

Что такое Capre?

Capre - язык императивного параллельного программирования, созданный на собственной независимой платформе.

Первая версия языка была разработана в 1994 году.

На сегодняшний день разработана и используется четвертая версия языка.

Capre - язык программирования, позволяющий описывать и реализовывать вычисления в той стилистике параллелизма, которая в наибольшей степени соответствует решаемым задачам. По-существу, Capre обеспечивает представление и реализацию вычислений по всем классам архитектур по Флинну.

Capre независим от архитектурной платформы, обладает собственной параллельной виртуальной машиной, не зависит от средств распараллеливания операционных систем, в то время как подавляющее число языков и библиотек параллельного программирования реализуют задачи параллельных вычислений только с помощью средств операционных систем и аппаратных особенностей.

Параллельные процессы в Capre могут исполняться как на однопроцессорном и одноядерном вычислителе, так и на многоядерных, многопроцессорных и многомашинных комплексах.

Capre обладает средствами динамической компоновки выполняемых программ, т.е. программа может компоноваться в зависимости от хода вычислений загрузкой и выгрузкой объектных модулей Capre-а (свойство самоорганизации). Тем самым, на каждом этапе вычислений мы можем иметь выполняемую программу, наиболее эффективно использующую ресурсы вычислительной установки.

В виртуальную машину Capre встроен компилятор времени исполнения (jitter), позволяющий компилировать исходные модули программ в ходе вычислений и исполнять их.

Capre обладает возможностями транспортирования программных модулей (как исходных, так и откомпилированных) в вычислительных сетях для последующего выполнения на других компьютерах, тем самым осуществляя распределение вычислений.

Capre основан на теоретически неограниченном параллелизме, т.е. количество запускаемых параллельных процессов ограничено только ресурсами вычислителя.

В новейшей версии виртуальной машины языка Capre была реализована концепция распределения виртуальных машин, реализующих одно вычисление, по реально параллельным устройствам (ядра процессора, процессоры, в перспективе - вычислительные машины многомашинного комплекса).

Согласно данной концепции псевдопараллельные процессы программы на Capre распределяются и перераспределяются для реализации между виртуальными машинами, функционирующими на физических устройствах.

Программы на Cарег могут компоноваться следующим образом:

- 1) созданием псевдоисполнимого (псевдо-ехе) кода, в который встраивается базовая виртуальная машина, компилятор времени исполнения (jitter), откомпилированный код стартового модуля программы (остальные объектные модули могут быть имплантированы уже в ходе вычисления);
- 2) созданием приложения на любом другом языке и включением в исполнимый код машины и jitter языка Cарег с последующими вызовами виртуальной машины из собственного кода для исполнения фрагментов на Cарег;
- 3) виртуальная машина и jitter представлены в виде динамической библиотеки, тем самым могут быть загружены из любого приложения на любом языке и использованы на основе стандартного протокола вызова функций библиотек;
- 4) программа на Cарег может загрузить любую динамическую библиотеку, созданную на любом другом языке программирования, и использовать функции такой библиотеки.

Тем самым, программы на Cарег имеют, практически, полный набор средств взаимодействия с программными кодами на других языках программирования.

С учетом масштабируемости на многопроцессорных и многомашинных системах количество параллельных процессов может исчисляться многими миллионами.