

DOI: 10.51790/2712-9942-2021-2-2-8

ОПЫТ СОВМЕСТНОЙ РАЗРАБОТКИ СЛОЖНЫХ ПРОГРАММНО-АППАРАТНЫХ СИСТЕМ ОБРАБОТКИ БОЛЬШИХ ПОТОКОВ ДАННЫХ

М. М. Лаврентьев^{1,a}, М. Ю. Шадрин^{1,2}, И. Г. Таранцев^{1,2}

¹ Институт автоматики и электрометрии СО РАН, Новосибирск, Российская Федерация

² ООО «СофтЛаб-НСК», Новосибирск, Российская Федерация

^a lavrentiev@iae.nsk.su

Аннотация: в работе рассматриваются вопросы построения простых и надежных решений для обработки непрерывных потоков данных на обычных операционных системах, которые не являются системами реального времени. Результаты получены в ходе плодотворного сотрудничества между Институтом автоматики и электрометрии СО РАН и ООО «СофтЛаб-НСК». Предлагаются необходимые и достаточные программно-аппаратные решения, объясняются принципиальные требования к функциональности программных и аппаратных компонент предлагаемых решений. Также в статье предлагается метод разделения программных компонент, обеспечивающий высокую надежность результирующей системы.

Ключевые слова: потоки данных, непрерывная обработка.

Для цитирования: Лаврентьев М. М., Шадрин М. Ю., Таранцев И. Г. Опыт совместной разработки сложных программно-аппаратных систем обработки больших потоков данных. *Успехи кибернетики*. 2021;2(2):90–95. DOI: 10.51790/2712-9942-2021-2-2-8.

COLLABORATIVE DEVELOPMENT OF COMPLEX SOFTWARE AND HARDWARE SYSTEMS FOR HIGH DATA RATE PROCESSING

Mikhail M. Lavrentiev^{1,a}, Mikhail Y. Shadrin^{1,2}, Igor G. Tarancev^{1,2}

¹ Institute of Automation and Electrometry, Siberian Branch, Russian Academy of Sciences, Novosibirsk, Russian Federation

² SoftLab-NSK, OOO, Novosibirsk, Russian Federation

^a lavrentiev@iae.nsk.su

Abstract: this study covers the development of simple and robust solutions for processing continuous data flows on conventional (not real-time) operating systems. It is collaborative research by the Institute of Automation and Electrometry and SoftLab-NSK. The necessary and sufficient software and hardware solutions are proposed, the basic requirements for the functionality of the software and hardware components of the proposed solutions are explained. This paper also proposes a software components separation method that ensures the high reliability of the resulting system.

Keywords: high data rate, continuous data processing.

Cite this article: Lavrentiev M. M., Shadrin M. Y., Tarancev I. G. Collaborative Development of Complex Software and Hardware Systems for High Data Rate Processing. *Russian Journal of Cybernetics*. 2021;2(2):90–95. DOI: 10.51790/2712-9942-2021-2-2-8.

Начало сотрудничества

В далеком 1991 году сотрудниками лаборатории машинной графики ИАиЭ СО РАН была образована компания «СофтЛаб-НСК». Первым продуктом компании стало устройство «Видеояма», обеспечивающее запись синтезированной компьютерной графики на видеомаягнитофоны. Оно было создано на базе разработанного в лаборатории тренажера для обучения головок наведения крылатых ракет в рамках конверсии военных разработок в гражданскую промышленность. В то время алгоритм JPEG только создавался, поэтому для устройства в рамках института был разработан оригинальный алгоритм сжатия синтезированных изображений [1, 2]. Компания же выпустила на рынок соответствующий продукт.

Опыт эксплуатации первых устройств показал необходимость резкого удешевления их стоимости. Тогда-то и родилась идея использования персонального компьютера как основы для создания

системы воспроизведения синтезированной графики. В 1993 году была совместно разработана и вышла в свет плата расширения в персональный компьютер IBM-PC, который можно было свободно купить и который уже имели компании, работающие в области 3D-графики. Программное обеспечение обеспечивало воспроизведение синтезированного фрагмента ролика длиной до нескольких секунд в виде стандартного телевизионного сигнала (ПАЛ, СЕКАМ, Y/C, YUV или RGB). В 1996 году вышла вторая плата — «Форвард», на которой использовался аппаратный кодек JPEG, что позволило на порядок понизить требования к передаваемым потокам данных и обеспечило непрерывный режим работы системы. Решения на платах «Форвард» могли воспроизводить ролики непрерывно неограниченное время в режиме 24/7/365 — 24 часа в сутки, 7 дней в неделю, 365 дней в году (раз в год нужно выключить и обслужить ПК, чтобы он надежно работал весь следующий год). «Форварды» разошлись не только по всей России, но по всему миру.



Рис. 1. Плата «Форвард» с комплектом программного обеспечения

Компания «СофтЛаб-НСК» и лаборатория института активно сотрудничают: лаборатория обеспечивает исследование и разработку новых технологий, а компания отвечает за их внедрение и создание коммерческих продуктов. Основной причиной успеха стало грамотное разделение работы между программной и аппаратной компонентами системы [3]. Решения «Форвард» базировались на операционной системе Windows. В те времена именно на Windows работали большинство компаний, занимающихся компьютерной графикой. Топовые системы компании «Silicon Graphics» были очень дороги для российских потребителей. «Apple» пришли в видеомонтаж слишком поздно. Системы на базе UNIX всегда были слишком сложными для творческих работников (художников, дизайнеров). На российском рынке компьютерной графики тогда царил 3DStudio Max под Windows. Операционная система Windows не является системой реального времени. Задержка выполнения любой программы может достигать многих сотен миллисекунд или даже секунд. Тем не менее для пользователя эти задержки являются приемлемыми, и пришлось разрабатывать решение именно под Windows. Телевизионный сигнал жестко таймирован — система обязана каждые 40 миллисекунд генерировать очередное изображение. Эта задача была возложена на плату ввода-вывода. Плата использует внутреннюю память для буферизации очередного изображения и занимается формированием телевизионного сигнала в соответствии с требованиями стандарта. По DMA (Direct Memory Access) плата самостоятельно копирует данные из памяти компьютера во внутренний буфер и, наоборот, из внутреннего буфера — в память компьютера (входные и выходные потоки видео). Работа обычных программ на компьютере не может помешать работе DMA, поэтому данные успевают скопироваться за время кадра. Также плата каждые 40 миллисекунд инициирует через аппаратные прерывания вызов программного обеспечения — драйвера платы. Хотя обработка прерывания может задерживаться на какое-то время, практически всегда это время не превышает десятка миллисекунд. Получив управление, драйвер передает в плату информацию о том, откуда нужно копировать следующий блок данных.

Плата обеспечивает буферизацию данных на уровне нескольких кадров: пока плата выводит на выход один кадр и копирует внутрь себя следующий кадр, драйвер успевает дать ей задание еще на кадр вперед. Однако из опыта эксплуатации известно, что операционная система может «зависнуть»

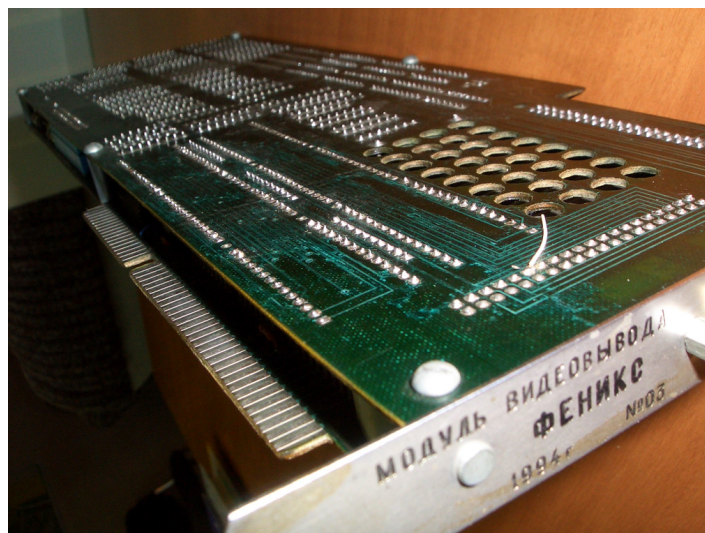


Рис. 2. «Феникс» — первая плата расширения в персональный компьютер для вывода видео

на десятки кадров, вплоть до секунды. Поэтому следующий уровень буферизации реализован на программном уровне. Драйвер платы выполняет роль микшера/коммутатора. По запросу программы он отводит в оперативной памяти компьютера большой буфер (на несколько десятков кадров) и организует на этом буфере очередь. Программа генерирует последовательность изображений и записывает их в очередь, следя за тем, чтобы очередь была постоянно заполнена. Драйвер в прерывании удаляет из очереди уже показанные изображения и прописывает в плату информацию о следующем изображении. Как правило, приложение создает пару очередей: одна очередь используется для воспроизведения текущего списка роликов, а вторая очередь заполнена первыми кадрами альтернативного списка роликов. В любой момент времени оператор может нажать кнопку старта альтернативного списка, тогда программа даст команду драйверу и он переключится с одного буфера на другой. Поскольку данные в буфере уже заполнены, переключение происходит максимально быстро — буквально в следующем кадре. Задержка переключения соответствует привычным для пользователя задержкам и не превышает времени реакции человека. Таким образом, аппаратная буферизация гарантирует стабильность выдачи данных на уровне десятков миллисекунд, а программная буферизация в оперативной памяти компьютера — гарантирует стабильность работы на несколько секунд вперед. Эта технология успешно работает и сейчас.

За 30 лет сотрудниками компании «СофтЛаб-НСК» и лаборатории № 13 ИАиЭ СО РАН было разработано много разных решений, ориентированных на обработку больших потоков данных в темпе поступления этих данных. Для большинства из них технология буферизации данных на очередях в оперативной памяти компьютера является ключевой. Вот некоторые из них:

- «TitleEngine» — технология генерации и совмещения многослойных полупрозрачных титров, позволяющая совмещать гарантированную плавность движения с интерактивностью управления (в любой момент можно включить или выключить любой слой титров) [4, 5, 6];
- «PostPlay» — технология буферизованной записи сжатых видео- и звуковых данных на жесткие диски персонального компьютера, позволяющая минимизировать нагрузку на диски и получить максимальную скорость чтения/записи данных на диск при многоканальной записи/чтении;
- «AutoDetect» — технология анализа потока видео/звука для детектирования в нем записанных ранее фрагментов видео/звука. При совпадении входного видео/звука с одним из фрагментов система генерирует управляющее событие, которое может, например, начинать с точностью до кадра воспроизведение списка роликов.

Отметим, что разработанное для удобного воспроизведения видеоповторов программно-аппаратное решение нашло свое применение в ФГБУ «Научно-исследовательский испытательный центр подготовки космонавтов имени Ю.А. Гагарина». Как институтом, так и компанией выполнено несколько проектов, поставленные системы визуального контроля эксплуатируются и получили высокую оценку.



Рис. 3. Решение для визуального контроля процесса тренировок космонавтов в НИИ ЦПК имени Ю. А. Гагарина

Проблема надежности

Другой проблемой создания сложных программно-аппаратных систем для обработки больших потоков данных является проблема надежности [7]. А именно, постепенно, с усложнением систем обработки, приходится использовать программное обеспечение (библиотеки) от сторонних производителей. Если программный код собственной разработки можно отладить и быть уверенным в его высокой надежности, то внешние библиотеки выступают в роли «черного ящика». К сожалению, такие технологии Windows, как DirectShow, славятся своей нестабильностью. Как создать надежную систему из ненадежных компонент?

Сотрудники лаборатории ИАиЭ СО РАН предложили решение в ответе на вопрос о максимально допустимом времени отказа системы, то есть о том, как долго системе можно не работать. Если время отказа не должно превышать 100 миллисекунд, то в решениях компании «СофтЛаб-НСК» рекомендуется использовать горячее резервирование и специальное устройство «watch dog». Горячее резервирование означает полное дублирование системы вторым компьютером с идентичным набором «железа» и софта. Выходы обеих систем поступают на коммутатор резерва, который управляется устройством «watch dog». Логика управления устройством «watch dog» следующая. Пользовательское приложение регулярно сообщает драйверу, что у него все будет хорошо еще некоторое время. Драйвер в каждом прерывании сравнивает это время с текущим и оценивает глубину заполненности очереди на воспроизведение. Если данные в очереди есть и приложение недавно рапортовало о хорошем статусе, то драйвер посылает команду на устройство «watch dog», что все хорошо. Если в течение пары кадров на устройство «watch dog» не приходила команда, то оно автоматически переходит в режим «авария» и переключает коммутатор резерва. Традиционно устройство «watch dog» подключают к основному компьютеру, поэтому переключение происходит при любом отказе: при отключении питания, при отказе платы ввода-вывода, при «зависании» компьютера, при «зависании» пользовательских программ. Для упрощения работы оператора и уменьшения числа человеческих ошибок пользовательские приложения компании «СофтЛаб-НСК» поддерживают режим «зеркалирования», когда действия оператора в одном приложении автоматически пересылаются в приложение на зеркалируемом компьютере. То есть оператору не нужно повторять действия на обоих компьютерах.

Если же время отказа может быть достаточно большим, в пределах десятков секунд или минут, то в решениях компании достаточно использовать разделение системы на отдельные программные модули, исполняющиеся в независимых процессах. При «взрыве» любого процесса специальная система

автоматически запускает новый процесс вместо выбывшего из работы. Такая модульность системы на уровне отдельных процессов, конечно же, порождает дополнительные трудности, связанные не только с управлением, но и с передачей больших потоков данных между процессами. Для этого опять же необходимо использовать буферизацию данных в оперативной памяти компьютера. При этом решения компании избегают лишних копирований данных: модуль «поставщик данных» заполняет изображение в памяти буфера, а модуль «приемник данных» берет данные для анализа прямо из памяти буфера.

Также разделение на процессы успешно решает проблему надежности работы системы в целом при непродолжительной перегрузке. Задавая разные приоритеты разным программным модулям, можно управлять временем работы каждого модуля. При полной загрузке процессора в первую очередь будут работать только важные модули с высочайшим приоритетом. К сожалению, в обычных операционных системах нет аналогичного управления приоритетом выполнения файловых операций. Поэтому в решениях компании используется собственный модуль приоритизации запросов к файловой системе. Это решает проблему при следовании рекомендованному рабочему процессу, но если пользователь запустит стороннее приложение, то система в целом может не успеть своевременно отработать все запросы на чтение/запись. Технология «PostPlay» как раз гарантирует своевременность записи и чтения данных.

Заключение

В настоящее время компания «СофтЛаб-НСК» выпускает широкий спектр плат ввода-вывода, включая платы для работы с телевизионным сигналом с разрешением UHD 4K (2840x2160 60 к/с). По-прежнему созданием алгоритмов обработки потоков данных занимаются сотрудники ИАиЭ СО РАН, а их встраиванием в рыночные продукты – компания. В современных платах все больший акцент делается на программном обеспечении: в них не используются аппаратные кодеры видео, поскольку вся обработка, включая кодирование и декодирование видео, выполняется стандартными средствами современного персонального компьютера. Решения компании мигрируют на операционную систему Linux, в первую очередь, это касается решений для современного цифрового телевидения. Увеличение потоков данных, передаваемых в современном телевидении, ставит перед сотрудниками лаборатории новые задачи, которые они успешно решают совместно с сотрудниками компании «СофтЛаб-НСК».

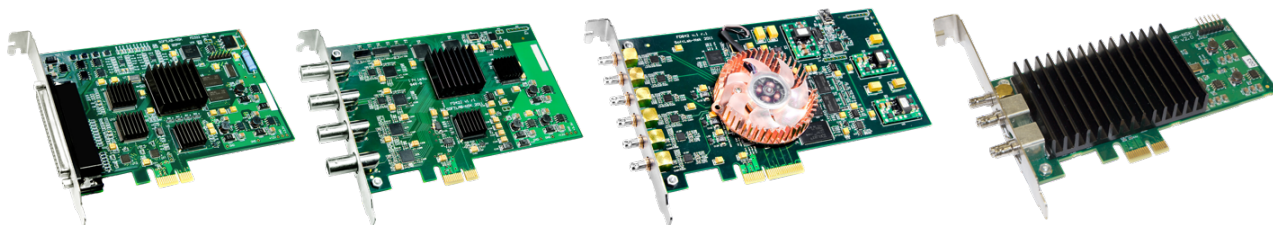


Рис. 4. Линейка современных плат компании «СофтЛаб-НСК»

ЛИТЕРАТУРА

1. Арсенин И. М., Морозов Б. Б., Роговой И. В., Таранцев И. Г., Тарасов Ю. В., Токарев А. С. Сжатие синтезированных изображений для систем реального времени. *Автометрия*. 1994;6:78–83.
2. Таранцев И. Г. Оптимизированный алгоритм сжатия синтезированных изображений для систем реального времени. *Автометрия*. 1997;2:70–74.
3. Арсенин И. М., Черепанов В. В. Методы управления с точностью до кадра параметрами воспроизведения мультимедиа-данных в ОС Windows. *Наукоемкое программное обеспечение*: семинар, 10–11 июля 2003 года, Новосибирск.
4. Арсенин И. М., Морозов Б. Б., Садов В. Г., Таранцев И. Г. Метод построения изображений в системах генерации титров. *Автометрия*. 1997;2:75–82.
5. Арсенин И. М., Таранцев И. Г. Ядро системы генерации многослойных динамических титров. *Графикон-2006*: труды 16-й Междунар. конф. по компьютерной графике и ее приложениям, 1–5 июля 2006 года, Новосибирск.

6. Гагарин А. А., Шишкин Д. Ю., Таранцев И. Г. Архитектура генераторов изображений для системы генерации многослойных динамических титров. *Графикон-2006*: труды 16-й Междунар. конф. по компьютерной графике и ее приложениям, 1–5 июля 2006 года, Новосибирск.
7. Арсенин И. М., Комаров А. С., Таранцев И. Г. Способы повышения надежности приложений при использовании технологии DirectShow. *Ершовская конференция по информатике — 2011, Научное программное обеспечение*: семинар, 27 июня – 1 июля 2011 года, Новосибирск.