

Евгений Зобнин

# Ионизация консоли

Долгое время местом моего обитания была обычная Linux-консоль с запущенной программой GNU/Screen. Все бы было хорошо, не возникни потребность в графическом браузере, чтении книг в популярном ныне формате djvu и других графических приложениях. И приходилось метаться между консолью и «иксами», так как ни один из оконных менеджеров не позволял достичь той эффективности работы, какая предоставлялась той самой Screen. Так и не наступил бы покой, если бы на глаза мне не попался менеджер окон Ion.

## | Почему Ion? |

Что же такое Ion и чем он так привлек мое внимание? Дело в том, что его разработчики отказались от привычной концепции оконного окружения, придуманной еще компанией Хегох и благополучно взятой на вооружение Apple, Microsoft и 99% менеджеров окон. Вместо нее в Ion используется уже известная, но не заслужившая должного внимания модель фреймов и табов (закладок). Фрейм в данном случае представляет собой область экрана (или весь экран), внутри которой может быть запущено несколько приложений-клиентов. Переключение между ними происходит при помощи табов (напоминающих по функциональности табы веб-браузеров). Понятие окна в данном случае отсутствует вообще, но можно сказать, что окно всегда занимает весь фрейм целиком.

Таким образом, рабочее пространство разбивается на несколько неперекрывающихся областей, а проблема с перетаскиванием и изменением размеров окон, зачастую выводя-

щая из себя, исчезает вовсе. Стоит заметить, что не все пользователи Ion предпочитают подобный стиль расположения фреймов. Многие обычно используют всего один фрейм, занимающий всю рабочую площадь.

Управление фреймами и табами происходит полностью с клавиатуры, поэтому немного освоившийся с сочетаниями клавиш пользователь сможет решать многие задачи буквально за доли секунды, не прибегая к помощи меню и иконок. Плюс ко всему Ion позволяет изменить практически все параметры, разместить запуск часто используемых приложений на клавиатурные комбинации, обладает модульной архитектурой, поддерживает dockapp (из WindowMaker и AfterStep), может быть расширен за счет простого скриптового языка lua, использует скины и содержит в себе еще множество других возможностей. Просуммировав все вышесказанное, мы получаем идеальный рабочий стол для людей, ценящих свое время, а не красоту интуитивно понятного интерфейса.

## Инсталляция

На данный момент существуют две текущие версии программы: стабильная вторая и разрабатываемая третья. Между собой они не совместимы, поэтому конфигурационные файлы от третьей версии не будут работать во второй, и наоборот. Помимо этого новая версия содержит в себе множество полезных функциональных дополнений и улучшений. По своему опыту могу сказать, что она уже достаточно стабильна и вполне подходит для повседневного использования. Поэтому дальнейшее повествование будет посвящено именно Ion 3. Взять его исходные коды можно с официальной страницы <http://modeemi.cs.tut.fi/~tuomov/ion>. Там же расположен архив с документацией, размер которого, что показательно, больше самой программы. Установка весьма тривиальна (./configure && make && make install), но требует наличия интерпретатора lua ([www.lua.org](http://www.lua.org)). Для тех, кто не в курсе, lua — это простой объектно-ориентированный интерпретируемый язык программирования, созданный специально для встраивания в другие приложения. Помимо Ion он используется в текстовом веб-браузере ELinks и других менее популярных приложениях.

## Первое знакомство

По окончании установки выходим из X Window, копируем файл ~/.xinitrc в укромное местечко и выполняем команду `echo "exec ion3" > ~/.xinitrc`. Эти действия направлены на то, чтобы научить команду `startx` загружать Ion вместо предыдущего оконного менеджера.

После весьма непродолжительного процесса загрузки Ion встретит нас приветственным сообщением и предложит прочитать map-страницу. От чтения документации отказываться не советую, так как стандартная настройка горячих клавиш крайне запутанна и не поддается интуитивному освоению (в следующем разделе мы это исправим).

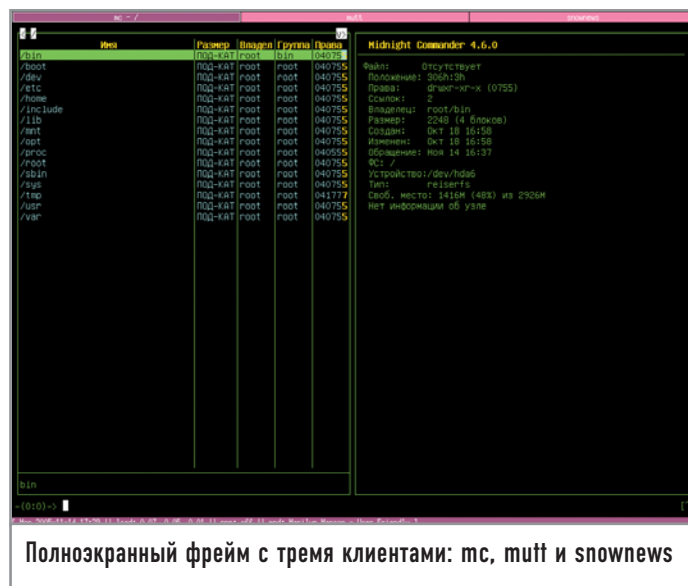
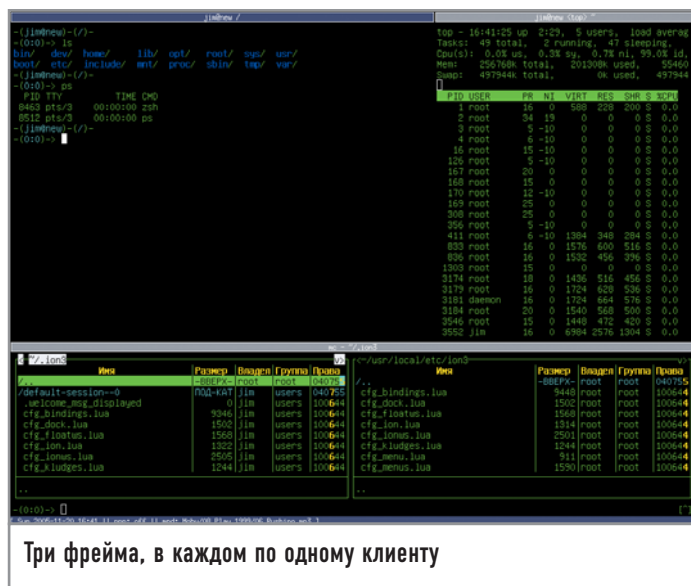
После ознакомления с документацией можно начинать работу. Для тех, кто не привык читать мануалы или не смог разобраться с терминологией, осветю основные моменты.

По умолчанию Ion запускается с одним-единственным фреймом, растянутым во весь экран. Его верхняя строка содержит имена табов, за переключение между которыми отвечают комбинации «Alt+K N» (следующий таб), «Alt+K P» (предыдущий таб) или «Alt+K X», где X — номер таба. Внизу — строка состояния, которую отрисовывает модуль `statusbar` (мы еще займемся его настройкой). Аналог кнопки «Закрыть окно», присутствующей в заголовке окна обычного оконного менеджера, — комбинация «Alt+C». К функциональным клавишам («F1–F12») привязаны различные полезные действия, например: «F1» — просмотр map-страниц, «F2» — запуск `xterm`, «F3» — запуск выбранного приложения. По клавише «F12» доступно основное меню, из которого можно запустить программы, выбрать стиль оформления или выйти из X Window.

А как же хваленый принцип разделения рабочего пространства на области? И правда, по умолчанию у нас на экране присутствует всего один фрейм, но достаточно нажать «Alt+S», экран разделится по горизонтали на две области — и в нашем распоряжении уже два фрейма; еще одно нажатие комбинации — и три. Переключение между ними — «Alt+P» (выше) и «Alt+N» (ниже). Вертикальное разбиение экрана — «Alt+K S», переключение — «Alt+Tab» (вправо), «Alt+K Tab» (влево). Поймите, но симметричные фреймы — это же совсем неудобно. Для включения режима изменения размера фрейма нажимаем «Alt+R», а затем — клавиши направления для увеличения размера или те же самые, но в связке с «Shift» для уменьшения. Да, уничтожает фрейм комбинация «Alt+K X». Не забудьте сохранить сессию («F12 → Session → Save») перед выходом из X Window, чтобы при следующем запуске Ion положение всех фреймов не изменилось.

Вполне естественно, что Ion поддерживает популярную концепцию множества рабочих столов. Вот только эта тема не так проста, потому что здесь предлагается на выбор аж четыре вида рабочих столов:

► WIonWS — стандартный рабочий стол, исповедующий концепцию табов и фреймов. Именно он используется по умолчанию и определяет облик оконного менеджера.



► WPaneWS — еще один схожий подход к организации рабочего пространства. В этом случае экран разделен на несколько областей фиксированного размера. Каждая область предназначена для запуска определенного типа приложений. Например, dockapp будет автоматически помещен в область небольших размеров, где ему и место, а веб-браузер — в основную. Причем размеры и количество областей, а также алгоритм размещения приложений полностью настраиваемы. Рекомендую поэкспериментировать, очень занятно.

► WFloatWS — специальный рабочий стол, как это ни парадоксально, со стандартным оконным интерфейсом, хотя и не лишенный табов и фреймов. Предназначен для запуска многооконных приложений, таких как GIMP и ImageMagic, управляться с которыми при помощи табов очень проблематично.

► Рабочим столом может стать одна-единственная программа! Для этого используйте комбинацию клавиш «Alt+Enter».

Чтобы получить доступ к новому рабочему столу, следует его создать, нажав «F9», а затем указать имя и тип. Переключение между столами — «Alt+X», где X — порядковый номер. Так как потребность в создании рабочего стола WIonWS возникает чаще, чем других типов, в стандартной настройке комбинация «Alt+F9» привязана к функции создания рабочего стола по умолчанию. Уничтожается рабочий стол, как и все остальное в Ion, при помощи незамысловатой комбинации «Alt+C».

Конечно же, клавиатура — не единственное средство управления. Мышь вполне корректно и предсказуемо работает и позволяет перемещаться между табами и вызывать различные контекстные меню кликом правой клавиши. Причем мышь полностью поддается процессу настройки. С другой стороны, она все-таки остается дополнительным средством управления и полезна разве что на рабочем столе типа WFloatWS.

Как уже было сказано, в нижней части рабочего стола находится строка статуса. Она очень похожа на аналогичную строку консольного Screen и предназначена для отображения различной динамически меняющейся информации. Теоретически модуль statusbar, отвечающий за отображение этой строки, можно запрограммировать на показ чего угодно.

В Интернете доступны скрипты, отображающие зарядку батареи ноутбука, состояние сетевого интерфейса, название композиции, проигрываемой XMMS, и т. д. По умолчанию в строке отображаются время, общая загрузка системы и количество писем в ящике.

Так как по своей сути статусная строка — это всего лишь строка, она не может отрисовывать графики, индикаторы и прочую графическую информацию. Эту проблему можно побороть, используя так называемый dock. Он хорошо знаком пользователям WindowMaker, AfterStep и Fluxbox (правда в этом оконном менеджере он называется Slit). Dock представляет собой небольшое окошко (обычно не имеющее рамок) и позволяет разместить в своем пространстве несколько dockapp — приложений с размером окна примерно 64x64, которые как раз и занимаются отрисовкой различных графиков (загрузка процессора, количество памяти, объем дискового пространства и т. д.). Знаменитый Gkrellm тоже может стать dockapp, если в настройках поставить галочку напротив опции «Установить тип окна dock или panel». Модуль dock в Ion размещает dock в определенном углу экрана так, чтобы он перекрывал все фреймы, независимо от выбранного типа рабочего стола. Чтобы dock не мешал работать в других приложениях, его можно спрятать, нажав «Alt+D».

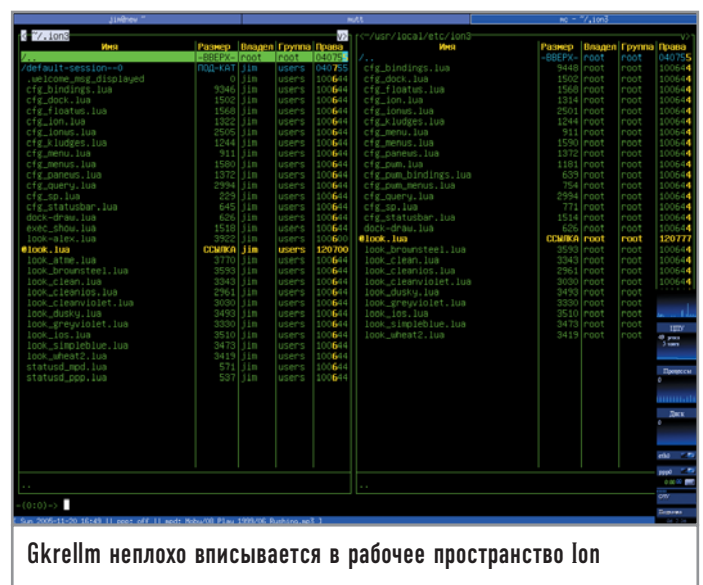
## Настройка

Повсюду в Ion можно обнаружить следы языка lua, и даже все настроечные файлы являются lua-скриптами и содержат множество вызовов процедур, объявлений функций и комментариев. Обычному пользователю, привыкшему к графическим конфигураторам, такой способ настройки может показаться безумно сложным и неоправданным. На самом деле lua — очень простой язык, позволяющий без особого труда добавить программе уникальной функциональности.

Для начала предлагаю скопировать рабочую конфигурацию из каталога /usr/local/etc/ion3 (каталог может быть иной, если во время установки был указан другой префикс) в ~/.ion3. Главный конфигурационный файл носит имя cfg\_ion.lua и включа-



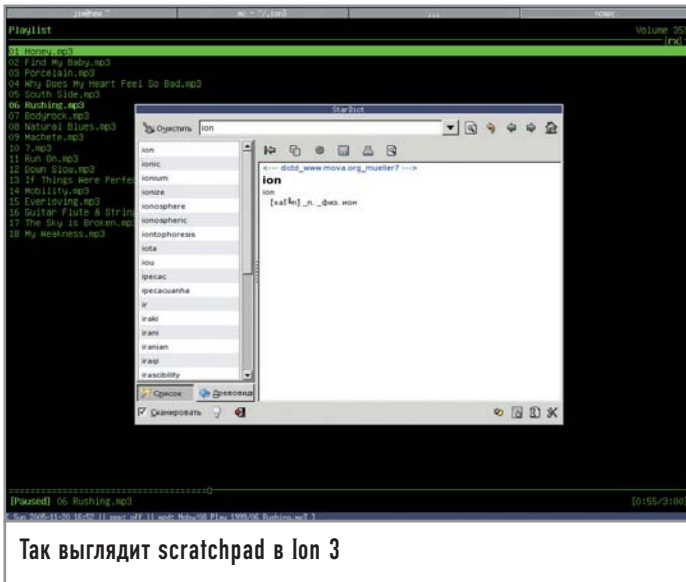
Рабочий стол в стиле Matrix



Gkrellm неплохо вписывается в рабочее пространство Ion







```
kpress("T", "ioncore.exec_on(_, 'xterm)'),
kpress("M", "ioncore.exec_on(_, 'mc)'),
kpress("E", "ioncore.exec_on(_, ':elinks)'),
})
```

Теперь по комбинации «Alt+E T» будет запускаться xterm, по «Alt+E M» — mc и т. д., можете добавить строки для запуска нужных вам программ. Кстати, знак «:» перед командой указывает на то, что ее следует запускать в новом терминале. Как видите, функция submap позволяет создавать многоуровневые комбинации, благодаря чему мы получаем огромное разнообразие оных. Например, клавиатурные комбинации можно использовать для управления аудиоплеером, запуска сетевых функций (подключение, отключение, SSH и т. д.), выдвижения лотка CD-ROM, монтирования файловых систем и многого другого.

Теперь создадим свое меню (то, которое вызывается по «F12»). Для этого опять же предусмотрен особый скрипт cfg\_menus.lua, в него мы и впишем нужные нам строки. Чтобы понять, как выглядит меню в исходном (на языке lua) виде, рассмотрим следующий пример:

```
defmenu("mainmenu", {
    submenu("programs", "appmenu"),
})
```

```
defmenu("appmenu", {
    menuentry("xterm", "ioncore.exec_on(_, 'xterm)'),
})
```

Здесь мы при помощи функции defmenu() создаем основное меню (оно всегда носит имя mainmenu) и добавляем в него подменю appmenu с именем programs. Далее по описанной выше схеме создается подменю appmenu, которое будет содержать пункт xterm, запускающий одноименную программу. Вот и все, за подсказками обращайтесь к содержимому оригинального скрипта.

Статусная строка конфигурируется в файле cfg\_statusbar.lua, и только в нем. Содержимое строки задается в переменной template в виде текста и специальных слов.

Рассмотрим пример:

```
template="[ %date || load: %load || mail: %mail_new/%mail_total ]"
```

Специальные слова %date, %load, %mail\_new и %mail\_total при выводе будут заменены на показания соответствующего счетчика, все остальное пойдет на экран в неизменном виде. Все четыре перечисленных счетчика являются встроенными. Чтобы создать свой счетчик, нужно написать специальный скрипт с именем «statusd\_имя.lua», поместить в каталог ~/.ion3 и вставить его название в переменную template (не забыв указать знак %). Написание такого скрипта я рассматривать не буду, так как это требует описания синтаксиса lua.

Ну и напоследок внесем несколько косметических изменений. Заглянем в файл cfg\_dock.lua и изменим позицию dock. Для этого запишем в переменную pos одно из четырех значений: tl (верхний левый угол), tr (верхний правый угол), bl (нижний левый угол) и br (нижний правый угол). Также полезно изменить направление роста (с какой стороны будет добавляться новый dockapp), присвоив переменной grow значение left, right, up или down. В Ion 3 есть интересный модуль scratchpad, отключенный по умолчанию. Scratchpad — это не зависимый от рабочих столов фрейм размером 640x480, появляющийся в центре экрана после нажатия комбинации клавиш «Alt+Пробел». Его удобно использовать для запуска аудиоплеера, словаря и других приложений, которые не должны присутствовать на экране постоянно. Чтобы активировать этот модуль, раскомментируйте строку dopath("mod\_sp") в скрипте cfg\_ion.lua. |

## Дополнительная информация

### Работа с диалоговыми и уведомляющими окнами

Может быть, у кого-то из читателей возник вопрос: как Ion поступает с диалоговыми и уведомляющими окошками? Ведь если такие окна будут открываться в новом табе, то идея «каждой программе по табу» пойдет коту под хвост, а работать в таком оконном менеджере станет невыносимо. Но и с этой про-

блемой программисты справились, придумав интересное решение. По определенным признакам Ion распознает окна такого типа, уменьшает их размеры до одной трети экрана и отрисовывает прямо поверх существующей картинке. Таким образом, диалоговое окно как бы вливается в основное.

