С.И. КИРКОРОВ

ВЫБОР ЯЗЫКА ПРОГРАММИРОВАНИЯ ДЛЯ ЗАЩИЩЕННЫХ СРЕДСТВ СВЯЗИ

В настоящее время именно среда выполнения предопределяет тот инструментарий с помощью которого специалисты реализуют свои идеи в науке, промышленности и обучении, при подготовке кадров в научной и технических областях. Желание абстрагироваться от технических деталей и сосредоточиться специфике задачи привело к бурному росту, в том числе и узко прикладных языков программирования. Характерной чертой, которых является их жесткая привязка к аппаратно-программной платформе (АПП). Таким образом, целесообразно определить аппаратно-программную платформу как связку: аппаратура-ОС-среда выполнения. В технической и экономической сфере наиболее популярные платформы: виртуальная машина (VM) Java, GCC, .NET, WEB-вычисления (сервер Apache, MS IIS). Достижения физико-математических наук аккумулируются в таком инструментарии, как Mathematica (фирма WOLFRAMRESEARCH), Mathcad (фирма PTC), Statistica (фирма STATSOFT), Matlab (фирмы MATHWORKS).

Не смотря на удобство использования выше описанного инструментария, возникают трудности в создании реальных систем или проведения вычислительного эксперимента для разработанной математической модели.

Одной из причин ограничений применения такого инструментария это сложность или высокая стоимость обеспечения соответствующего уровня гарантий конечного продукта в области технической защиты информации по всем трем составляющим – конфиденциальность, целостность и доступность. Это связано с различиями оптимизированного под прикладную область алгоритмом и его математической моделью реализованной такими пакетами.

Сложность и стоимость возрастет, когда стоит задача переноса модели алгоритма и ее реализации на другую доверенную платформу. Под доверенной платформой здесь подразумевается собственный вычислитель, например специализированный процессор с другой операционной системой или без нее или программируемые пользователями вентильные матрицы. Для тестирования доверенной платформы и параллельных вычислений реализуемые ее, как правило, создается стендовое оборудование. Стендовое оборудование как средство измерения также требует соответствующего уровня гарантий правильности своего функционирования.

Правильным выбором в этом случае было бы использовать инструментальное средство, которое:

1. Само могло бы пройти сертификационные испытания в области технической защиты информации, то есть, как минимум имела открытые спецификации, коды программ и так далее.

2. Соответствовало стандарту, строго его выполняло и имело стандарт, регламентирующий эти проверки. Было реализовано (или могло быть адаптировано) для целевых платформ.

3. Обеспечивало поддержку многозадачности (для моделирования алгоритмов, реализующих параллельные вычисления) на уровне языка высокого уровня. Этим достигается стабильность семантики, разработчик избавляется от необходимости использования разнородных внешних библиотек или собственных решения для обеспечения многозадачности.

4. Влияет, конечно, и экономический фактор. Стоимость приобретения и совокупного владения такого инструментария не должна превышать 1/3 от всей стоимости разработки.

Встроенная поддержка многозадачности является уникальной и широко известной особенностью языка программирования Ada, которая выгодно отличает его от большинства современных языков программирования

Одним из решением задачи – интеграция подходов и АПП на самом раннем этапе подготовки к проведению вычислительного эксперимента и при создания реальных систем, для которых необходимо проводить сертификационные испытания и аттестацию по требованиям информационной безопасности. Своеобразным "программным клеем" целесообразно использовать язык программирования Ada. Реализацией идеи интеграции является расширяемая платформа OEM-2011 for Windows разработанная на языке программирования Ada.