

Дмитрий Костюк, Павел Луцюк

Брест, Брестский государственный технический университет

Применение виртуальных машин в составе иллюстрированных обзоров истории программного обеспечения

Аннотация

Рассматривается опыт использования виртуальных машин в качестве замены копий экрана в иллюстрированной истории графических интерфейсов. Охарактеризована степень наполненности материалов. Рассмотрена проблематика применения вложенной виртуализации с внешней виртуальной машиной, взаимодействующей с документом и активирующей снапшоты. Описано решение проблемы свободного пространства иллюстрированных обзоров с сохранением возможности включения гостевых систем с несвободной лицензией за счет автоматического перестроения итогового документа. Используемое на хост-системе ПО включает гипервизор QEMU, а также свободные HTML-фреймворки для работы с протоколом VNC и отображения интерактивной иллюстрированной хронологии.

Хотя технически для того, чтобы работать с новыми информационными технологиями, не обязательно знать историю их развития, тем не менее специалист, который формулирует либо применяет современную теорию без знания ее истории, рискует лично повторять ошибки предшественников одну за другой.

В русле данного утверждения находится настоящая разработка, предоставляющая пользователю локальной сети либо рабочей станции комплект связанных хронологически HTML-документов, каждый

из которых содержит описание особенностей конкретной графической операционной системы и ее живую иллюстрацию в виде встроенного фрейма с экраном работающей виртуальной машины (благодаря производительности современных ноутбуков и настольных ПК такая задача оказывается достаточно легко выполнимой). Информационный контент разработки основан на лекционном материале по истории графического интерфейса, включающем 40 настольных и 30 мобильных операционных систем и графических оболочек.

Техническая инфраструктура, позволяющая реализовать такие информационные материалы, рассмотрена нами в [1] и включает следующие компоненты:

- виртуальная машина QEMU с аппаратной поддержкой виртуализации;
- VNC-клиент noVNC, написанный на JavaScript и HTML5;
- JavaScript-фреймворк для отображения информационных материалов.

Запуск схемы, показанной рис. 1-а, выполняется стартовым скриптом, сканирующим вложенные подкаталоги в поисках элементов обзора: страниц информационного контента, образов виртуальных машин и скриптов для их запуска. Стартовый скрипт выдает найденным виртуальным машинам номера портов и перестраивает HTML-документ для включения страниц с информационными материалами в хронологию. Такой компонентный подход позволяет разделить информационный обзор на свободно распространяемую часть и дополнения, распространение которых ограничено условиями коммерческих лицензий.

Виртуальная машина используется как полностью изолированный контейнер для хранения мгновенных снимков запущенных ОС [2]. Выбор в пользу QEMU был сделан из-за предельно простого переноса образов виртуальных машин между компьютерами, а также из-за его способности помимо платформы x86 эмулировать процессоры SPARC, PowerPC, Motorola 68k, MIPS и ARM, что необходимо для запуска многих ОС 80-х и 90-х годов. Однако на текущий момент данная многоплатформенность осталась практически незадействованной, т. к. степень поддержки устройств на альтернативных платформах QEMU редко оказывается достаточной для запуска нужной ОС.

В то же время для поддержки практически всех устаревших Intel-несовместимых ОС либо разработаны свободные эмуляторы, либо

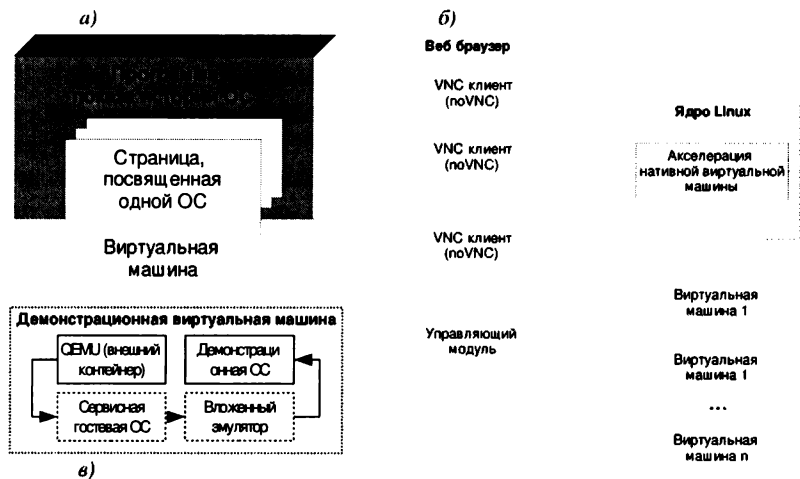


Рис. 1: Схема построения документа (а), взаимодействие компонентов системы (б) и схема вложенной виртуализации (в)

имеются эмуляторы из комплекта SDK. Первый вариант более характерен для настольных ОС, благодаря чему оказалось возможным включение в информационные материалы ОС Xerox Alto, Amiga, RiscOS, Apple Lisa, MacOS версий 1.x, 7.x и X. Проприетарные эмуляторы настольных ОС редки и принадлежат производителю самой ОС, как в случае Xerox GlobalView. В случае ОС мобильных устройств преобладают эмуляторы из состава SDK: Psion EPOC16 и EPOC32, PalmOS, Magic Cap, Windows CE, пред релизные версии Android SDK. Исключением из списка является свободный эмулятор Open Einstein, позволяющий запускать NewtonOS.

Однако перечисленные эмуляторы не поддерживают ни снапшоты, ни необходимый в нашем случае протокол VNC. В результате запуск большей части демонстрируемых ОС осуществляется по схеме вложенной виртуализации (рис. 1-в), где QEMU играет роль внешнего контейнера.

Многие ОС не требуют вложенной виртуализации и потому обходятся без внутреннего эмулятора. В нашем случае это настольные Windows версий 1.x, 2.x, 3.x и 95, GEM от Digital Research, GEOS от Berkeley Softworks, а также ряд мобильных ОС: Pen Windows, Maemo,

Android, WebOS (не в последнюю очередь благодаря тому, что QEMU часто входит в состав SDK).

Предназначенная для свободного распространения часть обзора содержит ОС, являющиеся свободным ПО (изначально, по причине крайнего устаревания, либо будучи свободным клоном). Это различные графические оболочки Unix и Linux для настольных и мобильных компьютеров, а также GEM, Amiga, RiscOS, HaikuOS.

Представляемый материал все еще находится на стадии наполнения; на данный момент в обзоре пропущена часть объектов, играющих важную роль в истории развития графических ОС. Главным образом это новые версии ОС от Microsoft и Apple. Кроме того, DOS-оболочка Visi On, IBM OS/2 всех версий и NeXTSTEP проявили несовместимость с актуальными версиями QEMU. В качестве замены NeXTSTEP в данный момент использован Linux-аналог GNUStep. Проблема с запуском Visi On и OS/2 может быть решена использованием Bochs или VirtualBox, что нежелательно как с точки зрения использования ресурсов, так и в плане портативности информационных материалов.

Число сыгравших важную роль в истории GUI графических оболочек, для которых не существует доступных версий, оказалось весьма невелико: на текущий момент таковыми можно признать многооконный интерфейс Smalltalk конца 70-х, Xerox Star Document Processor, а также мобильные системы PenPoint OS и IBM Simon.

В схеме вложенной виртуализации можно заметить еще один компонент — сервисную гостевую ОС, которая используется для запуска эмулятора. В каждом случае на ее выбор влияли требование минимального потребления памяти, возможность использования циклов бездействия процессора, а также поддержка шины USB для возможности эмуляции позиционирования в абсолютных координатах. Последнее требование важно для комфортного управления мышью в виртуальной машине [1]. В качестве сервисных гостевых ОС помимо нескольких версий Linux использованы FreeDOS и ReactOS. В отношении ReactOS можно дополнительно заметить, что она идеально отвечает всем трем требованиям, и таким образом в нашем собственном опыте это первый случай ее удачного применения.

Литература

- [1] *Костюк Д. А.* Особенности использования виртуализованных окружений, внедренных в презентационные материалы // Восьмая конференция «Свободное программное обеспечение высшей школе»: тез. докл. / Переславль, 26–27 января 2012 года. М.: Альт Линукс, 2012. — С. 83–86.
- [2] *Костюк Д. А., Дереченник С. С.* Построение прозрачных виртуализованных окружений для изоляции уязвимых программных систем // Комплексная защита информации: матер. XVI научно-практич. конф., Гродно, 17–20 мая 2011 г. Гродно, 2011. — С. 209–212.