

Урок 5. Подключение сторонних репозиториев, ручная установка пакетов (Д/З №5).

Задание:

1. Подключить дополнительный репозиторий на выбор: Docker, Nginx, Oracle MySQL. Установить любой пакет из этого репозитория.
2. Установить и удалить deb-пакет с помощью dpkg.
3. Установить и удалить snap-пакет.
4. Добавить задачу для выполнения каждые 3 минуты (создание директории, запись в файл).
5. * Подключить PPA-репозиторий на выбор. Установить из него пакет. Удалить PPA из системы.
6. * Создать задачу резервного копирования (tar) домашнего каталога пользователя. Реализовать с использованием пользовательских crontab-файлов.

1. Начните с загрузки пакета репозитория с помощью инструмента **wget** из командной строки.

```
$ wget -c https://dev.mysql.com/get/mysql-apt-config\_0.8.16-1\_all.deb
```

Затем установите пакет репозитория **MySQL**, используя следующую команду **dpkg**.

```
$ sudo dpkg -i mysql-apt-config_0.8.16-1_all.deb
```

Затем загрузите последнюю версию для пакета из всех сконфигурированных репозиториев, включая недавно добавленный репозиторий **MySQL**.

```
$ sudo apt update
```

2. Существуют два метода установки deb пакетов при помощи утилиты dpkg. Первый метод заключается в том, что перед установкой deb пакета необходимо указать путь к папке которая содержит deb пакет.

Например необходимо установить пакет `pinta_1.3-1_all.deb` который находится в папке `/home/имя_пользователя/Загрузки`.

Установка deb пакета:

указываем путь к папке содержащий deb пакет

```
cd /home/имя_пользователя/Загрузки/
```

устанавливаем deb пакет

```
sudo dpkg -i pinta_1.3-1_all.deb
```

Во втором методе можно сразу установить deb пакет, без предварительного указания папки содержащий deb пакет, вместо этого мы сразу указываем путь к пакету:

```
sudo dpkg -i /home/имя_пользователя/Загрузки/pinta_1.3-1_all.deb
```

Оба метода правильные, первый в основном используется когда необходимо установить несколько deb пакетов которые находятся в одной папке, но для того чтобы не устанавливать deb пакеты по одиночке используют команду `sudo dpkg -i *.deb`, например:

```
cd
```

```
/home/имя_пользователя/папка_содержащая_несколько_deb_пакетов/  
dpkg -i *.deb
```

Команда `dpkg -i *.deb` установить все deb пакеты находящиеся в указанной папке.

Помимо команды `dpkg -i *.deb` для установки нескольких deb пакетов можно просто указать какие пакеты необходимо установить, если все они содержатся в одной папке:

```
sudo dpkg -i пакет1.deb пакет2.deb пакет3.deb
```

Удаление deb пакета:

```
sudo dpkg -r имя_deb_пакета
```

Например:

```
sudo dpkg -r pinta
```

3. Для использования snap пакетов сначала надо установить менеджер пакетов snap. Для этого выполните:

```
sudo apt install snapd
```

После установки для включения возможности установки классических приложений через snap надо создать символическую ссылку на `/var/lib/snapd/snap` в корне:

```
sudo ln -s /var/lib/snapd/snap /snap
```

После этого можно установить магазин приложений snap-store:

```
sudo snap install snap-store
```

Или устанавливать пакеты snap через терминал.

Удалить snap пакет из системы очень просто. Для этого достаточно использовать команду **remove**:

sudo snap remove chromium

4. 1. Редактировать/добавить задание для Crontab

Для редактирования (существующих заданий) или добавления нового cron используйте команду **crontab -e**. Это откроет редактор для текущего пользователя.

\$ crontab -e

Чтобы отредактировать файл crontab другого пользователя, введите:

\$ crontab -u username -e

Или просто запустите команду **crontab -e**, войдя в систему от имени этого пользователя.

2. Список задач Cron

Чтобы просмотреть список всех заданий текущего пользователя, введите:

\$ crontab -l

Чтобы просмотреть список всех заданий cron для указанного пользователя, введите:

\$ crontab -u username -l

3. Удалить задачу cron

Чтобы удалить все задания cron текущего пользователя, выполните следующую команду:

crontab -r

Примечание: Это приведет к удалению всех заданий cron без запроса подтверждения.

Если вы хотите удалить в интерактивном режиме, вам нужно использовать следующую команду:

crontab -ir

Запускать задание cron каждые 3 минут

Иногда вам нужно следить за своей системой каждые n минут.

Используйте следующее выражение для выполнения задачи каждые 3 минут.

***/3 * * * * /scripts/test.sh**

5. Если вы хотите получить список всех репозиторий, доступных в вашей системе Ubuntu, введите следующую команду в терминале:

apt policy

Другой способ получить список всех добавленных PPA-репозиториях - распечатать содержимое каталога `/etc/apt/sources.list.d`. В этом каталоге содержится список всех репозиториях. Этот каталог содержит список всех репозиториях, доступных в вашей системе.

Вот как использовать эту команду:

ls -l /etc/apt/sources.list.d

Добавить PPA-репозиторий

Только авторизованный пользователь может добавлять/удалять и настраивать программное обеспечение и репозитории в Ubuntu. Вот синтаксис команды, которую вы можете использовать как `sudo`, чтобы добавить PPA в вашу систему:

add-apt-repository ppa:[ppa_name]

А затем установите необходимый пакет из добавленного репозитория с помощью `sudo` следующим образом: Advertisement

apt install [имя_пакета]

Начиная с Ubuntu 18.04 вам не нужно использовать команду `apt-get update` после добавления нового репозитория, так как система автоматически выполняет эту операцию. В более ранних версиях Ubuntu перед установкой пакета необходимо было выполнить следующую команду, чтобы установить последнюю доступную версию программы:

sudo apt-get update

Приведем пример, в котором мы добавим PPA-репозиторий Thomas Schiex в нашу систему, которую впоследствии будем использовать для установки Blender, пакета для создания 3D.

Мы добавим PPA с помощью следующей команды `sudo`:

sudo add-apt-repository ppa:thomas-schiex/blender

Теперь, когда PPA добавлен, для установки последней версии приложения Blender достаточно ввести следующую команду:

sudo apt-get install blender

Удаление репозитория PPA

Если вы хотите удалить добавленный PPA-репозиторий из вашей системы, вы можете использовать следующий синтаксис команды как `sudo`:

`add-apt-repository -remove ppa:[ppa_name]`

Приведенная выше команда удаляет только PPA из вашей системы, но не пакет программного обеспечения, который вы установили через него.

Например, чтобы удалить **репозиторий**, который я добавила в предыдущем разделе, я выполняю следующую команду в терминале:

`sudo add-apt-repository --remove ppa:thomas-schiex/blender`

Очистка хранилища

Очистка репозитория означает удаление его из вашей системы, а также понижение версии пакетов программ, которые вы загрузили через него. Существует утилита PPA Purge, которую вы можете использовать для этой цели.

Вы можете установить эту утилиту из официальных репозиториях Ubuntu следующим образом:

`sudo apt-get install ppa-purge`

Затем вы можете использовать следующий синтаксис команды для очистки добавленного репозитория:

`ppa-purge ppa:[ppa_name]`

Пример:

`ppa-purge ppa:thomas-schiex/blender`

Удаление репозитория вручную

Как упоминалось выше, файлы `.list` добавленного репозитория сохраняются в каталоге `/etc/apt/sources.list.d`. Вы можете вручную удалить PPA из этого каталога с помощью команды `rm` следующим образом:

`rm /etc/apt/sources.list.d/ppa_list_file`

Пример:

`sudo rm /etc/apt/sources.list.d/thomas-schiex-ubuntu-blender-*.list`

6. В простейшем случае скрипт для сохранения backup-копии будет выглядеть следующим образом:

```
#!/bin/sh
tar -zcf /backup/home.tar.gz /home
tar -zcf /backup/etc.tar.gz /etc
```

В этом скрипте нет ничего особенного, но основные принципы он демонстрирует:

- Сохранение и сжатие директории /home (домашние директории пользователей) в отдельный файл;
- Сохранение и сжатие директории /etc (общие системные настройки) в отдельный файл.

Рассмотрим использованные опции команды tar:

-z

Сжать файл с использованием gzip

-c

Создать новый архив

-f

Использовать указанный файл

Следующий скрипт реализует более широкие возможности для сохранения backup-копии:

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

```
tar -zcvpf /backup/backup-`date '+%F'`.tar.gz --directory / \
    --exclude=proc --exclude=var --exclude=mnt --exclude=usr --
exclude=backup .
```

В данном примере сохраняются не отдельные директории, а корневая директория /, исключая /proc, /var, /mnt, /usr и, конечно, /backup. Также к имени файла добавляется дата создания резервной копии.

Дополнительно к уже рассмотренным были задействованы следующие опции команды tar:

-v

Выводить список обработанных файлов

-p

Сохранять информацию о правах доступа

—directory

Директория для сохранения в архив

—exclude

Исключить директорию при сохранении в архив

Вы можете комбинировать оба предложенных варианта. Например, использовать первый скрипт ежедневно для ручного резервного

копирования, а второй запускать по расписанию раз в неделю для полного сохранения.

Другой вариант исключить примонтированные файловые системы — использовать параметр

—one-file-system

Осторожно со сжатыми копиями!

В случае порчи хотя бы нескольких байт сжатого резервного архива исключается возможность восстановления файлов из этого архива, даже если восстанавливаемый файл не находится в поврежденной области.

Поэтому, на такие носители как магнитная лента резервные копии предпочтительнее записывать в не сжатом виде. Для этого не нужно указывать опцию -z команды tar.