V14.02 (№6)
ISSN: 2223-6988

Облако в Linux своими руками

User And LINUX

Больше чем user

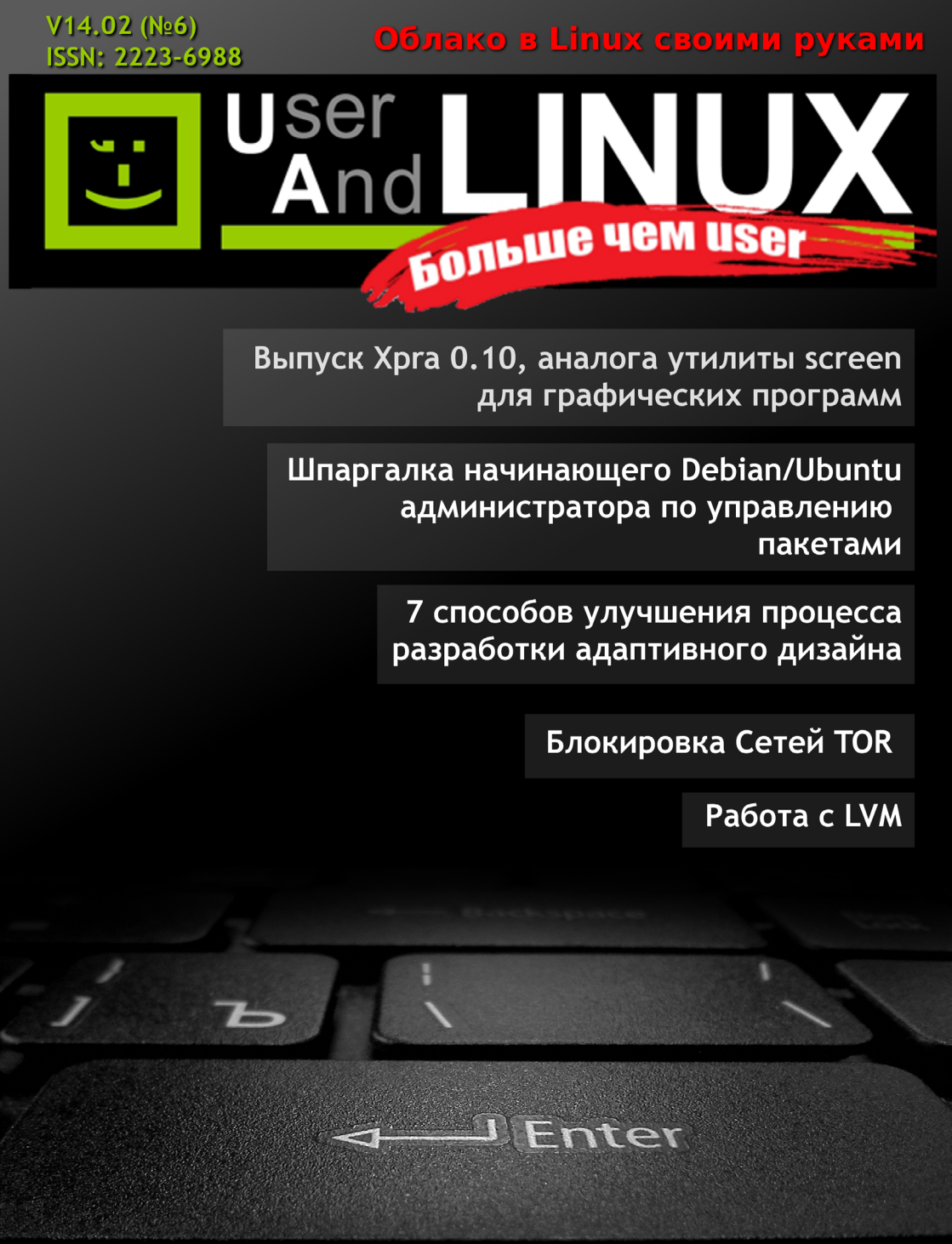
Выпуск Xpra 0.10, аналога утилиты screen для графических программ

Шпаргалка начинающего Debian/Ubuntu администратора по управлению пакетами

7 способов улучшения процесса разработки адаптивного дизайна

Блокировка Сетей TOR

Работа с LVM



Выпуск первого Forensic-дистрибутива из комплекта Ubuntu CyberPack (IRF) предназначенного для снятия дампа оперативной памяти компьютера.

UA Linux
http://ualinux.com

ubuntu BusinessPack

Операционная система, которая идеально подходит для использования на персональных компьютерах и ноутбуках. Она ориентирована на простоту использования и удобство работы.

Включена необходимая подборка программного обеспечения, которая позволяет создать удобное рабочее окружение в корпоративной среде предприятия или на домашнем компьютере.

Ubuntu Business Pack это:

- **простая установка операционной системы не требующая особых знаний;**
- **уверенность в том, что на компьютере установлено только лицензионное программное обеспечение;**
- **это низкая цена по сравнению с аналогами;**
- **создание рабочего места без дополнительных финансовых затрат. Это существенно экономит бюджет организаций;**
- **идеальное решение для перехода на Linux с Windows, если вы все еще используете windows-приложения и игры;**
- **полная поддержка в системе русского, украинского и английского языков;**
- **отсутствие необходимости затрат на антивирусную защиту.**

Программное обеспечение имеет понятный графический интерфейс и полностью совместимо с популярными форматами документов, поэтому переход не вызывает никаких проблем с переносом данных и переквалификацией сотрудников.

поддержка широкого спектра современного оборудования;
дополнительные драйвера для видео-карт, wi-fi адаптеров и принтеров;
возможность использовать Windows-драйвера для WiFi-адаптеров USB;
управление веб-камерами.

безопасность и надежная защита от вирусов;
проверка файлов на вирусы в режиме реального времени (актуально в случае запуска windows-приложений);
защита от вирусных атак системы и электронной почты;
проверка на спам.

поддержка мультимедия (аудио - видео) различных форматов (avi, divX, mp4, mkv, amr, aac, Adobe Flash и многие другие)
просмотр защищенных, зашифрованных лицензионных, двухслойных DVD и Bluray дисков

полный набор офисных компонент (тексты, таблицы, презентации) совместимых с форматами MS Office
включена поддержка импорта файлов MS Visio
поддержка различных типов архивов (RAR, ACE, ARJ и других);

поддержка windows-приложений (гарантированный запуск более 130 приложений и более 600 игр)

полноценная поддержка Java-приложений;
гарантированная работа онлайн банк-клиентов, таких как Приват24
гарантированная работа онлайн-бухгалтерии, таких как iFin.

Добрый день, уважаемые читатели!

Весь наш коллектив поздравляет вас с прошедшими Рождеством и Новым Годом и желает всего самого лучшего и приятного. Надеемся что в прошлом году мы не обманули ваших ожиданий.

Мы приветствуем вас в первом номере нашего журнала в 2014 году.

Сегодня будут материалы на различные темы — тут и создание своего собственного облачного сервиса, и веб-интерфейс для системы виртуализации, и статья об оконном менеджере DWM и многое другое.

Однако особенно хотелось бы обратить ваше внимание на статью «Получаем образ оперативной памяти». В мире существует много дистрибутивов и программ предназначенных для контроля IT- безопасности и расследованию соответствующих инцидентов. На этот раз мы рады сообщить, что подобные разработки есть и у нас в Украине. Здесь будет рассмотрен дистрибутив Ubuntu-CyberPack и, естественно, некоторые его возможности.

Оставайтесь с нами. Следующие номера будут не менее интересными.

С уважением,
Якимчук Сергей

НАД ВЫПУСКОМ РАБОТАЛИ:

Якимчук Сергей
Попов Владимир
Шарай Игорь
Россошанский Андрей
Звенигородская Анастасия
Кирильчук Виктор
Безруков Марк

Содержание

Облако в Linux своими руками

Многие люди пользуются такими сервисами хранения файлов, как Dropbox, Ubuntu One и др. Но намного интереснее создать своими руками собственное хранилище. Проект Owncloud позволит за несколько минут установить и настроить облако.

Основные возможности Owncloud

• наличие клиентских программ для Windows, Mac и Linux;

• шифрование на стороне сервера (особенно удобно, если хостинг предоставляет третья сторона, а файлы требуется держать в секрете);

• система контроля версий (можно откатиться на предыдущие версии хранимых файлов);

• просмотр ODF- и PDF-файлов прямо в окне браузера;

• подключение сторонних сервисов хранения (Dropbox, GDrive и т.п.) как директории внутри вашего хранилища;

• доступ через WebDAV;

• возможность подключения к хранилищу музыкальных проигрывателей по протоколу Ampache.

Установка Owncloud

Для установки введите с терминала следующие команды:

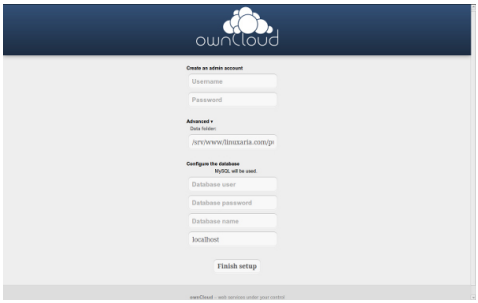
```
cd

apt-get update && apt-get -y
install apache2 php5 php-pear
php-xml-parser php5-sqlite php5-
json sqlite php5-mysql mp3info
curl libcurl3 libcurl3-dev php5-
curl zip php5-gd
wget http://download.owncloud.
org/releases/owncloud-4.0.2.tar.
bz2 && tar xjf owncloud-
4.0.2.tar.bz2  &&  cp -r owncloud
/var/www/  &&  chown -R www-
data:www-data /var/www/owncloud
/etc/init.d/apache2 restart
```

Теперь потребуется настроить базу данных MySQL:

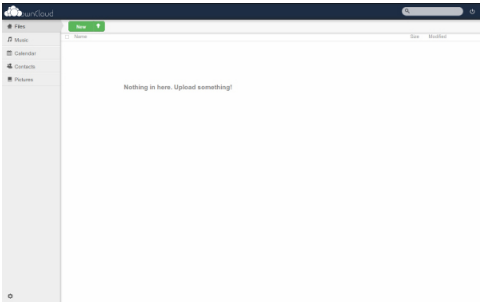
```
$ mysql -u adminusername -p
Enter password:
Welcome to the MySQL monitor.
Commands end with ; or \g.
Your MySQL connection id is 5340
to server version: 3.23.54

Type 'help;' or '\h' for help.
Type '\c' to clear the buffer.

mysql>CREATE DATABASE owncloud;
Query OK, 1 row affected (0.00
sec)

mysql> GRANT ALL
PRIVILEGES ON owncloud.* TO
"mycloud"@"localhost"
IDENTIFIED BY "mypassword";
Query OK, 0 rows affected (0.00
sec)
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01
```

```
sec)

mysql> EXIT
Bye
$
```

На этом работа в терминале заканчивается и дальнейшие настройки выполняются через удобный web-интерфейс. Наберите в браузере http://[имя-вашего-хоста]/owncloud. Вас попросят заполнить информацию для подключения к базе данных и указать имя и пароль администраторского аккаунта, который будет создан.



После этого вас ожидает простой и удобный web-интерфейс для загрузки файлов:



Как уже было сказано, помимо работы через браузер имеется возможность использовать клиенты под разные платформы. Скачать их можно с официального сайта проекта Owncloud.

По материалам сайта:
liberatum.ru

phpVirtualBox - GUI для VirtualBox

Существует множество систем виртуализации, некоторые работают только в консоли некоторые поддерживают графическое управление например VirtualBox.

А что делать если охота VirtualBox но на сервере нет графики. Можно из консоли, но можно через web интерфейс, такой как phpVirtualBox тем более что он рекомендован на официальном сайте гипервизора.

И так начнем. Скачиваем phpVirtualBox на свой сервере в папку /var/www , распаковываем и даем соответствующие права.

Редактируем файл config.php

```
/* Username / Password for system
user that runs VirtualBox */
var $username = 'vbox';
var $password = 'pass';
/* SOAP URL of vboxwebsrv (not
phpVirtualBox's URL) */
```

```
var $location =
'http://127.0.0.1:18083/';
```

Логин пароль вашего юзера и адрес с портом сервера, где работает virtualbox.

Далее создадим на сервере файл /etc/default/virtualbox с таким содержанием:

```
VBOXWEB_USER=vbox
VBOXWEB_HOST=0.0.0.0
VBOXWEB_PORT=18083
VBOXWEB_TIMEOUT=300
VBOXWEB_CHECK_INTERVAL=5
VBOXWEB_THREADS=100
VBOXWEB_KEEPALIVE=100
VBOXWEB_LOGFILE=»/var/log/
vboxwebservice.log»
INSTALL_DIR=/usr/lib/virtualbox
```

тут все предельно ясно, кому нет – на сайте есть пояснения.

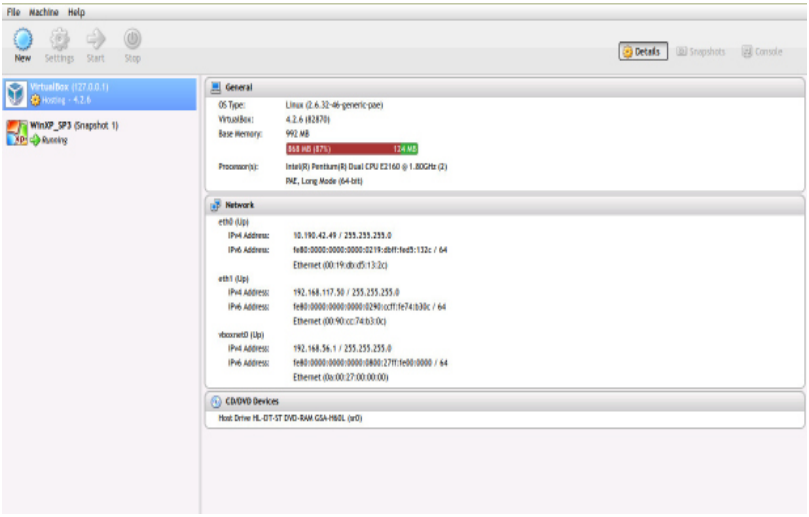
Последний шаг:

```
/etc/init.d/vboxweb-service
restart
```

Все готово, веб морда готова, заходим http://IP/phpvirtualbox/ логин пароль admin admin, если не подходит – правим файл config.php и заходим без авторизации, затем возвращаем авторизацию, заходим опять admin admin и создаем своего пользователя веб интерфейса.

phpVirtualBox постоянно обновляется, есть версии на все версии VirtualBox-сов. Рекомендую переодически обновлять, т.к. добавляются новые фичи и не требут рестарта сервера.

По материалам сайта:
linuxcenter.kz



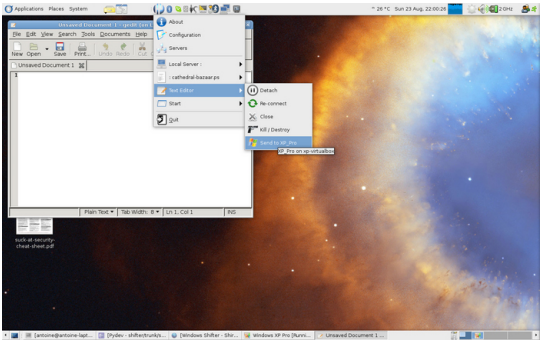
Выпуск Xpra 0.10, аналога утилиты screen для графических программ

Представлен новый выпуск утилиты Xpra 0.10, позволяющей обеспечить выполнение графических приложений без привязки к текущему X-сеансу. Xpra является аналогом консольных оконных менеджеров tmux и screen, нацеленным на работу с X11. В частности, xpra позволяет выполнить графическое приложение на удалённом хосте с трансляцией интерфейса на текущую машину, затем, не завершая выполнение программы, отсоединить её от текущего сеанса, через какое-то время подключиться к удалённому хосту с другого компьютера и продолжить работу с программой. Например, можно начать работу с программой на одной машине и продолжить на другой.

В новой версии значительно увеличена производительность кода на стороне клиента и сервера, обеспечена поддержка многопоточного кодирования с использованием x264, по умолчанию включён режим рендеринга на стороне клиента с использованием OpenGL. Добавлена поддержка прозрачности окон. Возможность задания произвольной оконной раскладки (--window-layout). Поддержка перехода в полноэкранный режим. Новый экспериментальный клиент, написанный с использованием библиотеки Qt.

Для упрощения использования xpra развивается специальный графический фронтэнд Winswitch, который кроме xpra поддерживает работу по протоколам VNC, NX и RDP. Для Xpra, NX и VNC в Winswitch реализована интересная функция клонирования, позволяющая организовать совместный доступ нескольких пользователей к одному приложению (в том числе и мультимедийному) одновременно с разных компьютеров. Другой примечательной особенностью Winswitch является автоматическое определение десктоп-систем в локальной сети при помощи протокола ZeroConf, что позволяет обеспечить взаимодействие компьютеров, на которых запущен Winswitch, без предварительной конфигурации.

Возможно использование xpra и из командной строки. На удалённом хосте, на



котором необходимо запустить приложение (в примере запускается xterm), выполняем:

```
xpra start :100 --start-child=xterm
```

после чего подсоединяемся с другой машины к созданному сеансу

```
xpra attach ssh:имя_сервера:100
```

По материалам сайта: opennet.ru

Внедрение динамического оконного менеджера dwm для использования на рабочих станциях

Рассматривается процесс настройки, разворачивания и использования динамического оконного менеджера DWM на рабочих станциях под управлением ОС Ubuntu 10.04.2 LTS.

В настоящее время всё большее применение находят фреймовые (или мозаичные) оконные менеджеры [1] – менеджеры окон X Window System, разбивающий рабочее пространство экрана на взаимно непересекающиеся прямоугольные области – фреймы, используемые для вывода информации отдельным приложением. Применение фреймовых оконных менеджеров минимизирует использование мыши (применяя клавиатурные комбинации), уменьшая время выполнения рутинных операций, повышая эффективность работы, полностью используя всё видимое пространство. Одним из таких менеджеров является DWM [2, 3] – динамический фреймовый оконный менеджер для X11. Особенностью является способ конфигурирования – основные настройки можно изменить в заголовочном файле.

Необходимо выполнить следующие шаги для установки и настройки рабочей среды под управлением DWM [4-7]:

1. Скачать dwm с официального сайта [3] по ссылке http://dl.suckless.org/dwm/dwm-5.9.tar.gz. Распаковать в удобную для дальнейшей работы папку (например, ~/dwm-5.9). Установить дополнительное программное обеспечение командой `sudo apt-get install halevt xloadimage`, где halevt – демон автомонтирования томов, xloadimage – установка изображения рабочего стола. Помимо этого требуется установленная среда разработки GCC с необходимыми файлами для сборки dwm.

2. Создать файл ~/.xinitrc (touch ~/.xinitrc && chmod 755 ~/.xinitrc) со следующим содержимым:

```
#!/bin/sh
pgrep halevt >> /dev/null ||
halevt
xloadimage -fullscreen -onroot /
home/путь_к_картинке{.png,.jpg}
test -x /usr/bin/uxterm && /usr/
bin/uxterm &
test -x /usr/bin/google-chrome &&
/usr/bin/google-chrome &
test -x /usr/bin/ooffice && /usr/
bin/ooffice &
test -x /usr/bin/k3b && /usr/bin/
k3b &
test -x /home/username/
virtualbox.sh && /home/username/
virtualbox.sh &
test -x /usr/local/netbeans-7.0/
bin/netbeans && /usr/local/
netbeans-7.0/bin/netbeans &
test -x /usr/bin/wxmaxima && /
```

```
usr/bin/wxmaxima &
while true
do
   LOCALTIME=$(date +%e/%b/%Y\
%a\ %k:%M)
   xsetroot -name «$LOCALTIME»
   sleep 20s
done &
exec dwm
exit 0
```

Данный скрипт производит запуск демона автомонтирования, устанавливает рисунок рабочего стола, запускает необходимые прикладные программы, в бесконечном цикле формирует текст заголовка главного окна (дата, время), и, собственно, запускает dwm.

3. Создать файл /usr/share/xsessions/dwm.desktop (sudo touch /usr/share/xsessions/dwm.desktop) со следующим содержимым (требуются права root):

```
[Desktop Entry]
Encoding=UTF-8
Name=dwm
Comment=Use this session to run
dwm as your desktop environment
Exec=/home/username/.xinitrc
Icon=
Type=Application
```

Это позволит произвести вход через GDM.

4. На основании ~/dwm-5.9/config.def.h создать файл ~/dwm-5.9/config.h (cp ~/dwm-5.9/config.def.h ~/dwm-5.9/config.h) и внести в него следующие изменения:

```
 /* tagging */
-static const char *tags[] = {
«1», «2», «3», «4», «5», «6»,
«7», «8», «9» };
+static const char *tags[] = {
«XTerm», «WWW», «OOffice», «4»,
«5», «6», «7», «8», «VBox» };

 static const Rule rules[] = {
      /* class      instance
title       tags mask
isfloating   monitor */
      { «Gimp»,     NULL,
NULL,       0,            True,
-1 },
      { «Firefox»,  NULL,
NULL,       1 << 8,       False,
-1 },
+     { «UXTerm»,   «xterm»,
NULL,       1,            False,
-1 },
+     { «Google-chrome»,
«google-chrome», NULL, 1 << 1,
False,       -1 },
+     { «OpenOffice.org
3.2», NULL, NULL,   1 << 2,
False,       -1 },
+     { «K3b»,      «k3b»,
NULL,       1 << 4,       False,
-1 },
+     { «VirtualBox»,NULL,
NULL,       1 << 8,       False,
-1 },
+     { «java-lang-
Thread»,NULL,NULL,       1 << 5,
False,       -1 },
+     { «Wxmaxima», «wxmaxima»,
NULL,       1 << 3,       False,
-1 },
 };

 /* key definitions */
-#define MODKEY Mod1Mask
+#define MODKEY Mod4Mask
```

Здесь приведен отредактированный вывод diff -u. При этом в контексте изменения знаками «-» помечены строки, которые надо удалить, знаками «+» помечены строки, которые следует добавить в файл config.h.

Первое изменение. Окна распределяются по именованным рабочим пространствам (тэгам) согласно определённым правилам, поэтому, для удобства работы, вместо чисел устанавливаются названия тэгов.

Второе изменение. Задаются правила размещения для отдельных программ. Правила задаются в формате: Класс Программы, Функция, Имя Окна, Тэги, Плавающее Или Тайловое, Номер Дисплея. С помощью вызова xprop | grep -e NAME -e CLASS можно получить WM_CLASS – значение первого поля (в случае, если в выводе есть несколько разных слов, второе слово идёт в первое поле, а первое – во второе), WM_NAME – третьего. «Тэги» указывают номера именованных рабочих пространств, где будут размещены окна программы: 0 –отображать на текущем тэге, ~0 – отображать на всех тэгах одновременно, 1<<n – отображать на (n+1)-м тэге, либо комбинацией, для одновременного отображения на нескольких тегах.

Третье изменение. Чтобы не было конфликтов с переключением раскладки на русский язык, необходимо изменить используемую по-умолчанию MODKEY с клавиши Alt на Windows-key, т.е. на Mod4Mask.

5. Собрать и установить dwm командой sudo make clean install.

6. Установить шрифты sudo apt-get install xfonts-terminus.

7. Выйти из текущего сеанса (выключить, перезагрузить компьютер) и в экранном менеджере GDM выбрать в качестве сеанса dwm.

Текущее окно выбирается фокусом мыши. Управление осуществляется следующими комбинациями клавиш:

1. [Shift]+[Modkey]+[Enter] – запустить терминал.
2. [Alt]+[p] – запуск программ.
3. [Modkey]+[Цифра] – переход к рабочему пространству с номером «Цифра»
4. [Shift]+[Modkey]+[Цифра] – переместить окно на рабочее пространство с номером «Цифра».
5. [Shift]+[Modkey]+[c] – убить текущее окно.
6. [Modkey]+[Enter] – переключает окно между «master» и «stack».
7. [Shift]+[Modkey]+[q] – выйти из DWM.
8. [Modkey]+[t] – включить тайловый режим (установлен по-умолчанию).

В дальнейшем планируется рассмотреть перенос в среду на базе DWM ряда возможностей среды GNOME. Выражаю благодарность участнику форума http://forum.ubuntu.ru под псевдонимом altprint за активное участие в первоначальном накоплении материала, связанного с dwm [5].

Библиографический список

1. Фреймовый оконный менеджер X Window System [Электронный ресурс]. – Электрон. текстовые данные. – Режим доступа: http://ru.wikipedia.org/wiki/Фреймовый_оконный_менеджер_X_Window_System.

2. dwm [Электронный ресурс]. – Электрон. текстовые данные. – Режим доступа: http://ru.wikipedia.org/wiki/Dwm.

3. dynamic window manager [Электронный ресурс]. – Электрон. текстовые данные. – Режим доступа: http://dwm.suckless.org/.

4. dwm [Электронный ресурс]. – Электрон. текстовые данные. – Режим доступа: https://wiki.archlinux.org/index.php/Dwm.

5. DWM [Электронный ресурс]. –Электрон. текстовые данные. – Режим доступа: http://forum.ubuntu.ru/index.php?topic=161499.0.

6. Dynamic Window Manager - Динамический Менеджер Окон [Электронный ресурс]. — Электрон. текстовые данные. – Режим доступа: http://www.calculate-linux.org/blogs/ru/212/show.

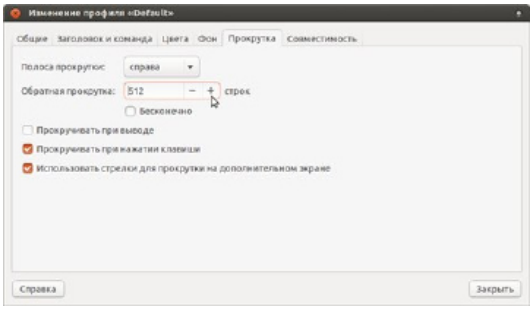
7. Dwm [Электронный ресурс]. – Электрон. текстовые данные. – Режим доступа: http://ru.gentoo-wiki.com/wiki/Dwm.

Кирилл Ткаченко

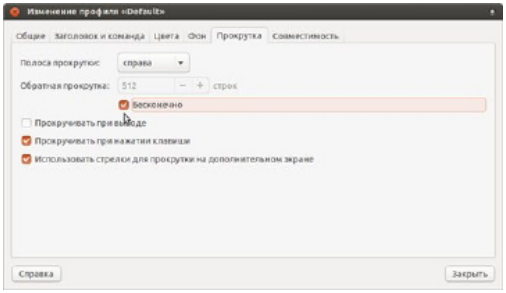
Данный материал написан для «Первого конкурса русскоязычных статей для ОС Ubuntu 2011 года» (konkurs.ubuntu.ru)

Бесконечная терминальная прокрутка в Ubuntu/Linux Mint

По умолчанию количество строк для отображения в терминале установлено в значении 512, но иногда вывод команды может быть очень длинным и вы не сможете его увидеть, прокручивая скрол. Эту проблему очень легко решить, если можно назвать это проблемой. Просто нужно отредактировать значение в Меню - Параметры профиля - Прокрутка - Обратная прокрутка, увеличив цифровое значение строк:



Но мне хотелось бы предложить вам другой вариант, а именно установить флажок на опции Бесконечно:



На работу терминала это ни коим образом не влияет, но в данном случе вы сможете увидеть весь вывод команды/команд, произведённых в терминале, прокручивая/поднимая скроль вверх.

По материалам сайта:
compizomania.blogspot.com

Шпаргалка начинающего Debian/Ubuntu администратора по управлению пакетами

Долгое время меня глодало незнание того, как сделать некоторые элементарные вещи в дебиановских менеджерах пакетов, но, как часто бывает, спросить рядом было не у кого, а до написания куда-либо руки не доходили. И вот наконец вопросы вызрели и я написал свой вопрос в дебиановскую рассылку. Естественно оказалось что пропустил что-то очевидное, но и узнал много неочевидных полезностей, посему решил набросать шпаргалку, авось кому пригодится.

Краткая справка Debian администратора

Основное и общеизвестное

получение информации о новых/обновлённых пакетах

```
sudo aptitude update
```

Обновление

```
sudo aptitude safe-upgrade
```

Поиск пакета по именам пакетов

```
aptitude search key_word
```

Поиск пакета по точному названию

```
aptitude search «^name$»
```

Поиск по описанию

```
aptitude search
«?description(«key_word»)»
```

Информация о пакете

```
aptitude show package_name
```

Установка

```
sudo aptitude install package_
name
```

Удаление

```
sudo aptitude remove package_name
```

Полное удаление (вместе с конфигами)

```
sudo aptitude purge package_name
```

Очистить кэш загруженных пакетов (освободить место)

```
aptitude autoclean # удалятся
только пакеты неактуальных версий

aptitude clean # очистится весь
кэш
```

Установка отдельно скачанного/созданного пакета (для создания пакета из сторонних исходников нужно использовать утилиту checkinstall с флагом -D)

```
sudo dpkg -i /path/to/package.deb
```

Для получения доп информации

```
man aptitude
```

```
sudo aptitude install aptitude-doc-en
```

и смотрим документацию (/usr/share/doc/aptitude/html/en/index.html), кому быструю справку по поисковым шаблонам, тому сюда — /usr/share/doc/aptitude/html/en/ch02s04.html. Если лень ставить доку, то в сети она есть. Вводная на Debian Wiki: wiki.debian.org/Aptitude.

А теперь то что не очевидно или требует полного прочтения документации

1. Как после update посмотреть какие пакеты будут обновлены?

```
aptitude search ?upgradable
```

также можно юзать (если поставить)

```
sudo daptup
```

но после его установки точно также будет себя вести и обычный update

2. Как узнать что изменилось в пакетах которые будут обновлены? Можно пробовать

```
sudo aptitude changelog package_name
```

для каждого пакета.

Но лучше поставить apt-listchanges, тогда перед любой установкой обновлений будет показан список изменений, по умолчанию настройки не очень удобные, поэтому лучше перенастроить под себя, например, выбрать формат вывода (пока использую текст, при больших обновлениях наверно pager лучше), не слать писем, спрашивать подтверждения, выводить всю информацию. Для этого нужно запустить

```
sudo dpkg-reconfigure apt-listchanges
```

3. Что делать если обновление что-то поломало и нужно откатиться?

Отката нет, можно попробовать найти предыдущую версию пакета

```
sudo aptitude version package_name
```

и установить её

```
sudo aptitude install package_name=version
```

4. Как найти все пакеты установленные вручную? Есть вариант команды (aptitude search '~i!~M') но, к сожалению, он не даёт желаемого результата, так что вопрос остаётся открытым, есть куча способов основанных на анализе логов

/var/log/aptitude (+ ротированные куски)

/var/log/installer/initial-status.gz

/var/log/dpkg.log (+ ротированные куски)

но простого и готового решения нет, да информация теоретически может быть потеряна при ротациях, нужно конфигурить

5. Как посмотреть список файлов в пакете?

Если пакет установлен

```
dpkg -L package_name
```

для любых пакетов поставить apt-file и

```
apt-file list package_name
```

6. Как посмотреть какому пакету принадлежит файл?

```
dpkg -S file_name
```

7. Как удалить все пакеты, где есть key в названии пакета?

```
sudo aptitude purge ~ikey
```

8. Как удалить оставшиеся конфиги от удалённых пакетов?

```
sudo aptitude purge ~c
```

9. Как найти пакет пакет, в котором содержится файл lib.so:

apt-file search lib.so

10. Как сконвертировать rpm пакет в deb?

```
alien --to-deb /path/to/file.rpm
```

11. Как найти список установленных ядер?

```
dpkg --list linux-* | grep ii
```

12. Как установить пакет из testing или experimental? На эту тему нужно писать отдельно (например так), но если кратко, то команды для этого есть

```
sudo aptitude -t testing package_name
```

или

```
sudo aptitude package_name/testing
```

13. Как удалить метапакет, но оставить одну из зависимостей? Придётся почитать документацию про ключ unmarkauto или глянуть сюда.

14. Как узнать что попало в файловую систему мимо системы управления пакетами? Есть утилита cruft, хотя вопрос интерпретации результатов (файла report) пока открыт

```
sudo cruft -d / -r report
--ignore /home --ignore /var
--ignore /tmp
```

15. Какие есть дополнительные репозитории? Debian – wiki.debian.org/UnofficialRepositoriesUbuntu – множество всяких PPA

16. Что есть ещё?

apt-cdrom
apt-spy
auto-apt
apt-key
apt-add-repository

Некоторые вещи умеет только apt-get. Есть альтернативные утилиты для управления пакетами, например wajig, который пытается вобрать в себя функционал всех остальных утилит.

По материалам сайта:
habrahabr.ru

Работа с LVM

Logical Volume Manager (LVM) - это очень мощная система управления томами с данными для Linux. Она позволяет создавать поверх физических разделов (или даже неразбитых винчестеров) логические тома, которые в самой системе будут видны как обычные блочные устройства с данными (т.е. как обычные разделы). Основные преимущества LVM в том, что во-первых одну группу логических томов можно создавать поверх любого количества физических разделов, а во-вторых размер логических томов можно легко менять прямо во время работы. Кроме того, LVM поддерживает механизм снапшотов, копирование разделов «на лету» и зеркалирование, подобное RAID-1.

LVM увеличивает гибкость файловой системы, однако, являясь просто промежуточным слоем, не отменяет ограничения и использование других слоёв, а также усложняет работу. То есть, по-прежнему нужно создавать и изменять разделы, форматировать их; изменение размера должно поддерживаться также и самой файловой системой.

Так же хотелось бы обратить внимание на один немаловажный момент. При использовании LVM не следует стараться распределить все имеющееся дисковое пространство в логические тома. Следует создать разделы с минимально необходимым размером, а потом, при необходимости нарастить их до требуемого размера из резервного свободного места.

Создание разделов на LVM

Для начала нужно установить поддержку LVM в нашей системе.

```
# apt-get install lvm2
```

У нас есть три свободных физических диска – sdb, sdc и sdd. Создадим LVM-раздел на первых двух из них.

Сначала на этих дисках создадим физические тома LVM:

```
# pvcreate /dev/sdb
# pvcreate /dev/sdc
```

Теперь создадим группу томов с произвольным названием, например, study:

```
# vgcreate study /dev/sdb /dev/
sdc
```

В результате мы должны получить вывод

```
Volume group «study» successfully
created
```

После этого можно создавать логические тома:

```
# lvcreate -n lv1 -L 5G study
# lvcreate -n lv2 -L 6G study
```

Теперь у нас есть блочные устройства /dev/study/lv1 и /dev/study/lv2. С ними можно работать так же, как и с обычными разделами.

```
# mkfs.ext4 /dev/study/lv1
# mkfs.ext4 /dev/study/lv2
```

И можем их примонтировать в систему
Для этого создадим в системе точки монтирования:

```
# mkdir /mnt/lv1
# mkdir /mnt/lv2
```

И примонтируем разделы

```
# mount /dev/study/lv1 /mnt/lv1
# mount /dev/study/lv2 /mnt/lv2
```

Введя команду df -h мы увидим информацию об этих разделах

```
/dev/mapper/study-lv1 5,0G 138M
4,6G 3% /mnt/lv1
/dev/mapper/study-lv2 6,0G 140M
5,5G 3% /mnt/lv2
```

То есть их размер 5 и 6 Gb, как мы и указывали при создании.

Добавление физических томов

Если нам потребуется увеличить размер какого-то раздела внутри LVM, то для этого нам понадобится, естественно, свободное место. Добавим в LVM диск sdd.

Создадим на нем физический том:

```
# pvcreate /dev/sdd
```

И добавим его в группу томов study:

```
# vgextend study /dev/sdd
```

Теперь можно создать ещё один логический диск при помощи lvcreate или увеличить размер существующего с помощью команды lvresize.

Увеличим раздел lv1 до 15 Gb.

```
# lvresize -L 15 G study/lv1
```

Далее увеличить размер файловой системы

```
# resize2fs /dev/study/lv1
```

Если после этого посмотреть информацию о разделах командой df -h, то мы увидим

```
/dev/mapper/study-lv1 15G 141M
14G 1% /mnt/lv1
```

То есть размер раздела действительно изменился.

Аварийная работа с LVM

Любая система, и тем более винчестеры, по определению не надежна. И рано или поздно вы встретитесь с ситуацией, когда винчестер начинает сбоить. При использовании LVM есть несколько вариантов защиты себя в таких ситуациях.

Замена диска налету.

Предположим, что у нас начались проблемы с диском /dev/sdd. Заменим его на диск /dev/sde без выключения системы.

Для этого создадим физический том на новом диске

```
# pvcreate /dev/sde
```

И добавим его в группу томов study:

```
# vgextend study /dev/sde
```

Теперь нам нужно переместить информацию с диска sdd на новый sde:

```
# pvmove /dev/sdd /dev/sde
/dev/sdd: Moved: 0,7%
/dev/sdd: Moved: 30,7%
/dev/sdd: Moved: 63,5%
/dev/sdd: Moved: 97,3%
/dev/sdd: Moved: 100,0%
```

Перемещение информации займет некоторое время в зависимости от объема диска.

По окончании перемещений удалим сбойный диск из группы томов:

```
# vgreduce study /dev/sdd

Removed «/dev/sdd» from volume
group «study»
```

Теперь у нас все работает, сбойный диск отключен и все это не прекращая работы сервера, буквально по живому.

Создание снапшотов LVM

Сферы применения снапшотов могут быть самыми разнообразными. Например, резервное копирование базы данных. Если не использовать LVM – базу данных необходимо останавливать, копировать ее файлы куда-нибудь для последующего резервного копирования, а затем запускать ее заново. То есть делать это придется в нерабочее время. С LVM все проще – следует сделать снимок раздела с файлами базы данных и уже можно начинать делать резервную копию. Остановка базы данных не нужна.

Самая интересная особенность LVM при работе со снимками – это то, что снимок может занимать меньше дискового пространства, чем оригинал. Для этого используется режим Copy-on-Write, при котором реальное использование дискового пространства начинается только при изменении данных на томе-оригинале. То есть при попытке модификации файла на томе-оригинале неизмененный файл сначала сохраняется на томе-снимке, а уж затем модифицируется.

ВНИМАНИЕ! При заполнении тома-снимка до конца, происходит его уничтожение. То есть том продолжает существовать, но ни смонтировать его, ни просмотреть его содержимое (если он был смонтирован до этого) уже не получится. Эту особенность следует обязательно учитывать при задании размера тома-снимка в момент его создания.

Создание снимка производится командой lvcreate:

```
# lvcreate -s -L 2M -n backup /
dev/study/lv2
Rounding up size to full physical
extent 4,00 MiB
Logical volume «backup» created
```

Ключ -s указывает, что создаем мы именно снапшот, -n указывает имя создаваемого тома, а /dev/study/lv2 показывает с какого именно тома мы делаем снимок.

Команда lvscan покажет нам, что мы создали снапшот:

```
# lvscan
ACTIVE ‘/dev/study/lv1’ [15,00
GiB] inherit
ACTIVE Original ‘/dev/study/lv2’
[6,00 GiB] inherit
ACTIVE Snapshot ‘/dev/study/
backup’ [4,00 MiB] inherit
```

Теперь можете убедиться в том, что изменения, происходящие с оригиналом, никак не повлияют на снапшот.

Информационные утилиты LVM

Получить информацию о группе томов

```
# vgdisplay
```

Получить информацию по созданным логическим томам

```
# lvdisplay
```

Получить информацию по физическим томам

```
# pvdisplay
```

По материалам сайта:
http://yakim.org.ua/articles/servers/152-lvm-ubuntu.html

Декомпиляторы, или Что делать, если нужно восстановить исходники из бинарников?

Проблема восстановления исходного кода из скомпилированных бинарников возникает сравнительно часто и протекает остро, с воем и рыданиями. Здесь нам, до некоторой степени, помогут замечательные программы-декомпиляторы, и в этом посте автор собрал свои скромные попытки выдрать исходники (или хотя бы намёки на них) из скомпилированных из С-шного кода бинарников.

Задача для декомпилятора бинарников, собранных из С кода

Классический случай: один деятель на факультете написал на правильном ANSI C (и используя библиотеки BLAS и LAPACK) нужные и хорошие алгоритмы, и скомпилировал их в виде MEX-файлов для использования (это С-шный код, который можно вызывать из МАТЛАБ).

Но потом он повздорил с народом, разозлился и свалил в частную контору, унеся все исходники с собой. Документации нет. Копий исходников нет. Есть обрывки личной переписки и намёки в сопровождающих файлах на тип алгоритма. Вариант физического воздействия на автора тупыми тяжёлыми предметами не рассматривается.

Нужно восстановить исходники если и не до компилируемого состояния, то во всяком случае выудить оттуда алгоритмы и ключевые методы, использованные при реализации.

Кратко: суть и сложность проблемы

Декомпилятор (Decompiler) пытается перевести скомпилированный бинарный файл обратно в некое подобие исходного кода. Качество выхлопа зависит от особенностей языка исходника:

• для C# или Java есть много декомпиляторов - байткод на java содержит много информации. Это помогает восстанавливать декомпилятору исходник до состояния, пригодного к повторной компиляции.

• совершенно другая история с двоичными файлами, в которых, как правило, отладочной информации нет. Тем не менее, динамически связанные библиотеки функций, как правило, вызываются по имени. Часто, типы параметров библиотечных функций известны, и это может помочь до известных пределов.

Декомпиляторы пытаются восстановить информацию, которая частично утрачена при компиляции в бинарный файл - в этом и заключается основная сложность.

Если вы думаете, что с помощью декомпилятора вы получите обратно красивый код на C - вы будете сильно разочарованы. В лучшем случае на выходе будет общая (и довольно грубая) структура программы, в худшем - только имена функций и немного мусора.

Так что ответ на вопрос заголовка поста: «Обхватить голову руками и закричать #$@@@@@!» :-).

Программы для декомпилирования (decompilers) для C/C++

Декомпиляторов для C/C++ немного, и ниже список из наиболее работоспособных. Здесь нет разделения на опенсорс или Linux-only - для такого дела, как вскрытие исходников, можно (и нужно) поступиться своими светлыми идеалами и наступить на горло собственной песне.

Сразу замечу: скорее всего, ни один декомпилятор не выдаст вам сразу компилируемый код. Придётся потратить порядком времени и сил, чтобы это месиво превратить в код, который можно читать (желательно, не только компилятору).

Boomerang

Boomerang это C decompiler с открытыми исходами:

- поддерживаемые бинарные форматы: ELF PE-COFF Mac-OS
- платформы: Windows/Linux
- поддерживаемые архитектуры: IA32 MIPS PPC
- метод работы: поточный, есть жалкий графический интерфейс, лучше использовать CLI.

Весьма продвинутый набор алгоритмов анализа кода, что не удивительно - один из соавтором защитил на этом докторскую диссертацию [Michael James Van Emmerik, Static Single Assignment for Decompilation, Ph.D. thesis, the University of Queensland, Australia. PDF, mirror PDF].

Качество кода, выдаваемого декомпилятором:

- Структурирование: очень хорошее
- Переменные: хорошее
- Типы данных: очень хорошее

Выдаваемое качество кода сильно варьируется: некоторые функции почти идеально восстановлены, хорошо видна структура кода и есть указание типа переменных. В других случаях функции сильно запутаны и их почти невозможно прочитать.

Программа всё ещё в состоянии бета-версии и для больших проектов не подходит.

RecStudio

Интерактивный декомпилятор RecStudio для C и (отчасти) C++, закрытая разработка:

- поддерживаемые бинарные форматы: ELF PE-COFF AOUT RAW PS-X
- платформы: Windows/Linux/MacOS
- поддерживаемые архитектуры: x86 (ia32) x86_64 Mips PowerPC mc68k
- метод работы: поточный и интерактивный, есть графический интерфейс.

Использует продвинутый набор алгоритмов анализа (partial Single Static Assignment, SSA), хотя до сих пор (и это после 20 лет!) в стадии разработки. Качество кода, выдаваемого декомпилятором:

- Структурирование: хорошее

• Переменные: частично
• Типы данных: частично или никак

Выдаваемое качетсво кода, как правило, хуже, чем у Boomerang, хотя обновлённый RecStudio более подробен.

Программа работает вполне стабильно, есть сборки под Linux.

dcc - DOS to C decompiler

Поточный декомпилятор Dcc, только ANSI C и для exe-файлов, с открытым исходным кодом под GPL:

• поддерживаемые бинарные форматы: EXE/COM
• платформы: Windows
• поддерживаемые архитектуры: x86
• метод работы: поточный.

Один из первых декомпиляторов вообще, и только под DOS. Сильная сторона - структурирование кода. Качество кода, выдаваемого декомпилятором:

• Структурирование: хорошее
• Переменные: частично
• Типы данных: частично или никак

Разработка Cristina Cifuentes, которая защитила PhD в Queensland University of Technology на этом деле [можно полистать, если что - C. Cifuentes, Reverse Compilation Techniques].

Hex Rays - plugin для IDA Pro

На самом деле Hex Rays не является отдельной программой - это плагин-декомпилятор для IDA Pro. Комбинация продвинутых возможностей IDA Pro (это дизассемблер) и Hex Rays в качестве декомпилятора очень впечатляет, как и аэрокосмическая цена. По причине закрытости продукта (нет даже демо-версии) и нереальной цены в этом разделе про Hex Rays больше ничего написано не будет.

Ходовые испытания в реальных условиях

Для начала попробуем декомпилировать что-нибудь совсем простенькое и написанное на ANSI C и с использованем библиотеки BLAS для векторых и матричных операций.

1. Простенький C-шный бинарник + BLAS

Собственно, код на C для перемножения матрицы и вектора (используется CBLAS).

Исходник:

```
#include <stdio.h>
#include <cblas.h>

double m[] = {
  3, 1, 3,
  1, 5, 9,
  2, 6, 5
};

double x[] = {
  -1, -1, 1
};

double y[] = {
  0, 0, 0
};

int main(void)
{
  int i, j;

  for (i=0; i<3; ++i) {
    for (j=0; j<3; ++j)
printf(«%5.1f», m[i*3+j]);
    putchar('\n');
  }

  cblas_dgemv(CblasRowMajor,
CblasNoTrans, 3, 3, 1.0, m, 3, x,
```

```
1, 0.0, y, 1);


  for (i=0; i<3; ++i)
printf(«%5.1f\n», y[i]);

  return 0;
}
```

После перемножения выдаст результат на консоль.

Выхлоп Boomerang

У него много ключей и параметров, часть которых не знает даже официальная, скажем так, документация. Тем не менее, для ключа -Td (Use data-flow-based type analysis) мы имеем выхлоп в стиле дзен:

```
double y;
double m = 3.;

// address: 0x80484e4
int main(int argc, char **argv,
char **envp) {
    void *local23;   // r28

    for(;;) {
        proc1();
    }
}
```

Скажем так, не слишком ободряюще. Ключ -Tc (Use old constraint-based type analysis) выдаёт больше информации к размышлению:

```
char y[24];
long long m[9] = {
0x4008000000000000LL,
0x3ff0000000000000LL,
0x4008000000000000LL,
0x3ff0000000000000LL,
0x4014000000000000LL,
0x4022000000000000LL,
0x4000000000000000LL,
0x4018000000000000LL,
0x4014000000000000LL };

// address: 0x80484e4
int main(int argc, char **argv,
char **envp) {
    int local10;   // r24
    int local13;   // r28
    int local14;   // r29
    int local15;   // r32
    int local2;   // m[r28 + 72]
{60}
    int local3;   // r28{181}
    int local6;   // m[r28 + 72]
{119}
    int local7;   // m[r28 + 72]
{149}
    int local8;   // m[r28 + 76]
{9}
    int local9;   // m[r28 + 76]
{49}

    *(int*)(local13 - 4) =
local14;
    *(int*)(local13 - 12) = 0;
    local3 = local13 - 84;
    if (*(int*)(local3 + 72) > 2)
{
        *(int*)(local3 + 52) = 1;
        *(int*)(local3 + 48) =
0x8049868;
        *(long long*)(local3 +
40) = 0.;
        *(int*)(local3 + 36) = 1;
        *(int*)(local3 + 32) =
0x8049848;
        *(int*)(local3 + 28) = 3;
        *(int*)(local3 + 24) =
0x8049800;
        *(long long*)(local3 +
16) = 1.;
        *(int*)(local3 + 12) = 3;
        *(int*)(local3 + 8) = 3;
        *(int*)(local3 + 4) =
111;
        *(int*)local3 = 101;
```

```
        cblas_dgemv();
        local6 = 0;
        for(;;) {
            *(long long*)(local3
+ 4) = y[0];
            *(int*)local3 =
0x80486b6;
            proc1();
            local7 = *(int*)
(local3 + 72) + 1;
        }
    }
    local8 = 0;
    for(;;) {
        local10 = *(int*)(local3
+ 72) + *(int*)(local3 + 72) +
*(int*)(local3 + 72);
        local15 = m[local10];
        *(long long*)(local3 + 4)
= local15;
        *(int*)local3 =
0x80486b0;
        proc1();
        local9 = *(int*)(local3 +
76) + 1;
    }
}
```

Подробностей тут больше, и тут выловлен самый главный ключик - cblas_dgemv()

Выхлоп RecStudio
Намного более обилен и представляет собой следующий поток сознания:

```
// Generated by Rec Studio 4 -
build Oct 20 2012

_init()
{// addr = 0x08048398
    _unknown_ __ebx;
// r1
    _unknown_ __ebp;
// r6
    _unknown_ _t2;
// _t2

    __esp = __esp - 4;
    L1();
    _pop(__ebx);
    if( *((intOrPtr*)(_t2 +
0x1414)) != 0) {
        __gmon_start__();
    }
    frame_dummy();
    __do_global_ctors_aux();
    _pop(__eax);
    return;
}

L080483A4()
{
    _unknown_ _t2;
// _t2

    _pop(__ebx);
    if( *((intOrPtr*)(_t2 +
0x1414)) != 0) {
        __gmon_start__();
    }
    frame_dummy();
    __do_global_ctors_aux();
    _pop(__eax);
    _pop(__ebx);
    __esp = __ebp;
    _pop(__ebp);
    return;
}

__gmon_start__()
{// addr = 0x080483D8
    goto __imp____gmon_start__;
}

putchar()
{// addr = 0x080483E8
    goto __imp__putchar;
}

__libc_start_main()
{// addr = 0x080483F8
    goto __imp____libc_start_
```

```
main;
}

cblas_dgemv()
{// addr = 0x08048408
    goto __imp__cblas_dgemv;
}

printf()
{// addr = 0x08048418
    goto __imp__printf;
}

_start(
    signed int __eax,
// r0
    _unknown_ __edx
// r3
)
{// addr = 0x08048430
    _unknown_ __ebx;
// r1
    signed int _t5;
// _t5
    _unknown_ _t6;
// _t6
    _unknown_ _t10;
// _t10

    __edx = __edx;
    _t4 = __eax;
    _pop(__esi);
    __ecx = __esp;
    __esp = __esp & 240;
    _push(__eax);
    _push(__esp);
    _push(__edx);
    _push(__libc_csu_fini);
    _push(__libc_csu_init);
    _push(__ecx);
    _push(_t10);
    _push(main);
    __libc_start_main();
    asm(«hlt «);
    0;
    0;
    _push(0);
    _push(_t6);
    __esp = __esp - 4;
    if(completed.5982 != 0) {
    } else {
        _t4 = dtor_idx.5984;
        _t6 = ( &__DTOR_END__ -
&__DTOR_LIST__ >> 2) - 1;
        if(_t4 >= _t6) {
        } else {
            do {
                _t5 = _t4 + 1;
                dtor_idx.5984 =
_t5;
                *((intOrPtr*)
(_t5 * 4 +  &__DTOR_LIST__))();
                _t4 = dtor_
idx.5984;
            } while(_t4 < _t6);
        }
        completed.5982 = 1;
    }
    __esp = __esp + 4;
    _pop(__ebx);
    _pop(__ebp);
    return;
}

__do_global_dtors_aux(
    _unknown_ __esi
// r5
)
{// addr = 0x08048460
    _unknown_ __ebx;
// r1
    _unknown_ __ebp;
// r6
    _unknown_ _t4;
// _t4
    signed int _t5;
// _t5
    signed int _t6;
// _t6
    _unknown_ _t10;
// _t10

    if(completed.5982 == 0) {
        _t5 = dtor_idx.5984;
        _t10 = ( &__DTOR_END__ -
&__DTOR_LIST__ >> 2) - 1;
        if(_t5 >= _t10) {
```

```
L4:
                completed.5982 = 1;
                return;
            }
            do {
                _t6 = _t5 + 1;
                dtor_idx.5984 = _t6;
                 *((intOrPtr*)(_t6 *
4 +  &__DTOR_LIST__))();
                _t5 = dtor_idx.5984;
            } while(_t5 < _t10);
            goto L4;
        }
        return;
}

frame_dummy()
{// addr = 0x080484C0
    _unknown_ __ebp;
// r6

    __eax = __JCR_LIST__;
    if(__JCR_LIST__ == 0) {
    } else {
        __eax = 0;
        if(__eax != 0) {
            *__esp =  &__JCR_
LIST__;
            *__eax();
            return;
        }
    }
    return;
}

main(
    _unknown_ __fp0
// r28
)
{// addr = 0x080484E4
    signed int _v8;
// _cfa_ffffffff8
    signed int _v12;
// _cfa_ffffffff4
    intOrPtr _v32;
// _cfa_fffffffe0
    char* _v36;
// _cfa_ffffffdc
    intOrPtr _v48;
// _cfa_ffffffd0
    char* _v52;
// _cfa_ffffffcc
    intOrPtr _v56;
// _cfa_ffffffc8
    char* _v60;
// _cfa_ffffffc4
    intOrPtr _v72;
// _cfa_ffffffb8
    intOrPtr _v76;
// _cfa_ffffffb4
    intOrPtr _v80;
// _cfa_ffffffb0
    _unknown_ __ebp;
// r6

    __fp0 = __fp0;
    __esp = __esp & 240;
    __esp = __esp - 80;
    _v12 = 0;
    while(_v12 <= 2) {
        _v8 = 0;
        while(_v8 <= 2) {
            __fp0 ?_?  *((long
long*)((_v12 + __edx + __edx +
_v8) * 8 +  &m));
            asm(«fstp qword
[esp+0x4]»);
             *__esp = 134514352;
            printf();
            _v8 = _v8 + 1;
        }
         *__esp = 10;
        putchar();
        _v12 = _v12 + 1;
    }
    _v32 = 1;
    _v36 =  &y;
    asm(«fldz «);
    asm(«fstp qword [esp+0x28]»);
    _v48 = 1;
    _v52 =  &x;
    _v56 = 3;
    _v60 =  &m;
    asm(«fld1 «);
    asm(«fstp qword [esp+0x10]»);
    _v72 = 3;
    _v76 = 3;
```

```
    _v80 = 111;
     *__esp = 101;
    cblas_dgemv();
    _v12 = 0;
    while(_v12 <= 2) {
        __fp0 ?_?  *((long long*)
(_v12 * 8 +  &y));
        asm(«fstp qword
[esp+0x4]»);
         *__esp = «%5.1f\n»;
        printf();
        _v12 = _v12 + 1;
    }
    return 0;
}

__libc_csu_fini()
{// addr = 0x080485F0
    _unknown_ __ebp;
// r6

    return;
}

__libc_csu_init(
    intOrPtr _a4,
// _cfa_4
    intOrPtr _a8,
// _cfa_8
    intOrPtr _a12
// _cfa_c
)
{// addr = 0x08048600
    intOrPtr _v36;
// _cfa_ffffffdc
    intOrPtr _v40;
// _cfa_ffffffd8
    _unknown_ __ebx;
// r1
    _unknown_ __edi;
// r4
    signed int __esi;
// r5
    _unknown_ __ebp;
// r6
    _unknown_ _t14;
// _t14
    _unknown_ _t15;
// _t15
    signed int _t18;
// _t18

    __i686.get_pc_thunk.bx();
    _t15 = _t14 + 4529;
    __esp = __esp - 28;
    _init();
    _t18 = _t15 + -248 - _t15 +
-248 >> 2;
    if(_t18 == 0) {
    } else {
        __esi = 0;
        do {
            _v36 = _a12;
            _v40 = _a8;
             *__esp = _a4;
             *((intOrPtr*)(_t15 +
-248 + __esi * 4))();
            __esi = __esi + 1;
        } while(__esi < _t18);
    }
    __esp = __esp + 28;
    return;
}

__i686.get_pc_thunk.bx()
{// addr = 0x0804865A
    return;
}

__do_global_ctors_aux()
{// addr = 0x08048660
    intOrPtr* __ebx;
// r1
    _unknown_ __ebp;
// r6

    __eax = __CTOR_LIST__;
    if(__eax == 255) {
    } else {
        __ebx =  &__CTOR_LIST__;
        asm(«o16 nop «);
        do {
            __ebx = __ebx - 4;
             *__eax();
            __eax =  *__ebx;
```

```
        } while(__eax != 255);
    }
    return;
}

_fini()
{// addr = 0x0804868C
    _unknown_ __ebx;
// r1
    _unknown_ __ebp;
// r6
    _unknown_ _t1;
// _t1

    __esp = __esp - 4;
    L1();
    _pop(__ebx);
    __do_global_dtors_aux(__esi);
    _pop(__ecx);
    return;
}

L08048698()
{
    _unknown_ _t1;
// _t1

    _pop(__ebx);
    __do_global_dtors_aux(__esi);
    _pop(__ecx);
    _pop(__ebx);
    __esp = __ebp;
    _pop(__ebp);
    return;
}

L08048698()
{
    _unknown_ _t1;
// _t1

    _pop(__ebx);
    @rec __do_global_dtors_aux@__
do_global_dtors_aux@(__esi);
    _pop(__ecx);
    _pop(__ebx);
    __esp = __ebp;
    _pop(__ebp);
    return;
}

// Statistics:
//      74 Register nodes
//      35 Temporaries nodes
//       5 Casts
//     207 Statements
//       2 Labels
//       1 Gotos
//      17 Blocks
//     469 Nodes
//      10 Assembly nodes
//      27 Unknown Types


Total time: 0 seconds.
```

Структура программы (вначале) в общем несколько лучше, чем у Boomerang, и куда больше подробностей.

2. Бен, ай нид хелп: MEX-файл, написанный на C + BLAS, исходников которому нет.

Этот пример в посте приводить не стану, так как он длинный, но желающим попробовать своё декомпиляйшн-кунфу такая возможность предоставится:

Некоторые входные данные: это оптимизационный алгоритм для Quadratic Programming типа Branch-and-Bound. Алгоритм в целом прост и незатейлив, но самая сложная часть в нём - определить lower/upper bound через решение упрощённой оптимизационной задачи, и делать это быстро. Как такое сделать - хороший вопрос, и именно он меня интересует более всего.

Короче, важен не столько алгоритм, сколько его составные компоненты (стратегия и подпрограммы для lower bound estimation).

Автор этих строк, поковыряв выхлоп RecStudio, нашёл для себя подсказку на строчке 12319:

qps_mq_sub(.....

и особенно на строчке 12088:

getalp(....

что позволило автору предположить, что для lower/upper bounds используется Liner Programming. Не слишком много, но по крайней мере понятно, в какую сторону копать.

EОбновление: теперь можно сравнить выдачи Boomerang, RecStudio, и (спасибо, Григорий!) IDA Pro. В самом деле, выдача IDA Pro куда лучше того, что дают остальные, особенно boomerang. Можно выудить (до некоторой степени) структуру программы и даже сообщения об ошибках.

Ссылки

Интересующийся читатель может попробовать полистать вебстраницы автора RecStudio с полезной информацией, сходить на wiki-ресурс по обратной разработке.

Помимо познавательных диссертаций Michael James Van Emmerik (Boomerang) и Cristina Cifuentes [C. Cifuentes, Reverse Compilation Techniques], есть хорошие книжки по теме:

- «Compilers - Principles, Techniques and Tools», Aho, Sethi, Ullman, 1986 Addison-Wesley Publishing Co. ISBN 0-201-10088-6.
- «Advanced Compiler Design & Implementation», Steven Muchnick, 1997 Morgan Kaufmann Publishers, ISBN 1-55860-320-4.
- «How debuggers work - Algorithms, Data Structures, and Architecture», Jonathan Rosemberg, 1996 John Wiley and Sons, ISBN 0-471-14966-7.

Дело это интересное, но весьма утомительное, хотя может помочь при раскопках очередного legacy-software и сэкономить вам полжизни.

По материалам сайта:
mydebianblog.blogspot.com

Магазин "TOTAL"

- персональные компьютеры;
- компьютерные комплектующие;
- ноутбуки, нетбуки, планшеты;
- принтеры, МФУ, расходники;
- сетевое оборудование;
- CD/DVD диски, флеш-накопители;
- и многое другое.

г. Кривой Рог, ул. Адмирала Головко, 40, Терновской р-н
тел. (067)-698-87-79, (097)-692-73-38

Блокировка Сетей TOR

Список точек выхода сети Tor, с которых инициируются исходящие соединения, можно загрузить на официальном сайте проекта Tor:

https://check.torproject.org/cgi-bin/TorBulkExitList.py

или используя неофициальные архивы:

http://torstatus.info/ip_list_all.php/Tor_ip_list_ALL.csv

https://www.dan.me.uk/torlist/

По указанным адресам выдаются списки IP-адресов, которые можно использовать для блокирования межсетевым экраном/HTTP-сервером или привязки метки через модуль geo_ip.

Пример блокирования через ipfw:

```
ipfw table 1 flush
curl https://check.torproject.org/cgi-bin/TorBulkExitList.py?ip=8.8.8.8 | grep -v «^#» | while read cnt ip; do
/sbin/ipfw table 1 add $ip 1
done
ipfw add deny all from «table(1)» to any via em1
```

Для выставления флага через модуль ngx_http_geo_module в nginx:

```
curl https://check.torproject.org/cgi-bin/TorBulkExitList.py?ip=8.8.8.8 |
grep -v «^#» | sed ‘s/$/ tor;/’ > /usr/local/etc/nginx/tor.conf
```

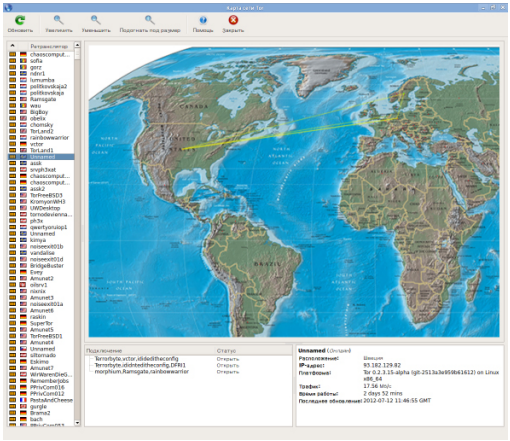
в nginx.conf:

```
geo $country {
include /usr/local/etc/nginx/tor.conf
}
```

далее для блокировки можно добавить проверку:

```
if ($country = ‘tor’) {
...
}
```

По материалам сайта:
armanenshaft-linux.blogspot.com



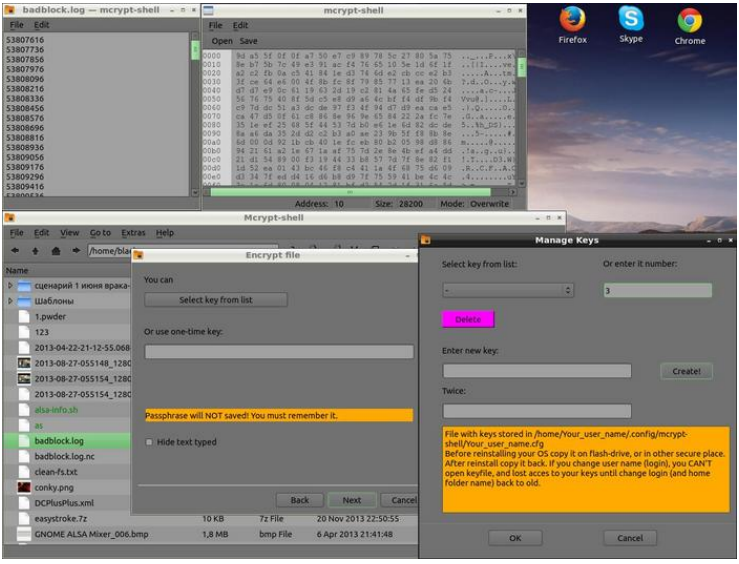
В рамках проекта mcrypt-shell подготовлена графическая оболочка к mcrypt

Опубликован первый выпуск Mcrypt-shell, графического интерфейса, позволяющего обычным пользователям, не знакомым с консолью, шифровать файлы с использованием более 10 симметричных алгоритмов, поддерживаемых mcrypt. Интерфейс построен с использованием Qt. Код распространяется под лицензией GPLv3.

Работа с программой организована в форме последовательных шагов, понятных даже неискушенному пользователю. Ключи, с помощью которых производится шифрование могут быть сохранены в зашифрованном мастер-паролем файле, либо каждый раз вводиться вручную. Во время работы не создаются временные файлы и не ведутся логи. В зашифрованный файл не сохраняется информация о типе шифрования, контрольной сумме, и прочие сигнатуры, в отличие от pgp/gnupg. Таким образом доказать, что файл был зашифрован, а не просто поврежден в результате сбоя диска, невозможно.

Имеется встроенный текстовый и шестнадцатеричный редактор, поддерживается архивирование в *.tar.gz, возможно добавление пользовательских команд в контекстное меню. На данный момент поддерживаются русский и английский языки. При желании сообщества будут добавлены другие переводы.

По материалам сайта:
www.opennet.ru



7 способов улучшения процесса разработки адаптивного дизайна

Адаптивный дизайн стремительно становится индустриальным стандартом, что влечет за собой целую карусель передового опыта, платформ и инструментов. В результате происходит сдвиг в мышлении специалистов и, в частности, в понимании того, как адаптировать рабочий процесс для повышения его эффективности.

Конечно, из-за того, что процесс работы над адаптивным проектом очень индивидуален и итеративен, проанализировать его и дать решения на все случаи жизни довольно трудно. Тем не менее, некоторые общие способы и техники можно применять практически всегда.

Мы рассмотрим семь техник по улучшению адаптивного дизайна начиная со структуры контента и заканчивая масштабируемыми изображениями.

Мобильная версия прежде всего

Подход «mobile first» с его постепенными улучшениями, охватывающими основные аспекты проектирования интерфейса, поможет вам сфокусироваться и понять ограничения мобильной среды для более эффективной и инновационной работы.

Это значит, что сначала создается дизайн и контент, оптимизированные для простейших устройств. Затем расширяется оформление для девайсов с небольшими экранами и поддержкой Media query. В заключение шаблон и контент улучшаются для десктопов. Количество пользователей, выходящих в сеть со смартфонов, продолжает расти бешеными темпами, и подход «mobile first» внедрил даже Google.

Проектирование для мобильных устройств должно отныне стать нашей стартовой точкой, а не концом работы. Этот подход заставляет фокусироваться на нуждах пользователей, обращать внимание на важный контент, проектировать оптимизированные и быстро загружае-

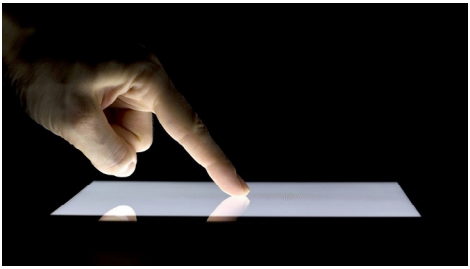
мые страницы, а затем улучшать все шаг за шагом.

Как отмечают члены команды ZURB, стоящей за популярным CSS-фреймворком Foundation:

Подход mobile first не в том, чтобы концентрироваться на проектировании для мобильных устройств, он также используется для улучшения юзабилити сайтов, более эффективного использования веб-пространства и сокращения ненужных элементов на веб-страницах.

Несмотря на молодость этой техники проектирования и большого количества технических сложностей, которые возникают при ее реализации, использование этого подхода означает, что вы создаете адаптивную, сфокусированную, лаконичную и перспективную основу своих продуктов.

Контентная стратегия



Цель адаптивного дизайна заключается в том, чтобы сделать возможным наилучший пользовательский опыт в любомконтексте. Создание адаптивного сайта – отличный повод для того, чтобы взглянуть на контент другими глазами, сделать его более читаемым и легкодоступным, вне зависимости от того, на каком устройстве его просматривает пользователь. Неважно, выберите ли вы постепенное ухудшение или прогрессивное улучшение, применение методологии «content out» поможет разработать контентную стратегию, которая ставит во главу угла пользователя.

Как сказали в UXMag: для того чтобы в каждом конкретном случае показывать пользователю правильный контент, необходима стратегия, которая обеспечит пользователя нужным контентом на каждом этапе его пути.

Разработка информационной инфраструктуры и контентной стратегии означает, что контент должен создаваться на базе каркаса и структуры, основанных на исследовании пользователей и их потребностей.

Эти знания и минимально необходимое для создания полезного дизайна количество контента станут основой, к которой можно добавлять большие экраны и расширения. Одинаково важно ответить на ключевые вопросы о том, как контент каждого типа соотносится с целями сайта, и установить, какой контент будет способствовать выполнению этих целей. Если контент хорошо структурирован на начальных стадиях проекта, будьте уверены, что его легко перенести на другие платформы.

Как красноречиво отметила автор издания Content Everywhere Сара Вахтер-Бётчер (Sara Wachter-Boettcher): если говорить о контенте, как о чем-то, что определяет форму и содержание, то станет ясно, что контент это даже не первоочередная, а основополагающая вещь.

Скетч и Прототип

После разработки контентной стратегии и инструментария, начинайте внедрение скетчинга в процесс адаптивного проектирования – это поможет создать умный, организованный макет, который легко масштабируется.

Сегодняшнее разнообразие размеров экранов, разрешений и возможностей устройств означают большее количество макетов, которые нужно спланировать. При помощи скетчей, можно вынести на обсуждение грубые наброски и идеи, чтобы в результате заложить основы для будущего каркаса и прототипа проекта. Создание скетчей портативно, доступно и дает гибкость для творчества и создания итеративных, эволюционных макетов, без траты огромного количества времени на рисование бесконечных мокапов в Photoshop. Кроме того, визуальная и контентная точность бумажных макетов довольно низка, а это значит, что вы действительно сможете сфокусироваться на взаимодействиях пользователя и потоке контента, вместо того, чтобы обращать внимание на то, как это будет в итоге выглядеть. В общем, ключевая идея скетчинга заключается в том, чтобы ставить во главу угла адаптируемость содержания, а не формы.

Существует громадное количество ресурсов для скетчинга, а в Responsive Sketchsheets от ZURB также включен и подход mobile first, так что с его помощью можно двигаться от полноразмерных мобильных страниц к миниатюрам эскизов для десктопных макетов. Или, при желании, просто двигаться вниз от десктопной версии. Можно даже заниматься скетчингом на iPad. Инструмент не имеет особого значения – выбирайте тот, который подходит именно вам и позволяет осуществлять быстрые итерации.

Получив набор скетчей, можно переходить к прототипированию, то есть созданию простого HTML-шаблона (или работающей черновой версии), с целью определить, работают ли в заданном контексте структура контента, навигация, разметка и точки прерывания. Они дают возможность совершенствования функциональности и взаимодействия и, благодаря минимальному визуальному оформлению, не отвлекают внимание от структуры и контента. Для прототипирования можно использовать многочисленные фреймворки и платформы, а можно и делать все самостоятельно.

В определенный момент на стадии скетчинга и прототипирования вы можете подумать о том, что было бы хорошо создатьбиблиотеку паттернов – динамическую, документированную проектную библиотеку базовых элементов и паттернов, которая используются в качестве отправной точки и затем дополняется. Такая библиотека поможет обеспечить более гибкий рабочий процесс и будет особенно полезна группам, работающим вместе. Существует несколько ресурсов – один из самых впечатляющих это Rock Hammer от Stuff and Nonsense. Используйте его в качестве базы паттернов разработки и дизайна или как основу для создания адаптивного сайта.

Как отметил Мэтт Гриффин (Matt Griffin)

в тексте для A List Apart, скетчинг и прототипирование дают нам возможность «переосмыслить весь подход к адаптивному проектированию для веба и помогают перестать разрабатывать «вебсайты» и «мобильные сайты», а вместо этого сконцетрироваться на создании гибких систем доставки контента с набором правил, которые определяют представление контента при изменениях контекста».

Фреймворки

Выбор CSS фреймворка зависит от идеологии проекта и личных предпочтений, но внедрение его в рабочий процесс создания адаптивного сайта имеет массу преимуществ. Использование фреймворков может ускорить процесс разработки, нормализовать стандартные проблемы кросс-браузерной совместимости и привнести структурированный подход в проектирование на основе сетки – все, что так необходимо в процессе создания адаптивного сайта. Пожалуй, главным плюсом фреймворков является необходимость минимального тестирования и дебаггинга, что возможно благодаря тому, что фреймворк исправляет ошибки, относящиеся к конкретному браузеру, и уже был протестирован на большинстве браузеров и мобильных устройств.

При выборе фреймворка важно обращать внимание на несколько вещей. Какой объем обучения необходим для начала работы? Каково количество доступной документации, и на каких условиях предоставляется поддержка? Каковы функциональные возможности фреймворка?

Один из фрейморков, реализующих подход mobile first – это Foundation от команды ZURB. Его последняя версия была перестроена с нуля, на основе методологии mobile first Люка Вроблевски (Luke Wroblewski) – теперь макет, который вы делаете, будет изначально построен для небольшого устройства. Его сетка с 12-ю столбцами может масштабироваться до значительного размера и при этом легко «сворачивается», так что вы можете создавать сложные макеты, с сеткой, которая уменьшена, чтобы уместить все необходимые для мобильных устройств столбцы один над другим. Она адаптируется к различным размерам экранов посредством как процентных ширин, так и media queries. Теперь Foundation поддерживает большее количество устройств, и работа с ними стала более интеллектуальной. Этот фреймворк сочетает в себе гибкость, мощь и использует препроцессор Sass, так что вы можете удалить все презентационные классы и написать точную семантическую разметку по своему выбору, сохранив дополнительные преимущества компонентов фреймворка. Сейчас Foundation поставляется в комплекте с дополнениями и расширениями Sass почти для каждого компонента.

Одно лишь использование адаптивного CSS-фреймворка не решит всех проблем проектирования и разработки, однако, эти средства стоят того, чтобы потратить время на их изучение, и могут послужить основой ваших будущих адаптивных проектов.

Точки прерывания

Распространено мнение, что точки прерывания должны основываться на общих размерах экранов (мобильных, планшетных, десктопных). Однако при разработке под конкретные разрешения устройств мы не полностью раскрываем потенциал адаптивного дизайна, который строится вокруг подвижности, гибкости и приспособляемости.

Поскольку контент – это краеугольный камень веба, имеет смысл использовать подход «отрицания устройств» (device-agnostic) и задавать точки прерывания в соответствии с контентом. Изучите ваш дизайн, найдите точки, где действительно возникают проблемы, и улучшайте слабые места. Пытаться подобрать каждому новому расширению свою точку прерывания – непрактично, ведь новые устройства появляются постоянно.

Плюс этого подхода, по мнению дизайнера и автора Эллиота Джей Стокса (Elliot Jay Stocks) заключается в том, что:

Если вы улучшаете свой дизайн, когда он выглядит хорошо, вам не придется беспокоиться о модернизации media queries для новых устройств.

Так что же означает «контентный» подход к точкам прерывания?

Это значит, что вы определяете основу структуры исходя из контента, а затем фокусируетесь на том, что происходит с макетами. Этот подход дает больше информации о микродеталях поведения контента в подвижном контексте, потому что вы фокусируетесь не на контроле дизайна в формате страниц, а на принципах дизайна, объединяющих страницы, считает дизайнер Марк Боултон (Mark Boulton).

Еще один подход к определению точек прерывания был озвучен автором книги Implementing Responsive Design Тимом Кадлецем (Tim Kadlec) и заключается в том, чтобы менять ширину и высоту окна браузера и смотреть, какие моменты можно улучшить, позволяя контенту выступать в роли проводника. Начните с мобильного представления (около 300px) и увеличивайте окно браузера до тех пор, пока вещи не начнут выглядеть требующими улучшения. Это будет первой точкой прерывания – установите для нее media query и решайте возникающие вопросы. Повторяйте эти действия, пока не добьетесь приемлемого диапазона между точками. Устанавливая точки прерывания в величинах REM и EM, вместо пикселей (вопрос на Хабре по теме — прим. перев.), вы добьетесь большой степени гибкости.

Кроме того, этот метод не отнимает много времени. Дрю Томас (Drew Thomas) утверждает:

На то, чтобы добавить «специальную» media query требуется столько же времени, что и на поиск решения, которое позволит контенту работать без дополнительных media query.

Мы не защищаем тезис о том, что «точки прерывания мертвы», скорее мы считаем, что контент должен наполнять наши media queries; единственный способ создать дизайн, который будет работать на любом устройстве – это забыть обо всех устройствах. Вы даже можете пойти дальше и обратиться к классической теории читабельности, чтобы определять точки прерывания на ее основе.

Масштабируемые изображения

Один из вызовов адаптивного дизайна – это работа с изображениями. Они должны быть подвижными для того, чтобымасштабироваться для соответствия как проекциям, так и тексту. Официальной спецификации от W3C по этому вопросу до сих пор не существует, так что остается только самостоятельно искать новые решения и работать с ними.

Одним из таких решений является использование ПО Adaptive Images, которое построено на эксперименте компании Filament Group. Adaptive Images использует один файл .htaccess, один php-файл и одну строчку кода JavaScript для опре-

деления размера экрана посетителя сайта. Затем модуль автоматически создает, кэширует и показывает подходящее конкретному устройству масштабированную версию встроенных HTML-изображений. Важно подумать и о том, как изображения будут урезаться при уменьшении – средства вроде Focal Point и ReSRC.it помогают интеллектуально обрезать изображение, так чтобы главный фокус изображения не терялся при просмотре на устройствах меньшего размера.

С появлением нового поколения retina-дисплеев с высокой плотностью пикселей, увеличилась важность оптимизации изображений для правильного масштабирования. Одним из способов создания адаптивных retina-изображений это использование средства CSS Sprites. Для работы с дисплеями с высоким разрешением вам нужны два изображения – в обычном (@1x) и высоком (@2x) разрешении, что означает дублирование файлов, селекторов и ссылок в CSS. При использовании CSS Sprites, однако, «вам всего лишь нужно переопределить ссылку на @1x спрайт-файл для всех селекторов, которые включают атрибуты высокого разрешения», говорит Мэйкел Луманс (Maykel Loomans). Эта техника снижает количество сетевых запросов, уменьшает размер файла стилей и позволяет добиться более эффективного процесса создания изображений для retina.

Тем не менее CSS Sprites работает только с изображениями, отмеченными в вашем CSS. Для картинок на веб-страницах Imulus разработала крайне полезный плагин Retina.js, который проверяет ваш сервер на наличие путей к изображениям с @2x на конце.

Еще один совет – урезание ненужных изображений с помощью Icon Fonts, которые очень легко масштабировать, редактировать и поддерживать, благодаря чему получаются более легкие сайты.

Еще одним медиа-ресурсом, требующим правильного масштабирования, является видео – оно должно быть гибким и масштабироваться от экрана к экрану. Здесь все чуть сложнее, чем в случае с изображениями, но jQuery-плагины

(напр.Fitvid.js и Fluidvid.js) помогут облегчить разработку.

Минификация

Вы потратили время на создание красивого адаптивного сайта, но предприняли ли вы что-то для оптимизации производительности? Изображения, JavaScript, CSS, библиотеки – все это нужно загружать, что значит большее время загрузки и количество HTTP-запросов.

Минификация – это практика удаления ненужных символов из кода (без потери его функциональности) для уменьшения его размера и повышения скорости загрузки. Этими ненужными символами могут быть пробелы, табы, переводы строк и комментарии, которые удаляются для минификации кода, оставляя при этом все его качества неизменными, но сокращая общий вес. Чтобы сократить вес еще больше, можно установить уровень минификации (особенно для JavaScript), который будет более агрессивно заменять переменные и имена функций на одиночные буквы. Независимый веб-разработчик Майк Беквит (Michael Beckwith) рекомендует сохранять неминифицированные копии CSS и JavaScript-файлы для рабочей копии сайта, а на «боевой» версии использовать минифицированные.

Минифицровать CSS и JavaScript-файлы можно с помощью большого количества инструментов: CSSTidy, Minify, JSMin, YUI Compressor и SquishI – это лишь некоторые из них. Эти средства можно даже сравнить. В конечном счете, выбор инструмента минификации зависит от вашего кода и рабочего процесса.

Помимо минификации кода, можно объединять CSS и JavaScript в общие файлы для сокращения количества HTTP-запросов. Это может быть довольно трудно реализуемо, если скрипты отличаются на разных страницах, но для уменьшения времени отклика – дело стоящее. Еще один метод оптимизации производительности – это включение Gzip-компрессии. После активации на сервере, она отправляет браузеру .zip файл, вместо файла .html. Затем браузер загружает архив, распаковывает его и показывает содержимое пользователю. В результате, вместо загрузки 100 килобайтного файла html, после компрессии загружается файл весом в 15 килобайт, а значит посетитель сайта будет взаимодействовать с более компактным, быстрым, сжатым контентом, и время загрузки уменьшится. Не стоит забывать, что снижение времени загрузки напрямую влияет на выручку, так что налицо сразу несколько плюсов.

По материалам сайта:
habrahabr.ru

Использование фильтров в Wireshark

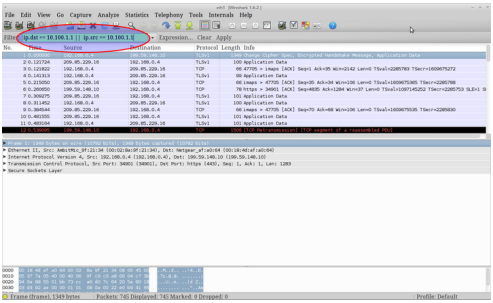
Некоторые люди считают искусство захвата и интерпретации пакетов, передающихся по сети таким же сакральным знанием, как чтение матрицы из одноименного фильма, но вам не нужно быть новым Нео для того, чтобы иметь возможность исследовать сетевой поток. Мощным союзником в этой миссии является Wireshark - программный инструмент для анализа сетевого трафика.

Wireshark сочетает в себе несколько инструментов. Вы можете использовать эту программу для анализа структуры вашей беспроводной сети и поиска возможных ошибок в ее конфигурации. Она может идентифицировать множество типов инкапсуляции, при этом разделить и показать все поля, из которых состоит сетевой пакет. Она также работает как сниффер пакетов аналогично программе tcpdump.

Учитывая мощные возможности Wireshark, можно подумать, что будет сложно обучиться использовать эту программу. В каком-то смысле это так, но вы можете без лишних усилий разобраться с тем, как использовать часть фильтров из комплекта поставки программы, и сконцентрироваться на исследовании интересующих клиентов и видов трафика. В этой статье я продемонстрирую ряд вариантов использования Wireshark для облегчения ваших поисков.

Когда я говорю «фильтры», я имею в виду Фильтры Пакетов Беркли (Berkeley Packet Filters (BPF)). На самом деле BPF является микроязыком программирования (содержащим мнемоники на ассемблере!), выражения которого компилируются и выполняются в отношении пакетов, обрабатываемых такими программами, как tcpdump и Wireshark. Фильтры просто необходимы в том случае, когда вам хочется отделить небольшую часть пакетов из сотен тысяч пакетов, передающихся по сети с предельной скоростью 100 Мбит/с. Выражения фильтров компилируются, поэтому обработка пакетов происходит с максимально возможной скоростью, что очень важно при осуществлении захвата в реальном времени.

Фильтры в Wireshark достаточно просто использовать. Вам необходимо знать только названия полей для каждого протокола, такие, как http, icmp и ftp. Например, если вы хотите, чтобы были показаны только пакеты ICMP, вы можете просто написать icmp в строке фильтра в главном окне Wireshark. Если вы хотите выделить все пакеты, идущие в обоих направлениях относительно узла с адресом, к примеру, 10.100.1.1, выражение фильтра будет записано в форме ip.dst == 10.100.1.1 || ip.src == 10.100.1.1, что расшифровывается как указание показывать только те пакеты, поле адреса назначения (ip.dst) или (||) поле адреса источника (ip.src) IP-протокола для которых соответствует (==) 10.100.1.1.



Фильтры захвата и отображения

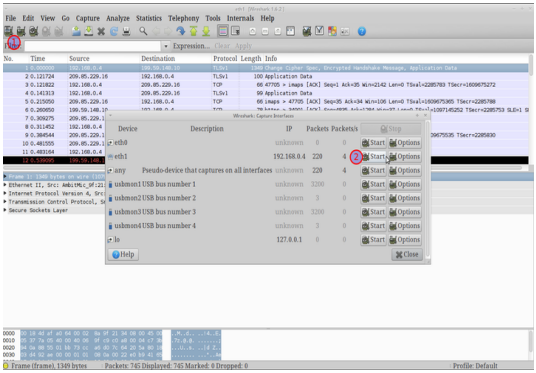
Wireshark поддерживает два типа фильтров. Фильтры захвата, как понятно из названия, используются для захвата части трафика, в то время как фильтры отображения могут применяться к захваченному трафику для показа только части пакетов в зависимости от использованного выражения. В этой статье поговорим об обоих типах фильтров.

Давайте начнем с установки Wireshark. Приложение доступно в виде бинарных пакетов для всех основных дистрибутивов, поэтому вы можете использовать ваш любимый менеджер пакетов: sudo apt-get install wireshark в Debian или Ubuntu, emerge wireshark в Gentoo или yum install wireshark в Red Hat или CentOS.

Давайте начнем с классического примера, демонстрирующего то, почему использование протокола FTP может считаться плохой идеей. Запустите Wireshark при помощи команды в терминале:

```
sudo wireshark
```

Вы можете начать захват трафика, перейдя на левую панель окна Wireshark и нажав кнопку «Capture/Interfaces». Выберите интерфейс, который «направлен в сеть» (например, eth1) и нажмите кнопку «Start», при этом Wireshark начнет исследование всех пакетов, передающихся в сети.



Теперь перейдите во второе окно терминала и установите обычную сессию FTP. Введите имя пользователя и пароль, выполните несколько команд FTP, после чего закройте сессию. Вернитесь к окну Wireshark, в котором вы должны увидеть множество пакетов, переданных по сети с момента начала захвата. Нажмите на кнопку «Stop Capture» (или используйте сочетание клавиш Ctrl+E); после этого вы можете исследовать полученный трафик.

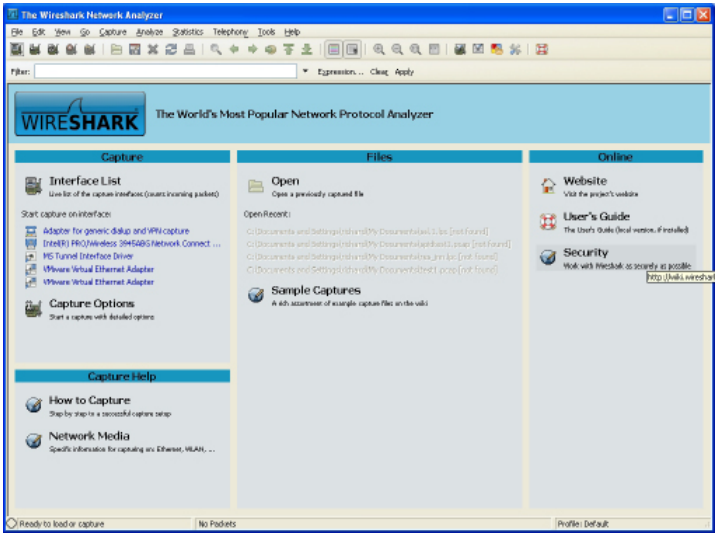
Понять «что есть что» в большом объеме захваченного трафика не так просто до того момента, как вы используете фильтр BPF. Хотелось бы сделать так, чтобы были показаны только пакеты, участвующие в работе FTP-соединения, поэтому в поле фильтра следует просто написать «ftp». Незамедлительно после этого будет выделен трафик, относящийся к вашей сессии FTP и в качестве потрясающей демонстрации проблем с безопасностью, вы сможете увидеть ваше имя пользователя и пароль.

Вывод будет похож на следующий:

```
356     101.676753
10.100.1.1      192.168.0.4
FTP     86      Response: 220
(vsFTPd 1.1.3)
360     104.546659
192.168.0.4     10.100.1.1
FTP     77      Request: USER
wazi
362     104.594520
10.100.1.1      192.168.0.4
FTP     100     Response: 331
Please specify the password.
366     106.530150
192.168.0.4     10.100.1.1
FTP     77      Request: PASS
mytest
371     108.922240
10.100.1.1      192.168.0.4
FTP     88      Response: 530
Login incorrect.
```

Если результат этого исследования не склонил ваших коллег к отказу от использования FTP и переходу на OpenSSH, они безнадежны.

По материалам сайта:
rus-linux.net



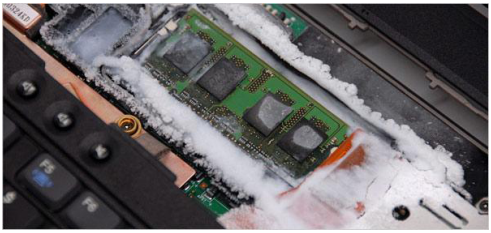
Получаем образ оперативной памяти

В рамках расследований компьютерных преступлений (http://ualinux.com) иногда возникает необходимость получить содержимое оперативной памяти.

Оперативная память - это важный источник информацией при изучении предыдущих действий с машиной. В ней могут содержаться как части самих исполняемых процессов, так и части удаленных файлов, пользовательских сессий, криптографических ключей. При современном распространении сложных систем защиты информации, основанных на криптовании восстановление их ключей становиться чуть-ли не одной из основных задач для исследования. В защищенных системах зачастую оперативная память это единственное место где могут сохраниться защитные ключи и другая временная, но очень важная информация.

Процесс получения информации, которая содержится в оперативной памяти состоит из двух этапов: изъятие содержимого оперативной памяти и анализ полученных данных после изъятия.

Обращая внимание на первый этап стоит заметить, что изъятие оперативной памяти может быть выполнено с помощью ряда средств: непосредственный доступ к памяти с использованием специальных плат расширения, порта FireWire, и даже физическом изъятии запоминающего устройства оперативной памяти (потребует замораживания плат), но в данном



материале мы рассмотрим программные средства, которые позволяют изъять содержимое оперативной памяти защищенных машин путем так называемой «горячей» перезагрузки и запуска машины в Live-режиме.

Для выполнения этой задачи будем использовать специальный дистрибутив **Ubuntu CyberPack (IRF) 1.0** (http://ualinux.com), состоящий из минимального набора компонент, а именно, только те, которые необходимы для

изъятия данных из памяти. Соответственно отсутствует и графический интерфейс.

Загрузить дистрибутив можно отсюда - http://ualinux.com

Использование такого подхода к изъятию содержимого оперативной памяти имеет ряд преимуществ и недостатков сравнительно с другими перечисленными выше средствами.

Положительные стороны:

- использование Live-дистрибутива позволяет проводить действие вне зависимости от того какая операционная система установлена на исследуемой машине;

- отсутствуют затраты на приобретение дорогостоящих специальных устройств, кабелей, плат, и др.

Недостаток:

- содержимое оперативной памяти будет неполным — ее часть будет перезаписана данными, необходимыми для запуска Live-дистрибутива (приблизительно 125 Мб).

Для использования доступны специально собранные дистрибутивы для машин с памятью объемом до 3 Гб (i386) и свыше 3 Гб (amd64). С их помощью можно создать загрузочный CD/DVD-диск или загрузочный USB-диск.

Замечания:

- **второго шанса система нам не дает - у нас есть только одна попытка**. т. е. при повторной перезагрузке исследуемого компьютера большая вероятность того что мы уже не найдем необходимой информации. Отсюда следует что не надо перезагружать его несколько раз, экспериментировать, прицеливаться.

Необходимо заранее подготовится и знать как компьютер себя поведет после перезагрузки.

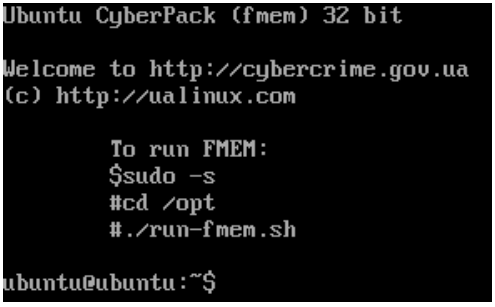
Большинство современных компьютеров позволяют прямо при старте указать откуда производить загрузку, но если этого нет, тогда необходимого настроить BIOS машины на загрузку с CD/DVD-привода или USB-привода/накопителя, после чего загрузить Live-дистрибутив с указанного устройства.

Итак, приступим.

Перезагружаем компьютер.

ВАЖНО: перезагрузка ни в коем случае не должна быть холодной (путем нажатия кнопки «ресет» или выключение\включение питания), а именно — перезагрузка должна быть осуществлена средствами самой работающей системы (например нажатием кнопок Ctrl-Alt-Del или путем выбора пункта «перезагрузка» в системе)

После загрузки дистрибутива пользователю доступна привычная строка консоли Linux, и краткая информация для запуска модуля.



Подготовка к работе программы fmem заключается в выполнении следующих команд:

```
$ sudo -s
```

```
# cd /opt (переход в папку где находиться программа);
# ./run-fmem.sh (скрипт запуска модуля съема памяти).
```

Замечание: Для дальнейших действий понадобиться примонтировать заранее подготовленный носитель (внешний жесткий диск, флеш-накопитель) с файловой системой ext2/3/4, в который будет сохраняться файл с содержимым оперативной памяти.

Для того, что бы узнать какой идентификатор присоединенному носителю присвоила система, необходимо после его подключения к компьютеру ввести следующую команду:

```
# dmesg (Команда выводит на экран информацию буфера сообщений ядра. Нас будет интересовать последняя запись.)
```

Как например вот это:

```
[16091.995428] sd 9:0:0:0:
Attached scsi generic sg2 type 0
[16091.995996] sd 9:0:0:0: [sdb]
32096120 512-byte logical blocks:
(16.4 GB/15.3 GiB)
[16091.998192] sd 9:0:0:0: [sdb]
Write Protect is off
[16091.998205] sd 9:0:0:0: [sdb]
Mode Sense: 0b 00 00 08
[16091.999433] sd 9:0:0:0: [sdb]
No Caching mode page found
[16091.999447] sd 9:0:0:0: [sdb]
Assuming drive cache: write
through
[16092.003486] sd 9:0:0:0: [sdb]
No Caching mode page found
[16092.003495] sd 9:0:0:0: [sdb]
Assuming drive cache: write
through
[16092.004251]
sdb: sdb1
```

(где «sdb» - присвоенное обозначение физического накопителя, а «sdb1» - присвоенное обозначение логического раздела накопителя).

Далее следует примонтировать логический раздел накопителя к папке /tmp загруженной в Live-режиме операционной системы:

```
# mount /dev/sdb1 /tmp
```

где

«mount» — команда монтирования устройства;

«/dev/sdb1» — адрес файла логического раздела присоединенного накопителя;

«/tmp» — папка в которую необходимо подключить накопитель.

Все подготовительные шаги сделаны — можно переходить к изъятию содержимого оперативной памяти:

```
# dd if=/dev/fmem of=/tmp/ram-image.mem bs=1K count=`head -1 /proc/meminfo | awk '{print $2}'`
```

где

«dd» - команда создания образа

«if=/dev/fmem» - источник данных, а именно оперативная память;

«of=/tmp/ram-image.mem» - запись в файл «ram-image.mem» в папку «/tmp»;

«bs=1K» - размер блока информации - 1 Кб;

«count=`head -1 /proc/meminfo | awk '{print $2}'`» - объем оперативной памяти, информация о которой извлекается из файла /proc/meminfo.

И ждем...

В результате удачного выполнения команды, мы получим сообщение похожее на это:

```
root@ubuntu:/tmp# dd if=/dev/fmem of=/tmp/ram-image.mem bs=1K count=`head -1 /proc/meminfo | awk '{print $2}'`
509232+0 records in
509232+0 records out
521453568 bytes (521 MB) copied, 158.405 s, 3.3 MB/s
root@ubuntu:/tmp#
```

```
521453568 bytes (521 MB) copied,
158.405 s, 3.3 MB/s
```

где

«521453568 bytes (521 MB) copied» - объем скопированной информации;

«158.405 s» - время в течении которого проводилась операция;

«3.3 MB/s» - скорость при которой проводилась операция.

В результате мы получили содержимое оперативной памяти машины в файле **ram-image.mem** на накопителе. Теперь его можно обрабатывать в т.ч. извлекая части исполняемых процессов, удаленных файлов, информацию о пользовательских сессиях, криптографических ключах и многое другое.

P.S.

Также стоит обратить внимание на то, что все современные системы используют в своей работе и swap-память (так называемый «файл подкачки»)

Файл подкачки – это своеобразное дополнение к оперативной памяти (которая занимается временным хранением данных для быстрой доставки их на обработку процессору) Вашего компьютера. Даже не столько дополнение, сколько её уширение или, можно сказать, продолжение. Дело в том, что когда не хватает оперативной памяти система может переносить данные из памяти на диск (так называемая дополнительная память), в котором соответственно также хранятся данные.

И для полной картины анализа памяти необходимо также получить и их.

Различные операционные системы используют разные способы их хранения.

В случае с Windows это обычно файлы в корне на системном диске C:

pagefile.sys для Win XP и Win 7 и достаточно просто скопировать файл

Для Linux - это отдельный раздел на носителе.

Например:

```
sudo fdisk -l /dev/sda покажет
нам все разделы в системе
/dev/sda1   *        2048
78125055    39061504   83  Linux
/dev/sda2         78125056
117186559    19530752   82  Linux
своп / Solaris
/dev/sda3        117186560
625141759   253977600   83  Linux
```

Исходя из чего мы видим что раздел подкачки находиться в /dev/sda2

Скопировать его можно также с помощию команды dd.

Например:

```
dd if=/dev/sda2 of=/media/<путь
куда записать>/linux-swap.dd
```

Для MacOS необходимо скопировать все файлы из директории **/private/var/vm/swapfile***

Обработка и анализ полученных результатов (как дампа оперативной памяти так и swap-памяти) может проводиться как в ручную с помощью например HEX-редактора, так и с помощью ряда программ о которых будет рассказано в следующий раз.

Марк Безруков
команда UserAndLINUX

ED.ua Школьный Электронный Дневник

Школа | Учительская | Профиль | Оплата | Обучение

Социальный проект компании "ВИТ" – Школьный электронный дневник

- Электронная база данных
- Персональный сайт школы
- Новости, события, праздники
- Связь с учителями и родителями
- Домашнее задание, оценки, замечания и поощрения
- Мобильная версия сайта
- Электронная очередь детских садов
- Отчеты, статистика, рейтинг школ

а также:

различные акции, скидки, праздники для наших пользователей!

Функции постоянно добавляются и модернизируются!

с ED.ua сбудется моя мечта!

ПОТОМУ ЧТО НА САЙТЕ ED.UA ЕСТЬ ПОЛНОЕ ДОМАШНЕЕ ЗАДАНИЕ!

ФЕОДОСИЯ
ФЛП Касьянова О. В. :
тел: +380991605920
+380950244989
http://ed.ua

ЛУГАНСК
ФЛП Турецкая З. В. :
тел: +380500311340
+380990631993
http://m.ed.ua

98112, Украина, АР Крым, г.Феодосия, ул. Крымская, 21-А Тел.: (06562) 7-44-79

Хотите стать частью команды UserAndLINUX?

Если вы давно используете Linux, либо только начали интересоваться продуктами OpenSoure, либо же просто интересуетесь компьютерными и техническими новинками, то мы с радостью примем вас в нашу дружную команду!

Каждый из нас именно так и попал в команду журнала UserAndLINUX – мы просто любим Linux и считаем, что обязаны нести эту любовь в массы. И мы знаем, как отплатить нашей любимой операционной системе – создать сообщество людей с общими интересами, поддерживать друг друга и помогать новичкам в этом интересном деле.

И не имеет значения, опытный ли вы программист, или одаренный школьник, или дизайнер, инженер, секретарь... Ваши идеи в совокупности с нашими могут помочь другим людям узнать, что такое Linux и с чем его едят!

Чем же вы можете нам помочь?

У вас есть творческие способности, креативное чувство стиля? Обладаете вкусом? Создавайте красивые темы и фоны (людям нравятся красивые картинки!) с командой наших дизайнеров!

Владеете иностранным языком? Переводите статьи с англоязычных ресурсов, чтобы наши читателя всегда имели удовольствие читать свежие интервью и новости со всего мира!

Любые навыки, которыми вы владеете, могут помочь команде UserAndLINUX – присоединяйтесь!
Предлагайте свои идеи. Учитесь вместе с нами. Развивайте новые умения, способности.

Спрашивайте. Задавайте вопросы, не стесняйтесь! Нам всегда нужны люди. Не думайте, что вы не сможете помочь, потому что вы не умеете программировать, администрировать или только начинаете работать в Linux как в системе, которая установлена у вас на компьютере.

Существует миллион способов внести свой вклад.
Присоединяйтесь к работе над журналом UserAndLINUX и приложением «Больше чем USER»!

Оставляйте свои вопросы, координаты и описание того, чем бы вы хотели помочь:
magazine@ualinux.com
на форуме ***http://ualinux.com/forum/userandlinux***
в нашей группе Вконтакте - ***https://vk.com/userandlinux***
или группе на Facebook - ***https://www.facebook.com/groups/userandlinux/***

С уважением,
коллектив журнала UserAndLINUX

User And LINUX

UA Linux
open source ■ open future

Адрес журнала в Интернете:
http://ualinux.com/journal

Обсуждение журнала
на форуме:
http://ualinux.com/forum

По вопросам
приобретения журнала:
http://ualinux.com/pay

Адрес редакции:
Украина, 03040,
г. Киев, а/я 56
Email: magazine@ualinux.com

Тип издания:
электронный

Регулярность: ежемесячный
Дата выпуска: 11.02.2014 г.
Тираж: свыше 60 000 загрузок*

*указано среднестатистическое ежемесячное суммарное значение, сформированное из полученной статистики загрузок журнала с официального сайта и других известных источников распространения (ftp, http и torrent)

Государственный реестр СМИ
Свид-во: КВ 18270-7070Р от 24.10.2011

Международный стандартный
серийный номер
ISSN: 2223-6988


User
And
LINUX
Больше чем user