УДК 519.681.3

ПЕРЕНОС ПРИЛОЖЕНИЯ WIN32 НА LINUX И ЕГО ОПТИМИЗАЦИЯ

Белоус Д.В.
Научный руководитель докт. техн. наук Глушков С.В.
Морской государственный университет им. адм. Г.И. Невельского

В настоящее время ведется разработка программного комплекса для визуализации показаний с датчиков, которые устанавливаются в танки нефтеналивных судов для определения давления и температуры груза. При дальнейшей разработке такое приложение позволит автоматизировать контроль состояния транспортируемого груза. Изначально приложение писалось на платформе WIN32 с использованием Open Graphic Library (OpenGL) на языке высокого уровня C++ (ISO/IEC 14882 C++). Поскольку платформа Windows является коммерческой, то для снижения стоимости разрабатываемого программного комплекса принято решение о переходе на платформу Linux. При переносе были выявлены следующие различия:

— интерфейсов прикладного программирования (API);
— оптимизации с помощью ассемблерных вставок.

Различие интерфейсов прикладного программирования. Самое главное различие для оконного приложения – это создание самого окна. В WIN32 создание окна начинается с описания экземпляра класса окна. На этом этапе происходит описание базовых свойств будущего окна, таких как:

— имя класса окна;
— используемое меню;
— стили окна;
— фон окна;
— адрес функции для обработки событий окна;
— идентификатор приложения.

После описания экземпляра класса окна необходимо этот экземпляр зарегистрировать, чтобы система в дальнейшем могла взаимодействовать с окном. Затем необходимо создать само окно, при этом передаётся идентификатор окна, по которому будет происходить дальнейшее с ним взаимодействие. На этом этапе задаются следующие второстепенные свойства окна:

— дополнительные стили окна;
— имя окна;
— размеры окна;
— адрес функции, которую необходимо выполнить до создания окна.

Далее создаётся бесконечный цикл для обработки событий окна. События окна представлены в виде зашифрованных сообщений. Для работы с такими сообщениям необходимо вызывать функции API WIN32 для дешифровки и отправки в функцию обработки событий окна.

Отличие создания окна на Linux заключается в явной выраженности «клиент-сервер». В роли клиента выступает само приложение, а в роли сервера – X Server. Первое что необходимо сделать – установить соединение с X Server. При создании самого окна также имеются отличия. Вместо экземпляра класса используется структура для описания свойств окна. Обязательным свойством является карта цветов (color map). Получить её можно из окна root. Также необходимо указать маску обработки сообщений окна. Обработка событий происходит в два этапа. Необходимо получить количество сообщений для окна, которые находятся в очереди, а затем изъять из

очереди само сообщение. С помощью операторов выбора исполняются необходимые инструкции.

Оптимизация с помощью ассемблерных вставок. Первое, что необходимо сделать на данном этапе, – это выбрать синтаксис ассемблерных вставок. Выбор стоит перед синтаксисами AT&T и Intel. В настоящее время компиляторы GNU поддерживают оба синтаксиса.

При создании ассемблерных вставок необходимо учитывать разрядность операционной системы на базе ядра Linux. Если Windows может запускать 32-х разрядные приложения в 64-х разрядной среде, то Linux на это не способен.

Необходимо также учитывать, что в 64-х разрядной операционной системе будут использоваться 64-х битные регистры, а размер адреса памяти будет составлять не четыре байта, а восемь.

Сама ассемблерная вставка для компиляторов GNU GCC выполняется в виде макроса, начинающегося со служебного слова «asm». Входной параметр макроса представляет собой строку с дополнительным листом параметров для синтаксиса AT&T и просто строку для синтаксиса Intel.

Серьезный недостаток ассемблерных вставок при использовании компиляторов GNU GCC состоит в ограниченности входных параметров. В то время как компиляторы для Windows вставляют в программу дополнительные инструкции для создания локальных переменных, компилятор будет распознавать адресацию локальных переменных по обращению к ним по мнемоническим именам.

В операционных системах на ядре Linux происходит сохранение адресов локальных переменных в регистры общего назначения и, частично, в регистры специального назначения. Компилятор не распознаёт дальнейшую адресацию по мнемоническим именам локальных переменных. Здесь появляется жёсткое ограничение на количество входных параметров и локальных переменных. Обойти данное аппаратное ограничение можно, используя структуры.

Создавая две структуры: одну для входных параметров, вторую для локальных переменных, – возможно работать с неограниченным числом переменных с помощью прямой адресации, используя всего два регистра общего назначения.

Необходимость в оптимизации может быть обусловлена скоростью работы приложения. В современных компиляторах заложен принцип избыточности операций, что позволяет компилировать программы любой сложности и любого вида, частично жертвуя при этом производительностью и размером самой программы.

Благодаря использованию языка высокого уровня C++ (ISO/IEC 14882 C++) и OpenGL, при переносе программного комплекса потребовалось лишь решить описанные выше проблемы. Программный комплекс успешно перенесен на платформу Linux и ведётся его дальнейшая доработка до коммерческого продукта. Перенос на платформу Linux позволил сократить стоимость проекта и устранить многие возможные проблемы с авторскими правами при реализации проекта в будущем.

Список литературы

1. Саймон Р. Microsoft Windows API. Справочник системного программиста. Второе издание, дополненное: Пер. с англ./ Ричард Саймон – К.: ООО «ТИД «ДС», 2004. – 1216 с.