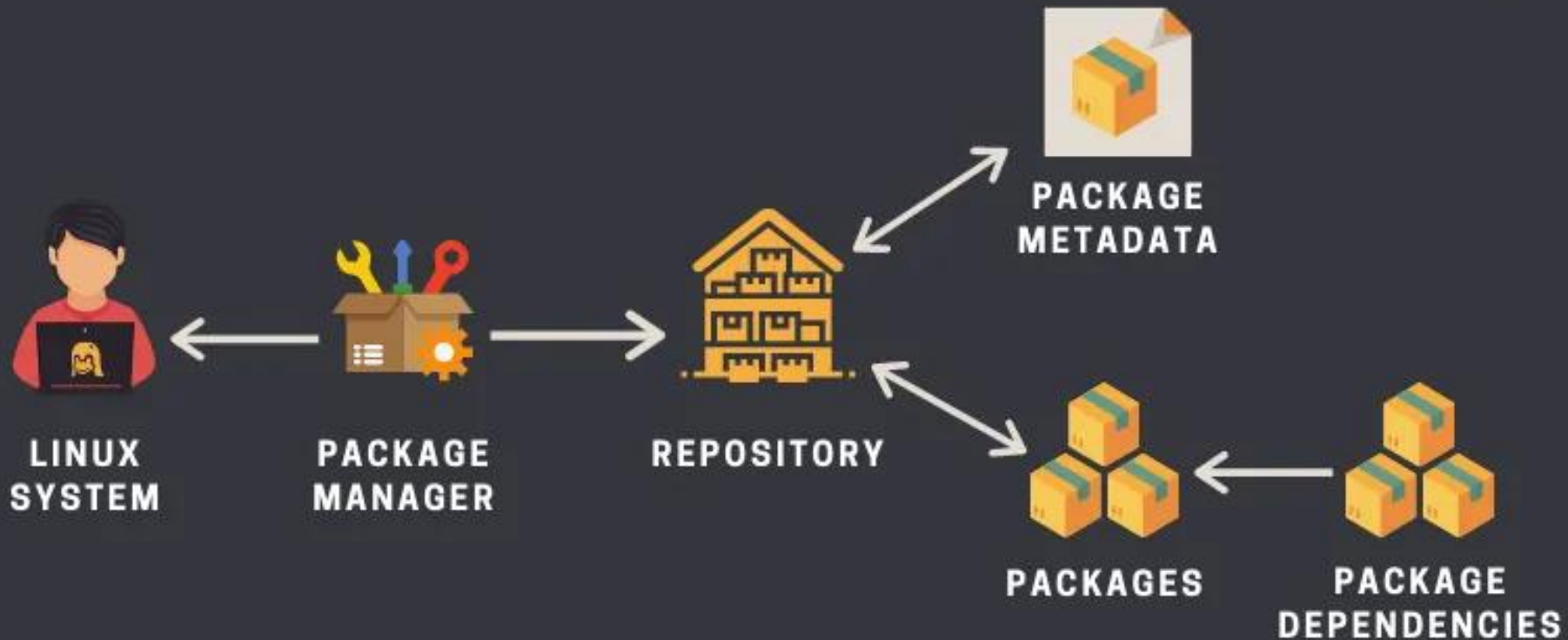


Тема: Управление пакетами и репозиториями.



План занятия:

1. Репозитории и управление репозиториями.
2. Подключение репозитория.
3. Управление пакетами.

1. Репозитории и управление репозиториями.

Пакет — под пакетами в Linux подразумевается программное обеспечение, которое можно установить, то есть набор файлов, объединённых для выполнения определённого функционала. Пакеты, как правило, хранятся в репозиториях.

Репозиторий — хранилище пакетов, то есть файлов и библиотек, которые мы можем установить в ОС. Репозиторий может размещен локально, может храниться на носителе (флешке, DVD-диске), но чаще всего размещен в интернете.

РЕПОЗИТОРИИ

локальные

в интернете

РЕПОЗИТОРИИ по типу

стандартные

дополнительные

неофициальные

Условно репозитории можно разделить на три группы:

1. Стандартные репозитории — это репозитории, поддерживаемые разработчиками операционных систем. Включают в себя стабильные версии программного обеспечения. Зачастую эти версии отстают на несколько шагов от последних версий пакетов.
2. Дополнительные репозитории — репозитории, поддерживаемые разработчиками программного обеспечения. Включают в себя последние стабильные версии ПО. Зачастую узкоспециализированы под конкретный пакет и библиотеки, необходимые для этого пакета. Например: Docker, Node.js и т.д.
3. Неофициальные репозитории — репозитории, созданные сообществом или одним человеком. Могут содержать в себе как последние стабильные, так и тестируемые версии программного обеспечения.

Ubuntu имеет несколько официальных репозиторий, которые содержат пакеты программного обеспечения, поддерживаемые разработчиками Ubuntu. Вот основные официальные репозитории Ubuntu:

1.Main: Этот репозиторий содержит основные пакеты, которые полностью поддерживаются разработчиками Ubuntu. Он включает в себя основные приложения и утилиты, необходимые для установки и работы с Ubuntu.

2.Restricted: Репозиторий Restricted содержит пакеты, которые не являются свободными по лицензии и требуют дополнительных ограничений или драйверов с закрытым исходным кодом. Включает в себя проприетарные драйверы для графики и другие ограниченные компоненты.

3.Universe: Репозиторий Universe содержит пакеты, которые поддерживаются сообществом Ubuntu, но не получают официальной поддержки от разработчиков Ubuntu. Он включает в себя множество дополнительных приложений, инструментов и библиотек.

4.Multiverse: Репозиторий Multiverse содержит пакеты, которые не свободны по лицензии и не поддерживаются сообществом Ubuntu. Он включает в себя проприетарные приложения, плагины и другие компоненты, которые могут быть полезными, но не являются частью официального дистрибутива.

Программное обеспечение Ubuntu

```
graph LR; A[Программное обеспечение Ubuntu] --- B[Main — свободное ПО, официально поддерживаемое компанией Canonical.]; A --- C[Restricted — проприетарное ПО (в основном драйверы устройств), официально поддерживаемое компанией Canonical.]; A --- D[Universe — свободное ПО, официально не поддерживаемое компанией Canonical, но поддерживаемое сообществом пользователей.]; A --- E[Multiverse — проприетарное ПО, не поддерживаемое компанией Canonical.];
```

Main — свободное ПО, официально поддерживаемое компанией Canonical.

Restricted — проприетарное ПО (в основном драйверы устройств), официально поддерживаемое компанией Canonical.

Universe — свободное ПО, официально не поддерживаемое компанией Canonical, но поддерживаемое сообществом пользователей.

Multiverse — проприетарное ПО, не поддерживаемое компанией Canonical.

Некоторые из типов записей, которые могут встречаться в репозиториях Ubuntu:

1. `$release` — пакеты на момент выхода релиза.

```
deb http://archive.ubuntu.com/ubuntu/ $release main
```

2. `$release-security` — пакеты критических обновлений безопасности.

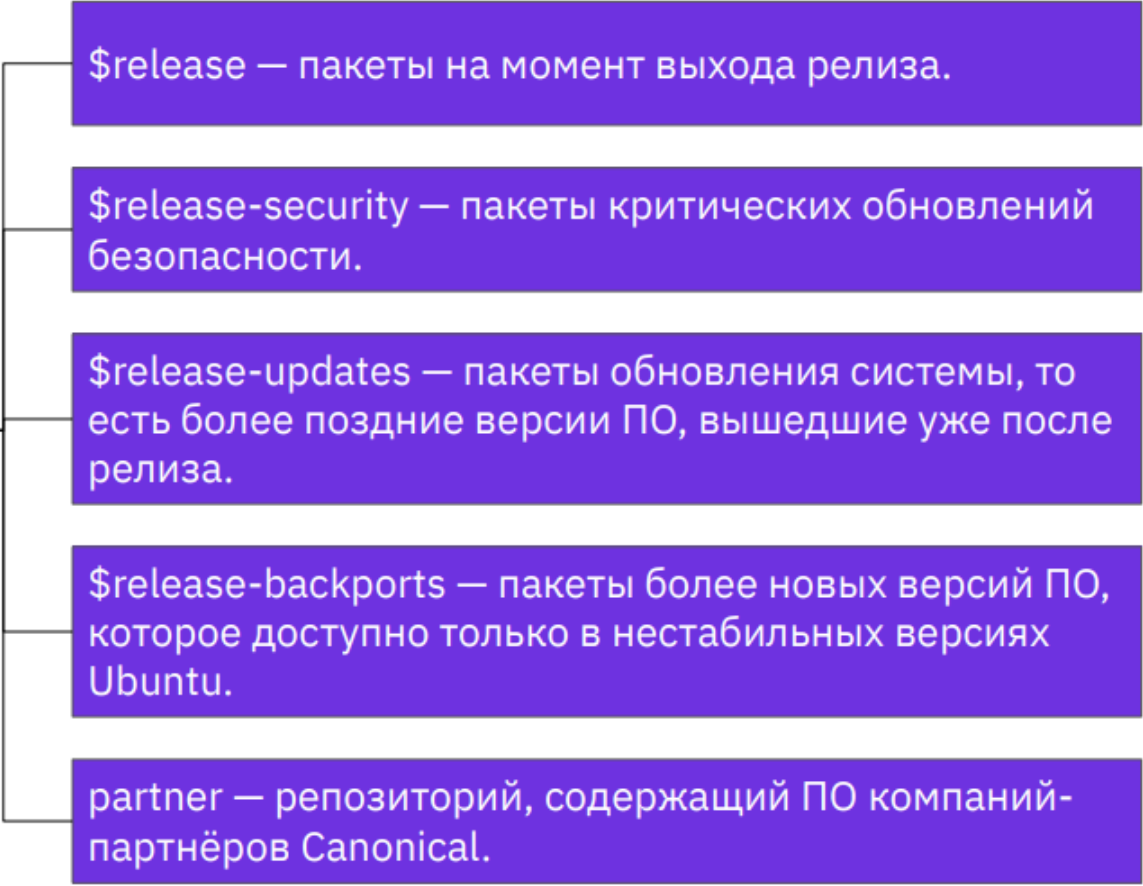
```
deb http://security.ubuntu.com/ubuntu/ $release-security main
```

3. `$release-updates` — пакеты обновления системы, то есть более поздние версии ПО, вышедшие уже после релиза.

```
deb http://archive.ubuntu.com/ubuntu/ $release-updates main
```

4. `$release-backports` — пакеты более новых версий ПО, которое доступно только в нестабильных версиях Ubuntu.
5. `partner` — репозиторий, содержащий ПО компаний-партнёров Canonical.

Официальные репозитории



```
graph LR; A[Официальные репозитории] --- B["$release — пакеты на момент выхода релиза."]; A --- C["$release-security — пакеты критических обновлений безопасности."]; A --- D["$release-updates — пакеты обновления системы, то есть более поздние версии ПО, вышедшие уже после релиза."]; A --- E["$release-backports — пакеты более новых версий ПО, которое доступно только в нестабильных версиях Ubuntu."]; A --- F[partner — репозиторий, содержащий ПО компаний-партнёров Canonical.];
```

`$release` — пакеты на момент выхода релиза.

`$release-security` — пакеты критических обновлений безопасности.

`$release-updates` — пакеты обновления системы, то есть более поздние версии ПО, вышедшие уже после релиза.

`$release-backports` — пакеты более новых версий ПО, которое доступно только в нестабильных версиях Ubuntu.

`partner` — репозиторий, содержащий ПО компаний-партнёров Canonical.

\$release, это обычно заменяется на кодовое имя версии Ubuntu (например, "**focal**" для Ubuntu 20.04 LTS). Таким образом, строки в итоге примут вид, подобный:

```
deb http://archive.ubuntu.com/ubuntu/ focal main
deb http://security.ubuntu.com/ubuntu/ focal-security main
deb http://archive.ubuntu.com/ubuntu/ focal-updates main
deb http://archive.ubuntu.com/ubuntu/ focal-backports main
deb http://archive.ubuntu.com/ubuntu/ focal-proposed main
```

Кодовое имя Ubuntu 22.04 LTS — Jammy Jellyfish.

Способы подключения репозиториев в системе

ВАЖНО: после подключения
репозитория не забудьте
обновить информацию о пакетах,
выполнив `sudo apt update`!

1. Редактирование файла `/etc/apt/sources.list`.
2. Команда `apt-add-repository`.

Существует несколько способов подключения репозитория к системе в Linux. Вот некоторые из наиболее распространенных методов:

1. Редактирование файла `sources.list`:

Основной метод в большинстве дистрибутивов Linux. Редактирование файла `/etc/apt/sources.list` или создание новых файлов в каталоге `/etc/apt/sources.list.d/` позволяет добавлять или удалять репозитории вручную.

Пример добавления строки для основного репозитория Ubuntu:

```
sudo nano /etc/apt/sources.list
```

Добавляем строку:

```
deb http://archive.ubuntu.com/ubuntu/ focal main
```

Left	File	Command	Options	Right			
<-	/etc/apt			<- ~			
.n	Name	Size	Modify time	.n	Name	Size	Modify time
/..		UP--DIR	Feb 1 07:41	/..		UP--DIR	Jan 30 10:40
/apt.conf.d		4096	Jan 30 09:12	/.cache		4096	Feb 1 07:28
/auth.conf.d		4096	Apr 8 2022	/.config		4096	Feb 1 07:28
/keyrings		4096	Apr 8 2022	/.local		4096	Feb 1 07:28
/preferences.d		4096	Apr 8 2022	/.ssh		4096	Jan 30 10:40
/sources.list.d		4096	Jan 30 09:04	.bash_logout		220	Jan 6 2022
/trusted.gpg.d		4096	Aug 10 00:20	.bashrc		3771	Jan 6 2022
sources.list		56	Jan 30 09:04	.profile		807	Jan 6 2022
sources.list.curlin.old		2403	Aug 10 00:20	.selected_editor		72	Feb 1 07:29
				.sudo_as_admin_successful		0	Jan 30 10:52
				*hello.sh		33	Feb 1 07:29
UP--DIR				UP--DIR			
5854M/11G (50%)				5855M/11G (50%)			
Hint: Use M-p and M-n to access the command history.							
srv2@srv2:/etc/apt\$							
1	Help	2	Menu	3	View	4	Edit
5	Copy	6	RenMov	7	Mkdir	8	Delete
9	PullDn	10	Quit				

/etc/apt/sources.list [-M--] 47 L:[1+ 1 2/ 3] *(103 / 104b) 0010 0x00A [*] [X]

deb [check-date=no] file:///cdrom jammy main restricted

deb http://archive.ubuntu.com/ubuntu jammy main

deb [trusted=yes] <http://archive.ubuntu.com> jammy restricted main universe

Вот разбор этой записи:

deb: Это префикс, который указывает, что источник является репозиторием, содержащим бинарные пакеты (не исходные коды).

[trusted=yes]: Этот параметр указывает APT, что этот источник является доверенным, и что пакеты, загружаемые из него, можно считать безопасными для установки.

<http://archive.ubuntu.com>: Это URL-адрес репозитория, из которого APT будет загружать пакеты. В данном случае указано официальное зеркало Ubuntu.

jammy: Это кодовое имя версии Ubuntu. В данном случае jammy обозначает Ubuntu 22.04 LTS (Jammy Jellyfish). Каждая версия Ubuntu имеет свое собственное кодовое имя.

Main: содержит основные пакеты, поддерживаемые Ubuntu;

Restricted: содержит пакеты, имеющие ограниченные права использования;

Universe: содержит сообщество-поддерживаемые пакеты.

```
deb http://archive.ubuntu.com/ubuntu/ jammy-updates main  
restricted universe multiverse
```

```
# deb-src http://archive.ubuntu.com/ubuntu/ jammy-updates main  
restricted universe multiverse
```

```
deb http://archive.ubuntu.com/ubuntu/ jammy-security main  
restricted universe multiverse
```

```
# deb-src http://archive.ubuntu.com/ubuntu/ jammy-security main  
restricted universe multiverse
```

```
deb http://archive.ubuntu.com/ubuntu/ jammy-backports main  
restricted universe multiverse
```

```
# deb-src http://archive.ubuntu.com/ubuntu/ jammy-backports main  
restricted universe multiverse
```

deb-src - это префикс, используемый в файле `sources.list` для указания источников, содержащих исходные файлы пакетов.

2. Добавление репозитория с помощью apt-add-repository:

Некоторые дистрибутивы Linux предоставляют команду `add-apt-repository` для упрощенного добавления репозитория. Эта команда также обеспечивает обновление списка пакетов и ключей.

```
srv2@srv2:~$ sudo apt-add-repository "deb http://archive.canonical.com/ubuntu jammy partner"
Repository: 'deb http://archive.canonical.com/ubuntu jammy partner'
Description:
Archive for codename: jammy components: partner
More info: http://archive.canonical.com/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Found existing deb entry in /etc/apt/sources.list.d/archive_uri-http_archive_canonical_com_ubuntu-jammy.list
Adding deb entry to /etc/apt/sources.list.d/archive_uri-http_archive_canonical_com_ubuntu-jammy.list
Found existing deb-src entry in /etc/apt/sources.list.d/archive_uri-http_archive_canonical_com_ubuntu-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-http_archive_canonical_com_ubuntu-jammy.list
Hit:1 http://kz.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://archive.canonical.com/ubuntu jammy InRelease
Hit:3 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:4 http://kz.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://kz.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:6 http://kz.archive.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
```

Теперь нужно обновить информацию о пакетах (`sudo apt update`) и обновить установленные пакеты (`sudo apt upgrade`).

```

srv2@srv2:~$ sudo apt update
Get:1 file:/cdrom jammy InRelease
Ign:1 file:/cdrom jammy InRelease
Get:2 file:/cdrom jammy Release
Err:2 file:/cdrom jammy Release
  File not found - /cdrom/dists/jammy/Release (2: No such file or directory)
Hit:3 http://kz.archive.ubuntu.com/ubuntu jammy InRelease
Get:4 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:5 http://kz.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:6 http://kz.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:7 http://kz.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1,395 kB]
Get:9 http://kz.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,326 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy/main Translation-en [510 kB]
Get:11 http://kz.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [869 kB]
srv2@srv2:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
  ubuntu-pro-client-l10n
The following packages will be upgraded:
  apparmor apt apt-utils base-files bind9-dnsutils bind9-host bind9-libs cloud-init cryptsetup
  cryptsetup-bin cryptsetup-initramfs distro-info distro-info-data git git-man initramfs-tools
  initramfs-tools-bin initramfs-tools-core irqbalance kpartx libapparmor1 libapt-pkg6.0
  libcryptsetup12 libldap-2.5-0 libldap-common libmm-glib0 libnetplan0 libnss-systemd
  libpam-systemd libsgutils2-2 libssl3 libsystemd0 libudev1 linux-firmware modemmanager
  motd-news-config multipath-tools netplan.io openssl python3-distro-info

```

Где взять адреса репозиториев?

Обычно адреса репозиториев предоставляются разработчиками ПО или сообществом дистрибутива.

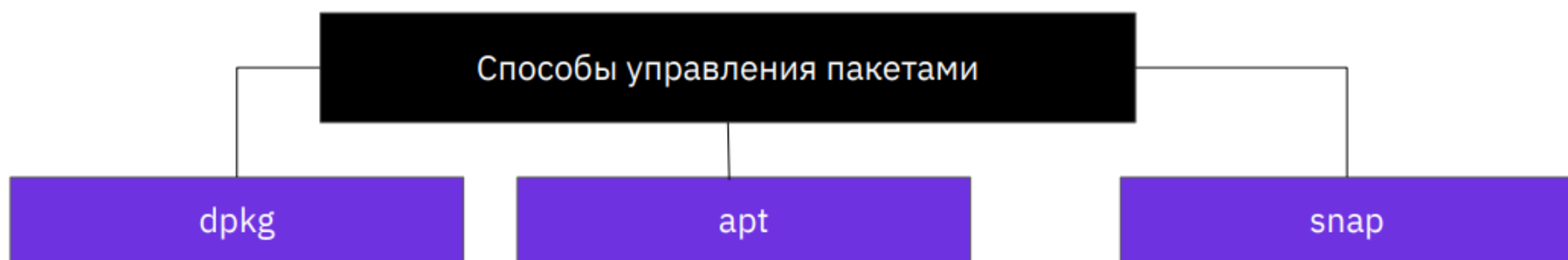
Для Ubuntu, вы можете посетить официальную страницу
<https://ubuntu.com/>

3. Управление пакетами.

Пакет — под пакетами в Linux подразумевается программное обеспечение, которое можно установить, то есть набор файлов, объединённых для выполнения определённого функционала. Пакеты, как правило, хранятся в репозиториях.

Репозиторий — хранилище пакетов, то есть файлов и библиотек, которые мы можем установить в ОС. Репозиторий может размещен локально, может храниться на носителе (флешке, DVD-диске), но чаще всего размещен в интернете.

apt — это пакетный менеджер, который включает в себя набор команд для управления пакетами. Он позволяет осуществлять поиск, установку и удаление пакетов, обновлять операционную систему, подключать репозитории.



Управление пакетами в Linux может осуществляться различными методами в зависимости от используемого дистрибутива.

APT (Advanced Package Tool):

Применение: Ubuntu, Debian и другие дистрибутивы, основанные на Debian.

Команды:

`sudo apt update`: Обновление списка пакетов.

`sudo apt upgrade`: Обновление установленных пакетов.

`sudo apt search <запрос>`: Поиск пакета

`sudo apt install <пакет>`: Установка пакета.

`sudo apt remove <пакет>`: Удаление пакета.

`sudo apt autoremove`: Удаление ненужных зависимостей.

```
svv@server22:~$ sudo apt search netstat
```

```
Sorting... Done
```

```
Full Text Search... Done
```

```
bwm-ng/jammy 0.6.3-2 amd64
```

```
small and simple console-based bandwidth monitor
```

```
gnome-nettool/jammy 42.0-1 amd64
```

```
network information tool for GNOME
```

```
golang-github-aelsabbahy-gonetstat-dev/jammy 0.0~git20160428.0.edf89f7-3 all
```

```
Netstat implementation in Go
```

```
golang-github-mwitkow-go-conntrack-dev/jammy 0.0~git20190716.2f06839-2 all
```

```
Go middleware for net.Conn tracking
```

```
libparse-netstat-perl/jammy 0.14-1.1 all
```

```
module to parse the output of the "netstat" command
```

```
net-tools/jammy 1.60+git20181103.0eebece-1ubuntu5 amd64
```

```
NET-3 networking toolkit
```

dnf (Dandified Yum):

Применение: **Fedora, Red Hat, Centos 9** и другие дистрибутивы, использующие **dnf**.

Команды:

`sudo dnf update`: Обновление списка пакетов.

`sudo dnf upgrade`: Обновление установленных пакетов.

`sudo dnf install <пакет>`: Установка пакета.

`sudo dnf remove <пакет>`: Удаление пакета.

yum:

Применение: Старые версии Red Hat, CentOS.

Команды:

`sudo yum update`: Обновление списка пакетов.

`sudo yum upgrade`: Обновление установленных пакетов.

`sudo yum install <пакет>`: Установка пакета.

`sudo yum remove <пакет>`: Удаление пакета.

dpkg — пакетный менеджер в Debian-подобных системах. Главное отличие от утилиты apt состоит в том, что dpkg работает только с **локальными** пакетами в формате .deb, он не умеет искать и устанавливать пакеты с репозиториев.

Установка пакета:

```
sudo dpkg -i имя_пакета.deb
```

Удаление пакета:

```
sudo dpkg -r имя_пакета
```

Показать информацию о пакете:

```
dpkg -l | grep имя_пакета
```

Проверка зависимостей пакета:

```
sudo dpkg -I имя_пакета.deb
```

Зависимости пакета - это условия, которые должны быть выполнены для корректной установки и работы данного пакета. Эти условия могут включать в себя другие пакеты, библиотеки, версии программного обеспечения или другие компоненты, необходимые для правильного функционирования пакета.

Проверка зависимостей пакета:

```
sudo dpkg -I имя_пакета.deb
```

Зависимости пакета - это условия, которые должны быть выполнены для корректной установки и работы данного пакета. Эти условия могут включать в себя другие пакеты, библиотеки, версии программного обеспечения или другие компоненты, необходимые для правильного функционирования пакета.

Snap (Snapcraft) — это система управления пакетами и формат пакета, разработанные Canonical для использования в операционной системе Ubuntu и других дистрибутивах Linux. Она предоставляет упакованные приложения со всеми их зависимостями, что облегчает установку и поддержание программного обеспечения.

Вот основные характеристики и особенности Snap:

1.Пакетирование со всеми зависимостями:

1. В отличие от традиционных пакетных форматов, таких как DEB и RPM, Snap-пакет включает в себя все необходимые зависимости. Это делает его более переносимым и устраняет проблемы совместимости.

2.Isolation (изоляция):

1. Snap-пакеты упаковываются в контейнер, что обеспечивает изоляцию от других частей системы. Это может повысить безопасность и стабильность приложения.

3.Обновления:

1. Snap-пакеты могут автоматически обновляться, что помогает пользователям всегда использовать последние версии программ без необходимости ожидания, когда они будут добавлены в репозитории.

4. Кросс-платформенность:

1. Snap-пакеты можно использовать на различных дистрибутивах Linux, не только на Ubuntu. Это делает Snap более универсальным в сравнении с форматами пакетов, специфичными для определенных дистрибутивов.

5. Удобство установки:

1. Установка Snap-пакетов проста и осуществляется одной командой. Например:

```
sudo snap install название_пакета
```

6. Сообщество и поддержка:

1. Snap имеет активное сообщество и широкую поддержку от разработчиков программного обеспечения. Многие популярные приложения теперь доступны как Snap-пакеты.

7. Snap Store:

1. Snap Store представляет собой централизованное место для поиска, установки и обновления Snap-пакетов.



Snap Store

Canonical Personalisation Utilities

Install

Q

AllInstalled


≡

—

□

✕

Featured Applications



GitKraken

For repo management, in-app code editing & issue tracking.

Categories

Games

Finance


Productivity

Utilities

News and Weather


Science

Editor's Picks




Rambox Pro

★★★★★




Ascii Patrol

★★★★★




Shotcut

★★★★★




OpenToonz

★★★★★



Atom

★★★★★




clementine

★★★★★


<Devel

ShowAllSortTop Rated




Sublime Text

★★★★★




PyCharm CE

★★★★★




Visual Studio ...

★★★★★




Android Studio

★★★★★




PhpStorm

★★★★★




Eclipse

★★★★★




notepadqq

★★★★★




kubectrl


★★★★★




powershell

★★★★★







Установку можно осуществить с помощью следующих команд:

```
$ sudo apt update
```

```
$ sudo apt install snapd
```

Snap store:

<https://snapcraft.io/>

Домашнее задание:

1. Изучить методичку.