

# Незаметный слуга

В предыдущем номере мы рассматривали udev и то, как он решает свою задачу — именование устройств, однако в наш динамичный век, когда все должно работать по мановению руки, этого явно недостаточно.

Аппаратура должна работать и при этом никого не утруждать своими настройками, пристройками и подстройками: воткнули USB-флеш — вот вам содержимое, подсоединили фотоаппарат — вот вам снимки, подключили принтер — он уже должен быть доступен для печати в CUPS. Для этого необходимо нечто большее, чем udev, — нужен условный «кто-то», кто сидел бы и наблюдал за нашей аппаратурой не только на предмет появления/исчезновения, но и следил за ее состоянием, предоставлял бы информацию о классах устройств в понятном для приложений виде. Именно этим и занимается HAL.

## | Что такое HAL? |

HAL расшифровывается как Hardware Abstraction Layer, то есть прослойка абстракции от аппаратуры. Как видно из названия, его основное дело — свести пользовательские приложения с устройствами, которые находятся в системе. Для этого HAL собирает массу информации об устройствах из hotplug, udev, sysfs и своей собственной базы данных об устройствах, которую пользователь может самостоятельно изменять.

Как обстояли дела до его появления? Приложения обращались к ядру (чаще всего посредством библиотек) напрямую и получали информацию об аппаратуре оттуда. Однако ядро не знает всего, а даже когда знает, применяет удобные с точки зрения ядра абстракции — именно так наша фотокамера может оказаться всего лишь еще одним блочным устройством в /dev, мало чем отличающимся, например, от винчестера. А ведь это фотокамера, это снимки, разве не было бы приятно увидеть запущенным менеджер фотоальбома сразу после подключения камеры? Еще раз вспомним про udev — теперь наши устройства могут называться в /dev как угодно, а приложения должны

понимать, что, например, /dev/my\_mega-printer все-таки является принтером, и на нем можно и нужно печатать.

HAL решает эту проблему ведением базы данных устройств, где все они представлены в виде объектов, каждому из которых присвоен свой уникальный идентификатор. Для любого объекта существует набор свойств в виде пар «ключ-значение», эти свойства собираются с различных уровней представления устройства, но все они могут быть одинаково интересны для приложений. Именно эти пары можно посмотреть в менеджере устройств Fedora Core 3/4 (или просто воспользовавшись командой lshal).

Наверное, самый интересный для устройств ключ — capabilities, в котором хранится набор качественных характеристик устройства. Так, например, у всех принтеров в capabilities обязательно будет записано «printer», а у всех фотокамер — «camera». Тем самым приложения могут спросить HAL о том, какие устройства определенных классов (интересующих приложение) есть в системе и в какие соответствующие файлы /dev надо бить челом.

## | Принципы работы |

При этом работает HAL на удивление просто: на сегодня он регистрирует скрипты-помощники в udev и hotplug, которые поставляют информацию об изменениях в аппаратуре. Хотя готовятся изменения, которые не только сделают взаимодействие более правильным (уже не придется работать с hotplug, HAL будет просто подхватывать всю информацию, которую уже успел собрать udev) и более быстрым (для передачи информации от udev к HAL будут использоваться сокеты; тесты с udevstart показывают, что время обработки может сократиться в два раза).

Но, конечно, информацией от `udev` и `hotplug` HAL не ограничивается, иначе в чем был бы смысл! Во-первых, HAL самостоятельно исследует доступную информацию для новых устройств в `sysfs`, что заполняет еще сразу несколько свойств, после чего в дело включается мощный механизм файлов информации об устройствах (`fdi`). Путем сопоставления уже полученной разными путями информации с описанными там устройствами HAL может добавлять новые свойства: именно так для карт-ридеров определяется тип поддерживаемых карточек, а для плееров по модели из информации в `sysfs` — данные о поддерживаемых форматах файлов.

Плюс к этому HAL может выполнять активное исследование. Устройства хранения данных, таким образом, проверяются на наличие типов имеющихся файловых систем, ведь эту информацию никак не предоставляет ядро, аналогично диагностируются устройства ввода, дискеты и прочая аппаратура.

Но даже это еще не все, HAL может динамически отслеживать состояние устройств и их свойства, для чего у него имеется механизм вызова `add-ons` (добавлений-включений). Это программы, которые работают весь период существования какого-то типа устройств в системе, соответственно, они запускаются из HAL при подключении первого устройства своего типа и закрываются им же после удаления последнего подобного устройства из системы.

В поставку HAL входит несколько включений: для ACPI, карт-ридеров, а также мышей USB CSR. Они просто выполняют активный опрос устройств и изменяют свойства в случае обновления состояний.

Именно таким образом мы можем узнать о появлении карточки в карт-ридере — сам он об этом ничего не сообщает (конечно, зависит от модели), аналогично выполняется мониторинг параметров ACPI и других. Важно, что любое приложение, запрашивающее информацию у HAL, всегда получает точные и актуальные сведения.

## | Взаимодействие с приложениями |

Сам HAL ничего не навязывает пользовательским приложениям, то есть, куда что монтировать, как что отображать — это не его дело, он всего лишь предоставляет удобную в использовании базу данных устройств. Пользовательские приложения могут работать с механизмом вызовов (`callout`) HAL, для того чтобы реализовать необходимые действия.

Если вернуться к избитому примеру с USB-флеш, то после его обнаружения нам бы хотелось иметь возможность монтировать его, не упражняя пальчики набором команд в консоли. Учитывая ненавязчивость HAL, для этой простой задачи уже есть два решения.

Одно из них применяется в Fedora Core и называется `fstab-sync` — эта программа просто заносит новое устройство в `/etc/fstab`, а дальше работа происходит в стандартном режиме. Второе уходит своими корнями в Debian и зовется `pmount` — это программа, позволяющая непривилегированным пользователям монтировать разделы, не описанные в `fstab`, `pmount` устанавливается как бинарный `suid`. Так что с точки зрения безопасности здесь присутствует какая-то доля риска, хотя у

такого подхода есть удобное преимущество, заключающееся в наличии утилиты `pmount` не только для взаимодействия с HAL; к тому же постоянно изменять `fstab` тоже некрасиво.

В любом случае, здесь уже есть из чего выбрать, а все, что касается вызываемых HAL программ, находится в следующих разделах: `/etc/hal/device.d/` (добавление/удаление устройств), `/etc/hal/capability.d/` (добавление/удаление возможностей устройств), `/etc/hal/property.d/` (добавление/удаление/изменение свойств устройств).

Но этим функционал HAL не ограничивается, он может работать и активным гидом, рассказывая о появлении/исчезновении устройств всем интересующимся приложениям через шину D-BUS. Впрочем, это было бы не слишком интересно в свете наличия `udev` (но стоит помнить, что `udev` не экспортирует информацию на D-BUS, хотя так было в ранних его версиях), поэтому самое интересное заключается в том, что точно таким же способом HAL может сообщать приложениям об изменениях состояний устройств.

Кроме того, HAL доступен и через D-BUS по адресу `/org/freedesktop/Hal/Manager`, описание интерфейсов вместе с исходниками HAL находится в `doc/spec/hal-spec.html`, там же есть пример на Python, использующий этот интерфейс.

## | Новые возможности |

Все это позволяет делать то, что раньше казалось трудно выполнимым, и это легко почувствовать, используя последние версии GNOME — именно эта настольная среда сегодня явно лидирует в освоении всех прелестей работы с HAL. Его используют библиотека для доступа приложений к файловым системам `GnomeVFS`, что позволяет создавать красивый вид по адресу «`computer:///`» (заметьте, что отображаемые иконки там четко соответствуют типу накопителя), апплет `NetworkManager`, который сам разбирается в том, какое сетевое подключение необходимо использовать в конкретный момент времени (владельцы ноутбуков оценят его особенно), а также `GNOME Volume Manager`, позволяющий задавать необходимые действия при появлении разных типов накопителей.

Впрочем, потихоньку HAL проникает и в KDE. Пользователи KDE 3.4.x, возможно, уже знают про новый `ioslave media:/`, предоставляющий доступ к устройствам хранения данных. Актуальность списка устройств и возможность их беспрепятственного монтирования обеспечивается именно HAL (в связке с `pmount` или `fstab-sync`). Даже свежий K3b теперь обновляет список доступных для записи устройств, используя информацию, полученную от HAL.

Кстати говоря, в будущем HAL, возможно, возьмет на себя функцию выбора и загрузки драйверов устройств, а также сможет предупредить пользователя об отсутствии того или иного драйвера в системе (а если продумать дальнейшую интеграцию с пакетным менеджером на предмет установки, будет очень красиво). Поэтому HAL на самом деле — только лишь начало настоящего Plug&Play в GNU/Linux, по мере адаптации к нему различных настольных сред и приложений мы увидим еще много интересного, а аппаратура... просто работает — разве нужно что-то еще? |