

## Локализация

```
apt-get install language-pack-ru
```

```
update-locale LANG=ru_RU.UTF-8
```

(Или в файле /etc/default/locale)

Меняем часовой пояс (сервер времени в след курсе)

```
# cp /usr/share/zoneinfo/Europe/Moscow /etc/localtime
```

```
# ntpdate time.apple.com
```

```
# date
```

## Локализация консоли.

apt install keyboard-configuration – локализует консоль

cat /etc/default/console-setup – смотрим настройки

```
-----ORIG-----
```

```
ACTIVE_CONSOLES="/dev/tty[1-6]"
```

```
CHARMAP="UTF-8"
```

```
CODESET="Lat15"
```

```
FONTFACE="VGA"
```

```
FONTSIZE="8x16"
```

```
VIDEOMODE=
```

```
-----
```

Меняем настройки локализации консоли:

```
dpkg-reconfigure console-setup
```

```
cat /etc/default/console-setup
```

```
dpkg-reconfigure console-setup
```

Или вручную подправить файл:

```
vim /etc/default/console-setup
```

```
update-initramfs -u
```

## Переменные окружения.

**set**

команда **set** - для вывода списка переменных окружения. В Ubuntu и Debian команда **set** также выведет список функций командной оболочки

**env**

выведет список экспортированных переменных окружения. Отличие данной команды от команды `set` с параметрами заключается в том, что команда `set` выводит список всех переменных окружения, включая те переменные, которые не экспортируются в дочерние командные оболочки.

Кроме того, команда `env` может также использоваться для запуска "чистой" командной оболочки (командной оболочки без наследования какого-либо окружения). Команда `env -i` позволяет очистить окружение дочерней командной оболочки.

Пример:

```
set | grep LANG
```

```
unset LANG
```

```
set | grep LANG
```

```
export LANG=ru_RU.UTF-8
```

## УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ

Учетные записи в

```
cat /etc/passwd
```

Пароли:

```
cat /etc/shadow
```

пароль (шифрованный хэш-пароля) и параметры (устаревание смена блокировка)

Создать пользователя

```
useradd user1 -m -s /bin/bash
```

Изменить параметры пользователя

```
usermod user1 -c 'Ivanov Ivan Ivanovich,239,45-67,499-239-45-33'
```

Задать/сменить пароль:

```
passwd
```

Посмотреть:

```
id userX
```

или

```
cat /etc/passwd
```

Создать группу

```
groupadd mygroupname
```

Посмотреть:

cat /etc/group

Добавить в группу

usermod -G группа пользователь

Удалить из группы

usermod -G '' userX

**Чтобы увидеть к каким группам вы принадлежите используйте команду *groups*:**

\$ groups

**Смена первичной группы пользователя**

**-g** – меняет первичную группу указанного пользователя

**newgrp группа** – временно меняет первичную группу пользователя в пределах нового открываемого этой командой сеанса.

**Добавить в группу множество пользователей**

root@server:~# gpasswd -M user1,user2 test

**Удалить из группы**

**gpasswd -d, --delete пользователь**

**Удалить пользователя из указанной группы**

**Locked Accounts:**

1. nologin shell (сообщения можно задать в /etc/nologin.txt)

# vi /etc/passwd

user2:x:1002:1002:Service account:/home/user2:/sbin/nologin

Включение-отключение учетной записи:

\$ sudo usermod -L user1       ##(-U – unlock)

Др. способ - в /etc/shadow заменить hash на !! .

Задать дату окончания действия учетной записи:

\$ sudo chage -E 2018-09-11 user1       ##(-1– unlock)

Password Aging (chage)

Смена пароля при первом входе:

\$ sudo chage -d 0 user1

Парольные политики

установить libpam-cracklib, прописать в /etc/pam.d/common-password:

password required pam\_cracklib.so retry=4 minlen=8 difok=1 dcredit=-1 lcredit=-1 ucredit=-1  
password required pam\_unix.so use\_authtok nullok md5 remember=4

параметр `retry=4` это количество попыток для смены пароля

`minlen=8` минимальную длину пароля 8 символов

`dcredit` наличия в пароле хотя бы одной цифры

`lcredit` и `ucredit` наличия в пароле хотя бы одной маленькой и большой букв

`difok=1` старый пароль должен отличаться от нового не менее чем на 1 символ

Файл конфигурации `sudo`

`sudo less /etc/sudoers`

`sudo visudo`

Редактирование `sudoers` (`visudo`)

`visudo` позволяет использовать любой редактор. Например, `vim`:

`EDITOR=vim visudo`

*Или*

`# ls -l /etc/alternatives/editor`

`# sudo update-alternatives --config editor`

Шаблон, для добавления пользователю доступа к `sudo` без пароля:

`user1 ALL=(operator) NOPASSWD: ALL, (root) ALL`

Пример делегирования задачи:

**`userX ALL = NOPASSWD: /bin/tar -cjf - etc/`**

### **запретить доступ по SSH**

Создаем группу `ssh-deny`.

`groupadd ssh-deny`

`usermod -G ssh-deny username`

Настраиваем конфигурационный файл:

`# vi /etc/ssh/sshd_config`

в конец файла добавляем строку:

`DenyGroups ssh-deny`

`# service ssh restart`

## Разрешения(права) в Linux/UNIX

a								
u			g			o		
r	w	x	r	w	x	r	w	x

	r	w	x
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

### Управление маской доступа

**Маска определяет разрешения по умолчанию на создаваемые файлы**

**У рута 0022 соотв разрешениям -644**

**У остальных 0002 соотв разрешениям -664**

Типовая = 660

Режим полного доступа для директорий — 777, для файлов — 666.

Для директории:

$777 \text{ И } (\text{НЕ } 022) = (111 \ 111 \ 111) \text{ И } \text{НЕ}(000 \ 010 \ 010) = (111 \ 111 \ 111) \text{ И } (111 \ 101 \ 101) = (111 \ 101 \ 101) = (\text{rwx} \ \text{r-x} \ \text{r-x}) = 755$

Для файла:

$666 \text{ И } (\text{НЕ } 022) = (110 \ 110 \ 110) \text{ И } \text{НЕ}(000 \ 010 \ 010) = (110 \ 110 \ 110) \text{ И } (111 \ 101 \ 101) = (110 \ 100 \ 100) = (\text{rw-} \ \text{r-} \ \text{r-}) = 644$

Т.е. сначала проводится операция побитового отрицания значения `umask`, а затем, полученное значение вычисляется как побитовое И со значением прав создаваемого файла или каталога.

`umask` – выводит текущее значение маски.

`umask 0022` – задает значение маски.

### ПРАВА ДОСТУПА К ПАПКЕ

Права доступа к папке задаются аналогично правам доступа к файлу, но их смысл не так прозрачен, как в случае с файлами:

Для директорий право на исполнение позволяет сделать данную директорию текущей, например, перейти в неё командой `"cd"`.

Также для получения подробной информации о файлах, находящихся в каталоге нужно иметь доступ на исполнение папки

**chmod** - программа для изменения прав доступа к файлам и директориям. Меняет маску доступа к файлу.

**chmod 666 file1**

Другой способ назначения прав - это символьное представление.

**u** - владельцу объекта;

**g** - группе объекта;

**o** - пользователю «все остальные»;

**a** - все вышеперечисленное.

Для назначения прав используются три знака: минус, плюс или равно:

- - убрать указанные права с объекта;
- + добавить указанные права к существующим правам объекта;
- = заменить права объекта на указанные.

Пример:

- **chmod g+w file.txt** - Добавить пользователям группы права на запись в этот файл;
- **chmod a=rwx file.txt** - Заменит существующие права на файл на полные права всем;

**-R** - рекурсивное назначение прав

Например:

**chmod -R 755 \*** - Назначение прав всем объектам текущего каталога, включая подкаталоги

Текущая маска доступа **ls -l**

**Идентификатор пользователя (UID) и текущий идентификатор пользователя (EUID)**

UID (User ID) — это идентификатор пользователя, создавшего данный процесс.

EUID (Effective User ID) — это текущий пользовательский идентификатор процесса.

SUID-программы всегда выполняются с правами владельца программы.

**setuid** и **setgid** являются флагами прав доступа, которые разрешают пользователям запускать исполняемые файлы с правами владельца или группы владельца исполняемого файла

Добавить **setuid**: **chmod u+s file1**

**Использование sticky-бита** в каталогах. Каталог с установленным sticky-битом означает, что удалить файл из этого каталога может только владелец файла или суперпользователь.

Восьмеричное значение sticky-бита: 1000

Символьное: +t

Установить sticky-бит на каталог:

```
chmod +t user1
```

Убрать sticky-бит:

```
chmod -t user1
```

## POSIX ACL

### # apt-get install acl

Пример:

```
setfacl -m u:user1:rw file.txt
```

- назначает пользователю user1 права на чтение и запись.

```
setfacl -m g:group1:r file.txt
```

- назначает группе group1 права на чтение

## УПРАВЛЕНИЕ ПО.

### Репозитории.

#### Официальные репозитории

настройки всех репозиториях находятся в каталоге /etc/apt.

Все официальные репозитории Ubuntu - /etc/apt/sources.list.

в Ubuntu есть несколько веток для различного рода программного обеспечения:

- **Main** - основная ветка - стабильное официально поддерживаемое Canonical ПО
- **Restricted** - официально поддерживаемое ПО распространяемое не под лицензией GPL
- **Universe** - программы поддерживаемые сообществом Ubuntu
- **Multiverse** - ветка проприетарного программного обеспечения

#### Персональные архивы пакетов (PPA)

для корректной работы с подключённым PPA необходимо импортировать его ключ в систему. Ключ - это часть строки «Signing key» после слеша. Рядом есть ссылка «What is this?», при нажатии на которую появится краткая справка о том, что такое подпись репозитория и как импортировать ключ в систему. Для импорта ключа выполнить команду:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 12345678
```

См <http://help.ubuntu.ru/wiki/ppa>

dpkg -l покажет, какие пакеты установлены в системе.

многие пакеты являются динамическими библиотеками

dpkg -l | grep ssh

**Утилита ldd - print shared library dependencies**

**# ldd /bin/bash**

Зависимости:

**Содержимое пакета**

# dpkg -L openssh-server

**В какой пакет входит файл**

# dpkg -S /etc/init/ssh.conf

Для анализа и установки/удаления пакета и зависимостей используется APT (Advanced Packaging Tool).

Конфиг файл /etc/apt/sources.list

APT позволяет:

- устанавливать, удалять и обновлять пакеты
- посмотреть зависимости
- искать пакеты по заданным критериям
- просматривать подробную информацию о пакете

Основные команды APT:

- list - список пакетов
- search - поиск пакетов по имени
- show - показать подробную информацию о пакете
- update - обновить списки доступных пакетов
- install - установить пакет
- remove - удалить пакет
- upgrade - установить доступные новые версии пакетов
- full-upgrade - полное обновление системы
- edit-sources - редактировать файл источников программного обеспечения

Все файлы настроек APT хранятся в директории /etc/apt.

**apt - объединяет в одной утилите такие команды как: apt-get, apt-cache и т.д.**

apt-get update используется для синхронизации с репозиториями, указанными в файле /etc/apt/sources.list



`apt-get upgrade` - для обновления всех установленных пакетов

`apt-get install` для установки или модернизации пакетов

получить список всех доступных пакетов.

`apt-cache pkgnames`

узнать имя пакета и описание

`apt-cache search имя_пакета`

для проверки зависимостей пакетов:

`apt-cache showpkg имя_пакета`

Для удаления программных пакетов, включая их конфигурационные файлы

`apt-get purge имя_пакета`

Для удаления программных пакетов, не включая их конфигурационные файлы

`apt-get remove имя_пакета`

**aptitude** является более новым инструментом, полностью заменяет функционал программ `apt-get` и `apt-cache`

**# aptitude** без параметров запустится в интерактивном режиме,

**# aptitude update**

**# aptitude safe-upgrade** - будут обновлены те пакеты, для обновления которых не требуется удалять другие пакеты

**# aptitude full-upgrade** - обновит все пакеты, для которых есть новые версии.

**# aptitude install**

**# aptitude remove**

Для полного удаления пакета вместе с файлами конфигурации выполняем:

**# aptitude purge**

зафиксировать версию пакета `gzip` выполним:

**# aptitude hold** имя\_пакета

Теперь пакет не будет обновляться при выполнении **safe-upgrade** или **full-upgrade**

снять фиксацию версии для пакета:

**# aptitude unhold**

**В CentOS.** YUM ( Yellowdog Updater Modified ) –разработанный в RedHat диспетчер для работы с пакетами в формате RPM. Позволяет устанавливать, удалять и обновлять пакеты в дистрибутивах, основанных на RedHat.

Опции:

update - обновить список репозитория;

list - показать список пакетов;

provides - поиск пакета по функциональности;

search - поиск пакета по имени;

info - информация о пакете

install - установить пакет;

update-to - обновить пакет до версии;

update-minimal - обновлять только исправления ошибок;

upgrade - полное обновление с обработкой дополнительных возможностей пакетов и их зависимостей;

remove - удалить пакет;

autoremove - очистить больше ненужные пакеты;

distro-sync - синхронизировать состояния системы или пакета с репозиторием. (Например, если установлена версия пакета 2, а в репозитории версия 1, будет выполнено понижение версии).

### **Установка ПО из исходных текстов**

**apt-get install gcc libncurses5-dev make**

<http://lynx.invisible-island.net/lynx.html>

cd /usr/src/

wget http://invisible-island.net/datafiles/release/lynx-cur.tar.gz

tar -xvf lynx-cur.tar.gz

cd lynx2.8.9dev.9/

./configure

make

make install

make clean

lynx <http://www.specialist.ru>

make uninstall

**Прим.** Скрипт configure пытается определить правильные значения для зависящих от системы переменных, которые используются в процессе установки. Он использует их для создания файлов Makefile. Он также может создавать один или несколько файлов .h содержащих зависящие от системы определения. В заключение, он создает скрипт командного процессора с именем config.status, который можно запускать для воссоздания текущей настройки, также создается файл config.cache, который сохраняет результаты тестов, для ускорения перенастройки, и файл config.log, содержащий вывод компилятора (полезен для отладки configure).

Файл configure.in используется для создания скрипта configure программой autoconf.

## Регистрация событий в Linux

### Использование утилиты logger

```
$ logger -t postfix -p mail.info 'Message from postfix'
```

Logger позволяет послать сообщение от имени (указанного) с указанным явно уровнем важности

Результат:

```
$ tail /var/log/syslog
```

```
$ tail /var/log/mail.log
```

### Резервное копирование и восстановление

#### Утилиты для резервного копирования

Для выполнения этих задач использовать средства самих приложений, если предусмотрены. Особенно для СУБД (mysqldump), служб каталогов и т.д.

#### Tape archiver – tar. Архивирует каталоги.

#cd / - переходим в родительский относительно архивируемого каталог (чтоб при восстановлении не перезаписать по абс пути данные)

-c #create (создать)

-t #type (просмотреть)

-x #extract (распаковать)

-j #сжимать

-f (файл архива)

Далее указываются каталоги для архивирования.

-v #(verbose)

```
# sudo tar-cvf etc.tar etc/
```

```
# sudo tar -xf /etc.tar -в текущий каталог
```

## Планирование заданий

Для задач по расписанию в linux средство crontab

`crontab -e`

отрывается редактор:

`0 */1 * * * <задача для выполнения раз в час>`

раз в минуту, тогда запись будет выглядеть так:

`*/1 * * * * xxxxx`

## ПРИМЕР ЗАДАЧИ РЕЗЕРВНОГО КОПИРОВАНИЯ

### АУТЕНТИФИКАЦИЯ НА КЛЮЧАХ

Генерируем ключевые пары

```
root@server:~# ssh-keygen
```

Копируем ключ на сервер:

```
# ssh-copy-id serverX
```

# ssh serverX – подключение без запроса ввода пароля.

```
vi /etc/cron.hourly/backup_conf.sh
```

```
#!/bin/sh
```

```
echo Backup conf
```

```
CMD="/bin/tar"
```

```
RCMD="ssh root@serverX"
```

```
DIRS="etc/"
```

```
cd /; $CMD -cjf - $DIRS | $RCMD "cat > `hostname`.backup_conf.`date '+%Y%m%d'`.tbz"
```

```
chmod +x /etc/cron.hourly/backup_conf.sh
```

```
# crontab -e
```

Делать каждые 5 минут:

```
*/5 * * * * /etc/cron.hourly/backup_conf.sh
```

```
# cat /etc/crontab
```

## **Работа с дисками**

Список оборудования (сообщения ядра) выводится:

```
dmesg | less
```

```
# dmesg | grep sdb
```

```
# lshw – вывод инфы об оборудовании
```

```
# lshw -C disk – вывод инфы о дисках
```

```
# fdisk -l /dev/sda
```

Подключаем второй диск к контроллеру SATA.

Создать раздел

```
# fdisk /dev/sdb
```

Предупредит, что разделов нет

n - Создать раздел

P – pri partition

1

p – посмотреть разделы

Далее по умолч (Enter)

w – записать изменения

## **Создать файловую систему**

```
# mkfs -t ext4 /dev/sdb1
```

```
# file -s /dev/sdb1
```

## **Смонтировать раздел**

```
# mkdir /disk2
```

```
# mount /dev/sdb1 /disk2
```

```
touch /disk2/file1.txt
```

```
ls -a /disk2
```

```
umount /disk2
```

```
ls -a /disk2
```

```
cat /etc/fstab
```

Чтоб не было путаницы с дисками после переподключения шлейфов, назначается UUID

Можно узнать командой

```
# blkid
```

нужно прописать монтирование при загрузке в файле /etc/fstab

...

Или UUID=xxxxxx /disk2 ext4 defaults 0 0

Или так /dev/sdb1 /disk2 ext4 defaults 0 0

Если ошибиться в fstab, система не загрузится.

Поэтому перед перезагрузкой проверяем:

```
mount /disk2/
```

```
ls -a /disk2
```

## **Настройка GUI**

**Установка X сервера, менеджера дисплеев, оконного менеджера и клиентских программ**

```
apt-get install xorg xdm xfce4 xfce4-terminal xfce4-xkb-plugin firefox thunderbird
```

## **Проблемы с разрешением экрана в VBox**

```
root@client1:~# apt-get install virtualbox-guest-utils virtualbox-guest-x11 virtualbox-guest-dkms
```

```
root@client1:~# init 6
```

## **Сборка ядра.**

```
cd /tmp
```

```
wget https://cdn.kernel.org/pub/linux/kernel/v4.x/testing/linux-4.8-rc6.tar.xz
```

```
cd /usr/src
```

```
/usr/src# tar -xvf /tmp/linux-*.tar.xz
```

```
apt-get install kernel-package libncurses5-dev libssl-dev
```

-----

```
cd /usr/src/linux-4.8-rc6/
```

```
cp /boot/config-4.4.0-31-generic ./config
```

```
make-kpkg clean
```

### **make menuconfig**

```
make-kpkg --initrd --append-to-version=-mykernel kernel_image kernel_headers
```

На выходе получим:

```
# ls -l /usr/src/
```

```
linux-headers-4.8.0-rc6-mykernel_4.8.0-rc6-mykernel-10.00.Custom_amd64.deb
```

и

```
linux-image-4.8.0-rc6-mykernel_4.8.0-rc6-mykernel-10.00.Custom_amd64.deb
```

Установим ядро:

```
# cd /usr/src/
```

```
dpkg -i linux-image-4.8.0-rc6-mykernel_4.8.0-rc6-mykernel-10.00.Custom_amd64.deb
```

и

```
# dpkg -i linux-headers-4.8.0-rc6-mykernel_4.8.0-rc6-mykernel-10.00.Custom_amd64.deb
```

Новое ядро появится в каталоге /boot.

```
# ls -l /boot
```

```
update-grub2
```

Теперь можно перезагрузиться, и проверить версию ядра

```
# uname -r
```