая и не вытесняющая. Разница между ними в том, что при невытесняющей многозадачности процессор переключается между задачами при наступлении события, а при вытесняющей переключение происходит по таймеру и каждой задаче выделяется определенный квант времени на обработку процессором. И в том, и в другом случае процессор не знает ничего о многозадачности.

3. Что такое процесс и поток?

Процесс — это сущность операционной системы, которая формируется при выполнении какой-либо программы и аккумулирует в себе информацию о подключенных библиотеках, данных, используемых программой, используемой памяти, потоках и пр., необходимое для работы программы.

Поток - сущность операционной системы, которая формируется при старте процесса. У каждого процесса есть хотя бы один поток. Множество потоков предоставляет возможность распараллеливать выполнение процесса, программы становятся “быстрее”, поскольку один процесс использует все адресованные ему ресурсы памяти и предоставляет их потокам без необходимости переключения ОС с одного адресного пространства в другое.

4. Что делает системный вызов fork()?

Системный вызов `fork()` - функция ОС, которая позволяет создавать процесс из другого процесса в случае, когда программе для последующего выполнения необходимо распараллеливание процессов. По сути, происходит его копирование. Процесс-потомок получает дополнительный идентификатор процесса-родителя (PPID), процессу-родителю будет известен PID потомка (системный вызов `fork()` передает таким образом код возврата).