tldr pages book

Simplified and community-driven man pages

Generated on Sun Oct 13 08:33:34 2024

Website: https://tldr.sh

GitHub: https://github.com/tldr-pages/tldr

Android

am

Менеджер активності Android.

Більше інформації: https://developer.android.com/tools/adb#am.

- Почати специфічну активність:

```
am start -n {{com.android.settings/.Settings}}
```

- Почати активність та передайте дані у неї:

```
am start -a {{android.intent.action.VIEW}} -d {{tel:123}}
```

- Почати активність яка є специфічною дією та категорією:

```
am start -a {{android.intent.action.MAIN}} -c
{{android.intent.category.HOME}}
```

- Перетворити значення в посилання:

```
am to-uri -a {{android.intent.action.VIEW}} -d {{tel:123}}
```

bugreport

Показати звіт багів в Android.

Ця команда може бути виконана тільки за допомогою **`adb shell`**.

Більше інформації: https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/bugreport.

- Показати повний звіт багів Android девайсу:

  ```
  bugreport
  ```

bugreportz

Згенерувати зіпований звіт багів.

Ця команда може бути виконана тільки за допомогою **`adb shell`**.

Більше інформації: https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/bugreportz.

- Згенерувати повний зіпований звіт багів Android девайсу:

  ```
  bugreportz
  ```

- Вивести прогрес виконуваної `bugreportz` операції:

  ```
  bugreportz -p
  ```

- Показати допомогу:

  ```
  bugreportz -h
  ```

- Вивести версію `bugreportz`:

  ```
  bugreportz -v
  ```

cmd

Менеджер сервісів Android.

Більше інформації: https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/cmd/.

- Вивести всі запущені девайси:

  ```
  cmd -l
  ```

- Викликати конкретний сервіс:

  ```
  cmd {{сервіс}}
  ```

- Викликати сервіс з заданими аргументами:

  ```
  cmd {{сервіс}} {{аргумент1 аргумент2 ...}}
  ```

dalvikvm

Віртуальна машина Android Java.

Більше інформації: https://source.android.com/docs/core/runtime.

- Запустити конкретну Java програму:

```
dalvikvm -classpath {{шлях/до/файлу.jar}} {{ім'я_класу}}
```

dumpsys

Отримати інформацію про системні сервіси Android.

Ця команда може бути виконана тільки за допомогою **`adb shell`**.

Більше інформації: https://developer.android.com/tools/dumpsys.

- Отримати дані діагностики для всіх системних сервісів:

  ```
  dumpsys
  ```

- Отримати дані діагностики для конкретного системного сервісу:

  ```
  dumpsys {{сервіс}}
  ```

- Вивести усі сервіси `dumpsys`, про які може бути надана інформація:

  ```
  dumpsys -l
  ```

- Вивести специфічні аргументи для сервісу:

  ```
  dumpsys {{сервіс}} -h
  ```

- Виключити специфічний сервіс з виводу діагностики:

  ```
  dumpsys --skip {{сервіс}}
  ```

- Визначити таймаут в секундах (стандартно 10 секунд):

  ```
  dumpsys -t {{8}}
  ```

getprop

Виводить інформацію про системні властивості Android.

Більше інформації: https://manned.org/getprop.

- Виводить інформацію про системні властивості Android:

```
getprop
```

- Виводить інформацію про специфічну системну властивість:

```
getprop {{властивість}}
```

- Виводить рівень SDK API:

```
getprop {{ro.build.version.sdk}}
```

- Виводить версію Android:

```
getprop {{ro.build.version.release}}
```

- Виводить модель девайсу Android:

```
getprop {{ro.vendor.product.model}}
```

- Виводить розблокований статус OEM:

```
getprop {{ro.oem_unlock_supported}}
```

- Виводить MAC адресу Wi-Fi карти Android:

```
getprop {{ro.boot.wifimacaddr}}
```

input

Відправляє коди подій чи жести на сенсорному екрані на Android девайс.

Ця команда може бути виконана тільки за допомогою **`adb shell`**.

Більше інформації: https://developer.android.com/reference/android/view/KeyEvent.html#constants_1.

- Відправити код події для одного знаку на Android девайс:

  ```
  input keyevent {{код_події}}
  ```

- Відправити текст на Android девайс (`%s` визначає пробіли):

  ```
  input text "{{текст}}"
  ```

- Відправити один дотик на Android девайс:

  ```
  input tap {{позиція_x}} {{позиція_y}}
  ```

- Відправити жест проведення(свайп) на Android девайс:

  ```
  input swipe {{x_старт}} {{y_старт}} {{x_кінець}} {{y_кінець}} {{тривалість_в_мс}}
  ```

- Відправити довге натискання на Android девайс, використовуючи жест проведення(свайп):

  ```
  input swipe {{позиція_x}} {{позиція_y}} {{позиція_x}} {{позиція_y}} {{тривалість_в_мс}}
  ```

logcat

Дамп журналу системних повідомлень, включаючи стек викликів, коли трапилась помилка, і інформаційні повідомлення, залоговані застосунками.

Більше інформації: https://developer.android.com/tools/logcat.

- Вивести системні логи:

```
logcat
```

- Записати системні логи в файл:

```
logcat -f {{шлях/до/файлу}}
```

- Вивести рядки, які підпадають під регулярний вираз:

```
logcat --regex {{регулярний_вираз}}
```

- Вивести логи для специфічного процесу (PID):

```
logcat --pid {{pid}}
```

- Вивести логи для процесу специфічного пакету:

```
logcat --pid $(pidof -s {{пакет}})
```

pkg

Утиліта менеджеру пакетів для Termux.

Більше інформації: https://wiki.termux.com/wiki/Package_Management.

- Оновити всі встановлені пакети:

  ```
  pkg upgrade
  ```

- Встановити пакет:

  ```
  pkg install {{пакет}}
  ```

- Видалити пакет:

  ```
  pkg uninstall {{пакет}}
  ```

- Перевстановити пакет:

  ```
  pkg reinstall {{пакет}}
  ```

- Пошук пакету:

  ```
  pkg search {{пакет}}
  ```

pm

Вивести інформацію про застосунки на Android девайсі.

Більше інформації: https://developer.android.com/tools/adb#pm.

- Вивести всі встановлені застосунки:

  ```
  pm list packages
  ```

- Вивести всі встановлені системні застосунки:

  ```
  pm list packages -s
  ```

- Вивести всі встановлені сторонні (3d-party) застосунки:

  ```
  pm list packages -3
  ```

- Вивести застосунки, які підпадають під специфічні ключові слова:

  ```
  pm list packages {{ключове_слово1 ключове_слово2 ...}}
  ```

- Вивести шлях до APK для специфічного застосунку:

  ```
  pm path {{app}}
  ```

screencap

Зробіть знімок мобільного екрану.

Ця команда може бути виконана тільки за допомогою **`adb shell`**.

Більше інформації: https://developer.android.com/tools/adb#screencap.

- Зробіть знімок мобільного екрану:

  ```
  screencap {{шлях/до/файлу}}
  ```

settings

Отримайте інформацію про операційну систему Android.

Більше інформації: https://adbinstaller.com/commands/adb-shell-settings-5b670d5ee7958178a2955536.

- Вивести список налаштувань в глобальному (`global`) просторі імен:

  ```
  settings list {{global}}
  ```

- Отримайте значення конкретного налаштування:

  ```
  settings get {{global}} {{airplane_mode_on}}
  ```

- Встановіть значення для налаштування:

  ```
  settings put {{system}} {{screen_brightness}} {{42}}
  ```

- Видаліть визначене налаштування:

  ```
  settings delete {{secure}} {{screensaver_enabled}}
  ```

wm

Показати інформацію про екран Android девайсу.

Ця команда може бути виконана тільки за допомогою **`adb shell`**.

Більше інформації: https://adbinstaller.com/commands/adb-shell-wm-5b672b17e7958178a2955538.

- Вивести фізичний розмір екрану Android девайсу:

  ```
  wm size
  ```

- Вивести фізичну щільність екрану Android девайсу:

  ```
  wm density
  ```

Common

7z

Архіватор файлів з високим ступенем стиснення.

Більше інформації: https://manned.org/7z.

- Додати ([a]dd) файл або каталог до нового або існуючого архіву:

```
7z a {{шлях/до/архіву.7z}} {{шлях/до/файлу_або_каталогу}}
```

- Зашифрувати існуючий архів (включаючи імена файлів):

```
7z a {{шлях/до/зашифрованого_архіву.7z}} -p{{пароль}} -mhe=on
{{шлях/до/архіву.7z}}
```

- Розпакувати (e[x]tract) архів, зберігаючи оригінальну структуру каталогів:

```
7z x {{шлях/до/архіву.7z}}
```

- Розпакувати (e[x]tract) архів у певний каталог:

```
7z x {{шлях/до/архіву.7z}} -o{{шлях/до/каталогу}}
```

- Розпакувати (e[x]tract) архів у `stdout`:

```
7z x {{шлях/до/архіву.7z}} -so
```

- Архівувати ([a]rchive) за допомогою певного типу архіву:

```
7z a -t{{7z|bzip2|gzip|lzip|tar|zip}} {{шлях/до/архіву}}
{{шлях/до/файлу_або_каталогу}}
```

- Вивести ([l]ist) перелік вмісту архіву:

```
7z l {{шлях/до/архіву.7z}}
```

- Встановити рівень стиснення (вище означає більше стиснення, але повільніше):

```
7z a {{шлях/до/архіву.7z}} -mx={{0|1|3|5|7|9}} {{шлях/до/
файлу_або_каталогу}}
```

7za

Архіватор файлів з високим ступенем стиснення.

Подібний до **7z**, за винятком того, що він підтримує менше типів файлів, але є кросплатформним.

Більше інформації: https://manned.org/7za.

- Архівувати ([a]rchive) файл або каталог:

```
7za a {{шлях/до/архіву.7z}} {{шлях/до/файлу_або_каталогу}}
```

- Зашифрувати існуючий архів (включаючи імена файлів):

```
7za a {{шлях/до/зашифрованого_архіву.7z}} -p{{пароль}} -
mhe={{on}} {{шлях/до/архіву.7z}}
```

- Розпакувати (e[x]tract) архів, зберігаючи оригінальну структуру каталогів:

```
7za x {{шлях/до/архіву.7z}}
```

- Розпакувати (e[x]tract) архів у певний каталог:

```
7za x {{шлях/до/архіву.7z}} -o{{шлях/до/каталогу}}
```

- Розпакувати (e[x]tract) архів у `stdout`:

```
7za x {{шлях/до/архіву.7z}} -so
```

- Архівувати ([a]rchive) за допомогою певного типу архіву:

```
7za a -t{{7z|bzip2|gzip|lzip|tar|...}} {{шлях/до/архіву.7z}}
{{шлях/до/файлу_або_каталогу}}
```

- Вивести ([l]ist) перелік вмісту архіву:

```
7za l {{шлях/до/архіву.7z}}
```

- Встановити рівень стиснення (вище означає більше стиснення, але повільніше):

```
7za a {{шлях/до/архіву.7z}} -mx={{0|1|3|5|7|9}} {{шлях/до/
файлу_або_каталогу}}
```

awk

Універсальна мова програмування для роботи з файлами.

Більше інформації: https://github.com/onetrueawk/awk.

- Вивести п’ятий стовпець (він же поле) у файлі, розділеному пробілами:

```
awk '{print $5}' {{шлях/до/файлу}}
```

- Вивести другий стовпець рядків, що містять "foo", у файлі, розділеному пробілами:

```
awk '/{{foo}}/ {print $2}' {{шлях/до/файлу}}
```

- Вивести останній стовпець кожного рядка у файлі, використовуючи кому (замість пробілу) як роздільник полів:

```
awk -F ',' '{print $NF}' {{шлях/до/файлу}}
```

- Підсумувати значення в першому стовпці файлу та надрукувати підсумок:

```
awk '{s+=$1} END {print s}' {{шлях/до/файлу}}
```

- Вивести кожен третій рядок, починаючи з першого:

```
awk 'NR%3==1' {{шлях/до/файлу}}
```

- Вивести різні значення залежно від умов:

```
awk '{if ($1 == "foo") print "Точний збіг foo"; else if ($1 ~
"bar") print "Частковий збіг bar"; else print "Baz"}' {{шлях/
до/файлу}}
```

- Вивести всі рядки, значення 10-го стовпця яких знаходиться між min і max:

```
awk '($10 >= {{min_value}} && $10 <= {{max_value}})'
```

- Вивести таблицю користувачів із UID >=1000 із заголовком і форматуванням, використовуючи двокрапку як роздільник («%-20s» означає: 20 символів рядка з вирівнюванням по лівому краю, «%6s» означає: 6 символів рядка з вирівнюванням по правому краю):

```
awk 'BEGIN {FS=":";printf "%-20s %6s %25s\n", "Name", "UID",
"Shell"} $4 >= 1000 {printf "%-20s %6d %25s\n", $1, $4, $7}'
/etc/passwd
```

ClamAV

Ця команда є псевдонімом для **`clamdscan`**.

Більше інформації: https://www.clamav.net.

- Дивись документацію для оригінальної команди:

  ```
  tldr clamdscan
  ```

clang-cpp

Ця команда є псевдонімом для **`clang++`**.

- Дивись документацію для оригінальної команди:

  ```
  tldr clang++
  ```

clojure

Ця команда є псевдонімом для **`clj`**.

- Дивись документацію для оригінальної команди:

  ```
  tldr clj
  ```

cola

Ця команда є псевдонімом для **`git-cola`**.

- Дивись документацію для оригінальної команди:

  ```
  tldr git-cola
  ```

dog

Утиліта пошуку DNS.

Вона має кольоровий вихід, підтримує протоколи DNS-over-TLS і DNS-over-HTTPS та може видавати JSON.

Більше інформації: https://dns.lookup.dog.

- Шукає IP-адреси пов'язані з іменем хоста (A records):

```
dog {{example.com}}
```

- Запитує тип записів MX, пов’язаних із заданим доменним ім’ям:

```
dog {{example.com}} MX
```

- Вкажіть конкретний DNS-сервер для запиту (наприклад, Cloudflare):

```
dog {{example.com}} MX @{{1.1.1.1}}
```

- Запит через TCP, а не UDP:

```
dog {{example.com}} MX @{{1.1.1.1}} --tcp
```

- Запитує тип записів MX, пов’язаних із заданим доменним ім’ям через TCP, використовуючи явні аргументи:

```
dog --query {{example.com}} --type MX --nameserver
{{1.1.1.1}} --tcp
```

- Шукає IP-адреси, пов’язані з іменем хоста (записи A), за допомогою DNS через HTTPS (DoH):

```
dog {{example.com}} --https @{{https://cloudflare-dns.com/
dns-query}}
```

dokku

Міні-Heroku на основі Docker (PaaS).

Легко розгортає кілька програм на власному сервері різними мовами за допомогою однієї команди **`git-push`**.

Більше інформації: https://github.com/dokku/dokku.

- Показати запущені програми:

  ```
  dokku apps
  ```

- Створити програму:

  ```
  dokku apps:create {{ім'я_програми}}
  ```

- Видалити програму:

  ```
  dokku apps:destroy {{ім'я_програми}}
  ```

- Встановити плагін:

  ```
  dokku plugin:install {{повний_url_на_репозиторій}}
  ```

- Зв'язати базу даних із програмою:

  ```
  dokku {{db}}:link {{ім'я_бази_даних}} {{ім'я_програми}}
  ```

find

Пошук файлів або каталогів в дереві каталогів, рекурсивно.

Більше інформації: https://manned.org/find.

- Знайти файли за розширенням:

```
find {{кореневий_шлях}} -name '{{*.ext}}'
```

- Знайти файли, що відповідають декільком шаблонам шляху/імен:

```
find {{кореневий_шлях}} -path '{{**/path/**/*.ext}}' -or -
name '{{*шаблон*}}'
```

- Знайти каталоги, що відповідають заданому імені, у режимі без урахування регістру:

```
find {{кореневий_шлях}} -type d -iname '{{*lib*}}'
```

- Знайти файли, що відповідають заданому шаблону, за винятком певних шляхів:

```
find {{кореневий_шлях}} -name '{{*.py}}' -not -path '{{*/
site-packages/*}}'
```

- Знайти файли, що відповідають заданому діапазону розмірів, обмеживши рекурсивну глибину до "1":

```
find {{кореневий_шлях}} -maxdepth 1 -size {{+500k}} -size
{{-10M}}
```

- Виконати команду для кожного файлу (використовуйте `{}` в команді, щоб отримати доступ до імені файлу):

```
find {{кореневий_шлях}} -name '{{*.ext}}' -exec {{wc -l}} {}
\;
```

- Знайти всі файли, змінені сьогодні, і передати результати одній команді як аргументи:

```
find {{кореневий_шлях}} -daystart -mtime {{-1}} -exec {{tar -
cvf archive.tar}} {} \+
```

- Знайти порожні файли (0 байт) або каталоги та докладно видалити їх:

```
find {{кореневий_шлях}} -type {{f|d}} -empty -delete -print
```

fossil ci

Ця команда є псевдонімом для **`fossil commit`**.

Більше інформації: https://fossil-scm.org/home/help/commit.

- Дивись документацію для оригінальної команди:

  ```
  tldr fossil-commit
  ```

fossil delete

Ця команда є псевдонімом для **`fossil rm`**.

Більше інформації: https://fossil-scm.org/home/help/delete.

- Дивись документацію для оригінальної команди:

  ```
  tldr fossil rm
  ```

fossil forget

Ця команда є псевдонімом для **`fossil rm`**.

Більше інформації: https://fossil-scm.org/home/help/forget.

- Дивись документацію для оригінальної команди:

```
tldr fossil rm
```

fossil new

Ця команда є псевдонімом для **`fossil init`**.

Більше інформації: https://fossil-scm.org/home/help/new.

- Дивись документацію для оригінальної команди:

  ```
  tldr fossil-init
  ```

gh cs

Ця команда є псевдонімом для **`gh codespace`**.

Більше інформації: https://cli.github.com/manual/gh_codespace.

- Дивись документацію для оригінальної команди:

```
tldr gh-codespace
```

gist

Завантажує код у https://gist.github.com.

Більше інформації: https://github.com/defunkt/gist.

- Увійти в gist на цьому комп'ютері:

```
gist --login
```

- Створити gist з будь-якої кількості текстових файлів:

```
gist {{ім'я_файлу.txt}} {{ім'я_файлу2.txt}}
```

- Створити приватний gist з описом:

```
gist --private --description "{{Змістовний опис}}"
{{ім'я_файлу.txt}}
```

- Прочитати контент з `stdin` і створити gist з цього:

```
{{echo "привіт світ"}} | gist
```

- Перелічити свої публічні та приватні gist:

```
gist --list
```

- Перелічити всі публічні gist будь-якого користувача:

```
gist --list {{ім'я_користувача}}
```

- Оновити gist за допомогою ID з URL:

```
gist --update {{GIST_ID}} {{ім'я_файлу.txt}}
```

git abort

Перериває поточне перебазування(rebase), злиття(merge) або вибір(cherry-pick).

Частина **`git-extras`**.

Більше інформації: https://github.com/tj/git-extras/blob/master/Commands.md#git-abort.

- Перериває Git перебазування(rebase), злиття(merge) або вибір(cherry-pick):

  ```
  git abort
  ```

git add

Додає змінені файли до індексу.

Більше інформації: https://git-scm.com/docs/git-add.

- Додає змінені файли до індексу:

```
git add {{шлях/до/файлу}}
```

- Додає усі файли (контрольовані та неконтрольовані):

```
git add -A
```

- Додає тільки ті файли, що вже контрольовані:

```
git add -u
```

- Додає й ті файли, що ігноруються:

```
git add -f
```

- Інтерактивно індексує частини файлів:

```
git add -p
```

- Інтерактивно індексує частини вказаного файлу:

```
git add -p {{шлях/до/файлу}}
```

- Інтерактивно індексує файл:

```
git add -i
```

git annotate

Показує хеш коміту і останнього автора на кожному рядку у файлі.

Дивіться **`git blame`**, якій варто віддати перевагу над **`git annotate`**.

`git annotate` призначена для тих, хто знайомий із іншими системами контролю версій.

Більше інформації: https://git-scm.com/docs/git-annotate.

- Виводить файл з ім'ям автора та хешем коміту доданими попереду кожного рядку:

  ```
  git annotate {{шлях/до/файлу}}
  ```

- Виводить файл з електронною поштою автора та хешем коміту доданими попереду кожного рядку:

  ```
  git annotate {{-e|--show-email}} {{шлях/до/файлу}}
  ```

- Виводить лише рядки, які відповідають регулярному виразу:

  ```
  git annotate -L :{{регулярний_вираз}} {{шлях/до/файлу}}
  ```

git blame

Показує хеш коміту та останнього автора на кожному рядку у файлі.

Більше інформації: https://git-scm.com/docs/git-blame.

- Виводить файл з ім'ям автора та хешем коміту на кожному рядку:

```
git blame {{шлях/до/файлу}}
```

- Виводить електронну пошту автора замість імені:

```
git blame {{-e|--show-email}} {{шлях/до/файлу}}
```

- Виводить файл з ім'ям автора та хешем коміту на кожному рядку у вказаному коміті:

```
git blame {{коміт}} {{шлях/до/файлу}}
```

- Виводить файл з ім'ям автора та хешем коміту на кожному рядку до вказаного коміту:

```
git blame {{коміт}}~ {{шлях/до/файлу}}
```

git checkout-index

Копіює файли з індексу до робочої директорії.

Більше інформації: https://git-scm.com/docs/git-checkout-index.

- Відновлює усі файли, що були видалені з часу останнього коміту:

  ```
  git checkout-index --all
  ```

- Відновлює усі файли, що були видалені чи змінені з часу останнього коміту:

  ```
  git checkout-index --all --force
  ```

- Відновлює усі файли, що були змінені з часу останнього коміту, ігноруючи файли, що були видалені:

  ```
  git checkout-index --all --force --no-create
  ```

- Експортує копію робочої директорії, у стані останнього коміту, до вказаного каталогу (слеш наприкінці обов'язковий):

  ```
  git checkout-index --all --force --prefix={{шлях/до/директорії_експорту/}}
  ```

git checkout

Перемикає на гілку чи шлях до робочої директорії.

Більше інформації: https://git-scm.com/docs/git-checkout.

- Створює та перемикає на нову гілку:

```
git checkout -b {{назва_гілки}}
```

- Створює та перемикає на нову гілку спираючись на певне посилання (приклади посилань: гілка, віддалена/гілка, тег):

```
git checkout -b {{назва_гілки}} {{посилання}}
```

- Перемикає на локальну гілку, що вже існує:

```
git checkout {{назва_гілки}}
```

- Перемикає на попередню гілку:

```
git checkout -
```

- Перемикає на віддалену гілку, що вже існує:

```
git checkout --track {{назва_віддаленого_сховища}}/
{{назва_гілки}}
```

- Відкидає усі неіндексовані зміни у поточній директорії (дізнайтесь більше про команди, як скасування, ознайомившись із `git reset`):

```
git checkout .
```

- Скасовує неіндексовані зміну у файлі:

```
git checkout {{ім'я_файлу}}
```

- Замінює файл у поточній директорії на його версію, яку було закомічено до вказаної гілки:

```
git checkout {{назва_гілки}} -- {{ім'я_файлу}}
```

git cherry-pick

Застосовує зміни, зроблені у наявних комітах, до поточної гілки.

Для застосування змін до іншої гілки спершу виконайте **`git checkout`**, аби переключитися на бажану гілку.

Більше інформації: https://git-scm.com/docs/git-cherry-pick.

- Застосовує коміт до поточної гілки:

```
git cherry-pick {{коміт}}
```

- Застосовує проміжок комітів до поточної гілки (дивіться також `git rebase --onto`):

```
git cherry-pick {{початковий_коміт}}~..{{кінцевий_коміт}}
```

- Застосовує декілька (непослідовних) комітів до поточної гілки:

```
git cherry-pick {{коміт1 коміт2 ...}}
```

- Додає зміни з коміту до робочої директорії без створення коміту:

```
git cherry-pick --no-commit {{коміт}}
```

git cherry

Виявляє коміти, які ще не були застосовані до першоджерела.

Більше інформації: https://git-scm.com/docs/git-cherry.

- Показує коміти (та їхні повідомлення) із відповідними комітами першоджерела:

  ```
  git cherry {{-v|--verbose}}
  ```

- Визначає інші першоджерело та тематичну гілку:

  ```
  git cherry {{origin}} {{topic}}
  ```

- Обмежує коміти до тих, що у наданих межах:

  ```
  git cherry {{origin}} {{topic}} {{base}}
  ```

git clone

Клонує репозиторій, що існує.

Більше інформації: https://git-scm.com/docs/git-clone.

- Клонує репозиторій, що існує, у задану директорію:

```
git clone {{шлях_до_віддаленого_репозиторію}} {{шлях/до/директорії}}
```

- Клонує репозиторій, що існує, та його підмодулі:

```
git clone --recursive {{шлях_до_віддаленого_репозиторію}}
```

- Клонує локальний репозиторій:

```
git clone --local {{шлях/до/локального/репозиторію}}
```

- Клонує тихо:

```
git clone --quiet {{шлях_до_віддаленого_репозиторію}}
```

- Клонує з репозиторію, що існує, тільки 10 останніх комітів з гілки по замовчанню (корисно для заощадження часу):

```
git clone --depth {{10}} {{шлях_до_віддаленого_репозиторію}}
```

- Клонує з репозиторію, що існує, тільки конкретну гілку:

```
git clone --branch {{ім'я}} --single-branch {{шлях_до_віддаленого_репозиторію}}
```

- Клонує репозиторій, що існує, використовуючи задану команду SSH:

```
git clone --config core.sshCommand="{{ssh -i шлях/до/приватного_ключа_ssh}}" {{шлях_до_віддаленого_репозиторію}}
```

git commit-graph

Записує та перевіряє файл графіку комітів Git.

Більше інформації: https://git-scm.com/docs/git-commit-graph.

- Записує файл графіку комітів для спакованих комітів у локальній директорії `.git`:

  ```
  git commit-graph write
  ```

- Записує файл графіку комітів, що містить набір усіх досяжних комітів:

  ```
  git show-ref --hash | git commit-graph write --stdin-commits
  ```

- Записує файл графіку комітів, що містить усі коміти у поточному файлі графіку комітів разом з тими, до яких можна отримати доступ з `HEAD`:

  ```
  git rev-parse {{HEAD}} | git commit-graph write --stdin-commits --append
  ```

git commit-tree

Низькорівнева утиліта для створення об'єктів комітів.

Дивись також: **`git commit`**.

Більше інформації: https://git-scm.com/docs/git-commit-tree.

- Створює об'єкт коміту із певним повідомленням:

```
git commit-tree {{tree}} -m "{{повідомлення}}"
```

- Створює об'єкт коміту читаючи повідомлення з файлу (використовуй `-` для читання зі стандартного введення):

```
git commit-tree {{tree}} -F {{шлях/до/файлу}}
```

- Створює GPG-підписаний об'єкт коміту:

```
git commit-tree {{tree}} -m "{{повідомлення}}" --gpg-sign
```

- Створює об'єкт коміту із певним батьківським об'єктом коміту:

```
git commit-tree {{tree}} -m "{{повідомлення}}" -p
{{sha_батьківського_коміту}}
```

git commit

Комітить файли до репозиторію.

Більше інформації: https://git-scm.com/docs/git-commit.

- Комітить індексовані файли до репозиторію з повідомленням:

  ```
  git commit --message "{{повідомлення}}"
  ```

- Комітить індексовані файли з повідомленням, що прочитано у файлі:

  ```
  git commit --file {{шлях/до/файлу_з_повідомленням}}
  ```

- Автоматично індексує усі змінені файли і комітить їх з повідомленням:

  ```
  git commit --all --message "{{повідомлення}}"
  ```

- Оновлює останній коміт додаючи до нього щойно індексовані зміни, також змінює геш коміту:

  ```
  git commit --amend
  ```

- Комітить тільки певні (вже індексовані) файли:

  ```
  git commit {{шлях/до/файлу1 шлях/до/файлу2 ...}}
  ```

- Створює коміт, навіть якщо немає жодного індексованого файлу:

  ```
  git commit --message "{{повідомлення}}" --allow-empty
  ```

git commits-since

Виводить коміти починаючи з певного періоду часу або дати.

Частина **`git-extras`**.

Більше інформації: https://github.com/tj/git-extras/blob/master/Commands.md#git-commits-since.

- Виводить коміти починаючи зі вчора:

```
git commits-since {{yesterday}}
```

- Виводить коміти починаючи з минулого тижня:

```
git commits-since {{last week}}
```

- Виводить коміти починаючи з минулого місяця:

```
git commits-since {{last month}}
```

- Виводить коміти починаючи зі вчора з 14:00:

```
git commits-since {{yesterday 2pm}}
```

git config

Керує спеціальними параметрами конфігурації для репозиторію Git.

Конфігурації можуть бути локальні (для поточного репозиторію) або глобальні (для поточного користувача).

Більше інформації: https://git-scm.com/docs/git-config.

- Надає перелік лише локальних налаштувань (що зберігаються у `.git/config` поточного репозиторію):

  ```
  git config --list --local
  ```

- Надає перелік лише глобальних налаштувань (що зберігаються у `~/.gitconfig`):

  ```
  git config --list --global
  ```

- Отримує значення для наданого параметру конфігурації:

  ```
  git config alias.unstage
  ```

- Встановлює глобальне значення для наданого параметру конфігурації:

  ```
  git config --global alias.unstage "reset HEAD --"
  ```

- Повертає значення по замовчанню для наданого глобального параметру конфігурації:

  ```
  git config --global --unset alias.unstage
  ```

- Відкриває для редагування файл конфігурацій поточного репозиторію у редакторі по замовчуванню:

  ```
  git config --edit
  ```

- Відкриває для редагування файл з глобальними конфігураціями у редакторі по замовчанню:

  ```
  git config --global --edit
  ```

git fetch

Завантажує об'єкти та посилання з віддаленого сховища.

Більше інформації: https://git-scm.com/docs/git-fetch.

- Отримує останні зміни з віддаленого сховища за замовчуванням (якщо встановлено):

  ```
  git fetch
  ```

- Отримує нові гілки з конкретного віддаленого сховища:

  ```
  git fetch {{назва_сховища}}
  ```

- Отримує останні зміни з усіх віддалених сховищ:

  ```
  git fetch --all
  ```

- Отримує, зокрема, й мітки з віддаленого сховища:

  ```
  git fetch --tags
  ```

- Видаляє локальні посилання на віддалені гілки, які були видалені з віддаленого сховища:

  ```
  git fetch --prune
  ```

git merge

Злиття гілок разом.

Більше інформації: https://git-scm.com/docs/git-merge.

- Злиття гілки з поточною гілкою:

```
git merge {{назва_гілки}}
```

- Редагує повідомлення при злитті гілок:

```
git merge {{-e|--show-email}} {{назва_гілки}}
```

- Зливає гілки і створює комміт злиття:

```
git merge --no-ff {{назва_гілки}}
```

- Перериває злиття у випадку конфлікту:

```
git merge --abort
```

git pull

Отримує дані з віддаленого репозиторію та зливає їх із локальним.

Більше інформації: https://git-scm.com/docs/git-pull.

- Завантажити зміни із типового віддаленого репозиторію та злити їх:

  ```
  git pull
  ```

- Завантажити зміни із типового віддаленого репозиторію та злити їх, використовуючи перемотання:

  ```
  git pull --rebase
  ```

- Завантажити зміни із певної гілки вказаного віддаленого репозиторію, а потім злити їх у HEAD:

  ```
  git pull {{назва_сховища}} {{назва_гілки}}
  ```

git push

Надсилає коміти до віддаленого репозиторію.

Більше інформації: https://git-scm.com/docs/git-push.

- Надіслати локальні зміни у поточній гілці до її типового віддаленого відповідника:

```
git push
```

- Надіслати зміни із вказаної локальної гілки до її віддаленого відповідника:

```
git push {{назва_сховища}} {{локальна_гілка}}
```

- Надіслати зміни із вказаної локальної гілки до її віддаленого відповідника та встановити цю віддалену гілку як типову для дій надсилання і стягування:

```
git push -u {{назва_сховища}} {{локальна_гілка}}
```

- Надіслати зміни із вказаної локальної гілки до вказаної віддаленої:

```
git push {{назва_сховища}} {{локальна_гілка}}:
{{віддалена_гілка}}
```

- Надіслати зміни з усіх локальних гілок до їх відповідників у вказаному віддаленому репозиторії:

```
git push --all {{назва_сховища}}
```

- Видалити гілку у віддаленому репозиторії:

```
git push {{назва_сховища}} --delete {{віддалена_гілка}}
```

- Видалити віддалену гілку, що не містить локального відповідника:

```
git push --prune {{назва_сховища}}
```

- Надіслати мітки, що відсутні у віддаленому репозиторії:

```
git push --tags
```

git rebase

Повторно застосовує коміти з однієї гілки поверх іншої.

Зазвичай використовується для дублювання комітів з однієї гілки до іншої, шляхом створення нових комітів у гілці призначення.

Більше інформації: https://git-scm.com/docs/git-rebase.

- Перебазовує активну гілку поверх іншої, вказаної гілки:

```
git rebase {{нова_базова_гілка}}
```

- Розпочинає інтерактивне перебазування, яке дозволяє змінювати порядок, оминати, об'єднувати чи редагувати коміти:

```
git rebase {{-i|--interactive}}
{{цільова_базова_гілка_або_хеш_коміту}}
```

- Продовжує перебазування перерване через збій злиття після виправлення конфліктних файлів:

```
git rebase --continue
```

- Продовжує перебазування призупинене через конфлікти при злитті, пропустивши конфліктний коміт:

```
git rebase --skip
```

- Перериває поточне перебазування (наприклад, якщо воно було перерване через конфлікт при злитті):

```
git rebase --abort
```

- Переносить частину поточної гілки поверх нової бази, використавши стару базу, як початок:

```
git rebase --onto {{нова_база}} {{стара_база}}
```

- Повторно застосовує останні 5 комітів, зупиняючись аби змінювати порядок, оминати, об'єднувати чи редагувати їх:

```
git rebase {{-i|--interactive}} {{HEAD~5}}
```

- Автоматично вирішує будь-які конфлікти надавши перевагу робочій версії гілки (ключ `theirs` має обернене значення в цьому випадку):

```
git rebase {{-X|--strategy-option}} theirs {{назва_гілки}}
```

git status

Показує зміни до файлів у Git-репозиторії.

Списки змінених, доданих та видалених файлів в порівнянні до поточного зареєстрованого коміту.

Більше інформації: https://git-scm.com/docs/git-status.

- Показує змінені файли які ще не додані до коміту:

  ```
  git status
  ```

- Виводить інформацію у стислому ([s]hort) форматі:

  ```
  git status -s
  ```

- Виводить інформацію без неконтрольованих файлів:

  ```
  git status --untracked-files=no
  ```

- Виводить інформацію у стислому ([s]hort) форматі разом з інформацією про гілку ([b]ranch):

  ```
  git status -sb
  ```

git

Розподілена система контролю версій.

Деякі команди, як от **`git commit`**, мають свою власну документацію.

Більше інформації: https://git-scm.com/.

- Виконує підкоманду Git:

  ```
  git {{підкоманда}}
  ```

- Виконує підкоманду Git у довільному репозиторії, вказавши шлях до нього:

  ```
  git -C {{шлях/до/репозиторію}} {{підкоманда}}
  ```

- Виконує команду Git із вказаними параметрами:

  ```
  git -c '{{config.key}}={{значення}}' {{підкоманда}}
  ```

- Показує базову допомогу:

  ```
  git --help
  ```

- Показує допомогу з певної підкоманди Git (наприклад, `commit`, `log` чи іншої):

  ```
  git help {{підкоманда}}
  ```

- Перевіряє версію Git:

  ```
  git --version
  ```

gnmic sub

Ця команда є псевдонімом для **`gnmic subscribe`**.

Більше інформації: https://gnmic.kmrd.dev/cmd/subscribe.

- Дивись документацію для оригінальної команди:

```
tldr gnmic subscribe
```

grep

Пошук шаблонів у файлах за допомогою регулярних виразів.

Більше інформації: https://www.gnu.org/software/grep/manual/grep.html.

- Знайти шаблон у файлі:

```
grep "{{шаблон_пошуку}}" {{шлях/до/файлу}}
```

- Знайти точний рядок (відключає регулярні вирази):

```
grep {{-F|--fixed-strings}} "{{точний_рядок}}" {{шлях/до/
файлу}}
```

- Знайти шаблон у всіх файлах рекурсивно в каталозі, виводячи номери рядків збігів, ігноруючи бінарні файли:

```
grep {{-r|--recursive}} {{-n|--line-number}} --binary-files
{{without-match}} "{{шаблон_пошуку}}" {{шлях/до/каталогу}}
```

- Використовувати розширені регулярні вирази (підтримує `?`, `+`, `{}`, `()` та `|`), у режимі без урахування регістру:

```
grep {{-E|--extended-regexp}} {{-i|--ignore-case}}
"{{шаблон_пошуку}}" {{шлях/до/файлу}}
```

- Вивести 3 рядки контексту навколо, до, або після кожного збігу:

```
grep --{{context|before-context|after-context}} 3
"{{шаблон_пошуку}}" {{шлях/до/файлу}}
```

- Вивести назву файлу та номер рядка для кожного збігу з кольоровим виводом:

```
grep {{-H|--with-filename}} {{-n|--line-number}} --
color=always "{{шаблон_пошуку}}" {{шлях/до/файлу}}
```

- Шукати рядкі, що відповідають шаблону, виводячи лише відповідний текст:

```
grep {{-o|--only-matching}} "{{шаблон_пошуку}}" {{шлях/до/
файлу}}
```

- Знайти в `stdin` рядки, які не відповідають шаблону:

```
cat {{шлях/до/файлу}} | grep {{-v|--invert-match}}
"{{шаблон_пошуку}}"
```

help

Відображення інформації про вбудовані команди Bash.

Більше інформації: https://www.gnu.org/software/bash/manual/bash.html#index-help.

- Показати повний список вбудованих команд:

```
help
```

- Надрукувати інструкції щодо використання конструкції циклу `while`:

```
help while
```

- Надрукувати інструкції щодо використання конструкції циклу `for`:

```
help for
```

- Надрукуйте інструкції щодо використання `[[ ]]` для умовних команд:

```
help [[ ]]
```

- Надрукувати інструкцію щодо використання `(( ))` для обчислення математичних виразів:

```
help \( \)
```

- Надрукувати інструкції щодо використання команди `cd`:

```
help cd
```

llvm-ar

Ця команда є псевдонімом для **`ar`**.

- Дивись документацію для оригінальної команди:

  ```
  tldr ar
  ```

llvm-g++

Ця команда є псевдонімом для **`clang++`**.

- Дивись документацію для оригінальної команди:

  ```
  tldr clang++
  ```

llvm-gcc

Ця команда є псевдонімом для **`clang`**.

- Дивись документацію для оригінальної команди:

  ```
  tldr clang
  ```

llvm-nm

Ця команда є псевдонімом для **`nm`**.

- Дивись документацію для оригінальної команди:

  ```
  tldr nm
  ```

llvm-objdump

Ця команда є псевдонімом для **`objdump`**.

- Дивись документацію для оригінальної команди:

  ```
  tldr objdump
  ```

llvm-strings

Ця команда є псевдонімом для **`strings`**.

- Дивись документацію для оригінальної команди:

  ```
  tldr strings
  ```

mscore

Ця команда є псевдонімом для **`musescore`**.

Більше інформації: https://musescore.org/handbook/command-line-options.

- Дивись документацію для оригінальної команди:

```
tldr musescore
```

openssl

Набір криптографічних інструментів OpenSSL.

Деякі підкоманди, такі як **`openssl req`** мають власну документацію щодо використання.

Більше інформації: https://www.openssl.org.

- Вивести список доступних підкоманд:

  ```
  openssl help
  ```

- Вивести параметри для певної команди:

  ```
  openssl help {{x509}}
  ```

- Вивести версію OpenSSL:

  ```
  openssl version
  ```

pio init

Ця команда є псевдонімом для **`pio project`**.

- Дивись документацію для оригінальної команди:

  ```
  tldr pio project
  ```

piodebuggdb

Ця команда є псевдонімом для **`pio debug`**.

- Дивись документацію для оригінальної команди:

  ```
  tldr pio debug
  ```

platformio

Ця команда є псевдонімом для **`pio`**.

Більше інформації: https://docs.platformio.org/en/latest/core/userguide/.

- Дивись документацію для оригінальної команди:

```
tldr pio
```

r2

Ця команда є псевдонімом для **`radare2`**.

- Дивись документацію для оригінальної команди:

  ```
  tldr radare2
  ```

tar

Утиліта архівування.

Часто поєднується з методом стиснення, таким як **`gzip`** або **`bzip2`**.

Більше інформації: https://www.gnu.org/software/tar.

- Створити ([c]reate) архів і записати його у файл ([f]ile):

```
tar cf {{шлях/до/цілі.tar}} {{шлях/до/file1 шлях/до/file2 ...}}
```

- Створити ([c]reate) g[z]ipped архів і записати його у файл ([f]ile):

```
tar czf {{шлях/до/цілі.tar.gz}} {{шлях/до/file1 шлях/до/file2 ...}}
```

- Створити ([c]reate) g[z]ipped архів з каталогу, використовуючи відносні шляхи:

```
tar czf {{шлях/до/цілі.tar.gz}} --directory={{шлях/до/каталогу}} .
```

- Розпакувати (e[x]tract) стиснутий файл ([f]ile) архіву у поточний каталог детально ([v]erbosely):

```
tar xvf {{шлях/до/джерела.tar[.gz|.bz2|.xz]}}
```

- Розпакувати (e[x]tract) стиснутий файл ([f]ile) архіву у певний каталог:

```
tar xf {{шлях/до/джерела.tar[.gz|.bz2|.xz]}} --directory={{шлях/до/каталогу}}
```

- Створити ([c]reate) стиснутий архів і записати його у файл ([f]ile), використовуючи розширення файлу для автоматичного визначення програми стиснення:

```
tar caf {{шлях/до/цілі.tar.xz}} {{шлях/до/file1 шлях/до/file2 ...}}
```

- Вивести ([l]ist) перелік вмісту tar файлу ([f]ile) детально ([v]erbosely):

```
tar tvf {{шлях/до/джерела.tar}}
```

- Розпакувати (e[x]tract) файли, що відповідають шаблону, з файлу ([f]ile) архіву:

```
tar xf {{шлях/до/джерела.tar}} --wildcards "{{*.html}}"
```

tldr

Відображає прості сторінки допомоги для інструментів командного рядка з проекту tldr-pages.

Більше інформації: https://github.com/tldr-pages/tldr/blob/main/CLIENT-SPECIFICATION.md#command-line-interface.

- Показує типове використання команди (підказка: це те, як ви потрапили сюди!):

```
tldr {{команда}}
```

- Показує tldr сторінку для команди `cd` на вказаній платформі:

```
tldr -p {{android|linux|osx|sunos|windows}} {{cd}}
```

- Показує tldr сторінку для підкоманди Git `git checkout`:

```
tldr {{git-checkout}}
```

- Оновлює локальні tldr сторінки (якщо клієнт підтримує кешування):

```
tldr -u
```

tldrl

Ця команда є псевдонімом для **`tldr-lint`**.

Більше інформації: https://github.com/tldr-pages/tldr-lint.

- Дивись документацію для оригінальної команди:

```
tldr tldr-lint
```

tlmgr arch

Ця команда є псевдонімом для **`tlmgr platform`**.

Більше інформації: https://www.tug.org/texlive/tlmgr.html.

- Дивись документацію для оригінальної команди:

```
tldr tlmgr platform
```

vi

Ця команда є псевдонімом для **`vim`**.

- Дивись документацію для оригінальної команди:

  ```
  tldr vim
  ```

vim

Vim (Vi IMproved), консольний текстовий редактор, надає різні режими для різних маніпуляцій над текстом.

Натиснувши **`i`** потрапляємо в режим вставки (insert mode). **`<Esc>`** повертає у нормальний режим (normal mode), який дозволяє користуватися командами Vim.

Більше інформації: https://www.vim.org.

- Відкрити файл:

  ```
  vim {{шлях/до/файлу}}
  ```

- Відкрити файл на визначеноу рядку:

  ```
  vim +{{номер_рядку}} {{шлях/до/файлу}}
  ```

- Подивитися допомогу Vim:

  ```
  :help<Enter>
  ```

- Зберегти і вийти:

  ```
  :wq<Enter>
  ```

- Анулювати (undo) останню операцію:

  ```
  u
  ```

- Знайти паттерн у файлі (натисніть `n`/`N` щоб перейти до наступного/попереднього збігу):

  ```
  /{{паттерн_для_пошуку}}<Enter>
  ```

- Виконати регексп заміну в цілому файлі:

  ```
  :%s/{{регексп_вираз}}/{{заміна}}/g<Enter>
  ```

- Показати номери рядків:

  ```
  :set nu<Enter>
  ```

Linux

adduser

Утиліта додавання користувачів.

Більше інформації: https://manned.org/adduser.

- Створити нового користувача з домашнім каталогом за замовчуванням і попросити користувача встановити пароль:

```
adduser {{юзернейм}}
```

- Створити нового користувача без домашнього каталогу:

```
adduser --no-create-home {{юзернейм}}
```

- Створити нового користувача з домашнім каталогом за вказаним шляхом:

```
adduser --home {{шлях/до/дому}} {{юзернейм}}
```

- Створити нового користувача з указаною оболонкою, встановленою як оболонка входу:

```
adduser --shell {{шлях/до/оболонки}} {{юзернейм}}
```

- Створити нового користувача, що належить до вказаної групи:

```
adduser --ingroup {{група}} {{юзернейм}}
```

apt-add-repository

Керує взаємодією з репозиторіями **`apt`**.

Більше інформації: https://manned.org/apt-add-repository.1.

- Додайте новий репозиторій `apt`:

```
apt-add-repository {{репозиторій}}
```

- Видалити репозиторій `apt`:

```
apt-add-repository --remove {{репозиторій}}
```

- Оновити кеш пакетів після додавання репозиторію:

```
apt-add-repository --update {{репозиторій}}
```

- Увімкнути вихідні пакети:

```
apt-add-repository --enable-source {{репозиторій}}
```

apt-cache

Інструмент запиту пакетів Debian і Ubuntu.

Більше інформації: https://manned.org/apt-cache.8.

- Шукати пакет у ваших поточних джерелах:

  ```
  apt-cache search {{запит}}
  ```

- Показати інформацію про пакет:

  ```
  apt-cache show {{пакет}}
  ```

- Показати, чи встановлено та оновлено пакет:

  ```
  apt-cache policy {{пакет}}
  ```

- Показати залежності для пакета:

  ```
  apt-cache depends {{пакет}}
  ```

- Показати пакети, які залежать від конкретного пакета:

  ```
  apt-cache rdepends {{пакет}}
  ```

apt-file

Пошук файлів в пакетах **`apt`**, включно з тими, що ще не встановлені.

Більше інформації: https://manned.org/apt-file.1.

- Оновити базу метаданих:

```
sudo apt update
```

- Пошук пакетів, які містять вказаний файл або шлях:

```
apt-file {{search|find}} {{частковий_шлях/до/файлу}}
```

- Список вмісту конкретного пакета:

```
apt-file {{show|list}} {{пакет}}
```

- Пошук пакетів, які відповідають `регулярному_виразу`:

```
apt-file {{search|find}} --regexp {{регулярний_вираз}}
```

apt-get

Утиліта керування пакетами Debian і Ubuntu.

Шукати пакети за допомогою **`apt-cache`**.

Більше інформації: https://manned.org/apt-get.8.

- Оновити список доступних пакетів і версій (рекомендується запускати це перед іншими командами `apt-get`):

  ```
  apt-get update
  ```

- Встановити пакет або оновити його до останньої доступної версії:

  ```
  apt-get install {{пакет}}
  ```

- Видалити пакет:

  ```
  apt-get remove {{пакет}}
  ```

- Видалити пакет і файли його конфігурації:

  ```
  apt-get purge {{пакет}}
  ```

- Оновити усі встановлені пакети до найновіших доступних версій:

  ```
  apt-get upgrade
  ```

- Очистити локальний репозиторій - видалити файли пакетів (`.deb`) із перерваних завантажень, які більше не можна завантажити:

  ```
  apt-get autoclean
  ```

- Видалити усі пакети, які більше не потрібні:

  ```
  apt-get autoremove
  ```

- Оновити встановлені пакети (як `upgrade`), але видалити застарілі пакети та встановити додаткові, щоб відповідати новим залежностям:

  ```
  apt-get dist-upgrade
  ```

apt-key

Утиліта керування ключами для диспетчера пакетів APT в Debian та Ubuntu.

Примітка: **`apt-key`** застарілий (за винятком використання **`apt-key del`** у сценаріях підтримки).

Більше інформації: https://manned.org/apt-key.8.

- Список довірених ключів:

```
apt-key list
```

- Додати ключ до довіреного сховища ключів:

```
apt-key add {{public_key_file.asc}}
```

- Видалити ключ з довіреного сховища ключів:

```
apt-key del {{key_id}}
```

- Додайте віддалений ключ до надійного сховища ключів:

```
wget -qO - {{https://host.tld/filename.key}} | apt-key add -
```

- Додати ключ із сервера ключів лише з ідентифікатором ключа:

```
apt-key adv --keyserver {{pgp.mit.edu}} --recv {{KEYID}}
```

apt moo

Пасхалка від менеджеру пакетів **APT**.

Більше інформації: https://manned.org/apt.8.

- Друкує пасхалку з коровою:

```
apt moo
```

apt

Утиліта керування пакетами для дистрибутивів на основі Debian.

Рекомендована заміна для **`apt-get`** при інтерактивному використанні в Ubuntu версії 16.04 і пізніших.

Еквівалентні команди в інших менеджерах пакунків дивитися https://wiki.archlinux.org/title/Pacman/Rosetta.

Більше інформації: https://manned.org/apt.8.

- Оновити список доступних пакетів і версій (рекомендується запускати це перед іншими командами `apt`):

  ```
  sudo apt update
  ```

- Шукати заданий пакет:

  ```
  apt search {{пакет}}
  ```

- Відобразити інформацію про пакет:

  ```
  apt show {{пакет}}
  ```

- Встановити пакет або оновити його до останньої доступної версії:

  ```
  sudo apt install {{пакет}}
  ```

- Видалити пакет (використання `purge` натомість також видаляє його конфігураційні файли):

  ```
  sudo apt remove {{пакет}}
  ```

- Оновити усі встановлені пакети до найновіших доступних версій:

  ```
  sudo apt upgrade
  ```

- Відобразити список усіх пакетів:

  ```
  apt list
  ```

- Відобразити список усіх встановлених пакетів:

  ```
  apt list --installed
  ```

aptitude

Утиліта керування пакетами Debian і Ubuntu.

Більше інформації: https://manned.org/aptitude.8.

- Синхронізувати список доступних пакетів і версій. Це слід запустити спочатку, перш ніж запускати наступні команди aptitude:

```
aptitude update
```

- Встановити новий пакет і його залежності:

```
aptitude install {{пакет}}
```

- Шукати пакет:

```
aptitude search {{пакет}}
```

- Шукати встановлений пакет (`?installed` це термін пошуку aptitude):

```
aptitude search '?installed({{пакет}})'
```

- Видалити пакет і всі залежні від нього пакети:

```
aptitude remove {{пакет}}
```

- Оновити встановлені пакети до найновіших доступних версій:

```
aptitude upgrade
```

- Оновити встановлені пакети (як `aptitude upgrade`) включно з видаленням застарілих пакетів і встановленням додаткових, щоб відповідати новим залежностям пакетів:

```
aptitude full-upgrade
```

- Затримати встановлений пакет, щоб уникнути його автоматичного оновлення:

```
aptitude hold '?installed({{пакет}})'
```

cat

Зчитування та об'єднання файлів.

Більше інформації: https://www.gnu.org/software/coreutils/cat.

- Вивести вміст файлу в `stdout`:

```
cat {{шлях/до/файлу}}
```

- Об’єднати кілька файлів у вихідний файл:

```
cat {{шлях/до/файлу1 шлях/до/файлу2 ...}} > {{шлях/до/вихідного_файлу}}
```

- Додайте кілька файлів до вихідного файлу:

```
cat {{шлях/до/файлу1 шлях/до/файлу2 ...}} >> {{шлях/до/вихідного_файлу}}
```

- Записати `stdin` у файл:

```
cat - > {{шлях/до/файлу}}
```

- Пронумерувати всі вихідні рядки:

```
cat -n {{шлях/до/файлу}}
```

- Відобразити недруковані символи та пробіли (з префіксом `M-`, якщо не ASCII):

```
cat -v -t -e {{шлях/до/файлу}}
```

cp

Скопіювати файли і папки.

Більше інформації: https://www.gnu.org/software/coreutils/cp.

- Скопіювати файл в інше місце:

```
cp {{шлях/до/файлу_який_скопіювати.ext}} {{шлях/до/
файлу_в_який_скопіювати.ext}}
```

- Скопіювати файл в іншу папку, зберугіючи назву файлу:

```
cp {{шлях/до/файлу_який_скопіювати.ext}} {{шлях/до/
папки_в_яку_скопіювати}}
```

- Рекурсивно скопіювати вміст папки до іншого місця (якщо місце призначення існує, папка скопіюється всередину нього):

```
cp -r {{шлях/до/файлу_який_скопіювати}} {{шлях/до/
папки_в_яку_скопіювати}}
```

- Скопіювати папку рекурсивно у докладнішому режимі (показує файли у міру їх копіювання):

```
cp -vr {{шлях/до/папки_яку_спопіювати}} {{шлях/до/
папки_в_яку_скопіювати}}
```

- Скопіювати текстові файли в інше місце в інтерактивному режимі (запитує користувача перед перезаписом):

```
cp -i {{*.txt}} {{шлях/до/папки_в_яку_скопіювати}}
```

- Зберігає символічні посилання(symbolic link) перед копіюванням:

```
cp -L {{посилання}} {{шлях/до/папки_в_яку_скопіювати}}
```

- Використовує повний шлях файлу який потрібно скопіювати, створюючи будь-які відсутні проміжні папки під час копіювання:

```
cp --parents {{повний/шлях/до/файлу}} {{шлях/до/
бажаного_файлу}}
```

dmesg

Відобразити повідомлення ядра в **`stdout`**.

Більше інформації: https://manned.org/dmesg.

- Відобразити повідомлення ядра:

```
sudo dmesg
```

- Відобразити повідомлення про помилки ядра:

```
sudo dmesg --level err
```

- Відобразити повідомлення ядра та продовжити читати нові, подібно до `tail -f` (доступно в ядрах 3.5.0 і новіших):

```
sudo dmesg -w
```

- Відобразити, скільки фізичної пам'яті доступно в цій системі:

```
sudo dmesg | grep -i memory
```

- Відобразити повідомлення ядра по 1 сторінці за раз:

```
sudo dmesg | less
```

- Відобразити повідомлення ядра з міткою часу (доступно в ядрах 3.5.0 і новіших):

```
sudo dmesg -T
```

- Відобразити повідомлення ядра у формі, зрозумілій людині (доступно в ядрах 3.5.0 і новіших):

```
sudo dmesg -H
```

- Розфарбувати виведені дані (доступно в ядрах 3.5.0 і новіших):

```
sudo dmesg -L
```

dpkg-reconfigure

Змінює конфігурацію вже встановленого пакету.

Більше інформації: https://manned.org/dpkg-reconfigure.8.

- Змінити конфігурацію одного або декількох пакетів:

```
dpkg-reconfigure {{пакунок1 пакунок2 ...}}
```

ip route list

Ця команда є псевдонімом для **`ip route show`**.

- Дивись документацію для оригінальної команди:

  ```
  tldr ip-route-show
  ```

journalctl

Запити до журналу systemd.

Більше інформації: https://manned.org/journalctl.

- Показати всі повідомлення з рівнем пріоритету 3 (помилки) від цього завантаження ([b]oot):

```
journalctl -b --priority={{3}}
```

- Видалити записи журналу, які старіші за 2 дні:

```
journalctl --vacuum-time={{2d}}
```

- Слідкувати за новими повідомленнями (як `tail -f` для традиційного syslog):

```
journalctl -f
```

- Показати всі повідомлення за конкретним блоком ([u]nit):

```
journalctl -u {{блок}}
```

- Фільтрувати повідомлення в межах діапазону часу (мітка часу або покажчики місця заповнення, як-от «вчора»):

```
journalctl --since {{now|today|yesterday|tomorrow}} --until
"{{YYYY-MM-DD HH:MM:SS}}"
```

- Показати всі повідомлення за певним процесом:

```
journalctl _PID={{pid}}
```

- Показати всі повідомлення за певним виконуваним файлом:

```
journalctl {{шлях/до/виконуваного_файлу}}
```

locale

Отримайте інформацію, що стосується локалізації.

Більше інформації: https://manned.org/locale.

- Список усіх глобальних змінних середовища, що описують локалізацію користувача:

  ```
  locale
  ```

- Список всії наявних локалізацій:

  ```
  locale --all-locales
  ```

- Показати всі доступні локалізації та пов'язані метадані:

  ```
  locale --all-locales --verbose
  ```

- Відображення поточного формату дати:

  ```
  locale date_fmt
  ```

lsblk

Показує інформацію про пристрої.

Більше інформації: https://manned.org/lsblk.

- Показати усі пристрої зберігання даних у деревоподібному форматі:

```
lsblk
```

- Також показати порожні пристрої:

```
lsblk -a
```

- Показати стовпець SIZE у байтах, а не у форматі, зрозумілому людині:

```
lsblk -b
```

- Вивести інформацію про файлові системи:

```
lsblk -f
```

- Використати символи ASCII для форматування дерева:

```
lsblk -i
```

- Вивести інформацію про топологію блочного пристрою:

```
lsblk -t
```

- Виключити пристрої, указані в розділеному комами списку основних номерів пристроїв:

```
lsblk -e {{1,7}}
```

- Вивести налаштований підсумок за допомогою розділеного комами списку стовпців:

```
lsblk --output {{NAME}},{{SERIAL}},{{MODEL}},{{TRAN}},
{{TYPE}},{{SIZE}},{{FSTYPE}},{{MOUNTPOINT}}
```

man

Форматування та відображення сторінок посібника.

Більше інформації: https://manned.org/man.

- Відобразити довідкову сторінку для команди:

```
man {{команда}}
```

- Відобразити сторінку довідки для команди з розділу 7:

```
man {{7}} {{команда}}
```

- Відобразити усі доступні розділи для команди:

```
man --whatis {{команда}}
```

- Відобразити шлях пошуку довідкових сторінок:

```
man --path
```

- Відобразити розташування довідкової сторінки, а не довідкову сторінку:

```
man --where {{команда}}
```

- Відобразити довідкову сторінку з використанням певної локалі:

```
man --locale={{локаль}} {{команда}}
```

- Знайти довідкові сторінки, які містять рядок пошуку:

```
man --apropos "{{рядок_пошуку}}"
```

nala

Утиліта керування пакетами з кращим форматуванням.

Фронтенд для API **`python-apt`**.

Більше інформації: https://gitlab.com/volian/nala.

- Встановити пакет або оновити його до останньої версії:

  ```
  sudo nala install {{пакет}}
  ```

- Видалити пакет:

  ```
  sudo nala remove {{пакет}}
  ```

- Видалити пакет та його конфігураційні файли:

  ```
  nala purge {{пакет}}
  ```

- Пошук назв пакетів і описів за допомогою слова, регулярного виразу (за замовчуванням) або glob:

  ```
  nala search "{{паттерн}}"
  ```

- Оновити список доступних пакетів та оновити систему:

  ```
  sudo nala upgrade
  ```

- Видалити усі невикористовувані пакети та залежності з вашої системи:

  ```
  sudo nala autoremove
  ```

- Отримати швидші дзеркала, щоб покращити швидкість завантаження:

  ```
  sudo nala fetch
  ```

- Показати історію всіх транзакцій:

  ```
  nala history
  ```

ncal

Ця команда є псевдонімом для **`cal`**.

Більше інформації: https://manned.org/ncal.

- Дивись документацію для оригінальної команди:

  ```
  tldr cal
  ```

pacman

Утиліта для керування пакетами Arch Linux.

Дивіться також: **`pacman-database`**, **`pacman-deptest`**, **`pacman-files`**, **`pacman-key`**, **`pacman-mirrors`**, **`pacman-query`**, **`pacman-remove`**, **`pacman-sync`**, **`pacman-upgrade`**.

Для еквівалентних команд в інших менеджерах пакетів дивіться https://wiki.archlinux.org/title/Pacman/Rosetta.

Більше інформації: https://manned.org/pacman.8.

- Синхронізувати та оновити всі пакети:

```
sudo pacman -Syu
```

- Встановити новий пакет:

```
sudo pacman -S {{пакет}}
```

- Видалити пакет і його залежності:

```
sudo pacman -Rs {{пакет}}
```

- Шукати в базі даних пакети, що містять певний файл:

```
pacman -F "{{ім'я_файлу}}"
```

- Перелічити встановлені пакети та версії:

```
pacman -Q
```

- Перелічити лише явно встановлені пакети та версії:

```
pacman -Qe
```

- Перелічити безхазяйні пакети (встановлені як залежні, але фактично не потрібних для жодного пакета):

```
pacman -Qtdq
```

- Очистити весь кеш Pacman:

```
sudo pacman -Scc
```

pamac

Утиліта командного рядка для GUI менеджера пакетів pamac.

Якщо ви не можете побачити пакети AUR, увімкніть його **`/etc/pamac.conf`** або в GUI.

Більше інформації: https://wiki.manjaro.org/index.php/Pamac.

- Встановити новий пакет:

  ```
  pamac install {{назва_пакета}}
  ```

- Видалити пакет і його непотрібні залежності (сироти):

  ```
  pamac remove --orphans {{назва_пакета}}
  ```

- Шукати пакет в базі даних пакетів:

  ```
  pamac search {{назва_пакета}}
  ```

- Перелічити встановлені пакети:

  ```
  pamac list --installed
  ```

- Перевірити наявність оновлень пакетів:

  ```
  pamac checkupdates
  ```

- Оновити всі пакети:

  ```
  pamac upgrade
  ```

parted

Програма для роботи з розділами.

Дивіться також: **partprobe**.

Більше інформації: https://www.gnu.org/software/parted/parted.html.

- Перелічити розділи на всіх блокових пристроях:

```
sudo parted --list
```

- Запустити інтерактивний режим із вибраним диском:

```
sudo parted {{/dev/sdX}}
```

- Створити нову таблицю розділів указаного типу міток:

```
sudo parted --script {{/dev/sdX}} mklabel {{aix|amiga|bsd|
dvh|gpt|loop|mac|msdos|pc98|sun}}
```

- Показати інформацію про розділ в інтерактивному режимі:

```
print
```

- Вибрати диск в інтерактивному режимі:

```
select {{/dev/sdX}}
```

- Створити розділ на 16 ГБ із зазначеною файловою системою в інтерактивному режимі:

```
mkpart {{primary|logical|extended}} {{btrfs|ext2|ext3|ext4|
fat16|fat32|hfs|hfs+|linux-swap|ntfs|reiserfs|udf|xfs}}
{{0%}} {{16G}}
```

- Змінити розмір розділу в інтерактивному режимі:

```
resizepart {{/dev/sdXN}} {{кінцева_позиція_розділу}}
```

- Видалити розділ в інтерактивному режимі:

```
rm {{/dev/sdXN}}
```

rm

Видалити файли або директорії.

Дивіться також: **`rmdir`**.

Більше інформації: https://www.gnu.org/software/coreutils/rm.

- Видалити певні файли:

```
rm {{шлях/до/файлу1 шлях/до/файлу2 ...}}
```

- Видалити певні файли, ігноруючи неіснуючі:

```
rm --force {{шлях/до/файлу1 шлях/до/файлу2 ...}}
```

- Видалити певні файли інтерактивно запитуючи перед кожним видаленням:

```
rm --interactive {{шлях/до/файлу1 шлях/до/файлу2 ...}}
```

- Видалити певні файли, друкуючи інформацію про кожне видалення:

```
rm --verbose {{шлях/до/файлу1 шлях/до/файлу2 ...}}
```

- Видалити певні файли та директорії рекурсивно:

```
rm --recursive {{шлях/до/файлу_або_папки1 шлях/до/
файлу_або_папки2 ...}}
```

rpi-eeprom-update

Оновити EEPROM та переглянути іншу інформацію про EEPROM.

Більше інформації: https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#rpi-eeprom-update.

- Переглянути інформацію про поточний встановлений EEPROM raspberry pi:

  ```
  sudo rpi-eeprom-update
  ```

- Оновити Raspberry Pi EEPROM:

  ```
  sudo rpi-eeprom-update -a
  ```

- Скасувати незавершене оновлення:

  ```
  sudo rpi-eeprom-update -r
  ```

- Показати довідку:

  ```
  rpi-eeprom-update -h
  ```

sleep

Затримка на певний час.

Більше інформації: https://www.gnu.org/software/coreutils/sleep.

- Затримка в секундах:

```
sleep {{секунди}}
```

- Затримка в хвилинах (також можна використовувати інші одиниці ([д]ень, [г]одина, [с]екунда, вічність):

```
sleep {{хвилини}}m
```

- Затримка на 1 день 3 години:

```
sleep 1d 3h
```

- Виконати певну команду через 20 хвилин затримки:

```
sleep 20m && {{command}}
```

systemctl

Керуйте системою systemd і диспетчером служб.

Більше інформації: https://www.freedesktop.org/software/systemd/man/systemctl.html.

- Відобразити всі запущені служби:

```
systemctl status
```

- Відобразити список служб, які не запустилися через помилки:

```
systemctl --failed
```

- Запуск/зупинка/перезапуск/перезавантаження служби:

```
systemctl {{start|stop|restart|reload}} {{служба}}
```

- Показати статус служби:

```
systemctl status {{служба}}
```

- Увімкнути/вимкнути запуск служби під час завантаження:

```
systemctl {{enable|disable}} {{служба}}
```

- Маскування/розкриття служби, щоб запобігти ввімкненню та ручній активації:

```
systemctl {{mask|unmask}} {{служба}}
```

- Перезавантажити systemd, шукати нові або змінені пристрої:

```
systemctl daemon-reload
```

- Перевірити, чи ввімкнено службу до автозавантаження:

```
systemctl is-enabled {{служба}}
```

uname

Uname відображає інформацію про машину та операційну систему, на якій вона працює.

Більше інформації: https://www.gnu.org/software/coreutils/manual/html_node/uname-invocation.html.

- Відобразити всю інформацію:

  ```
  uname --all
  ```

- Вивести назву поточного ядра:

  ```
  uname --kernel-name
  ```

- Вивести поточне ім’я хоста мережевого вузла:

  ```
  uname --nodename
  ```

- Вивести поточний випуск ядра:

  ```
  uname --kernel-release
  ```

- Вивести поточну версію ядра:

  ```
  uname --kernel-version
  ```

- Вивести поточну назву апаратного забезпечення машини:

  ```
  uname --machine
  ```

- Вивести поточний тип процесора:

  ```
  uname --processor
  ```

- Вивести назву поточної операційної системи:

  ```
  uname --operating-system
  ```

Windows

cinst

Ця команда є псевдонімом для **`choco install`**.

Більше інформації: https://docs.chocolatey.org/en-us/choco/commands/install.

- Дивись документацію для оригінальної команди:

```
tldr choco install
```

clist

Ця команда є псевдонімом для **`choco list`**.

Більше інформації: https://docs.chocolatey.org/en-us/choco/commands/list.

- Дивись документацію для оригінальної команди:

```
tldr choco list
```

cuninst

Ця команда є псевдонімом для **`choco uninstall`**.

Більше інформації: https://docs.chocolatey.org/en-us/choco/commands/uninstall.

- Дивись документацію для оригінальної команди:

```
tldr choco uninstall
```

curl

Ця команда є псевдонімом для **`curl -p common`**.

Більше інформації: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/invoke-webrequest.

- Дивись документацію для оригінальної команди:

```
tldr curl -p common
```

iwr

Ця команда є псевдонімом для **`invoke-webrequest`**.

Більше інформації: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/invoke-webrequest.

- Дивись документацію для оригінальної команди:

```
tldr invoke-webrequest
```

pwsh where

Ця команда є псевдонімом для **`Where-Object`**.

Більше інформації: https://learn.microsoft.com/powershell/module/microsoft.powershell.core/where-object.

- Дивись документацію для оригінальної команди:

```
tldr Where-Object
```

sls

Ця команда є псевдонімом для **`Select-String`**.

Більше інформації: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/select-string.

- Дивись документацію для оригінальної команди:

```
tldr select-string
```