## Управление пакетами Linux

### Цель лабораторной работы

Научиться устанавливать новое программное обеспечение, как при помощи специальных программ инсталляторов, так и путем компилирования из исходных файлов.

### Задания к работе

1. Откройте окно менеджера программ и познакомьтесь с набором программного обеспечения, входящего в стандартный репозиторий вашей операционной системы.
2. Найдите и установите программу chromium-browser. Проверьте ее работоспособность.
3. Удалите установленную программу firefox.
4. Произведите обратные действия (установку firefox и удаление chromium) командами в терминале.
5. Откройте менеджер пакетов Synaptic в графическом режиме.
6. Выведите полный список пакетов, установленных в системе. Подсчитайте общее количество пакетов, установленных в системе.
7. Выберите из общего списка все пакеты, имеющие в названии слово gnome (т.е. те пакеты, которые принадлежат интегрированной среде GNOME) и подсчитайте их количество.
8. Проверьте, установлен ли в системе пакет mc и получите расширенную информацию об этом пакете.
9. Выполните имитацию удаления (НЕ УДАЛЕНИЕ!) пакета perl. Подсчитайте количество пакетов, которые от него зависят.
10. Выполните имитацию установки пакетов httpd, vsftpd и webmin, проанализировать результаты (убедиться, что все зависимости удовлетворены).
11. Повторите задания 6-10 в консольном режиме с использованием программы dpkg.
12. Установите программу PyCharm, следуя инструкциям на сайте производителя. Добавьте ярлык программы на рабочий стол или панель и убедитесь, что программа работает правильно.

### Методические указания

Одна из первых вещей, на которые вы обращаете внимание при установке Linux -- это большое количество входящих в дистрибутив пакетов. Большинство дистрибутивов содержат операционную систему Linux, средства для инсталляции и средства администрирования. Кроме того, в них включаются средства для работы в Интернете, средства разработки, офисные пакеты, игры, а также некоторые средства, о которых вы даже не слышали. Дистрибутивы Linux, содержащие тысячи доступных пакетов, не редкость. Если вы не выбрали "установить все", будет установлено некоторое подмножество этих пакетов.

Теперь у вас могут возникнуть вопросы "Как удалить ненужные пакеты? Как установить что-то недостающее? Могу ли я использовать программное обеспечение, не входящее в мой дистрибутив?"

#### Зависимости

Разные разработчики разрабатывают разные программы и библиотеки. Определенная программа может зависеть от десятка библиотек, которые разработали разные люди. Следить за всем этим вручную — неблагодарное занятие.

К счастью для нас, проблема эта решена. Каждый раз когда вы ставите какое-либо ПО из репозитория или вручную при помощи пакета-установщика, программа-установщик разбирает эти зависимости и устанавливает их вместе с нужным вам приложением. Но иногда бывает и такая ситуация, когда репозитории не подключены (например, если нет интернета) и единственный способ поставить программу — скачать ее предварительно где-нибудь и потом ставить вручную. В любом случае при установке вы будете предупреждены, если нужная зависимость не установлена и не может быть найдена в репозиториях, так что теперь вы знаете в чем дело.

И напоследок о зависимостях: допустим, вам надо скачать какую-то программу. Она тянет за собой 5 зависимостей. Каждый из них тянет за собой по 5 зависимостей еще. В итоге может получиться так, что сама программа занимает 1 мегабайт, а скачать надо 50.

#### .deb = setup.exe

В Windows, чтобы установить программу, необходимо скачать некий файл, наподобие setup.exe, который сам все установит. В дистрибутивах, основанных на Debian используется похожая система, только вместо setup.exe используются файлы с расширением .deb

Это пакет-установщик. В разных линукс-системах установщики могут быть разные. Например, в системе RedHat Linux для установки используются пакеты с расширением .rpm

Зачем это нужно знать? Когда вы ищите ПО в интернете (если его нету в репозиториях) — обычно на странице загрузки предоставляются различные пакеты. Вам нужен deb! Даже когда вы устанавливаете что-либо из репозитория, невидимо от вас программа скачивает deb файл (или несколько файлов, если имеются зависимости) и потом устанавливает его.

#### Программа apt.

Все архивы программного обеспечения, компакт-диски и т. д. не принесут никакой пользы, если вы не умеете устанавливать программы. Итак, вы хотите знать, как установить новый пакет в Debian.

Воспользуйтесь командой apt install:

# apt install tuxkart

Установка пакета с перезаписью файлов:

# apt install --reinstall tuxkart

Чтобы установить сразу несколько программ, перечислите их, разделяя пробелами:

# apt install tuxkart gltron frozen-bubble tuxracer nethack galaga

Загрузка программ без установки или распаковки:

# apt -d install tuxkart

Чтобы протестировать команду перед выполнением, присоедините к строке ключ - dry-run:

# apt install tuxkart gltron frozen-bubble tuxracer nethack galaga --dry-run

#### Удаление пакетов

Для удаления одного или нескольких пакетов воспользуйтесь командой apt remove:

# apt remove tuxpaint

Предварительное тестирование команды remove:

# apt remove tuxpaint -dry-run

Удаление всех следов существования пакета, включая конфигурационные файлы:

# apt --purge remove tuxpaint

Чтобы удалить сразу несколько программ, перечислите их, разделяя пробелами:

# apt remove tuxkart gltron frozen-bubble tuxracer nethack galaga

#### Обновление пакетов

Часто требуется обновить пакет в системе Debian, потому что новая версия обладает расширенными возможностями или в ней исправлены ошибки. Воспользуйтесь командой apt install:

# apt install gltron

Обновление нескольких пакетов:

# apt install tuxkart gltron frozen-bubble tuxracer nethack galaga

Команды обновления отдельных пакетов не существует. Команда apt install всегда устанавливает последнюю версию пакета.

#### Обновление системы

При обновлении системы полностью требуется обновить все пакеты в системе и заменить их новейшими версиями. Убедитесь в том, что файл /etc/apt/sourcs.list содержит ссылки на нужные источники, а затем выполните команду apt upgrade.

Всегда начинайте с обновления списков пакетов:

# apt update

Следующая команда обновляет все установленные пакеты, но не удаляет пакеты для разрешения зависимостей:

# apt -u upgrade

Обновление всех установленных пакетов с удалением или установкой пакетов по мере необходимости для разрешения всех зависимостей:

# apt -u dist-upgrade

Флаг -u позволяет заранее просмотреть список всех изменений. Обновление может занять несколько часов, в зависимости от скорости подключения к Интернету и количества загружаемых пакетов.

Чтобы флаг -u применялся по умолчанию, отредактируйте (или создайте) файл /etc/apt/apt.conf:

// Всегда перечислять обновляемые пакеты

// и запрашивать подтверждение у пользователя

APT::Get::Show-Upgraded "true";

Перед каждым выполнением команды apt\_get dist\_upgrade сначала выполните команду apt upgrade, чтобы снизить вероятность ошибок при выполнении distupgrade

#### Поиск установленных программ

Следующая команда выводит список всех установленных пакетов и направляет его в файл:

$ dpkg -l | tee dpkglist

Поиск всех пакетов, удовлетворяющих критерию поиска, и вывод информации об их состоянии:

$ dpkg -l '\*gnome\*'

Поиск установленных пакетов, удовлетворяющих критерию поиска:

$ dpkg -l | grep gnome

Вывод списка файлов, входящих в пакет:

$ dpkg -L gnome-applets

#### Операции с кэшем пакетов

Время от времени требуется обеспечить актуальность кэша пакетов и списков пакетов, чтобы программа apt работала правильно, без возникновения ложных проблем с зависимостями.

Воспользуйтесь программами apt и dpkg. Не забывайте выполнять команду apt update после модификации файла /etc/apt/sources.list, а также регулярно выполняйте ее, чтобы список пакетов оставался актуальным.

Следующая команда выводит список загруженных, но не установленных пакетов:

$ dpkg --yet-to-unpack

Проверка нарушенных зависимостей:

$ apt check

Удаление кэшированных пакетов, которые стали ненужными:

$ apt-cache autoclean

Удаление всех кэшированных пакетов:

$ apt-cache clean

Вывод списка частично установленных пакетов:

$ dpkg -audit

Если вызов dpkg --audit возвращает какие-либо результаты, как в следующем случае:

$ dpkg --audit

vpw (no information available)

для начала убедитесь в том, что возвращаемый пакет существует:

$ dpkg -l

vpw Package 'vpw' is not installed and no info is available

Если пакет существует, либо завершите установку, либо удалите ее. Если пакет не установлен, поищите соответствующую запись в /var/lib/dpkg/available и /var/lib/dpkg/status и удалите ее

#### Разрешение конфликтов зависимостей

Иногда может произойти такая ситуация, что какая-либо программа не устанавливается из-за проблем с зависимостями, или команда apt dist-upgrade оставляет пакеты, которые нужно удалить. Существует несколько команд, которые могут пригодиться для решения этой задачи; в этом разделе они перечисляются в том порядке, в котором их стоит попробовать.

Допустим, проблемы возникли с пакетом libpam-modules, который не желает обновляться:

# apt install libpam-modules

...

The following packages have unmet dependencies:

libpam-modules: Depends: libdb3 (>=3.2.9-19.1) but 3.2.9-19 is to be installed

E: Broken packages

Если вы работаете в смешанной системе, для начала попробуйте указать версию:

# apt install -t stable libpam-modules

Если это не помогает или система не является смешанной, попробуйте воспользоваться системой разрешения конфликтов Debian:

# apt -f install

Затем выполните команду:

# dpkg --configure -a

И снова повторите команду:

# apt -f install

Если будет получен следующий ответ, значит, попытка завершилась неудачей:

Reading Package Lists... Done

Building Dependency Tree... Done

0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.

Теперь проверьте, что произойдет при удалении существующего пакета libpammodules:

# apt remove --dry-run libpam-modules

Reading Package Lists... Done

Building Dependency Tree... Done

The following packages will be REMOVED: adduser adminmenu apache at base-config courier-imap courier-imap-ssl courier-pop courier-pop-ssl cron cupsys cupsys-driver-gimpprint dict-elements dict-foldoc dict-gcide dict-jargon dict-vera dict-wn dictd gdm2...

....

WARNING: The following essential packages will be removed

This should NOT be done unless you know exactly what you are doing!

login libpam-modules (due to login)

В данном случае исправление конфликтов зависимостей потребует едва ли не полной перестройки системы. Чаще проблемы ограничиваются несколькими пакетами. В этом случае начинайте удалять наименее важные из них, пока конфликты зависимостей не будут разрешены, а затем переустановите все пакеты, которые вам нужны.

Если apt -u dist-upgrage отображает какие-либо задержанные (held) пакеты, от них лучше избавиться. Пакеты удерживаются из-за конфликтов зависимостей, которые не удается разрешить apt. Попробуйте воспользоваться следующей командой для поиска и исправления конфликтов:

# apt -o Debug::pkgProblemResolver=yes dist-upgrade

Если исправить конфликты не удается, команда завершается с сообщением

0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.

Удаляйте held-пакеты один за одним, каждый раз запуская dist-upgrade, пока не останется ни одного задержанного пакета. Затем установите заново все необходимые пакеты. Не забудьте использовать ключ --dry-run, чтобы заранее знать обо всех последствиях: # apt remove --dry-run libsdl-perl В «монолитных» системах подобные конфликты версий встречаются редко. Пользователи смешанных систем сталкиваются с ними чаще. Чтобы предотвратить их, будьте чрезвычайно осмотрительны при выборе устанавливаемых программ — при первой установке Debian потратьте немного времени и выберите каждый необходимый пакет.

### Самостоятельное изучение

1. Установка пакетов из исходных кодов. Команды compile, make, make install
2. Установка и запуск программ Windows в системе Linux. Программа-эмулятор Wine/
3. Контейнерные пакеты snap и flatpack.

### Дополнительные задания:

1. Установите другой менеджер рабочего стола (например, KDE или XFCE). Проверьте его работоспособность, залогинившись в него.
2. Установите какую-либо программу по вашему выбору из исходных кодов.
3. Установите на вашу Linux-машину библиотеку tensorflow-gpu. Проверьте правильность установки, запустив в командной строке интерактивную консоль Python командой python и затем введя команду

import tensorflow

Вы не должны получить сообщение об ошибке импорта.