



user And LINUX

№4 11.2013

Больше чем user

ISSN: 2223-6988

Язык программирования
BASIC в Ubuntu Linux

Криптография почтовых
сообщений

Инструменты для
исследования
сетей с интерфейсом
командной строки
часть 1

Создание и настройка
виртуальных сетевых
интерфейсов в Linux

Использование
планировщика
cron в Linux

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.



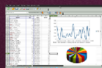
поддержка широкого спектра современного оборудования;
дополнительные драйвера для видео-карт, wi-fi адаптеров и принтеров;
возможность использовать Windows-драйвера для WiFi-адаптеров USB;
управление веб-камерами.



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 мультимедиа (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.

Мы рады приветствовать Вас на страницах приложения «Больше чем User». Мы постоянно модернизируем приложение для того, чтобы Вы не утратили к нему интерес. С этого номера у нас произошли изменения в дизайне и структуре самого журнала.

Сегодня мы представляем несколько рубрик: «Servers», «Workstations», «Console», «Programming» и «Security», в которых будут публиковаться статьи соответствующего направления.

Наше приложение ориентированно на программистов, разработчиков, начинающих системных администраторов и на всех тех, кто интересуется миром «IT». На страницах «Больше чем User» Вы найдете много интересного – профессиональные советы по настройке Вашей системы, «рецепты» для программистов и веб-разработчиков, особенности и подсказки по работе в консоли. Вместе с нами Вы узнаете о различных OpenSource-программах, дополнениях, которые сделают Вашу работу автоматизированной и легкой.

Наша команда – это простые ребята: сисадмины, программисты, веб-разработчики, инженеры, дизайнеры... Есть среди нас и люди, которые не работают в Linux. Но всех нас объединяет одно – тяга к инновациям и компьютерным технологиям. Мы – больше, чем пользователи. Мы – «айтишники», представляем Вам свежий выпуск приложения «Больше чем User».



С уважением, Якимчук Сергей.

НАД ВЫПУСКОМ РАБОТАЛИ:

Якимчук Сергей

Попов Владимир

Шарай Игорь

Россошанский Андрей

Звенигородская Анастасия

СОДЕРЖАНИЕ

SERVERS

Сканирование документов по сети.....5

Создание и настройка виртуальных сетевых
интерфейсов в Linux.....6

Apache Virtual Hosts несколько сайтов
на одном сервере UBUNTU / DEBIAN.....8

WORKSTATION

OpenOffice. Настройка замены прописных букв
на строчные по хоткею.....11

Битва за потерянные файлы.....11

CONSOLE

Использование планировщика cron в Linux.....16

18 советов по использованию команд apt и dpkg
в системах Debian и Ubuntu.....21

Пересборка пакетов в Debian.....25

PROGRAMMING

Обзор самых распространенных реализаций
языка Бейсик под Linux.....27

SECURITY

Криптография почтовых сообщений.....35

Настройка ufw в Ubuntu.....37

OTHERS

Инструменты для исследования сетей
с интерфейсом командной строки.....42

Сканирование документов по сети

В локальной сети у меня есть сканер. По своей конструкции он ни разу не сетевой. Но очень захотелось его таким сделать.

Для этого цепляем его к серверу и начинаем настраивать.

Для начала на сервере установим необходимые пакеты:

```
#aptitude install sane sane-utils\
libsane
```

Для доступа к сканеру по сети надо внести пользователя saned в группы scanner и lp:

```
#usermod -aG scanner,lp saned
```

Теперь для проверки правильности установки введем команду:

```
scanimage -Lq
```

Если все правильно, должна вывестись информация о подключенном сканере. Что-то типа:

```
device `xerox_mfp:libusb:001:007'
is a SAMSUNG ORION multi-function
peripheral
```

К сожалению, Sane это не сервис и принимать соединение по сети сам по себе не может. Поэтому установим еще и openbsd-inetd, который будет принимать соединения и передавать их Sane:

```
#aptitude install openbsd-inetd
```

Откроем файл /etc/services и ищем там строку sane-port 6566/tcp. Если ее там нет, то прописываем.

Еще в файле /etc/inetd.conf необходимо прописать:

```
sane-port stream tcp nowait
saned.saned /usr/sbin/saned saned
```

Теперь беремся за настройку самого Sane.

Открываем файл /etc/sane.d/saned.conf и прописываем в нем сеть, из которой будет разрешен доступ к сканеру. Например, 192.168.0.0/24.

Перестартовываем сервис openbsd-inetd:

```
#service openbsd-inetd restart
```

На этом настройка сервера закончена.

Переходим к настройке рабочей станции.

На ней установим нужные программы:

```
#aptitude install sane sane-utils\
libsane xsane xsane-common
```

Откроем на редактирование файл /etc/sane.d/net.conf и пропишем в нем адрес сервера со сканером. Что-то типа 192.168.0.1. Теперь в файле /etc/sane.d/dll.conf проверяем наличие строки net, если она есть и незакомментирована - все нормально. Иначе вписываем ее.

Проверить доступность сканера с рабочей станции можно при помощи все той же команды:

```
scanimage -Lq
```

Если и здесь сканер обнаружился - значит у нас все работает и мы можем начинать сканирование.

По материалам сайта:
yakim.org.ua

Создание и настройка виртуальных сетевых интерфейсов в Linux

Знаете ли вы, что можете присвоить более чем один IP-адрес физическому сетевому интерфейсу? Эта техника очень полезна, например при работе с Apache и виртуальными хостами, так как позволяет получить доступ к одному и тому же серверу Apache с двух разных IP-адресов.

Временный виртуальный сетевой интерфейс

Процесс создания виртуального сетевого интерфейса в Linux не занимает много времени. Он включает один запуск команды `ifconfig`.

```
$ ifconfig eth0:0 123.123.22.22
```

Приведенная выше команда создает виртуальный сетевой интерфейс, базирующийся на оригинальном физическом сетевом интерфейсе `eth0`. Самое важное условие для создания виртуального сетевого интерфейса - должен существовать физический сетевой интерфейс, в нашем случае `eth0`. Ниже приведен полный пример:

```
# ifconfig eth0
eth0      Link encap:Ethernet
HWaddr 3c:97:0e:02:98:c8
          inet
addr:192.168.100.23
Bcast:192.168.100.255
Mask:255.255.255.0

# ping 192.168.100.23
PING 192.168.100.23
```

```
(192.168.100.23) 56(84) bytes of
data.
```

```
64 bytes from 192.168.100.23:
icmp_req=1 ttl=64 time=0.023 ms
64 bytes from 192.168.100.23:
icmp_req=2 ttl=64 time=0.059 m
```

Теперь мы можем настроить новый виртуальный интерфейс на базе `eth0`. После выполнения команды `ifconfig` новый виртуальный интерфейс готов к немедленному использованию.

```
# ifconfig eth0:0
eth0:0    Link encap:Ethernet
HWaddr 3c:97:0e:02:98:c8
          UP BROADCAST MULTICAST
MTU:1500  Metric:1
          Interrupt:20
Memory:f1600000-f1620000
```

```
# ifconfig eth0:0 123.123.22.22
# ifconfig eth0:0
eth0:0    Link encap:Ethernet
HWaddr 3c:97:0e:02:98:c8
          inet
addr:123.123.22.22
Bcast:123.255.255.255
Mask:255.0.0.0
# ping 123.123.22.22
PING 123.123.22.22
(123.123.22.22) 56(84) bytes of
data.
64 bytes from 123.123.22.22:
icmp_req=1 ttl=64 time=0.060 ms
64 bytes from 123.123.22.22:
icmp_req=2 ttl=64 time=0.057 ms
```

Отключение виртуального сетевого интерфейса

Для отключения нашего, созданного ранее, временного сетевого интерфейса мы можем также использовать команду `ifconfig` с флагом `down`.

```
# ifconfig eth0:0 down
```

Присвоение виртуальному интерфейсу постоянного адреса

Описанные выше настройки не сохраняются после перезагрузки. Если вы хотите, чтобы виртуальный сетевой интерфейс работал постоянно, необходимо модифицировать конфигурационные файлы в соответствии с требованиями вашего дистрибутива Linux. Ниже описан этот процесс для самых распространенных дистрибутивов:

Debian / Ubuntu

Статический адрес

В Debian или Ubuntu вам необходимо отредактировать файл `/etc/network/interfaces`, добавив в него следующие строки:

```
iface eth0:0 inet static
address 123.123.22.22
netmask 255.0.0.0
broadcast 123.255.255.255
```

DHCP

Возможно также использовать виртуальный сетевой интерфейс с DHCP. В этом случае вам необходимо добавить в `/etc/network/interfaces` следующую строку:

```
iface eth0:0 inet dhcp
```

Для того чтобы изменения вступили в силу необходимо перезапустить сеть:

```
# /etc/init.d/networking restart
```

Redhat / Fedora / CentOS

Статический адрес

В Redhat, Fedora или CentOS Linux директория, отвечающая за присвоение постоянных IP-адресов, - это `/etc/sysconfig/network-scripts`. В этой директории необходимо создать файл, соответствующий вашему новому виртуальному интерфейсу. В нашем случае этот файл будет называться `ifcfg-eth0:0`. Создайте этот новый файл и вставьте в него приведенные ниже строки. После перезагрузки адрес будет присвоен виртуальному интерфейсу на постоянной основе.

```
DEVICE=eth0:0
IPADDR=123.123.22.22
NETMASK=255.0.0.0
NETWORK=123.0.0.0
BROADCAST=123.255.255.255
ONBOOT=yes
```

DHCP

```
DEVICE=eth0:0
BOOTPROTO=dhcp
ONBOOT=yes
```

Когда закончите, перезапустите ваши интерфейсы:

```
# service network restart
```

Заключение

Раньше один физический сервер обслуживал один веб-сайт. Сегодня такой способ хостинга уже не является жизнеспособным, поэтому способность операционной системы создавать виртуальные сетевые интерфейсы действительно необходима.

По материалам сайта:
rus-linux.net

Apache Virtual Hosts

несколько сайтов на одном сервере

UBUNTU / DEBIAN

Допустим на нашем сервере должны располагаться несколько сайтов и необходимо, чтобы при переходе по имени сайта открывались соответствующие этому сайту страницы, а при переходе по IP-адресу или по любому другому имени, кроме избранных, открывался сайт-заглушка, например со списком имеющихся сайтов или просто предупреждением. Все это можно организовать с помощью так называемых виртуальных хостов.

Установка

В Red-Hat-based дистрибутивах Apache называют httpd, в Debian и Ubuntu – apache2.

Команды установки:

- Fedora

```
yum install httpd
```

- Debian

```
apt-get install apache2
```

Основным конфигурационным файлом web-сервера является /etc/httpd/conf/httpd.conf и /etc/apache2/conf/apache2.conf соответственно.

В конфигурационный файл свежешустановленного web-сервера можно внести некоторые косметические исправления. Для этого среди многочисленных (и полезных) комментариев находим и раскомментируем директивы:

```
ServerAdmin admin@mysite.ru
# почтовый адрес администратора
```

```
ServerName mysite.ru:80 #имя сервера
```

В строке LanguagePriority переставляем ru на первое место:

```
LanguagePriority ru en ca cs da
de el eo es et fr he hr it ja ko
ltz nl nn no pl...
```

Структура

Сайты на сервере размещаются в отдельных каталогах. Удобнее, да и правильнее с точки зрения безопасности, создать для каждого сайта подкаталоги для хранения самих страниц, временных файлов, возможно cgi-скриптов и других полезностей.

Можно пойти ещё дальше и организовать для каждого сайта отдельного SSH-пользователя. В таком случае удобно размещать сайт в домашнем каталоге этого пользователя. Соответственно каталоги сайта будут иметь вид:

```
/home/site_user/mysite.ru/
httpdocs # для страниц
/home/site_user/mysite.ru/tmp
#для хранения сессий и временных
#файлов
/home/site_user/user/mysite.ru/
cgi-bin #для скриптов
```

Однако в небольших системах, где SSH-доступом обычно пользуется только администратор сервера, в этом нет необходимости. Поэтому сайты размещают внутри папки /var/www, которую по умолчанию использует web-сервер.

```
/var/www/vhosts/mysite.ru/
```

```
httpdocs # для страниц
/var/www/vhosts/mysite.ru/tmp
#для хранения сессий и временных
#файлов
/var/www/vhosts/mysite.ru/cgi-bin
#для скриптов
```

HOW-TO

Допустим наши сайты называются first.ru и second.ru.

Раскомментируем директиву
NameVirtualHost *:80

в третьей секции конфигурационного файла.

Создаем папки:

- общую

```
$ mkdir /var/www/vhosts
```

- для первого сайта

```
$ mkdir /var/www/vhosts/first.ru
$ mkdir /var/www/vhosts/first.ru/
httpdocs
```

- и для второго

```
$ mkdir /var/www/vhosts/second.ru
$ mkdir /var/www/vhosts/second.
ru/httpdocs
```

Для каждого сайта в конфигурацион-
ный файл добавляем отдельную секцию

```
<VirtualHost *:80>
ServerName first.ru
ServerAlias www.first.ru
DocumentRoot /var/www/vhosts/
first.ru/httpdocs
ErrorLog logs/first.ru-error.log
</VirtualHost>
<VirtualHost *:80>
ServerName second.ru
ServerAlias www.second.ru
DocumentRoot /var/www/vhosts/
second.ru/httpdocs
```

```
ErrorLog logs/second.ru-error.log
</VirtualHost>
```

С реальными сайтами разобрались, те-
перь что произойдет с обращении к сер-
веру по несуществующему адресу или
при обращении по IP? Если оставить все
как есть, то будет открываться первый
сайт first.ru , потому что, как ни стран-
но: The first VirtualHost section is used for
requests without a known server name.

Поэтому перед секциями реальных сай-
тов добавляем секцию-ловушку. Получа-
ется примерно так:

```
<VirtualHost *:80>
ServerName default
</VirtualHost>
```

Теперь все запросы с неправильными
именами сайтов и обращения по IP на-
правляются в папку /var/www/html, где
можно разместить простейшую индекс-
ную страницу.

Для вступления всех изменений в силу
перезапускаем сервер

```
# service httpd restart
или
# /etc/init.d/apache2 restart
```

Дальше больше

Приведенные выше секции содержат
минимальный набор директив. Со време-
нем возникает необходимость в допол-
нениях, которые описывают, например,
свойства PHP или возможности .htaccess.

Apache позволяет управлять доста-
точно большим количеством настроек.
При необходимости следует обра-
щаться к официальной документации
<http://httpd.apache.org>

Приведу пример секции VirtualHost для сайта domain.tld с синонимом another_domain.tld одного реально действующего сервера

```
<VirtualHost *\>
ServerName domain.tld
ServerAlias www.domain.tld\
another_domain.tld www.another_
domain.tld
DocumentRoot /var/www/vhosts/
domain.tld/httpdocs
ErrorLog logs/domain.tld.error_log
CustomLog logs/domain.tld.access_log
SymLinksBaseDir /var/www/vhosts/
ScriptAlias /cgi-bin/ /var/www/
vhosts/domain.tld/cgi-bin//
<Directory «/var/www/vhosts/
domain.tld/httpdocs»>
```

```
AllowOverride All
Options +Includes
</Directory>
php_admin_value open_basedir /
var/www/vhosts/domain.tld/
php_admin_value doc_root /var/
www/vhosts/domain.tld/httpdocs
php_admin_value upload_tmp_dir /
var/www/vhosts/domain.tld/tmp/
php_admin_value sendmail_path «/
usr/sbin/sendmail -t -i -fphp-\
sender@domain.tld»
</VirtualHost>
```

По материалам сайта:
linux.profiua.com



OpenOffice.

Настройка замены прописных букв на строчные по хоткею

В MS Офисе замена регистра символа делается не сложно. Там для этого уже настроены хоткеи. Если я правильно помню, то это Shift+F3.

В OpenOffice замена делается через меню **Формат-Регистр** ну и выбираем, что мы хотим. Такой вариант, конечно, не плох, но хотелось бы сделать циклическую замену. И на хоткей повесить.

Для этого идем в меню **Сервис-Настройка-Клавиатура** и далее в перечне Категория выбрать **Макросы OpenOffice.org – share – Capitalise**.

В окне **Комбинации клавиш** выбираем Shift+F3 и нажимаем **Заменить**.

Вполне вероятно, что после этого

попытка сменить регистр будет не работать, а выдавать ошибку. Это связано с тем, что для работы данного скрипта необходим BeanShell, который по умолчанию вместе с OpenOffice не ставится, по крайней мере в Ubuntu 10.04. По этому доставляем нужные пакеты:

```
$ sudo aptitude install\
openoffice.org-officebean\
libplexus-bsh-factory-java
```

После окончания установки перезапускаем OpenOffice и проверяем как это все работает.

По материалам сайта:
yakim.org.ua

Битва за потерянные файлы



Как и обещал раньше, вот я и созрел написать небольшую статью о крахе системы, о попытках восстановить файлы и директории и многом другом. Короче решил я создать себе мультизагрузочную флешку с многими осями и нужными программами и с одним общим меню. Идея воплотилась в жизнь, и я в итоге создал нужную мне флешку. После законченной работы я вставил флешку в комп и загрузился с нее. В открывшемся меню красовались: Parted Magic, Kaspersky Rescue Disc, DrWeb Live

CD, Ubuntu 12.04, Xubuntu 12.04, Linux Mint 13 KDE. Но были там еще несколько пунктов, которые появляются по умолчанию, при создании загрузочной флешки, это пункты: MemTest, Smart Boot Manager, Super Grub Disc, PloP Boot Manager, Ispsci.

Одним словом нажал я на один из этих пунктов, точно не помню на какой, но что-то связанное с Boot или Grub, и передо мной появился черный экран с мигающим в левом верхнем углу курсором. Прошло минут 5, но ничего не происходило и я решил перезагрузить компьютер. Что же я увидел после перезагрузки? Надпись, что системы нет и GRUB не найден. Дословно не помню, но смысл примерно такой же. Я решил, что просто напросто слетел GRUB и я его захотел переустановить по старой, проверенной схеме. Но и после этих действий ничего не получилось. В итоге я загрузился все с той же флешки и решил посмотреть на мой жесткий диск с системой. Какой же меня ждал шок, когда я увидел, что системный раздел был полностью пустой, а раздел home был занят всего на 10-15% и это была папка видео. Что в итоге произошло, я так и не понял. Важно было теперь не трогать мой жесткий диск, не пытаться ставить систему заново, а нужными средствами попробовать восстановить все что было потеряно, или хотя бы по максимуму вытянуть нужную информацию. Для этих целей я решил попробовать отличную программу TestDisk. Перед этим я уже не раз помогал людям с восстановлением потерянных файлов или папок. Теперь видать пришла и моя очередь. После сканирования харда, TestDisk выдал список множества удаленных файлов и па-

пок. Но вот ведь незадача, для файловой системы Ext4, в отличии от NTFS, FAT32, нет пункта восстановить удаленные файлы. Одним словом «Висит груша, нельзя скушать». И вот я принимаю, возможно необдуманное, решение «форматнуть» все разделы в тот же, что и был – Ext4. После десятков сканирований, программа более не находила потерянные файлы. Но мне никак нельзя было потерять все. Кстати для информации – восстановление файлов с Ext4 на данный момент очень проблематичное, поэтому у вас есть два выхода: держать нужную информацию на NTFS или Ext3, либо регулярно делать бэкапы и хранить свои данные на облачных сервисах, флешках или съемных носителях.

Пришлось зарываться по уши в интернет и гуглить нужную позарез информацию по восстановлению файлов в Линуксе. Раньше мне очень часто помогала на 100% чудесная программа для Windows – Zero Assumption Recovery. Она находила буквально все до копейки, с папками и файлами внутри. Загрузившись с Live CD я решил проверить ее еще раз. К сожалению, она ничего не нашла, в очередной раз доказав, что пока что файловая система Ext4 еще молодая и с ней мало кто работает, в плане программ по восстановлению информации. Тот же результат меня ждал с монстром по восстановлению – R-Studio. Она тоже отказалась работать. Оставались программы для проверки, которые искали и восстанавливали файлы по расширению, т.е. jpeg, bmp, mp3, pdf и другие. Конечно же не вариант рыться в тоннах картинок, которые попадутся в

папку восстановления после серфинга в интернете за многие месяцы, а то и годы, но ничего другого, к моему сожалению, не оставалось.

Из этих программ хочу выделить отличные программы в своем классе, это – Photorec, Scalpel и Foremost.

О Photorec я уже писал, поэтому сразу перейду к двум другим программам. Скажу лишь, что Photorec достаточно неплохо справляется с поставленными задачами. Список файлов, которые она поддерживает очень велик, но она мне не подходила по одной причине – она не может восстанавливать файлы из пакета LibreOffice, а на них у меня была очень ценная инфа, хотя и две других программы тоже, но об этом чуть позже. Точнее как то можно лично добавлять информацию о нужных расширениях в программу, но это весьма сложно и не на одном форуме мне так и не помогли с этой бедой. Хотя популярные файлы Photorec щелкает как орехи. Есть еще один минус в этих трех программах – восстановленные файлы имеют совсем другие названия отличные от оригинальных и не остаются метки тэги и прочая информация. В остальном же файлы, практически все, полностью читаемые.

Нужная информация по Photorec. Раз, два, три.



Foremost

Итак Foremost. Если верить информации из сети, то программу создали два специалиста из спецслужб США. Весьма обнадеживающая информация. Короче приступаем. Установим программу:

```
sudo apt-get install foremost
```

Программа поддерживает следующие типы файлов: avi, bmp, dll, doc, exe, gif, htm, jar, jpg, mbd, mov, mpg, pdf, png, ppt, rar, rif, sdw, sx, sxc, sxi, sxw, vis, wav, wmv, xls, zip. Также можно запустить софтинку с опцией `-all`, тогда foremost будет искать все поддерживаемые типы файлов. Чтобы программа без проблем восстановила нужные файлы, она должна быть запущена не с того раздела диска, на котором установлена. Это кстати касается любой программы этого типа. Тоже самое касается и для восстановления. Если у Вас потерянные файлы находятся на разделе HOME то их нужно восстанавливать на другой раздел, флешку или жесткий диск.

Это главное правило!

```
sudo foremost -v -T -t
jpeg,png,bmp -i /dev/sda3 -o
/home/ваш_логин/output/
```

Разберем все детально:

`sudo foremost` - запускаем программу от root

`-v` – информация о прогрессе сканирования

`-T` – покажет время в названии папок для восстановленных файлов.

`-t reg,png,bmp` – восстанавливать только файлы нужного типа, например `*jpg` (нужно перечислять несколько типов файлов через запятую).

`-i /dev/sda3/` – раздел жесткого диска, с

которого будете восстанавливать данные.

-o /home/ваш логин/output – место, куда будут сохраняться восстанавливаемые файлы, а именно в вашей домашней директории в папку output.

В моем случае я восстанавливал файлы с одного жесткого диска на другой, если у вас будет всего один жесткий диск, то можно восстановить данные, к примеру, на флешку или съемный жесткий диск, тогда последняя строчка будет выглядеть примерно так:

```
-o /media/disk/disk
```

То есть в папке /media, которая находится на системном разделе, при включенной флешке появится папка /disk, которая и есть вашей флешкой, хотя название может и отличаться, а вторая папка /disk, это папка на подключенной флешке, созданная вами и может носить другое название.

Scalpel

Scalpel — это программа для быстрого восстановления нужных файлов. Уникальность программы состоит в том, что она не зависит от файловой системы. Программа ищет по базе данных начало и конец файлов известных форматов и делает попытки отыскать их на диске. Из этого следует, что восстановление возможно как с FAT16(32), NTFS, ext2/3/4, так и с «голых» (raw) разделов. Основана программа на базе Foremost, поэтому алгоритм поиска схож у этих программ. Установим Scalpel командой в термине:

```
$ sudo apt-get install scalpel
```

Чтобы программа начала поиск нужных файлов, нужно отредактировать ее конфигурационный файл, который находится по адресу /etc/scalpel/scalpel.conf. В нем нужно раскомментировать нужные (восстанавливаемые) файлы. Что означает «раскомментировать»? Напротив каждого типа файлов стоит символ #. Именно его нужно будет убрать. Например. Внизу две строчки из конфига scalpel. Нам нужно «вытянуть» все файлы pdf. Для этого мы убираем символ #, и тем самым даем программе команду, что нужно искать именно этот тип файлов.

```
# txt y 100000 --BEGIN\040PGP
pdf y 500000
%PDF %EOF\x0a REVERSE
```

А как искать нужные файлы при помощи этой программы? В принципе команда практически не отличается от поиска программой foremost:

```
$ sudo scalpel /dev/sda1 -o /home/ваш логин/output
```

где - /dev/sda1 это откуда нужно извлекать файлы

-o/home/ваш логин/output показывает, что в домашней директории появится папка output.

Но для меня основным, точнее одним из основных, было найти документы из пакета LibreOffice (OpenOffice). На одном забугорных форумов я отыскал нужную мне инфу. Нужно было добавить следующие строчки в конфиг данной программы и раскомментировать нужные файлы. Вот собственно сам текст:

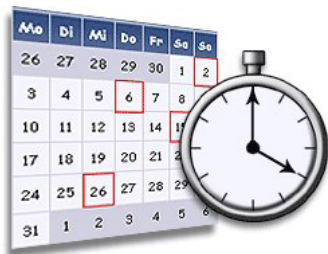
```
# Added to end of /etc/scalpel/scalpel.conf
```

```
#
#-----
-----
# OPENOFFICE FILES
#-----
-----
    odt y 20000000 PK?????
    ?????????????????????mimetypeap
    plication/vnd.oasis.opendocument.
    textPK META-INF/manifest.
    xmlPK????????????????????
    ods y 10000000 PK?????
    ?????????????????????mimetypeap
    plication/vnd.oasis.opendocument.
    spreadsheetPK META-INF/manifest.
    xmlPK????????????????????
    odp y 10000000 PK?????
    ?????????????????????mimetypeap
    plication/vnd.oasis.opendocument.
    presentationPK META-INF/manifest.
    xmlPK????????????????????
    # odg y 10000000 PK?????
    ?????????????????????mimetypeap
    plication/vnd.oasis.opendocument.
    graphicsPK META-INF/manifest.
    xmlPK????????????????????
    # odc y 10000000 PK?????
    ?????????????????????mimetypeap
    plication/vnd.oasis.opendocument.
    chartPK META-INF/manifest.
    xmlPK????????????????????
    # odf y 10000000 PK?????
    ?????????????????????mimetypeap
    plication/vnd.oasis.opendocument.
    formulaPK META-INF/manifest.
    xmlPK????????????????????
    # odi y 10000000 PK?????
    ?????????????????????mimetypeap
    plication/vnd.oasis.opendocument.
    imagePK META-INF/manifest.
    xmlPK????????????????????
    # odm y 10000000 PK?????
    ?????????????????????mimetypeap
    plication/vnd.oasis.opendocument.
    text-masterPK META-INF/manifest.
    xmlPK????????????????????
    # sxw y 10000000 PK??
    ?????????????????????mim
    etypeapplication/vnd.sun.xml.
    writerPK META-INF/manifest.
    xmlPK????????????????????
    #-----
    -----
    # THUNDERBIRD FILES
    #-----
    -----
    # msf y 10000000 //\s<!--
    -\s<mdb:mork:z\sv=»1.4»/>\s-->?<\
    s<(a=c)>\s//\s(f=iso-8859-1)
    //\s<!--\s<mdb:mork:z\
    sv=»1.4»/>\s-->?<\s<(a=c)>\s//\
    s(f=iso-8859-1) NEXT
    # actual Local Folder data
    files, no way to tell end so grab
    100MB
    NONE y 10000000 From
    ??????????????????????X-
    Mozilla-Status:\s?????X-Mozilla-
    Status2: NEXT
```

Я все так и сделал и начал поиски. В итоге программа scalpel нашла полностью все нужные файлы. После всех поисков и восстановлений мне оставалось из нескольких сотен гигабайт найти нужные, а весь мусор удалить. Но то уже другая история.

По материалам сайта:
softhelp.org.ua

Использование планировщика cron в Linux



Введение

Если вы уже имеете некоторый опыт системного администрирования Linux, то должны знать, что такое cron и для чего он предназначен. Если же вы только начинаете работать с Linux, то эти знания вам пригодятся. Даже если вы уже изучали использование cron, данная статья поможет вам освежить знания.

Название cron происходит от Cronos, древнегреческого олицетворения времени. И это правильно выбранное имя, так как основным предназначением cron является помощь в планировании различных задач, которые система должна выполнить в заданное время. Если вы пользовались Windows, то вероятно вам приходилось копаться в утилите Scheduled Tasks. В целом назначение у программ аналогичное, но реализация разная. Cron предназначен для серьезных задач системного администрирования и обладает большей гибкостью. В качестве примера реальной задачи можно привести резервное копирование. Понравится ли вам вручную выполнять резервное

копирование данных на нескольких сотнях машин? Думаю, нет. Вы просто пишете простой скрипт с использованием rsync, планируете его выполнение, например ежедневно, и забываете об этом. Все, что необходимо в дальнейшем – время от времени проверять логи. Некоторые даже используют cron для напоминания о важных личных событиях, например о днях рождения. Но cron – это просто демон, запускающий указанные вами задания. А имеется ли утилита, помогающая редактировать, добавлять, удалять эти задачи? Конечно имеется, и называется она crontab (от cron table). Однако будем двигаться вперед последовательно и начнем с установки.

Установка cron

Большинство дистрибутивов используют Vixie Cron и некоторые другие производные как дефолтную реализацию cron, и именно их мы будем рассматривать в этой статье. Кроме того, практически во всех дистрибутивах cron уже предустановлен, так как он является очень важной частью системы Linux. Хотя обычный пользователь, как правило, никогда не работает с ним напрямую, система и многие приложения используют cron (хорошим примером является периодическое обновление базы данных map-страниц). Значимым исключением является Gentoo, где необходимо вручную установить vixie-

cron и добавить его в runlevel по умолчанию. В Arch, Debian, Slackware, Ubuntu, Fedora и OpenSUSE cron установлен по умолчанию и автоматически запускается при загрузке системы. Мы надеемся, что вы изучите man-страницы cron и crontab, так как во многих дистрибутивах используются их пропатченные версии, что делается для лучшей интеграции программы под конкретный дистрибутив. Поэтому существует вероятность, что функция, которая имеется например в Gentoo, будет недоступна в Slackware. В любом случае, нет ничего лучше чтения man-страниц, особенно если имеешь дело с новой программой, которую собираешься достаточно часто использовать.

Использование cron

Во-первых, мы должны убедиться, что cron запущен:

```
$ ps faux | grep cron
```

Если в выводе ничего не будет, проверьте, запущен ли cron, и добавьте его в службы, запускаемые по умолчанию, чтобы он стартовал при следующей загрузке. Как это сделать, зависит от дистрибутива. В системах Debian команда должна вывести примерно следующее:

```
root      1424  0.0  0.0  22000
884 ?        ss   13:56   0:00 /
usr/sbin/cron
```

Теперь заставим cron работать. Сначала необходимо подумать о том, какую команду мы будем использовать. Требуется ли она привилегий root? Предположим, мы будем использовать crontab, чтобы

добавить простой скрипт на выполнение через каждые пять минут как простой пользователь.



Crontab

• Пример

Напишем простой скрипт, который будет выводить обычное диалоговое окно на базе curses в терминале (требуется dialog и любой эмулятор терминала – мы используем xterm):

```
#!/bin/sh
xterm -e «dialog --msgbox
'Testing cron...' 234 234»
#This is a very simple example,
#treat it as it is
```

Назовем файл crontest.sh и сделаем его исполняемым, так как все, что запускается с помощью cron, должно быть исполняемым:

```
$ chmod +x crontest.sh
```

Теперь нам нужно отредактировать пользовательскую таблицу crontab, чтобы добавить запись для нашего замечательного скрипта:

```
$ crontab -e
```

-e ставится для редактирования, -r - для удаления. Будьте очень осторожны! Многие системные администраторы по нео-

сторожности вводили -г вместо -е, тем более что эти клавиши расположены рядом, и удаляли все записи из crontab. В любом случае, скорее всего вы увидите полностью закоментированный файл (в crontab для обозначения комментариев используется #), в котором приведены примеры и описаны назначения полей записей. Да, запись состоит из полей, последнее из которых задает исполняемую команду, а остальные – временные характеристики. Итак, наша запись в crontab для скрипта, запускаемого каждые пять минут, выглядит следующим образом:

```
$ */5 * * * * export DISPLAY=:0
&& /home/$user/crontest.sh
```

Нам необходимо экспортировать переменную DISPLAY, чтобы xterm не выдавал ошибки вида «DISPLAY is not set». Теперь посмотрим назначение каждого поля.

• Поля crontab

Если вы работали ранее с символами для подстановки, то вам не составит труда освоить использование звездочки в crontab: она означает, что вместо нее подставляются все возможные значения поля. Поля по умолчанию в Linux (будьте внимательны, некоторые Unix-системы используют другие реализации cron) обозначают, слева направо: минуту, час, день месяца, месяц, день недели, год (необязательно) и исполняемую команду. Таким образом, если мы хотим, чтобы наш скрипт запускался каждые пять минут, каждый час, каждый день, каждый месяц и каждый год, наша запись должна выглядеть так:

```
5 * * * * export DISPLAY=:0 &&
/home/$user/crontest.sh
```

Мы призываем вас быть осторожнее с полем дня недели: в вашей стране первым днем недели может быть понедельник, а в других странах им может быть воскресенье. Если работаешь в мультикультурной, многонациональной среде, необходимо помнить о культурных и религиозных особенностях. Cron отправляет уведомления пользователю, которому принадлежит запись, на его почтовый адрес по умолчанию (\$user@\$hostname). Если вы хотите изменить адрес электронной почты, просто добавьте «MAILTO=\$email_address» в crontab. Если вы хотите отключить эту функцию, добавьте «>/dev/null 2>&1» в конец записи.

Теперь, если вы уже устали от всплывающего каждые пять минут окна, снова запустите crontab -е, чтобы удалить или, что проще, закоментировать запись. Чтобы просмотреть содержимое crontab, просто используйте опцию -l (list).

Взгляд на /etc/

• Скрипты cron

В предыдущих разделах мы решали простые и тривиальные задачи просто для разминки, а теперь мы уже займемся серьезными делами. Заголовок данного раздела относится к директориям в /etc, которые отвечают за периодические задачи по обслуживанию системы. Например, в директории /etc/cron.weekly имеется скрипт man-db, который обновляет базу данных man-страниц. Все подобные скрипты включены в состав дистрибутива и запускаются согласно файлу /etc/

crontab. Поэтому, если вы хотите разобраться их работе, лучше всего изучить содержимое этого файла. Вы увидите записи, похожие на те, что мы делали ранее, но исполняемые команды будут отличаться. «run-parts» - это небольшая утилита, созданная для запуска всех скриптов в заданном каталоге.

Внимательный читатель может заметить, что в /etc/crontab имеется одно поле, которого мы не видели при редактировании своего crontab – это поле «user». Это сделано по соображениям безопасности. Если вы вводите команду crontab -e как \$user, это значит, что любая команда запускается от имени \$user. Но так как /etc/crontab относится ко всей системе, необходимо явно задавать пользователя, так как может потребоваться запускать определенные скрипты или программы от имени другого пользователя, а не от root, особенно если системный администратор заботится о безопасности и добавляет пользователей и группы в систему по мере необходимости. Пример: для резервного копирования не нужны все привилегии root, требуются только права на чтение и запись в определенные директории, поэтому администратор создает группу backup и пользователя backupadmin с необходимыми правами и запускает ночные скрипты резервного копирования с помощью записи:

```
30 23 * * * backupadmin /home/
backupadmin/nightlybkup.sh
```

Эта запись запускает скрипт nightlybkup.sh каждую ночь в 23:30. Если мы захотим запускать резервное копирование только

с понедельника по пятницу, нужно отредактировать нашу запись:

```
30 23 * * 1-5 backupadmin /home/
backupadmin/nightlybkup.sh
```

Если вы захотите запускать резервное копирование только по понедельникам, средам и пятницам, замените «1-5» на «1,3,5». Когда вы точно знаете, что вам нужно, cron становится понятным и простым в использовании.

Размышления о безопасности

Существует множество ситуаций, когда вы не хотите, чтобы пользователь, имеющий доступ к вашей системе, имел возможность создавать записи crontab. Здесь в игру вступают файлы /etc/cron.deny и /etc/cron.allow. Их использование аналогично /etc/hosts.allow и /etc/hosts.deny, поэтому, если вы работали с этими файлами раньше, то будете чувствовать себя как дома. Этих двух файлов (cron.deny и cron.allow) по умолчанию нет в системе, по крайней мере на системе, в которой мы работаем, поэтому по умолчанию всем разрешено создавать записи в crontab. Вы можете проверить, какие файлы, относящиеся к cron имеются в вашем каталоге /etc с помощью команды:

```
$ ls /etc | grep cron
```

И снова напоминаем, что все сказанное справедливо для Linux, так как отсутствие этих файлов на Solaris означает прямо противоположное, кроме того, файлы расположены в других местах. Сначала проверяется cron.allow, поэтому обычно мы вводим «ALL» в cron.deny, а затем добавляем пользователей, которым мы хо-

тим предоставить доступ, в cron.allow.

Примеры использования cron

Каждую минуту:

```
* * * * /usr/local/bin/check-disk-space.sh
```

Команда будет выполняться каждую минуту, день и месяца.

По дням:

```
30 02 10 01,06,12 * /home/$user/bin/checkdrive.sh
```

Эта команда будет запускать скрипт checkdrive.sh в 02:30 каждое 10 число января, июня и декабря.

Каждые полчаса:

```
00,30 * * * * /home/$user/backupdata.sh
```

Только по выходным:

```
* * * 6,7 /usr/bin/weekend
```

Дважды в день:

```
20 11,16 * * * /usr/sbin/command
```

Команда будет выполняться каждый день в 11:20 и 16:20.

Каждый час в один конкретный день:

```
01 * 2 05 * /sbin/system_command
```

Команда будет запускаться каждый час каждое 2 мая.

Каждые 10 минут во время пяти рабочих дней (понедельник – пятница):

```
*/10 * * * 1-5 /usr/local/bin/check-disk-space.sh
```

Только в рабочее время:

```
00 09-17 * * 1-5 /usr/local/bin/check-disk-space.sh
```

Эта команда будет выполняться один раз в день с 9 до 17 часов.

Один раз в году, в полночь 1 января:

```
0 0 1 1 * /usr/local/bin/check-disk-space.sh
```

Дважды в год в 12:00 и в 24:00

```
0 0,12 1 */6 * /usr/local/bin/check-disk-space.sh
```

Эта команда будет исполняться дважды в год (каждые 6 месяцев) в 12:00 и в 24:00.

Каждый третий четверг любого заданного месяца в 10:00:

```
0 10 15-21 * 4 /usr/local/bin/check-disk-space.sh
```

Каждый день через 20 минут после каждого четного часа:

```
20 0-23/2 * * * /usr/local/bin/check-disk-space.sh
```

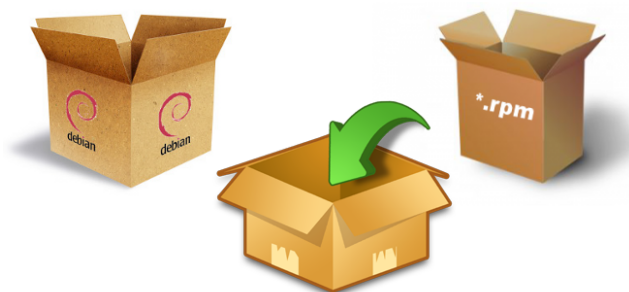
Эта команда будет выполняться каждый день через 20 минут после каждого четного часа (0:20, 2:20...22:20).

Заключение

Если в начале вам показалось, что записи crontab трудны для понимания и запутаны, уверяю, что при их постоянном использовании вы очень скоро будете наизусть помнить порядок и назначение всех полей, а также название и расположение всех файлов, которые необходимо редактировать. Все, что требуется - это практика.

По материалам сайта:
rus-linux.net

18 советов по использованию команд apt и dpkg в системах Debian и Ubuntu



В настоящем руководстве представлены наиболее популярные и полезные примеры использования команд APT и DPKG, которые можно применить как в Ubuntu, так и в Debian (а также в системах, производных от них). Явно указано, в каких случаях требуются привилегии суперпользователя, а для тех команд, где ничего не указано – команду может выполнить обычный пользователь. Если вы используете Ubuntu, то для получения привилегии суперпользователя, наберите перед командой префикс `sudo` (и введите свой пароль пользователя); в Debian наберите команду `su`, введите пароль, а затем вводите команды, которые описаны ниже:

```
$ dpkg -L package_name
```

Эта команда покажет, какие файлы для пакета с именем `package_name` установлены на вашем компьютере. Обратите внимание, что эта команда работает только для уже установленных пакетов.

Пример:

```
embryo@debian:~$ dpkg -L vorbis-  
tools  
./.  
/usr  
/usr/bin  
/usr/bin/ogg123  
apt-get build-dep package_name  
(необходимы привилегии root)
```

Это очень полезная команда: с ее помощью устанавливаются все библиотеки, необходимые для разработки приложения. Так что если вам нужно скомпилировать новую версию программы, отличающуюся от той, что есть в репозитории, для получения зависимостей используйте эту команду. Обратите внимание, что будут установлены библиотеки, необходимые для текущей версии приложения, которая взята из репозитория, поэтому вам, возможно, потребуется установить дополнительные библиотеки. Но для большинства приложений этого не требуется. Пример:

```
# apt-get build-dep wesnoth
```


С помощью этой команды будут установлены все пакеты разработчика, необходимые для приложения Wesnoth. Имейте в виду, что у вас должен быть прописан доступ к репозиториям исходных кодов (в Ubuntu по умолчанию доступ к ним есть).

apt-get clean (и apt-get autoclean)

(необходимы привилегии root)

С помощью этой команды удаляются все архивы, установленные в вашей системе. Всякий раз, когда вы устанавливаете приложение, оно будет загружено и его deb-пакет будет помещен в директорию /var/cache/apt/archives/. Данная команда удаляет эти пакеты и освобождает место.

```
# apt-get clean
# apt-get autoclean
```

Относительно apt-get autoclean: с помощью этой команды удаляются только те архивы, которых уже нет в репозиториях, и сохраняются те, которые в данный момент в репозиториях присутствуют. Эта команда полезна, если вы хотите избавиться от старых архивов, которые были обновлены, но обновленные архивы хотите на всякий случай сохранить. Для того, чтобы увидеть, сколько места занимают архивы, используйте команду `du -h` (аргумент `-h` указывает, что результат нужно выдать в формате, удобном для чтения):

```
$ du -h /var/cache/apt/archives/
```

apt-cache show package_name

Будет отображена информация о любом пакете `package_name`, который присутствует в репозитории. Вам достаточно набрать только название пакета, а не указывать название и

версию. Пример:

```
embryo@debian:~$ apt-cache show
vorbis-tools
Package: vorbis-tools
Priority: optional
Section: sound
Installed-Size: 580
```

apt-cache search pattern

Будет произведен поиск и будет выдан список всех пакетов, у которых в имени или в описании содержится шаблон поиска (pattern). Вы можете перенаправить этот список в `grep` для получения результата в более удобном варианте:

```
embryo@debian:~$ apt-cache search
gnome-terminal|grep terminal
gmfsk - MFSK, RTTY and other
digital mode terminal for HF/
amateur radio
gnome-terminal-data - Data files
for the GNOME terminal emulator
gnome-terminal - The GNOME 2
terminal emulator application
```

apt-get update && apt-get upgrade

(необходимы привилегии root)

Это одностроковая команда, которая обновит список ваших пакетов и выполнит обновление вашей системы. Поскольку для ее выполнения требуются привилегии root, то если ваш дистрибутив — Ubuntu, используйте:

```
$ sudo apt-get update && sudo
apt-get upgrade
```

Обратите внимание, что вторая команда (после двойного амперсанда) выполняется только в случае, если выпол-

нение первой команды было успешным, поэтому, если по какой-нибудь причине списка обновлений получить не удалось, вторая команда выполняться не будет. Если вы используете тестовый или бета-релиз, используйте вместо нее следующую команду:

```
sudo apt-get update && sudo apt-get dist-upgrade
```

Обновления будут получены даже в случае, если они зависят от более новых пакетов.

dpkg --get-selections

Эта команда выведет список всех пакетов, установленных в системе с помощью менеджера пакетов (более точно, это те пакеты, которые были выбраны в менеджере пакетов – прим.пер.).

Пример:

```
embryo@debian:~$ dpkg --get-selections
acpi-support-base install
acpid install
adduser install
akregator install
```

dpkg -S filename

Будет найден по образцу filename список файлов, имеющих в установленных пакетах.

Пример:

```
embryo@debian:~$ dpkg -S oggenc
vorbis-tools: /usr/share/man/
man1/oggenc.1.gz
vorbis-tools: /usr/bin/oggenc
```

apt-get autoremove

(необходимы привилегии root)

С помощью этой команды удаляются ненужные пакеты (например, одиночные пакеты), которые не используются никаким из пакетов, в текущий момент установленных в системе. Такая ситуация часто возникает после удаления определенного приложения, когда его зависимости не удаляются и по-прежнему остаются в системе. Используйте эту команду, чтобы избавиться от таких пакетов.

apt-cache dump

Эта команда выводит список всех пакетов, находящихся в кэше.

apt-cache depends package_name

Очень полезная команда, которая перечисляет все зависимости, которые есть у пакета, в том числе и рекомендуемые пакеты.

```
embryo@debian:~$ apt-cache
depends vorbis-tools
vorbis-tools
Depends: libao2
Depends: libc6
Depends: libcurl3-gnutls
```

apt-cache rdepends package_name

Команда покажет обратную зависимость пакетов, т.е. будет выдан список всех пакетов, которые зависят от указанного пакета.

```
# apt-cache rdepends mplayer
```

apt-cache policy package_name

Эта команда выдаст список репозиториев, в которых находится пакет. В Ubuntu, есть четыре таких репозитория: main, restricted, universe и multiverse. В Debian имеются репозитории main, contrib и non-free. Для каждого пакета

есть определенная политика его использования и она зависит от того, в каком из указанных репозиториях находится пакет.

```
$ apt-cache policy amarok
$ apt-cache policy sun-java6-jre
```

apt-cache showsrc package_name

Будет отображаться информация об исходном коде пакета.

apt-get -f install

(необходимы привилегии root)

Будет сделана попытка исправить нарушенные зависимости. Как правило, эта команда очень полезна в случае, если вы пытаетесь вручную установить deb-пакета, не устанавливая первоначально его зависимости. Выполните команду `apt-get -f install`, а затем снова попытайтесь установить пакет.

apt-cdrom add

(необходимы привилегии root)

Команда добавит CD-ROM к числу источников, откуда можно получать пакеты, так что вы сможете устанавливать с него пакеты. Эта команда особенно полезна, если у вас есть образы ISO, которые вы не хотите записывать на компакт-диск. Для того, чтобы сканировать образы ISO, используйте команду:

```
# apt-cdrom -d=/cdrom add
```

В результате в директории `/cdrom` будет смонтирован образ ISO, а затем его содержимое будет добавлено в `sources.list`. Обратите внимание, что директорий `/cdrom` должен быть пустым. Для этого вы можете использовать любую пустую директорию:

```
# apt-cdrom -d=/home/USER/mydir
```

dpkg -c local_file.deb

Аргумент `-c` указывает команде `dpkg` выдать список содержимого deb-пакета. Пример:

```
embryo@debian:~$ dpkg -c google-chrome-stable_current_i386.deb
```

...

```
drwxr-xr-x root/root 0 2010-07-30
23:54 ./usr/share/man/man1/
-rwxr-xr-x root/root 3098 2010-
07-30 23:54 ./usr/share/man/man1/
google-chrome.1
lrwxrwxrwx root/root 0 2010-07-30
23:54 ./usr/bin/google-chrome ->
/opt/google/chrome/google-chrome
dpkg -f local_file.deb
```

Эта команда покажет управляющую информацию бинарного пакета. Из нее менеджер пакетов получает данные о зависимостях, о том, кто осуществляет поддержку пакета, о домашней странице пакета, общее описание, а также многое другое.

```
embryo@debian:~$ dpkg -f google-chrome-stable_current_i386.deb
Package: google-chrome-stable
Version: 5.0.375.126-r53802
```

Это наиболее популярные примеры использования команд `apt` и `dpkg`

По материалам сайта:
rus-linux.net

Пересборка пакетов в Debian

Загрузка пакета с исходными текстами

Проверяем чтобы в `/etc/apt/sources.list` были активированы репозитории `deb-src`:

```
deb-src http://ftp.debian.org/
debian unstable main contrib non-
free
deb-src http://ftp.debian.org/
debian testing main contrib non-
free
deb-src http://ftp.debian.org/
debian stable main contrib non-
free
```

Для загрузки самых свежих исходных текстов пакета «publican» следует выполнить:

```
sudo apt-get update
apt-get source publican
```

Если необходимо установить исходные тексты пакета из определенной ветки дистрибутива, например, Debian Testing, что является разумным компромиссом между свежестью и стабильностью:

```
apt-get source publican/testing
```

Исходные тексты будут сохранены в текущую директорию:

```
ls -dF publican*
```

```
publican-2.1/ publican_2.1-2.dsc
publican_2.1-2.debian.tar.gz
publican_2.1.orig.tar.gz
```

В случае, когда пакет отсутствует в репозиториях, загрузить пакет можно командой:

```
dget -u dsc-url
```

где `dsc-url` представляет собой URL к `.dsc`-файлу с описанием параметров пакета, опция `-u` указывает на отмену стадии проверки валидности пакета. Для получения утилиты `dget` нужно установить пакет.

Установка зависимостей

Для установки пакетов, необходимых для сборки заданной программы, следует использовать команду «`apt-get build-dep пакет`», например:

```
apt-get build-dep publican/
testing
```

Если пакет взят не из репозитория, то для выполнения аналогичного действия в директории с исходными текстами пакета нужно выполнить команду:

```
dpkg-checkbuilddeps
```

Эта команда выведет список задействованных при сборке пакетов, которые затем нужно установить через:

```
apt-get install
```

Изменение исходных текстов перед сборкой

При необходимости код собираемой программы можно модифицировать, например, приложить дополнительный патч или отредактировать `debian/rules`. При внесении изменений во избежание конфликтов рекомендуется изменить номер версии, используя команду `dch` из па-

кета devscripts:
`dch --local имя`

где *имя* является коротким именем, идентифицирующим вносящего изменение. После выполнения команды будет предложено указать комментарий для помещения в debian/changelog.

Сборка пакета

Для сборки в директории с распакованным кодом выполняем команду `debuild` из пакета devscripts.

```
cd publican-2.1
debuild -us -uc
```

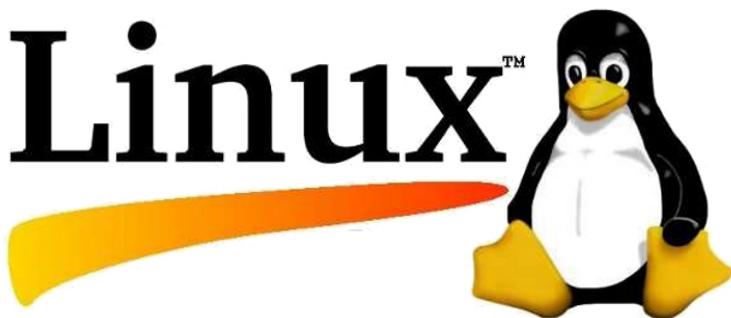
где опции `-us` `-uc` указывают пропустить связанный с проверкой сигнатур шаг, в случае если у сборщика отсутствует валидный GPG-ключ, совпадающий с данными в начале файла changelog.

После завершения сборки в предыдущей директории можно будет найти бинарный пакет:

```
cd ..
ls -dF publican*
```

```
publican-2.1/ publican_2.1-2rh1.
dsc
publican_2.1-2.debian.tar.gz
publican_2.1-2rh1_i386.changes
publican_2.1-2.dsc publican_2.1-
2rh1_source.changes
publican_2.1-2rh1_all.deb
publican_2.1.orig.tar.gz
publican_2.1-2rh1.debian.tar.gz
```

По материалам сайта:
armanenshaft-linux.blogspot.com



Обзор самых распространенных реализаций языка Бейсик под Linux

Бейсик – BASIC (Beginner's All-purpose Symbolic Instruction Code) – один из некогда самых популярных языков программирования, который ведет свою историю с 1964 года. Именно с языка Бейсик многие начинали свое знакомство с программированием, а для некоторых он остался единственным. Существует мнение, что Бейсик уже безнадежно устарел вместе с компьютерами Altair, ZX-Spectrum, IBM PC и операционными системами CP/M и DOS. В лучшем случае про Бейсик вспоминают, когда речь идет о VisualBasic или VBA, но в этом случае сфера его применения ограничивается Windows, а у профессиональных программистов упоминание о Бейсике способно вызвать лишь улыбку. Я решил сделать небольшой обзор самых распространенных реализаций языка Бейсик под Linux и показать, что не настолько уж плохо обстоят дела с этим незаслуженно забытым языком. Ведь если у вас есть желание или необходимость создать несложную программу для собственных нужд, но при этом нет времени, желания или возможностей изучать другой язык, вы можете «вспомнить молодость» и воспользоваться одним из описанных ниже инструментов.

Bywater BASIC Interpreter/Shell

Не зря я начинаю обзор именно с этой программы. Среди всех рассматриваемых, она наиболее похожа на первое поколение интерпретаторов BASIC, знакомых

многим еще по IBM PC, ZX-Spectrum, различным «Электроникам», «БК» и прочим. А название бинарного файла – bwbasic – вызывает ассоциации с небезызвестным GW-BASIC. Интерпретатор запускается прямо в терминале, даже не очищая экран и весь ввод / вывод проходит прямо там. Используется привычная для первых версий Бейсика нумерация строк, и, соответственно, команды RUN, LIST, NEW, SAVE, LOAD... Для выхода из интерпретатора используется команда SYSTEM.

```
malky@malky-desktop:/media/temp5$ bwbasic
Bywater BASIC Interpreter/Shell, version 2.20 patch level 2
Copyright (c) 1993, Ted A. Campbell
Copyright (c) 1993-1997, Jon S. Wilkoff

bwBASIC: 10 INPUT "Как тебя зовут?", Name$
bwBASIC: 20 PRINT "Привет", Name$
bwBASIC: list
      10: INPUT "Как тебя зовут?", Name$
      20: PRINT "Привет", Name$
bwBASIC: run
Как тебя зовут? Malky
Привет Malky
bwBASIC: system
malky@malky-desktop:/media/temp5$
```

Запуск самого интерпретатора, как видно на приведенном выше скриншоте, осуществляется командой bwbasic. Единственным параметром может быть имя текстового файла программы, например **bwbasic myprogram.bas**

Подспорьем в качестве справочника по операторам и функциям будет ман-страница программы. **man bwbasic**

Среди перечня операторов и функций есть как привычные для всех диалектов Бейсика, так и специфические,

предназначенные для работы с файловой системой Unix (Linux) / Windows. Встречаются и некоторые недочеты: например, привычная команда CLS действия в версии 2.2. не осуществляет (хотя упоминается в руководстве для версии 2.1), поэтому очистку экрана можно сделать при помощи CLEAR SCREEN.

Желательно также помнить о такой интересной особенности, что, хотя интерпретатор не различает регистр для операторов и функций, он при этом различает его для переменных. Поэтому NAME\$, Name\$ и name\$ – это три разных переменных с точки зрения bwbasic.

В отличие от рассматриваемых ниже версий BASIC, только этот интерпретатор не имеет инструментов для работы со звуком и графикой, что не так уж и удивительно, учитывая работу в консоли.

Установить Bywater BASIC можно сразу из репозитория Ubuntu:

```
sudo apt-get install bwbasic
```

Однако в Ubuntu 12.10 была доступна версия 2.2, тогда как на сайте разработчика заявлена 2.6.

В целом, интерпретатор может быть использован в учебных целях, для базового знакомства с навыками программирования; для написания программ для собственных целей, если знания программирования у вас ограничиваются только этим языком; да и просто для ностальгии по старым временам

Сайт: <http://bwbasic.sourceforge.net/>

Преимущества: стабильная работа, «классический» интерпретатор, работа из командной строки.

Недостатки: ограниченная функциональность.

X11-BASIC – Basic interpreter and shell

Вторая рассматриваемая в обзоре программа, не смотря на то, что также является интерпретатором, кардинально отличается от Bywater BASIC. X11-BASIC (или xbasic) представляет собой Бейсик второго поколения. Нумерация строк не используется (хотя, в комплект поставки включена программа, которая преобразовывает нумерованную программу в ненумерованную для совместимости с xbasic).

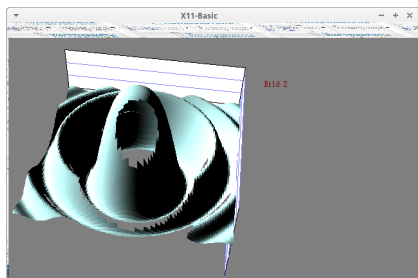
Как и в случае с bwbasic, xbasic запускается прямо в командной строке, и после выхода оставляет результаты своей работы на экране. Т.к. нумерация строк не поддерживается, то команда выполняется сразу после её ввода, а если же строка ввода начинается с числа, то используется режим «калькулятора» (пример видно на скриншоте ниже).

```

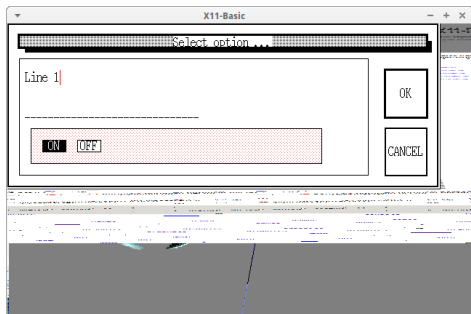
Терминал - mbk@mbk-desktop:/media/tempi
Файл Правка Вид Терминал Tabs Справка
mbk@mbk-desktop:/media/tempi$ xbasic
*****
*          V. 1.20          *
*        by Markus Hoffmann 1997-2013 (c) *
*          *****          *
* version date:   Thu Mar 28 08:06:27 CET 2013 *
* library V.1.20 date:   Thu Mar 28 08:06:27 CET 2013 *
*          *****          *
> list
> load "xbasprog.bas"
> 10
10
> 10+20
30
> print "Hello"
Hello
> quit
mbk@mbk-desktop:/media/tempi$

```

Чтобы ввести собственно код программы, необходимо выполнить команду EDIT. В результате будет вызван редактор по умолчанию для командной строки, при



Однако даже после этого проблемы наблюдались. Иногда некоторое изображение удавалось получить, «кликая» мышью по окну вывода, иногда даже попытка вызова какого-то графического оператора приводила к краху программы:



X11-BASIC поддерживает выполнение (или просто загрузку) файла с программой в качестве параметра запуска, выполнение отдельной строки с кодом, режим «калькулятора» и даже работу программы в качестве демона (!).

Еще одной интересной и полезной особенностью является возможность использования `xbasic` в качестве командной оболочки. Так, если возможностей `bash` вам мало, а изучать Perl или Python

нет желания, времени или возможности, вы можете подобным образом создавать скрипты на Бейсике, если вы владеете им.

Т.е., достаточно сообщить путь к интерпретатору в первой строке `bas`-файла:

`#!/usr/bin/xbasic` (путь к `xbasic` в моей системе, который вы сможете получить командой `whereis xbasic`)

И сделать его исполняемым командой `chmod +x имя_файла`, после чего, файл станет возможным запускать как и любой другой бинарный или скриптовый файл.

Для простоты установки интерпретатора можно использовать `deb`-файл, предлагаемый разработчиком по ссылке. К сожалению, доступна версия только для 32-битной системы, которая, однако, работает после установки необходимой 32-битной библиотеки:

```
sudo apt-get install  
libreadline6:i386
```

Руководство по операторам, функциям и системным переменным программы доступны через справочную службу `man`:

`man xbasic`

а на сайте автора доступны примеры различных программ, включая работу с графикой, звуком, сетью, и даже написанные на самом же Бейсике компиляторы для Windows и Unix.

Сайт: <http://x11-basic.sourceforge.net/>

Преимущества: очень широкие возможности и большой перечень реализованных функций; доступность под различные ОС; возможность использования в качестве командной оболочки.

Недостатки: сложность работы для малоподготовленного пользователя; сбой при работе с графическим режимом.

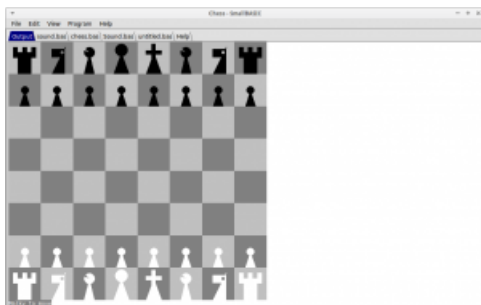
SmallBASIC

Следующий представитель семейства BASIC в обзоре также представлен версиями для различных операционных систем, где на фоне привычных Windows, Linux и даже Android, выделяется версия для PalmOS.

SmallBASIC можно установить, используя deb-файл на странице разработчика. Не смотря на то, что версия 32-битная, никаких дополнительных библиотек устанавливать не понадобилось. После установки, вы сможете найти программу через главное меню вашей графической оболочки в разделе «Разработка». SmallBASIC имеет графический интерфейс с поддержкой множества вкладок. При этом, для вывода (одновременно как графического, так и текстового) используется только одна вкладка, поэтому и запущена одновременно может быть только одна программа, хотя листинги на разных вкладках могут быть разные.

Хотя SmallBASIC является интерпретатором, это не значит, что он работает с построчным вводом. Вы можете свободно перемещаться по листингу программы, изменять код, шрифт редактора для удобства прочтения. Для сохранения / открытия / создания новых вкладок используется меню программы; запуск и останов осуществляется также не при помощи текстовых команд, а через меню программы или панель инструментов внизу. В целом, программа скорее напоминает такие реализации Бейсика, как QuickBASIC и PowerBASIC. Нумерация строк не нужна, однако выполнению кода совершенно не мешает, что позволяет использовать и старые программы без необходимости удаления нумерации. При этом, может использоваться нумерация только отдельных строк (в качестве меток), однако текстовые метки, (определенные через LABEL), также допустимы. Допустимы и разные применения операторов, которые могли различаться в различных диалектах Бейсика. Так, например, возможна, как классическая однострочная схема IF ... THEN ... ELSE..., так и «стаканчиком», где могут использоваться вариации ELSEIF / ELIF, ENDIF / FI, что обеспечивает хорошую совместимость с другими версиями BASIC.

В редакторе возможно отображение древовидной структуры программы (с отображением констант и подпрограмм), как на скриншоте ниже.




```
dev:i386 libxpm-dev:i386
```

Для 32-битной Ubuntu / Debian:

```
sudo apt-get install gcc g++  
lib32ncurses5-dev libx11-dev  
libxext-dev libxrender-dev  
libxrandr-dev libxpm-dev libffi-  
dev
```

Далее скачаем и распакуем архив с бинарным файлом FreeBASIC, распакуем его, сделаем исполняемым установочный файл в папке с программой

```
chmod +x install.sh
```

и установим его, выполнив

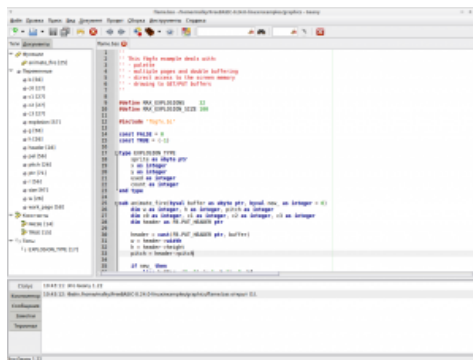
```
sudo ./install.sh -i
```

После этого, вы уже можете компилировать готовые bas-файлы (или же подготовленные в любом текстовом редакторе) при помощи компилятора fbc с файлом в качестве аргумента при запуске. При этом могут использоваться опции `-lang qb` для совместимости с QuickBASIC или же `-lang fb` для использования новых возможностей FreeBASIC.

На сайте FreeBASIC представлена документация по языку в различных форматах, в том числе и он-лайн. Там же можно найти примеры программ и уроки по FreeBASIC. Отдельно представлена сравнительная таблица для C и FreeBASIC, которая может быть полезна как для C-программистов, которые решили изучать BASIC (если таковые найдутся), так и для тех, кто был знаком ранее только с BASIC, но решил изучать C. FreeBASIC также поддерживает использование C-библиотек и встроенных вставок на ассемблере.

Не смотря на то, что это, пожалуй, самый мощный инструмент из уже рассмотренных в обзоре, я не буду останавливаться подробно на синтаксисе языка, т.к. документации в сети по этому языку достаточно много.

Однако впечатление немного смазывается отсутствием полноценной IDE для FreeBASIC, но эту проблему несложно решить, установив универсальную IDE Geany.



```
sudo apt-get install geany
```

При первом запуске Geany необходимо будет выбрать в меню **Документ – Установить тип файла – Языки программирования – Файл FreeBasic** и можно начинать работу. На скриншоте ниже показан пример выполнения графической программы из Geany.

При этом компилируется и бинарный файл, который потом можно использовать отдельно. Указать дополнительные параметры для компилятора (например, параметр совместимости с QuickBASIC) можно в меню **Сборка – Установить команды сборки**.

Сайт: <http://www.freebasic.net/> Geany:
<http://www.geany.org/>

Достоинства: очень мощный компилятор с широкими профессиональными возможностями.

Недостатки: ограничен i386 архитектурой.



Gambas

Gambas – представитель третьего поколения языка BASIC, объектно-ориентированный, и позиционируется как альтернатива Microsoft VisualBASIC под Linux. Название представляет собой рекурсивный акроним Gambas Almost Means BASIC, т.е. «Gambas — почти Бейсик».

Установить программу можно сразу из репозитория Ubuntu:

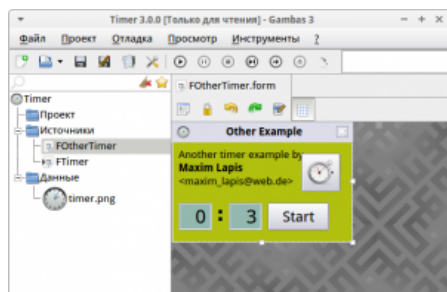
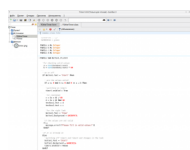
```
sudo apt-get install gambas3
```

На случай, если в вашем репозитории есть только Gambas 2.x, то третью версию вы можете получить в этом PPA.

Хотя автор и заявляет, что совместимость с VisualBASIC не является целью, в Gambas скорее себя будут чувствовать комфортно те, кто знаком с VisualBASIC, а не пользователи QuickBASIC / PowerBASIC. Отдельно представлены отличия от VisualBASIC.

По Gambas доступно достаточно документации в Wiki, в том числе и на русском языке. Если вам нужен объектно-ориентированный BASIC, то Gambas – это то, что вам подходит.

Сайт: <http://gambas.sourceforge.net>



Wiki: <http://gambasdoc.org>

Достоинства: объектно-ориентированный BASIC с большим количеством документации по языку; возможность работы с базами данных.

Недостатки: функционирование ограничено платформой Linux.

На этом я завершаю обзор пяти отобранных вариаций языка BASIC под Linux. Вы можете также обратить внимание на HBasic, wxBASIC и другие, однако я решил ограничиться только упомянутыми. Надеюсь, что статья вызовет приятные воспоминания у тех, кто как и я, начинал программирование с Бейсика, а кто-то, быть может, выберет какой-то из упомянутых проектов в качестве своего рабочего инструмента.

По материалам сайта:
ubuntu.sumy.ua

Криптография почтовых сообщений

Каждый день люди отправляют миллионы писем в разные концы земного шара. Отправляют, но лишь небольшая часть всех пользователей задумывается о том, что каждое отправленное письмо проходит через множество серверов, прежде чем оказывается непосредственно у получателя.

На сегодняшний день многие почтовые провайдеры побеспокоились о том, чтобы хотя бы немного обезопасить своих пользователей от утечек информации. И создали защищенные каналы связи между сервером и пользователем. То есть когда мы отправляем письмо, уже можно исключить целую сеть интернет-провайдеров, через которых проходит сообщение, пока оно не будет доставлено на почтовый сервер. Но это только в том случае, если используется безопасный почтовый сервер, к примеру Google Mail, в то время как в случае с Mail.ru, и ему подобным, любое письмо может быть перехвачено еще на этапе отправки.

Но о чем я говорю? О каком перехвате? И кого может заинтересовать та информация, что я пишу своим друзьям и близким? Именно такие вопросы чаще всего возникают, когда люди слышат о вопросах шифрования электронной почты.

Но давайте немного подумаем. Каждое письмо обычно проходит через целый ряд почтовых серверов, на каждом из которых оно анализируется, как минимум, на предмет спама. И никто не может

гарантировать того, что собранная информация нигде не собирается и никем не используется. И никто не может гарантировать того, что почтовый ящик будет взломан и вся корреспонденция окажется в руках третьих лиц.

Итак, что же можно сделать для того, чтобы устранить эту опасность? Естественно – шифровать свою почту. В этом случае мы сможем не только передавать защищенные сообщения, которые сможет прочесть только получатель, но еще и подтверждать с помощью цифровой подписи, что именно мы являемся отправителем данного сообщения. Что так же немаловажно!

Осталось только разобраться с самой технологией работы.

Технология работы

Для шифрования на сегодняшний день существует два пути – использование PGP (платное приложение) и GPG (бесплатное приложение). Я попробовал тестовый вариант PGP и могу сказать, что от бесплатного оно мало отличается, только стоит очень дорого. Поэтому на этом варианте останавливаться не будем.

Установить GPG в систему довольно просто. Достаточно скачать с официальной страницы *gnupg* пакет для той операционной системы, что используется на компьютере и установить его. На странице представлены пакеты для трех основных операционных систем. Кроме того, на

Linux можно установить GnuPG с использованием пакетного менеджера, что несколько упрощает ситуацию.

Практически вся работа с GnuPG ориентирована на использование командной строки. Абсолютно все операции, такие как работа с ключами, подписывание и шифрование, дешифрование, все это проводится с помощью командной строки.

Таким образом, если мы установили только сам `gpg`, то для того, чтобы отправить зашифрованное письмо нам необходимо создать текстовый файл, который затем в командой строке шифруется для определенного получателя. После чего его содержимое вставляется в новое сообщение почтовой программы и отправляется. И соответственно обратная ситуация в том случае, если мы получаем письмо.

Понятно, что совершать так много действий будет только тот человек, который заинтересован в защите своей рабочей переписке, но не тот, кого беспокоит анализ писем поисковой системой.

Дополнения почтовых программ

Есть целый ряд расширений для почтовых программ, которые позволяют упростить работу с зашифрованными сообщениями. При этом можно полностью отказаться от использования командной строки.

GPGMail – расширение для Mail.app, которое устанавливается пакетом MacGPG, позволяет шифровать и расшифровывать сообщения автоматически, работа с почтовой программой довольно прозрачна и удобна. И все бы ничего, только сама

программа Mail.app не умеет нормально работать с Gmail.

Проблема заключается в том, что все ярлыки Gmail интерпретируются как отдельные папки, и если письму было назначено несколько ярлыков, то в программе возникают сразу несколько дубликатов письма. Пробовал отключать на сервере показ ряда `imap`-папок, но в итоге стало просто неудобно работать.

Enigmail – расширение для почтовых программ Thunderbird и PostBox. Расширение значительно упрощает работу с зашифрованными сообщениями.

Но столкнулся с тем, что для того чтобы заставить нормально работать Thunderbird, приходится прикладывать значительные усилия. Как и практически все бесплатные продукты, данная почтовая программа очень гибкая и настраиваемая, но недоведенная до ума. Приходится много с ней работать, пока поведение программы будет адекватным. Неопытному пользователю это сделать будет довольно сложно.

PostBox же отлично работает с Gamil, достаточно только указать свои учетные данные. И все бы ничего, да только Enigmail для данной почтовой программы устанавливается с ошибкой, причем обновление расширения эту проблему не решает.

Выводы

Вся информация о нас давно уже стала достоянием общественности. Нужно лишь протянуть руку и взять то, что нужно.

А вот для того, чтобы устранить данную опасность практически нет инструментов. Почему нет?

Потому что нет еще ни одного нормального почтового клиента, который смог бы корректно работать с серверами Google, и при этом имел бы возможность работать с зашифрованными сообщениями.

В почтовых клиентах либо нет поддержки шифрования вообще, либо предоставляется с помощью расширений, которые часто работают не корректно.

Вы можете себе представить девушку, которая получив письмо, копирует его содержимое в отдельный файл, проводит из командной строки его дешифрацию, читает, пишет ответ в отдельном файле, после чего шифрует его, копирует в ответ на это письмо и отправляет его? Я тоже не могу такого представить. Да, и если задуматься, при такой сложности работы, проще плюнуть на все и просто надеяться на то, что данная информация никому не будет нужна.

До тех пор, пока не будет разработана отдельная программа, которая прекрасно работает с Gmail, как к примеру PostBox

или MailMate и при этом умела бы прозрачно работать с зашифрованными сообщениями, не требуя от пользователя познаний в криптографии, никто не будет использовать шифрование в своей повседневной жизни.

Но, если честно, появление такой программы выглядит как мечта... Надеюсь, что когда-нибудь эта мечта станет реальностью. А до тех пор придется использовать почту в ее обычном режиме, без криптографии.

Ведь даже найти единомышленников, которые использовали бы шифрование почтовых сообщений при общении очень проблематично. И все из-за того, что нет необходимого программного обеспечения, которое было бы достаточно простым для обычных людей.

По материалам сайта:
juev.ru

Настройка ufw в Ubuntu

Начиная с версии Ubuntu 8.04 в ее состав по умолчанию включена новая упрощенная оболочка для работы с файрволом – ufw (Uncomplicated firewall – «незамысловатый» брэндмауер). Как известно Linux по умолчанию имеет очень мощный встроенный сетевой фильтр - netfilter.

Стандартной утилитой для управления настройками netfilter является утилита iptables. Ufw создан для того чтобы сде-

лать более простым настройку netfilter-a. По сути ufw - это скрипт, который получает команды в одном формате и преобразует их в формат iptables.

Так как ufw является упрощенной утилитой настройки файрвола, он так же является и более ограниченным в своих возможностях по сравнению с iptables. Практика показывает что большинству пользователей вполне достаточно функционала, который предоставляет ufw. В то

же время, в случае необходимости всегда можно дописать нужное правило с помощью iptables.

Функционал ufw

Основными функциями файрвола ufw являются:

1. Работа в одном из двух режимов: разрешено все кроме того что описано правилами, запрещено все, кроме описанного правилами.
2. Создание правил для фильтрации пакетов с действиями allow, deny, reject.
3. Создание правил с учетом адреса и порта получателя и отправителя, используемого протокола.
4. Логирование пакетов подпадающих под правило.
5. Блокировка частых неправильных попыток авторизации по ssh.
6. Настраиваемые профили для приложений.

Запуск ufw

Ufw может быть включен или выключен системным администратором сервера, для этого используется команд

```
# ufw enable
```

Межсетевой экран активен и будет запущен при запуске системы.

```
# ufw disable
```

Файрвол остановлен и деактивирован при загрузке.

Просматривать состояние можно с помощью команд

```
# ufw status [numbered]
Status: active
```

Параметр numbered может быть использован для отображения правил файрвола с соответствующей нумерацией.

Изменение политики по умолчанию:

```
# ufw default allow [in|out]
# ufw default deny [in|out]
# ufw default reject [in|out]
```

Где allow подразумевает что все пакеты будут пропущены, кроме тех, для которых есть запрещающее правило: deny – все пакеты будут заблокированы, кроме тех, для которых есть разрешающие правила; reject – все пакеты будут заблокированы с соответствующим уведомлением подключающуюся сторону.

Параметры in|out указывают для какого направления мы хотим применять политику, для входящих или исходящих пакетов соответственно.

Управление правилами осуществляется с помощью команд allow, deny, reject, delete, insert.

allow – добавить разрешающее правило
 deny – добавить запрещающее правило (файрвол не предпринимает действий при попадании в это правило).

reject – добавить запрещающее правило (файрвол даст знать подключающемуся что этот порт закрыт).

limit – добавляет ограничивающее правило для указанного порта. В случае если с одного и того же адреса в течении 30 секунд будет произведено 6 и более неуспешных попыток авторизации в службу удаленного доступа ssh, ufw заблокирует удаленный адрес.

delete – удалить добавленное ранее правило

insert – вставить правило в определенную позицию в цепочке

Построение правил файрвола

Конструкция построения правил

```
ufw allow|deny|reject|limit
[in|out on INTERFACE] [log|log-
all] [protoprotocol] \
```

```
[from ADDRESS [port PORT]] [to  
ADDRESS [port PORT]]
```

Подробнее о параметрах:

allow|deny|reject|limit – указываем что будет делать это правило (см. описание-выше).

[in|out on INTERFACE] – указываем будет это правило относиться к входящему трафику (in) или исходящему (out) и хотим ли мы ограничить данное правило отдельным сетевым интерфейсом (к примеру eth0).

[log|log-all] – при желании можно логировать события которые подпадают под правило. По умолчанию лог пишется в /var/log/syslog. При использовании опции log – в файл будут записаны лишь пакеты участвующие при установке соединения с сервером. Используя опцию log-all в лог файл будут записаны все пакеты подпадающие под правило.

[proto protocol] - в правиле можно указать для какого протокола оно действует: tcp или udp.

[from ADDRESS [port PORT]] - адрес и порт источника пакета.

[to ADDRESS [port PORT]] - адрес и порт получателя пакета.

Профили приложений

Ufw позволяет создавать профили приложений. в которых можно указать все порты которые используются и удобно добавлять их в правила. Так к примеру можно создать профиль для веб-сервера Apache, указав что слушает он 80 и 443 порты по протоколу tcp.

Профили приложений являются текстовыми файлами и могут быть отредактированы любым текстовым редактором. все файлы хранятся в директории:

/etc/ufw/applications.d:

```
root@server:/etc/ufw/  
applications.d# pwd  
/etc/ufw/applications.d
```

```
root@server:/etc/ufw/  
applications.d# ls -l  
-rw-r--r-- 1 root root 406 2009-  
11-13 00:52 apache2.2-common  
-rw-r--r-- 1 root root 145 2009-  
10-22 22:50 openssh-server  
-rw-r--r-- 1 root root 241 2009-  
09-22 16:52 postfix
```

Рассмотрим пример профиля приложения apache2.2-common:

```
[Apache]  
title=web Server  
description=Apache v2 is the next  
generation of the omnipresent  
Apache web server.  
ports=80/tcp  
[Apache Secure]  
title=web Server (HTTPS)  
description=Apache v2 is the next  
generation of the omnipresent  
Apache web server.  
ports=443/tcp  
[Apache Full]  
title=web Server (HTTP,HTTPS)  
description=Apache v2 is the next  
generation of the omnipresent  
Apache web server.  
ports=80,443/tcp
```

В одном файле может находится несколько профилей. В данном примере 3: Apache, Apache Secure, Apache Full. Структура записей проста: сначала имя профиля в квадратных скобках ([Apache Full]), далее развернутое наименование приложения, далее описание приложения и список портов с указанием протокола которые задействованы в профиле (80,443/tcp).

Вывести список профилей можно с помощью команды:

```
# ufw app list
```

Доступные приложения:

```
Apache
Apache Full
Apache Secure
OpenSSH
Postfix
Postfix Submission
```

Детальную информацию по профилю можно просмотреть с помощью команды:

```
# ufw app info OpenSSH
|профиль: OpenSSH
|Название: Secure shell server,
an rshd replacement
|Описание: OpenSSH is a free
implementation of the Secure
Shell
protocol.
|Порт: 22/tcp
```

После добавления или изменения профиля приложения (в файле) необходимо дать знать фаерволу о новом профиле:

```
# ufw app update OpenSSH
```

Для того чтобы новый профиль сразу вступил в силу, необходимо добавить опцию --add-new:

```
# ufw app update --add-new
OpenSSH
```

По умолчанию добавленные с помощью --add-new профили не будут обрабатываться фаерволом. Рекомендуется вначале задать политику для новых профилей приложений:

```
ufw app default allow|deny
```

После чего все новые профили приложений будут добавляться в цепочку фаер-

вола с указанной политикой.

Использовать профили приложений можно также при создании обычных правил, например:

```
# ufw allow OpenSSH
```

Настройка профиля безопасности

1. Ограничение входящих подключений к FTP и веб – серверу.

Пример настройки фаервола для ситуации, когда необходимо заблокировать доступ к FTP и веб – серверу с адреса 192.168.100.100. В примере 192.168.1.11 – сервер на котором настраивается фаервол:

```
# ufw deny proto tcp from
192.168.100.100 to 192.168.1.11
port 21,80
```

Правило добавлено

```
# ufw status
```

```
|Статус: активно
```

```
|До Действие от
```

```
-----
192.168.1.11 21,80/tcp DENY
192.168.100.100
```

Теперь при попытках подключения с хоста 192.168.100.100 к 21 или 80 порту хоста 192.168.1.11 клиент будет висеть пока не отвалится по таймауту. Удалим это правило и добавим аналогичное но с действием reject:

```
# ufw delete deny proto tcp from
192.168.100.100 to 192.168.1.11
port 21,80
# ufw reject proto tcp from
192.168.100.100 to 192.168.1.11
port 21,80
```

Теперь при попытке подключения к 21 порту сервера удаленный клиент получит ответ Connection refused:

```
$ telnet 192.168.1.11 21
Trying 192.168.1.11...
telnet: Unable to connect to
remote host: Connection refused
```

Просканируем порты сервера:

```
# nmap -sT 192.168.1.100
Starting Nmap 5.00 ( http://nmap.
org) at 2009-12-28 22:56 EET
Interesting ports on
192.168.1.100:
Not shown: 992 closed ports
PORT STATE SERVICE
21/tcp filtered ftp
22/tcp open  ssh
80/tcp filtered http
139/tcp open netbios-ssn
445/tcp open microsoft-ds
2323/tcp open unknown
3128/tcp open squid-http
5900/tcp open vnc
MAC Address: 00:18:F3:67:85:9E
(Asustek Computer)
Nmap done: 1 IP address (1 host
up) scanned in 17.52 seconds
```

2. Ограничение исходящих подключений к указанному серверу

Цель – запретить исходящие подключения с сервера к адресу 195.214.195.105:

```
# ufw deny proto tcp to
195.214.195.105
```

3. Ограничение попыток авторизации по ssh

```
# ufw limit 22
```

Произведем несколько попыток подключения к 22 порту сервера с введением заведомо неправильного пароля:

```
$ ssh 192.168.1.100
user@192.168.1.100's password:
Permission denied, please try
again.
...
```

```
$ ssh 192.168.1.100
ssh: connect to host
192.168.1.100 port 22: Connection
refused
```

Как видим после нескольких неудачных попыток фаервол блокирует IP адрес с которого исходят попытки подключения и в логах появляется соответствующая запись:

```
Dec 28 22:59:13 fortress kernel:
[273659.578463] [UFW LIMIT BLOCK]
IN=eth0 OUT= MAC=00:18:f3:67:85:9
e:00:26:5e:46:40:86:08:00
SRC=192.168.1.156
DST=192.168.1.100 LEN=60 TOS=0x00
PREC=0x00
TTL=64 ID=3150 DF PROTO=TCP
SPT=33783 DPT=22 WINDOW=5840
RES=0x00 SYN
URGP=0
```

4. Логирование пакетов

```
# ufw allow log 22
```

С помощью этого правила фаервол будет логировать пакеты установки соединения по указанному 22 порту. В логах появятся следующие записи:

```
Dec 28 23:04:18 fortress kernel:
[273964.364715] [UFW ALLOW]
IN=eth0
OUT= MAC=00:18:f3:67:85:9e:00:26:
5e:46:40:86:08:00
SRC=192.168.1.156
DST=192.168.1.100 LEN=60 TOS=0x00
PREC=0x00
TTL=64 ID=6564 DF PROTO=TCP
SPT=58069 DPT=22 WINDOW=5840
RES=0x00 SYN URGP=0
```

Инструменты для исследования сетей с интерфейсом командной строки

часть 1

Ping

Ping является утилитой для проверки доступности узла в компьютерной сети, работающей по межсетевому протоколу (Internet Protocol) и измерения времени обращения сообщений, отправленных с компьютера исследователя на целевой компьютер. Название утилиты пришло из области активной эхолокации.

Функционирование утилиты ping связано с отправкой пакетов эхо-запроса протокола межсетевых управляющих сообщений (Internet Control Message Protocol – ICMP) к целевому узлу и ожиданием ICMP-ответов. Во время работы происходит измерение времени от отправки запроса до получения ответа (времени обращения) и запись информации обо всех потерях пакетов. Результаты проверки выводятся в форме статистической сводки, включающей в себя количество принятых пакетов, минимальное, максимальное и среднее время обращения, а также иногда среднеквадратичное отклонение времени обращения.

Использование утилиты ping обычно описывается как тестовый опрос компьютера.

ping может использоваться с различными параметрами командной строки, зависящими от конкретной реализации утилиты, которые позволяют использовать различные режимы работы, такие, как установка размера пакета для исследования, включение автоматического повторения отправки заданного количества пакетов, параметры учета времени или проведение атаки «наводнения пакетами

ping (ping-flood). Эта атака является простейшей разновидностью атаки отказа в обслуживании при которой атакующий блокирует работу целевой системы обработкой большого количества присланных пакетов с эхо-запросами ICMP.

Базовый синтаксис:

```
ping <имя узла | IP-адрес узла>
```

Пример:

```
ping www.linuxaria.com
PING linuxaria.com (66.7.205.168)
56(84) bytes of data.
64 bytes from solid.nseasy.
com (66.7.205.168): icmp_seq=1
ttl=46 time=220 ms
64 bytes from solid.nseasy.
com (66.7.205.168): icmp_seq=2
ttl=46 time=171 ms
64 bytes from solid.nseasy.
com (66.7.205.168): icmp_seq=3
ttl=46 time=169 ms
64 bytes from solid.nseasy.
com (66.7.205.168): icmp_seq=4
ttl=46 time=169 ms
64 bytes from solid.nseasy.
com (66.7.205.168): icmp_seq=5
ttl=46 time=168 ms
^C
--- linuxaria.com ping statistics
---
5 packets transmitted, 5
received, 0% packet loss, time
4005ms
rtt min/avg/max/mdev =
```

```
168.906/179.895/220.449/20.297
ms
```

Что означает этот вывод?

`time=220 ms` – время обращения; время, прошедшее между отправкой запроса «здесь ли ты» и получения ответа «да, я здесь!». В данном случае это 220 мс. Поскольку отправка запросов происходит несколько раз, можно заметить, что это время почти постоянно, что является хорошим знаком. Время может варьироваться в зависимости от ряда факторов, включающих в себя следующие: как далеко находится ваша система от удаленного сервера, как много маршрутизаторов и другого сетевого оборудования находится между вашей системой и сервером и многое другие.

`5 packets transmitted, 5 received` – одной из особенностей протокола TCP/IP является то, что он может работать с потерями пакетов. В идеальном случае любой отправленный вами пакет должен быть доставлен по назначению, но по ряду причин это не всегда происходит. Поскольку в нормальных условиях работы сети пакеты все равно будут доставлены по назначению после одного или нескольких повторений отправки, вы даже не заметите этого. `ping` отправляет множество пакетов и отдельно оценивает количество успешно отправленных пакетов, поэтому вы можете оценить, приводит ли данное соединение к потере пакетов.

`rtt min/avg/max/mdev` – примерное время обращения - в случае, когда в первый раз один и тот же тип пакета отправляется по одному и тому же адресу назначения, в следующий раз должно уйти примерно такое же время на отправку пакета, что не всегда так. Иногда по различным причинам, таким, как режимы функционирования сетевого оборудования и

маршрут следования пакета, на некоторые процессы отправки пакета и получения ответа уходит больше времени, чем на другие. `ping` выводит эту статистику, поэтому вы можете оценить, приводит ли данное соединение к неравномерности времени обращения.

Другие примеры:

```
ping -c 5 -q 127.0.0.1
```

Параметр `-c` используется для отправки только 5 пакетов, параметр `-q` используется для того, чтобы пользователю была показана только статистика.

```
for i in {1..254}; do ping -c 1
-w 1 192.168.0.$i | grep 'from'
| cut -d' ' -f 4 | tr -d ':';
done
```

Пинг-исследование без `ntar`

Эту команду удобно использовать в том случае, когда у вас нет возможности использовать `ntar` для поиска узла с неизвестным адресом.

Запросы отправляются по адресам из диапазона от 192.168.0.1 до 192.168.0.254, но вы можете легко отредактировать эту команду для использования в вашей подсети.

Время ожидания для повышения скорости работы установлено равным 1 секунде, а в случае использования низкоскоростного соединения рекомендуется повысить значение, чтобы избежать пропущенных ответов.

```
ping -q -c1 -w3 www.linuxaria.
com 2>&1 /dev/null || echo www.
linuxaria.com ping failed |
mail -ne -s'Server unavailable'
admin@linuxaria.com
```

Проверка работоспособности сервера. В случае отсутствия ответа, отправляется

электронное письмо администратору.

Эта команда полезна в том случае, если администратор сервера хочет получать электронные письма в моменты недоступности сервера. Следует создать задачу cron, выполняющуюся каждые 5 минут с этой командой, чтобы администратору отправлялось письмо при отсутствии ответа на запрос ping в течение более чем 3 секунд.

Telnet

Telnet является протоколом из набора протоколов TCP/IP. Он практически полностью идентичен протоколу программы rlogin из UNIX. Программа telnet позволяет управлять удаленным компьютером с вашего компьютера. Она является программным эмулятором терминала. В прошлом жесткие диски были сложны и дороги (я говорю об очень давнем времени) и не было персональных компьютеров. Для использования имеющихся компьютеров необходимо было выделить часть жесткого диска и использовать терминал для управления системой. Для разработчиков это было прекрасным решением, так как работа с компьютером обходилась дешевле. Необходим был только сервер, с которым можно было создать и поддерживать множество соединений. С telnet вы можете имитировать этот тип распределенных вычислений и, например, управлять суперкомпьютером на расстоянии.

TCP/IP работает с портами, и один из них зарезервирован за telnet. Его номер 23. Существует несколько спецификаций RFC. Спецификация номер 854 от 1983 года носит название «Спецификация протокола telnet» (Telnet protocol specification).

С telnet вы можете делать множество вещей, например, отправлять электронную

почту, использовать IRC-конференции или прокси-сервера, и даже (хотя, конечно, сложнее) просматривать и изменять содержимое сайтов. Существуют сервисы, предоставляющие возможность поиска в огромных базах данных при помощи telnet. Благодаря telnet, вы можете использовать всю мощь удаленного компьютера, не тратя ценных ресурсов своей системы.

А теперь давайте рассмотрим, что мы сможем сделать при помощи telnet. Синтаксис вызова:

```
telnet <имя узла | IP-адрес узла>
```

Telnet должен создать TCP-соединение с системой, IP-адрес и порт которой указаны пользователем для проверки возможности соединения. Если время соединения истекает, telnet не может установить соединение с узлом (возможно, соединение блокируется межсетевым экраном). Если соединение отклонено, telnet может установить соединение с узлом, но либо служба не работает на заданном порту, либо доступ к службе заблокирован.

```
telnet www.linuxaria.com 80
Trying 66.7.205.168...
Connected to linuxaria.com.
Escape character is '^['.
GET /index.php HTTP/1.1
HTTP/1.1 400 Bad Request
Date: Fri, 24 Sep 2010 21:02:46 GMT
Server: Apache/2.2.14 (Unix)
mod_ssl/2.2.14 OpenSSL/0.9.8e-fips-rhel5 mod_auth_
passthrough/2.1 mod_bwlimited/1.4
FrontPage/5.0.2.2635 PHP/5.2.11
Content-Length: 422
Connection: close
Content-Type: text/html;
```

`charset=iso-8859-1`

Использование telnet для соединения с HTTP-сервером по порту 80

После соединения отправлен HTTP-запрос GET для получения страницы сайта, что удобно для тестирования возможности доступа к сайту на удаленном сервере хостинг-провайдера с пользовательского сервера.

```
telnet towel.blinkenlights.nl
```

Просмотр Звездных Войн по telnet.

```
telnet towel.blinkenlights.nl
666
```

DIG

DIG (искатель информации о домене – domain information groper) является гибким инструментом для опроса серверов имен DNS. Он может проводить поиск записей и отображать результаты, возвращаемые сервером имен в ответ на запрос. Большинство администраторов DNS-серверов используют dig для поиска проблем с DNS потому, что dig отличается гибкостью, простотой использования и ясностью вывода. Другие аналогичные программы зачастую отличаются меньшей функциональностью.

Несмотря на то, что dig чаще всего используется с параметрами командной строки, предусмотрен и режим работы по сценарию, при котором производится чтение запросов из файла. Краткое описание параметров командной строки и аргументов выводится при передаче параметра -h. В отличие от ранних версий программы, реализация dig от разработчиков сервера BIND9 позволяет производить несколько поисков одновременно из командной строки.

При осуществлении запроса, dig использует список серверов из файла /etc/resolv.conf за исключением того случая, когда четко задан адрес сервера для за-

проса.

Базовый синтаксис:

```
dig @server name type
```

В качестве параметра «server» передается имя или IP-адрес сервера имен для осуществления запроса. В качестве параметра «name» передается имя записи для поиска.

```
dig +short myip.opendns.com
@resolver1.opendns.com
```

Получение вашего внешнего IP-адреса

Вместо того, чтобы открывать браузер, переходить в поисковик Google и вводить строку поиска «whatismyip», можно просто использовать эту команду в терминале. Она также иногда полезна в различных сценариях оболочки.

```
dig +short txt .wp.dg.cx
```

Запрос статьи из энциклопедии Wikipedia в терминале через записи DNS

Эта команда запрашивает статью из Wikipedia, выполняя запрос к DNS-серверу и выводит TXT-запись. Эта запись также содержит короткую ссылку URL на соответствующую запись в Wikipedia.

```
dig +short txt linux.wp.dg.cx
```

Вывод этой команды:

```
«Linux ( or ) is a Unix-like
computer operating system
assembled under the model of
free and open
source software development
and distribution. The defining
component of Linux is the Linux
kernel, an operating system
kernel first released 5 October
1991 by» « Linus Torvalds..
http://en.wikipedia.org/wiki/
Linux»
```

По материалам сайта:
rus-linux.net



Школьный
Электронный
Дневник



Школа



Учительская



Профиль



Оплата



Обучение

Социальный проект компании "ВИТ" – Школьный электронный дневник



**Функции постоянно
добавляются и
модернизируются !**



- **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 !**

**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>

