УДК 519.686

Неволин А.В. [1,2]

[1] Московский физико-технический институт

[2] ЗАО «Интел А/О»

Комплексная верификация системы команд, двоичного транслятора и окружения

В первом приближении, двоичную трансляцию можно рассматривать как перевод кода из одной системы команд (исходной) в другую (целевую) [1]. Программу, выполняющую такое преобразование, принято называть двоичным транслятором (ДТ).

На сегодняшний день, из-за огромной популярности архитектуры Intel x86, почти невозможно представить появление новой успешной микропроцессорной архитектуры, не совместимой на двоичном уровне с x86. В то же время, наиболее эффективным на сегодня способом обеспечения двоичной совместимости является применение двоичной трансляции. Именно эта технология легла в основу наиболее известных проектов последних лет: Itanium [2], Transmeta [3], Elbrus 2000. Без успешного развития двоичной трансляции, сегодня едва ли можно ожидать появления новой микропроцессорной архитектуры, способной составить конкуренцию x86. Ниже ДТ рассматривается как неотъемлемая часть микропроцессорной архитектуры, двоично совместимой с x86.

В данной работе представлен метод комплексной проверки корректности работы ДТ, корректности работы окружения ДТ и правильности целевой системы команд. Предполагается, что задача осложнена следующими особенностями: 1) ДТ может выполнять достаточно агрессивные оптимизации и планирование кода; 2) работа ДТ может опираться на использование профильной информации; 3) ДТ может обеспечивать различные уровни двоичной совместимости; 4) трансляция кода может выполняться как статически, так и динамически; 5) не существует машины «в железе», реализующей целевую систему команд (т.е. рассматривается

этап проектирования такой машины; при этом необходима проверка, изменение и дополнение системы команд).

Все перечисленные особенности сильно усложняют процесс отладки и требуют решения следующих задач: 1) реализация интерпретатора, полностью компенсирующего отсутствие «в железе» целевой платформы и отсутствие «родной» операционной системы; 2) реализация единого процесса проверки целевой системы команд и отладки длинной цепочки, состоящей из ДТ, ассемблера, компоновщика, дизассемблера, интерпретатора; 3) разработка способа проверки обеспечения как полной, так и частичной двоичной совместимости; 4) на длительно исполняющихся приложениях проверка цепочки инструментов и системы команд должна происходить за разумное время.

В данной работе были разработаны различные способы решения перечисленных задач, в том числе: предварительное статическое и динамическое профилирование исходного приложения с целью поиска кода, подлежащего трансляции; эмуляция загрузки x86 кода; эмуляция системных вызовов; прямая подстановка эффекта системного вызова; отслеживание и сравнение контекстов памяти и регистров; эмуляция рабочего окружения; использование контрольных точек для отладки долго исполняющихся приложений. См. рис. 1.

Все перечисленные методы строго дополняют друг друга. Не отдельные техники, но их определенная комбинация, делают возможным сравнение исполнения x86 кода на «родной» платформе и на проектируемой. И именно такое сравнение, в свою очередь, позволяет нам доказать правильность реализации БТ и корректность целевой системы команд.

СПИСОК ЛИТЕРАТУРЫ

1. *Ebcioglu K., Altman E.* Dynamic binary translation and optimization. // IEEE Transactions on computers. – 2001. – V. 50, N. 6. – P. 529-548.
2. *Baraz L., Devor T. and others.* IA-32 execution layer: a two-phase dynamic translator designed to support IA-32 applications on Itanium®-based systems // Microarchitecture, MICRO-36. Proceedings. 36th Annual IEEE/ACM

International Symposium on. – 2003. – P. 191-201.

3. *Dehner J.C., Grant B.K. and others.* The Transmeta Code Morphing™ Software: using speculation, recovery, and adaptive retranslation to address real-life challenges // Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization. – 2003. – P. 15-24

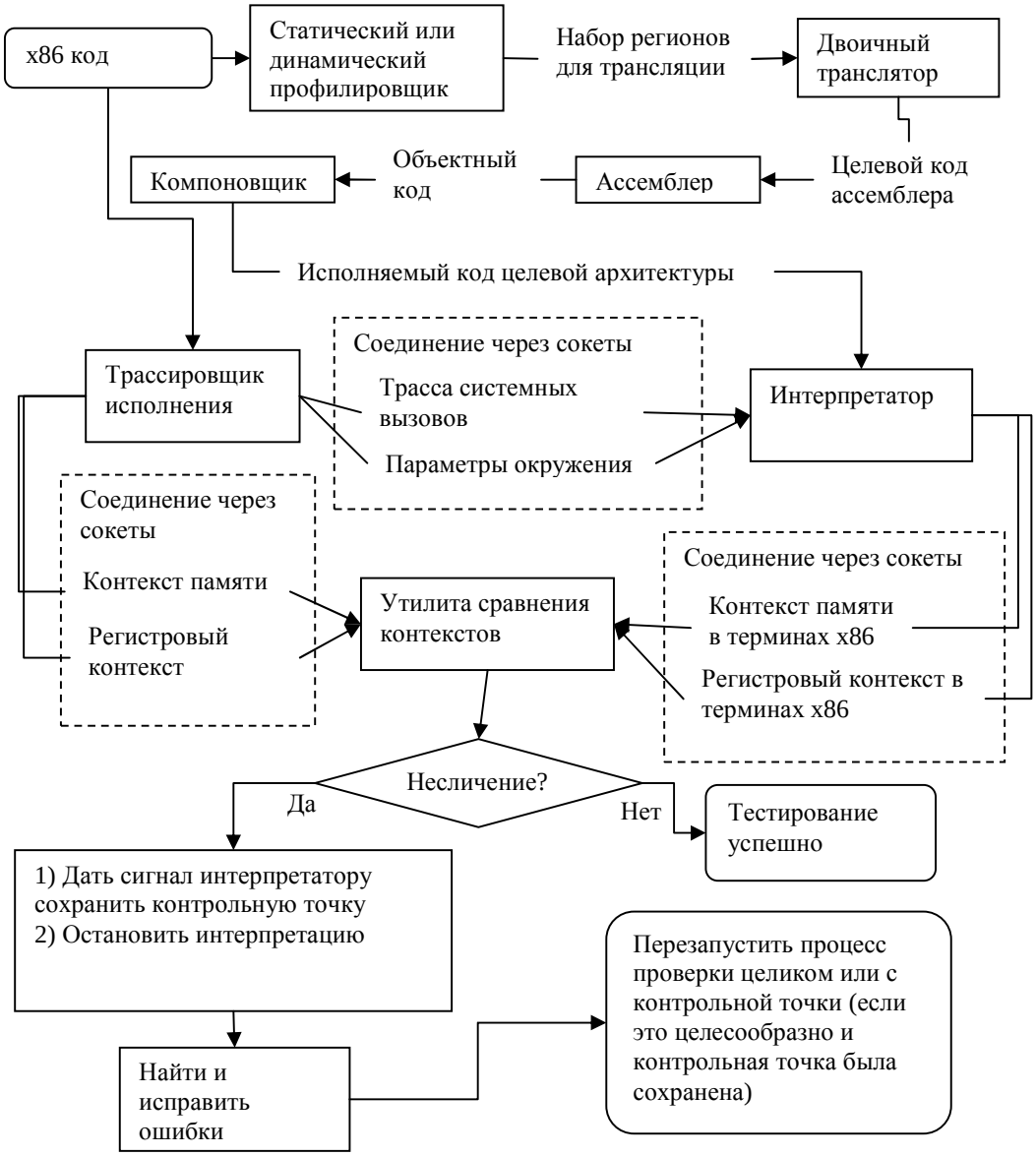

Рис. 1. Высокоуровневая схема верификации двоичного транслятора, системы команд и окружения.