

Построение документации с живыми иллюстрациями на основе виртуальных машин

Иван Волчецкий, Дмитрий Костюк, Павел Луцюк,
Юрий Сойко

Брестский государственный технический университет

Цель проекта

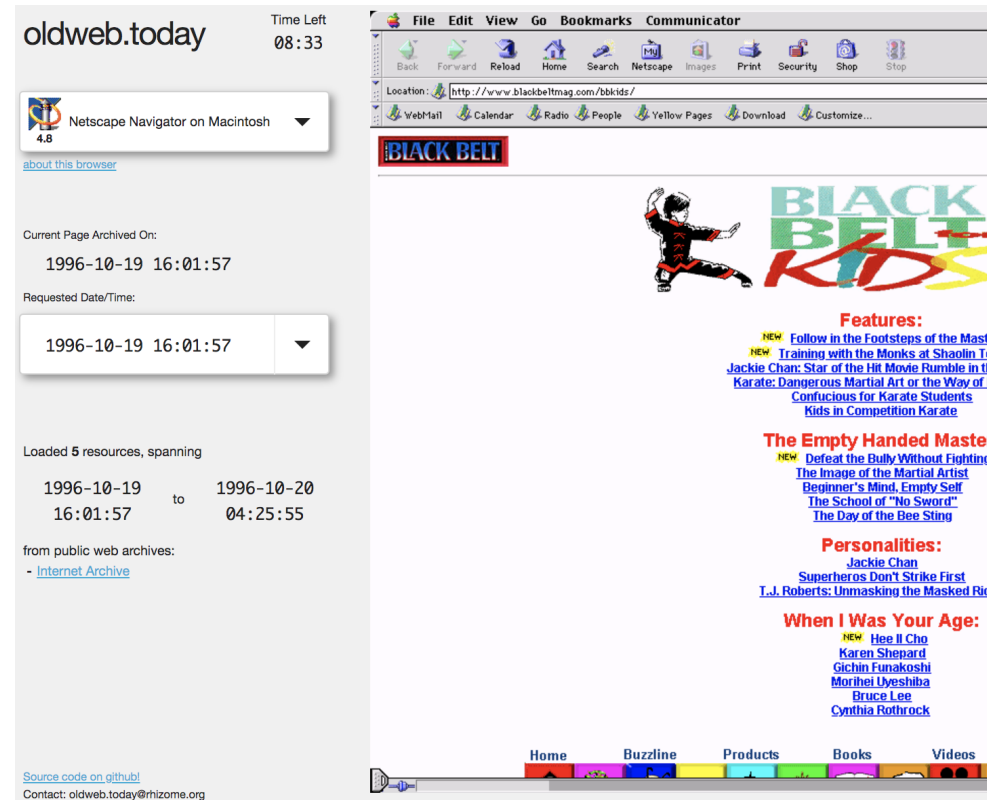
- Мультимедийные вставки — не интерактивны
 - Картинки показывают отдельные экраны ПО
 - Видео экрана передает впечатление от динамики
 - Впечатления от интерактивного взаимодействия через документацию не передаются
- Переключение с документации на ПО
 - Требуется работы с двумя объектами
 - Постоянные переключения сбивают фокус внимания

Интеграция ПО с документацией

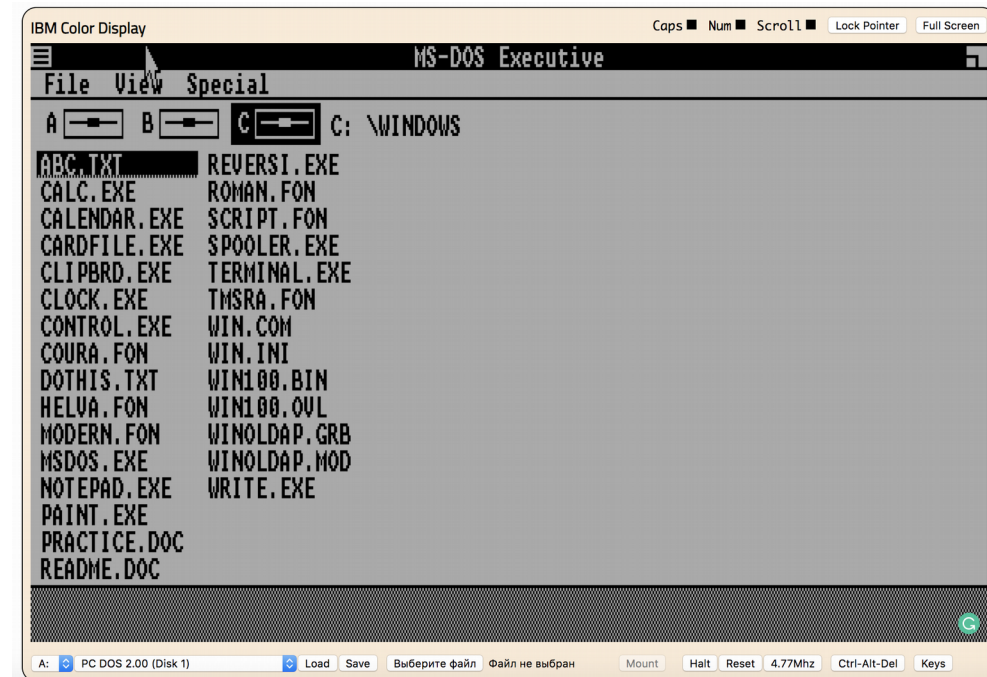
- Чаще реализовано в виде встроенной справки
 - Хорошо подходит для справочных материалов, и плохо для ознакомительных, рекламных, и др.
 - Документация становится частью программы, а не наоборот
 - Нельзя включить в один «документ» несколько разных программ
 - Очень слабая универсальность технологии разработки такой документации
- ПО встроенное в документ
 - Встречается крайне редко, в основном для ПО изначально предназначенного для встраивания
 - Редактор формул, диаграмм и т.д.

Примеры применения виртуальных машин

- <https://oldweb.today> - контейнерная виртуализация
- Старинные веб-браузеры со снапшотами старинных сайтов из веб-архива



- <https://pcjs.org>
 - эмулятор IBM PC на JavaScript
 - Демо нескольких десятков старинных ОС



Недостатки

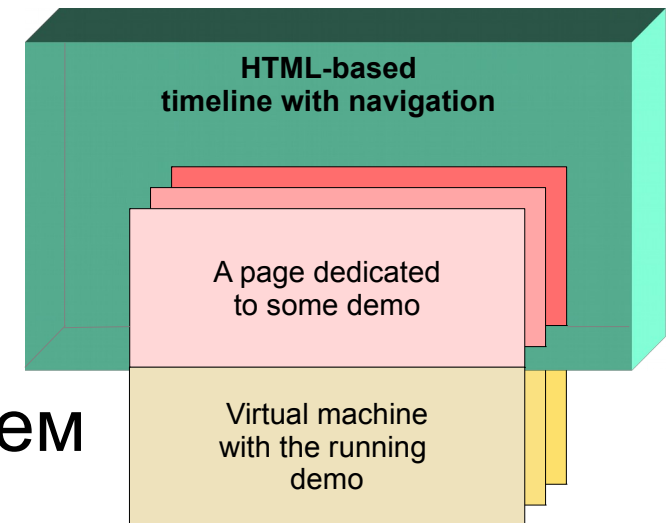
- Исключительно веб
- Или отдельные проекты вместо универсального фреймворка, или фреймворк, но узкоспециализированный

Наши собственные попытки

- Универсальный веб-фреймворк для демо на основе QEMU (<https://ostimeline.org>)
- Использование виртуальной машины в качестве «мультимедиа» через GStreamer

Ostimeline — виртуальные машины и HTML

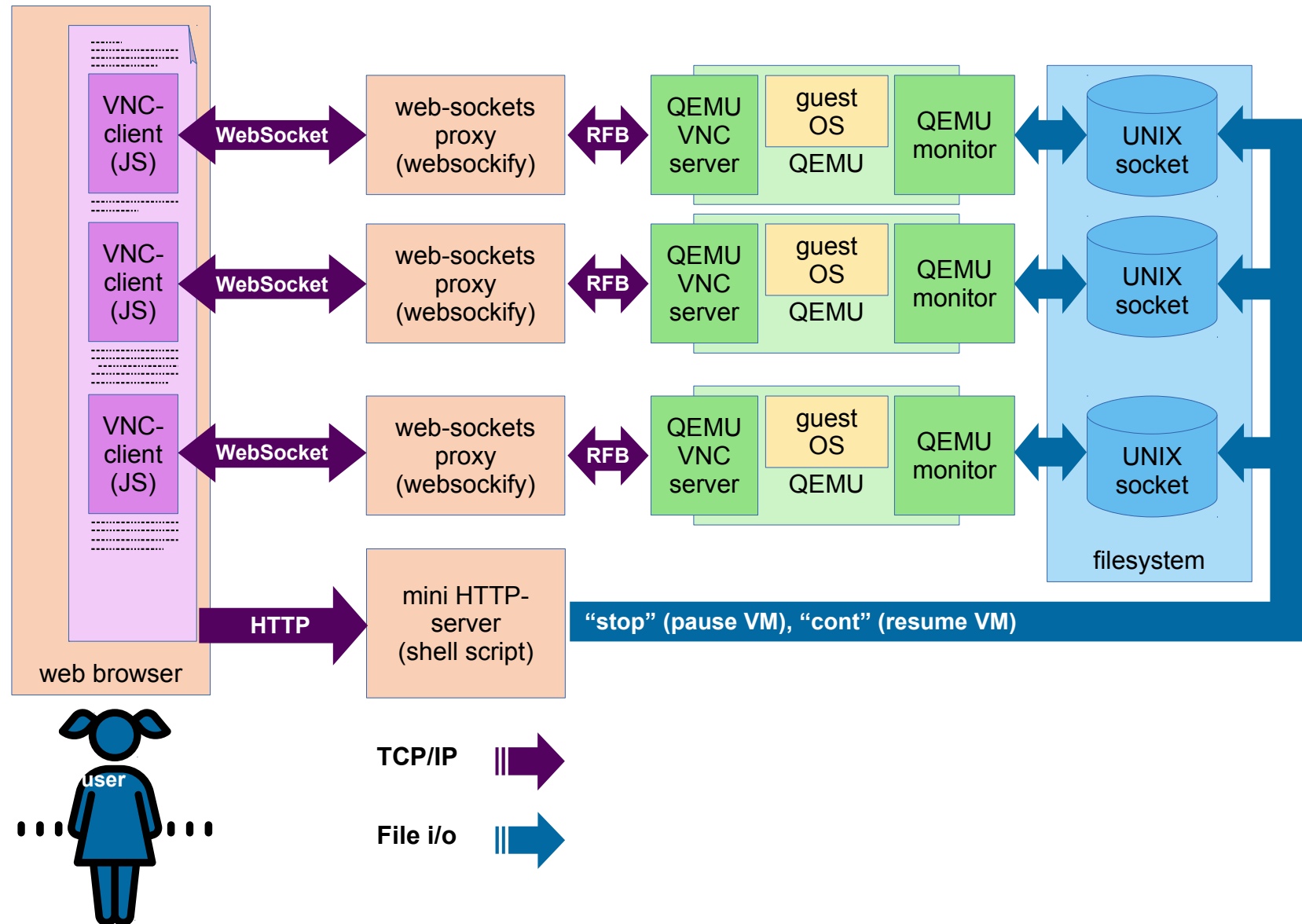
- Объединение в HTML-документе фреймов удаленного доступа к виртуальной машине
 - Изначально создавалось для задачи демонстрации истории графических операционных систем
- Каждый документ содержит:
 - описание особенностей ОС
 - ее живые иллюстрации в виде экранов виртуальных машин



Реализации

- ОС настольных компьютеров
 - около 40 виртуальных машин в документе
- ОС портативных устройств
 - около 30 виртуальных машин в документе
- История виджет-тулкитов
 - около 30 слайдов с примерами исходных кодов, 20+ виртуальных машин
- История текстовых процессоров
 - Still a work-in-progress, около 20 программных пакетов

Архитектура osthimeline



Техническая инфраструктура

- виртуальная машина QEMU
 - qemu-system-i386, qemu-system-x86_64, qemu-system-arm, qemu-powerpc;
- VNC-клиент noVNC на JavaScript и HTML5;
- websockify (прокси)
- примитивный http-сервер чтобы обрабатывать запросы на приостановку/запуск виртуальных машин по событию смены страницы пользователем
 - На основе утилиты socat (аналога netcat) и shell-скрипта
- JavaScript-фреймворк TimelineSetter для навигации по хронологии

Проблемы контента

- Распространение виртуальных образов
- Поддержка архитектур в QEMU
- Позиционирование курсора
- Совместимость снапшотов

Проблема распространения

Комплект файлов общим весом ~30-50Gb:

- СПО
- freeware
- abandonware
- Объекты, которые можно запускать, купив лицензию на вторичном рынке
- MacOS X («легитимна» только на маковском железе)

Так как мы не можем распространять весь набор...

...то он должен уметь собирать хронологию
из найденных у пользователя фрагментов :)

Реализация:

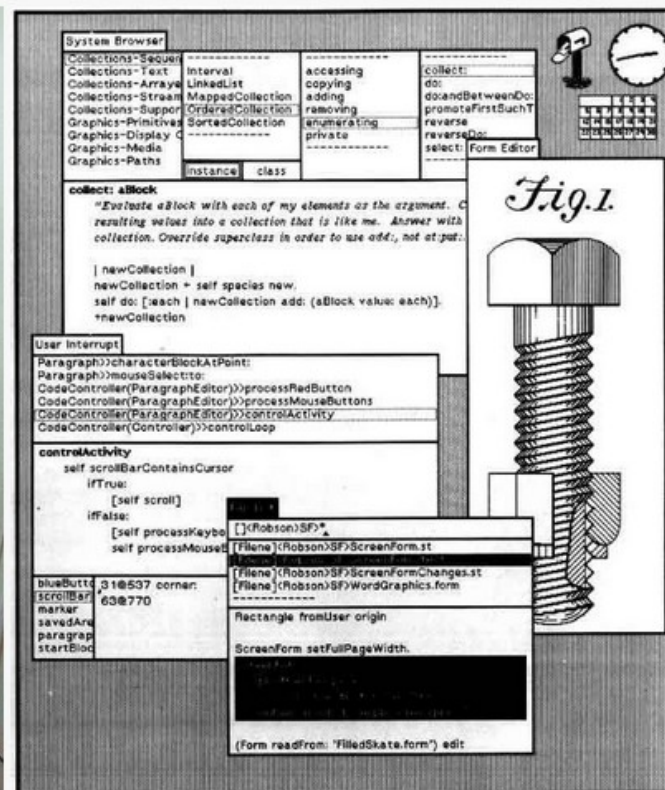
- Стартовый скрипт проводит перекомплектацию
 - находит скрипты запуска виртуальных машин
 - передает им при запуске номера портов
 - находит HTML-страницы с контентом и вписывает в них те же номера портов
 - включает исправленные страницы в хронологию

Проблемы инфраструктуры

- Совместимость снапшотов между версиями QEMU
 - Частично решается явным заданием архитектуры при вызове QEMU (нечто типа *-M pc-i440fx-1.4*)
 - Отсутствует между стандартным вызовом и вызовом с параметром *-enable_kvm*
- Унификация
 - Для использования по локальной сети используется опциональная замена *localhost* реальным адресом в обращениях к VNC-серверу

1974 - Интерфейс Smalltalk для Xerox Alto

Smalltalk - первый объектно-ориентированный язык программирования - будучи реализован для компьютеров Xerox Alto, включал также собственную среду разработки и собственный многооконный GUI для программ.



Новая веб-площадка проекта — <https://ostimeline.org>

Ostimeline

News

About

Downloads

Projects

Login

Wctablet

Gstreamer

Legacy OS mouse integration

Problem description

Running old-time OS in emulator often comes with a problem of non-coinciding cursors. This problem arises from the availability of two different types of cursor positioning devices: ones with relative coordinates (computer mice) and ones with absolute coordinates (tablets and touchscreens). Relative positioning devices provide operating system with the vector of the cursor movement instead of it's new coordinates, and as a result, different cursor acceleration formulas are breaking coincidence of host and guest systems cursor.

Mainstream visualization systems (including QEMU) can emulate as relative pointing devices (PS/2 and serial mice), so an absolute pointing device (USB Wacom tablet in case of QEMU). Mode with absolute coordinates allows host cursor to control guest OS and is called "mouse integration mode" in desktop virtualization systems. But mouse integration is available only if guest OS either has special driver from the virtual machine vendor, or supports USB tablet. In all other situations desktop virtual machines use "mouse lock mode," when host cursor is hidden until some dedicated hotkey is pressed, and user can interact with the guest system only. This mode is not only less convenient for operation, but it is usable for the web access to the virtualized system. So projects providing web demonstration of old operating systems have to show user two cursors moving with different speed:



Solution

Partners



2) QEMU как плагин фреймворка GStreamer

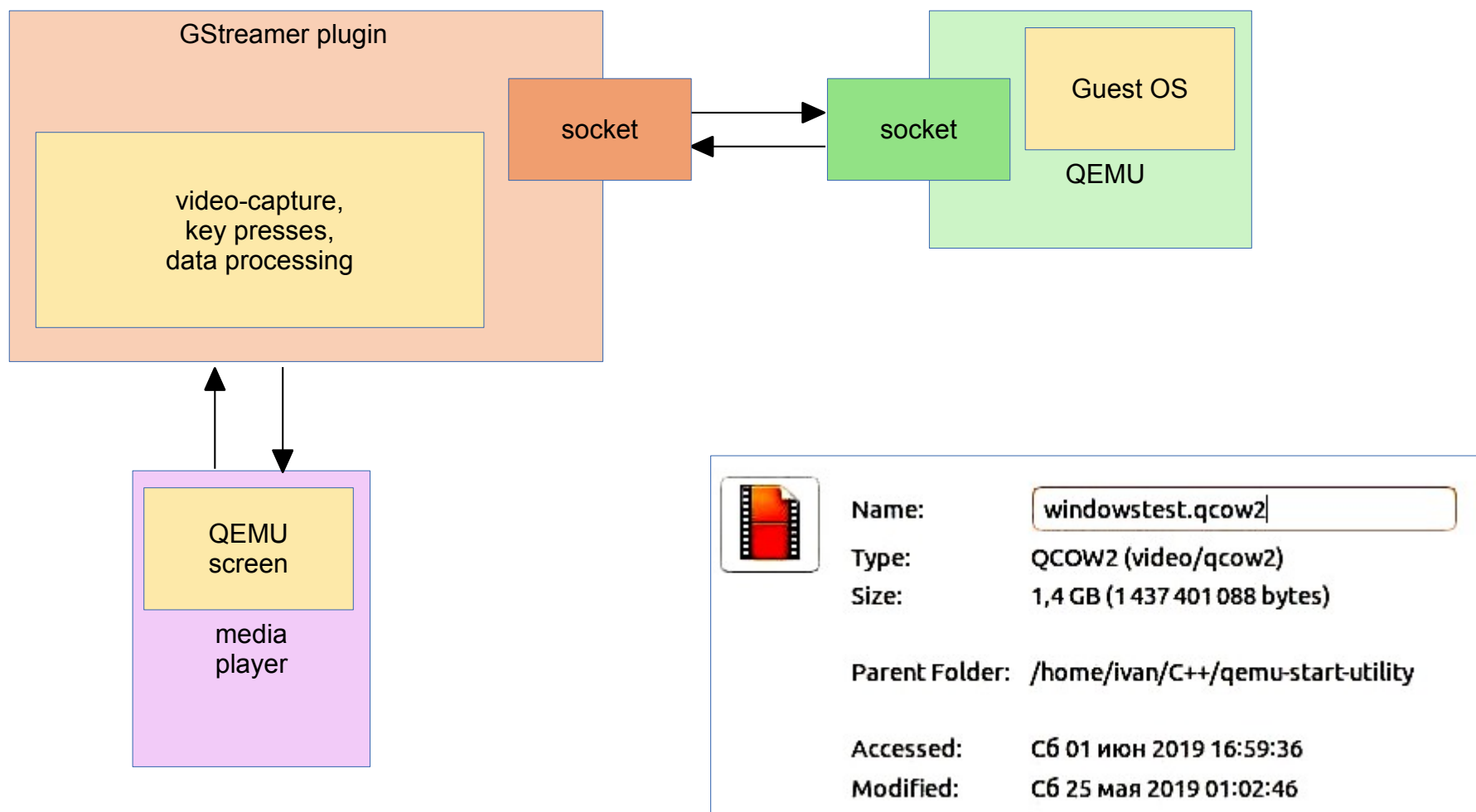
Недостаток решения на HTML

- Слишком много файлов
 - Сложна в развёртывании
 - Внешние зависимости
 - Скачивание и компиляция некоторых элементов
 - Сложна в переносе
 - Из-за предыдущего пункта, это та еще задача

Решение на основе GStreamer

- Образ виртуальной машины - «мультимедийный файл»
- Его проигрывает «кодек» с помощью QEMU
 - Кодеки тоже умеют обрабатывать пользовательский ввод — например, DVD-меню
 - GStreamer поддерживает плагины и хорошо документирован
- Действия плагина:
 - вывод изображения экрана гостевой ОС в медиа-плеер
 - подключение к QEMU через сокет;
 - Регистрация события (движения мыши и нажатия клавиш) пользователя и передача их в монитор QEMU

Решение на основе GStreamer



Как хранить строку запуска виртуальной машины?

- Мультимедийный файл — это QCOW-образ
- Строка запуска должна лежать в образе
 - Снапшоты там все равно уже лежат
- Для этого было разработано нестандартное расширение QCOW-формата...
 - оно хранится в первом кластере файла образа в области расширений, сразу за заголовком
- ... и несложная утилита запуска, читающая эти данные

Расширение заголовка QCOW-файла — формат

- Байт 0 - 3: Тип экстеншна:
 - 0x00000000 — Конец области расширений
 - 0xE2792ACA — Имя бэкинг-файла
 - 0x6803f857 — Таблица поддерживаемых опций
 - 0x23852875 — Расширение битовых карт
 - Все остальное - Нестандартное расширение, игнорируется QEMU
 -
- 4 - 7: Длина области данных расширения
 -
- 8 - n: Данные расширения
 -
- n - m: Выравнивание

Что пока не сделано

- Не решена проблема поддержки формата стандартными медиа-плеерами
 - без их перекомпиляции
- Не реализовано включение файлов QCOW в мультимедийные контейнеры
 - mkv, avi, ...
 - требуется доработка утилиты запуска для игнорирования «обёртки»