

## Лабораторная работа № 3

### Установка программного обеспечения

#### Задачи:

1. Научиться получать информацию об установленных и доступных пакетах и устанавливать пакеты программного обеспечения из репозитория в Linux.
2. Научиться устанавливать программные продукты из исходных кодов в Linux.
3. Научиться конфигурировать менеджер пакетов и создавать собственные репозитории.

#### Рекомендации:

1. Программные продукты под Linux распространяются или в виде готовых бинарных пакетов (с расширением `deb` для ветви Debian и расширением `rpm` для ветви RedHat) или в виде исходных кодов.
2. Из бинарного пакета программное обеспечение устанавливается с помощью специального ПО — менеджера пакетов (например, **dpkg** и **apt** для Debian ветви и **rpm** или **dnf** (в ранних версиях **yum**) для ветви RedHat). Менеджеры пакетов также позволяют работать с репозиториями — коллекциями пакетов, некоторые из которых поддерживаются производителями дистрибутивов. На работу с такими репозиториями, как правило, по умолчанию настроена система.
3. Для установки, обновления и просмотра информации о пакетах из сетевого репозитория используется менеджер пакетов **apt**. Например для того, чтобы установить пакет используется команда **apt install <пакет>**, команда **apt build-dep <пакет>** позволяет поставить зависимости для пакета, а команда **apt download <пакет>** - скачать сам бинарный пакет.
4. Сама команда **apt** — новая утилита, объединяющая функции старых утилит **apt-get**, **apt-cache** и других инструментов из пакета **apt-utils**. Но некоторые функции в **apt** не реализованы. Так для того, чтобы скачать исходные коды программ лучше использовать команду **apt-get source <пакет>**, а иногда **git**. Для получения прямых или обратных зависимостей можно использовать **apt-cache** или пакет **debtree**.
5. Исходный код программ, в большинстве случаев, распространяется в архивах. Для создания этих архивов, как правило, используются две программы: **tar** и **gzip**. Первая собирает все исходные файлы, включая структуру каталогов в один общий файл, а вторая сжимает этот файл для того, чтобы сэкономить время передачи архива по сети.
6. Сборка и установка пакетов из исходных кодов осуществляется с помощью последовательного запуска команд:  

```
./configure  
make  
make install
```

Эти команды должны быть запущены так, чтобы текущим каталогом при этом был каталог с исходным кодом устанавливаемого пакета, содержащий файл с именем **Makefile**.
  - **./configure** - подготавливает среду для сборки, проверяет наличие необходимых библиотек и настроек, это не обязательный этап.
  - **make** - компилирует программу, если нет необходимых для программы компонентов, то компилятор в сообщении об ошибке будет указывать какого именно.
  - **make install** - устанавливает скомпилированную программу в систему, но в только в том случае, если в **Makefile** есть секция **install**.
7. Для создания собственного репозитория должен быть установлена и использована специальная утилита **dpkg-scanpackages** из пакета **dpkg-dev** для Debian.
8. Для аутентификации в репозиториях используются ассиметричные ключи, для управления которыми используется утилита **gpg**, но можно настроить репозиторий без аутентификации, как доверенный.
9. Для каждого репозитория в системе должен быть конфигурационный файл. Эти файлы располагаются в каталоге: `/etc/apt/sources.list.d/` для Debian или в общем файле `/etc/apt/sources.list`

#### Задание:

Выполните указанные ниже действия и запишите все использовавшиеся команды с параметрами и изменения в файлах в отдельный файл протокола. Для удобства выполнения работы рекомендуется настроить доступ к виртуальной машине по `ssh`.

#### Порядок действий:

1. Выведите список всех подключенных репозитория
2. Обновите локальные индексы пакетов в менеджере пакетов.

3. Выведите информацию о метапакете **build-essential**,
4. Установите метапакет **build-essential**, при этом определите какие компоненты будут установлены, а какие обновлены.
5. Найдите пакет, в описании которого присутствует строка «clone with a bastard algorithm»
6. Скачайте в отдельную директорию в домашнем каталоге архив с исходными кодами найденного в п.5 пакета.
7. Установите пакет из исходных кодов, скаченных в п.6
8. Если в конфигурационном файле пакета нет параметров установки пакета в систему, то добавьте их, так, чтобы пакет устанавливался в `/usr/local/bin` и на него назначались права `---rwxr-xr-x`. Проверьте выполнения этих директив.
9. Проверьте, что любой пользователь может запускать установленный пакет, но не тратьте на это более 5 минут.
10. Создайте файл **task10.log**, в который выведите список всех установленных пакетов.
11. Создайте файл **task11.log**, в который выведите список всех пакетов (зависимостей), необходимых для установки и работы компилятора **gcc**.
12. Создайте файл **task12.log**, в который выведите список всех пакетов (зависимостей), установка которых требует установленного пакета **libgpm2**.
13. Создайте каталог **localrepo** в домашнем каталоге пользователя **root** и скопируйте в него с сайта <http://snapshot.debian.org/package/htop/> пять разных версий пакета **htop**. Это можно сделать с помощью **wget** или просто передав файлы на виртуальную машину используя протокол **ssh** и утилиту **scp**.
14. Сгенерируйте в каталоге репозитория файл **Packages**, который будет содержать информацию о доступных пакетах в репозитории и файл **Release**, содержащий описание репозитория. Например:  
Origin: My Local Repo  
Label: My Local Repo  
Suite: stable  
Version: 1.0  
Codename: myrepo  
Architectures: amd64  
Components: main  
Description: My local APT repository
15. Обновите кэш **apt**
16. Выведите список подключенных репозиториях и краткую информацию о них.
17. Создайте файл **task16.log** в который выведите список всех доступных версий **htop**
18. Установите предпоследнюю версию пакета
19. Скачайте из сетевого репозитория пакет **nano**. Пересоберите пакет таким образом, чтобы после его установки, появлялась возможность запустить редактор **nano** из любого каталога, введя команду **newnano**. Для работы с пакетом следует использовать **dpkg-deb**, а для установки **dpkg**. В файле протокола работы опишите использованные команды.
20. Бонусный вопрос с подвохом - что есть в АРТ?