Управление на софтуер

7. Управление на софтуер  
• Архивиране (tar,cpio);  
• Компресия ( gzip, bzip2, zip);  
• Инсталиране от сорс код (configure, make files);  
• Ръчно инсталирани библиотеки (ldd, ldconfig) ;  
• RPM, YUM, DPKG, APT, Aptitude;  
8. Текстови редактор VI  
• Oсновни команди за преглед и редактиране на текстови файлове.

Разбиране на Linux за управление и инсталиране на софтуер

Преглед

Посетете нашия раздел Уроци, за да научите повече за: [Разбиране на Linux: Управление и инсталиране на софтуер] https://www.udoo.org/tutorial/understanding-linux-software-management-installing/).

Ubuntu на UDOO, базирана на Linux операционна система, се справя с управлението на софтуера по начин, по който може да не сте свикнали. Това обаче е много удобно в края на деня. Ubuntu разполага със система от хранилища, която е софтуерна колекция, съхранявана на няколко сървъра, които можете да получите и инсталирате с няколко команди. Това гарантира, че актуализирането и инсталирането на софтуера е лесно, ефективно и безопасно.

За да цитирам от Уикипедия:

В софтуера, системата за управление на пакети, наричана още мениджър на пакети, е набор от софтуерни инструменти за автоматизиране на процеса на инсталиране, надстройка, конфигуриране и премахване на софтуерни пакети за операционната система на компютъра по един последователен начин. Обикновено тя поддържа база данни със софтуерни зависимости и информация за версията, за да предотврати несъответствия в софтуера и липсващи предпоставки. Пакетите са дистрибуции на софтуер, приложения и данни. Пакетите съдържат и метаданни, като например името на софтуера, описание на неговата цел, номер на версията, доставчик, контролна сума и списък на зависимости, необходими за правилното функциониране на софтуера. При инсталиране метаданните се съхраняват в локална база данни за пакети.

Софтуерният мениджър на UDOO на Ubuntu е инструмент от командния ред, наречен apt. Използването му е много лесно. apt има свой собствен списък на хранилището, съхранен в

/etc/apt/sources.list

Можете да добавяте или премахвате софтуерни източници, като редактирате този файл. За да видите съдържанието му, просто напишете:

cat /etc/apt/sources.list

Предлага се обаче да редактирате това само ако знаете какво правите. Можете да бъдете в безопасност само с тази, включена в Ubuntu на UDOO.

Преди да се опитате да инсталирате софтуер от хранилище, нека актуализираме apt кеша. Това е основно база данни на целия наличен софтуер, който включва зависимости и информация за версията.

Да се ​​актуализира:

sudo apt-get update

След това нека предположим, че искаме да инсталираме nano, много полезен и лесен текстов редактор:

sudo apt-get install nano

Ще бъдем попитани дали искаме да продължим, да отговорим с "да". Така че, въведете y

След това Apt ще изтегли, разопакова и инсталира всички необходими за нас файлове. Просто изчакайте и ще приключи.

Да предположим, че искаме да премахнем програма, например vim (друг текстов редактор)

sudo apt-get премахни vim

Отново, ние отговаряме с да напишете y.

Някои други полезни команди за apt и system housekeeping са:

apt-get autoremove

премахва автоматично пакетите, които вече не са необходими

ап-да се чисти

изтрили изтеглени пакети, които губят дисково пространство

Сега можем да използваме друг мениджър на пакети, с графичен потребителски интерфейс, Synaptic, който е мениджърът на пакетите по подразбиране в повечето версии на Ubuntu.

Ще използваме apt за инсталиране на synaptic

sudo apt-get install synaptic

Веднъж инсталиран, ще намерим Synaptic Package Manager в менюто „Други“. Нека го отворим.

Въведете паролата на root и въведете.

След това можете да търсите и инсталирате всеки пакет, който желаете, в приятна и удобна за потребителя среда.

За да започнете, ще активираме повече софтуерни източници, отколкото имаме по подразбиране.

За да направите това, отидете в „настройки“, след това в „хранилища“ и проверете всички хранилища, които искате да активирате; обикновено трябва да сте добър, като разрешите всички софтуерни източници от раздела „Софтуер на Ubuntu“.

След като приключите, затворете и натиснете “reload”, което е същото като “apt-get update”.

Когато приключите, можете да започнете да търсите и разглеждате хиляди различни софтуери, за да ги инсталирате, просто поставете отметка в квадратчето, изберете „Маркирай за инсталация“ и натиснете „приложи“.

Вече имате на разположение целия софтуер, от който се нуждаете, за да разширите възможностите на UDOO NEO.

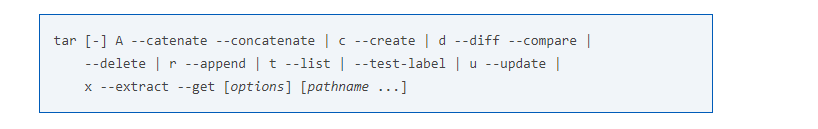
Linux tar команда

описание

"Тар" означава лентов архив. Това е архивен файлов формат.

Tar е първоначално разработен в ранните дни на Unix за целите на архивирането на файлове на устройства за съхранение на лента. По-късно той беше официално официализиран като част от стандарта POSIX, а днес се използва за събиране, разпространение и архивиране на файлове, като същевременно се запазват атрибутите на файловата система, като потребителски и групови разрешения, дати за достъп и модификация и структури на директории.

Тази документация обхваща GNU версията на tar, която е включена в най-модерните варианти на операционната система Linux.



операция

Първият аргумент на tar трябва да бъде спецификация на функцията: или една от буквите A, c, d, r, t, u, или x, или едно от дългите имена на функции. Функционалното писмо не трябва да има префикс с тире ("-") и може да бъде комбинирано с други опции с една буква. Дългото име на функция трябва да има представка с двойно тире ("-"). Някои опции приемат параметър; с еднобуквената форма те трябва да се представят като отделни аргументи. С дългата форма те могат да бъдат дадени чрез добавяне на "= value" към опцията.

Например следните команди са еквивалентни:



Функции

Определянето на една от следните функции избира основния режим на работа на tar-a:

A, - catenate, --concatenate Добавете tar файлове към архив.

Създайте нов архив.

d, --diff, --compare Изчислете разликите между архива и файловата система.

--delete Изтриване от архива. (Тази функция не работи с магнитни ленти).

r, --append Добавете файловете в края на tar архива.

t, --list Посочете съдържанието на архива.

--test-label Проверете етикета на архива и излезте.

u, --update Добавяне на файлове, но само тези, които са по-нови от копието в архива.

x, --extract, --get Извличане на файлове от архив.

Други възможности

Следните опции определят начина, по който работи tar:

- [0-7] [lmh] Указва диск и плътност. (Ако не сте сигурни какво означава това, не е необходимо да използвате тази опция.)

-a, --auto-compress Използвайте суфикса на архива, за да определите програмата за компресиране. Например, ако е зададена тази опция, винаги ще се обработва архив с разширение .tar.gz, както ако е зададена опцията --gzip (вижте --gzip, по-долу).

--add-file = file Добавяне на файл в архива. (Тази опция е особено полезна, когато името на файла започва с тире.)

--anchored Шаблоните с имена на файлове трябва да съвпадат от началото на името на файла.

--no-anchored Шаблоните с имена на файлове могат да съвпадат след всяко "/" (това е по подразбиране за изключване на файлове; вижте --exclude, по-долу).

--atime-save Запази време за достъп до архивирани файлове, или чрез възстановяване на времето, или (ако операционната система го поддържа) не ги променя на първо място.

--no-auto-compress Не определя автоматично програмата за компресиране, използвайки суфикса с име на архивния файл.

-B, --blocking-factor BLOCKS Използвайте BLOCKS x 512 байта на запис при работа с архива.

-B, - чете-пълни записи "Re-block" всички входни данни. Тази опция може да помогне за четене на повредени архиви.

--backup [= backup-method Вместо да изтрива файлове от файловата система, tar ще ги архивира, като използва споменатия резервен метод за архивиране, който може да бъде един от следните:

t, numbered Винаги правете номерирани архиви.

nil, съществуващ Направете номерирани архиви на файловете, които вече имат, и прости архиви на другите.

никога, просто Винаги правете прости архиви.

Ако не е зададен, методът на резервната копие по подразбиране е стойността на променливата на обкръжението VERSION\_CONTROL; ако VERSION\_CONTROL е недефинирана, методът на резервния файл по подразбиране е съществуващ.

-C, --directory DIR Преди да извършите каквито и да е операции, променете директорията DIR.

- проверете номерата на устройствата за проверка на устройства, когато създавате допълнителни архиви. Това е поведението по подразбиране.

--no-check-device Не проверявайте номерата на устройствата, когато създавате инкрементални архиви.

--checkpoint [= NUMBER] Използвайте "контролни точки": показвайте съобщение за напредъка на всеки NUMBER запис (по подразбиране 10).

--checkpoint-action = ACTION Изпълнете ACTION на всяка контролна точка. ДЕЙСТВИЕ може да бъде едно от следните:

Звънец Звучи звънец на конзолата.

точка,. Отпечатайте една точка.

echo Показване на текстово съобщение на конзолата (до стандартна грешка).

echo = string Показване на низ от стандартни грешки; преди изхода, всички метасимволи в низ се разширяват.

exec = команда Изпълни дадената команда.

sleep = time Изчакайте секунди за време.

ttyout = string Изходен низ в текущата конзола ('/ dev / tty').

Могат да бъдат зададени няколко опции --checkpoint-action. Предоставените действия се изпълняват в реда на появата им в командния ред.

Използването на --checkpoint-action без --checkpoint ще поеме по подразбиране интервал за проверка на 10 записа.

--delay-directory-restore Забавяне настройка на времето за промяна и разрешения на извлечени директории до края на извличането.

--no-delay-directory-restore Модификационните времена и разрешения на извлечените директории се задават, когато всички файлове от тази директория са извлечени. Това е по подразбиране.

--exclude = PATTERN Избягвайте да работите с файлове, чиито имена съответстват на модела с име на файл PATTERN.

--exclude-backups изключва архивиране и заключване на файлове от всички операции.

--exclude-caches Причинява tar да изключва всички директории, които съдържат маркер за кеш директория.

Тагът на кеш директорията е кратък файл с името CACHEDIR.TAG и има стандартен заглавие, посочено в http://www.brynosaurus.com/cachedir/spec.html. Тази опция изключва съдържанието на директорията, но архивира самата директория и файла CACHEDIR.TAG.

--exclude-caches-all Изтрийте директории, съдържащи файл CACHEDIR.TAG изцяло.

--exclude-caches-under Изключете всичко под директории, съдържащи файл CACHEDIR.TAG, включително файла CACHEDIR.TAG; но архивирайте директорията.

--exclude-tag = FILE Изключва съдържанието на всяка директория, съдържаща файл FILE, но архивира директорията и файла FILE.

--exclude-tag-all = FILE Изцяло отхвърля директории, съдържащи файл FILE.

--exclude-tag-under = FILE Изключва всичко в директориите, съдържащи файл FILE, включително файла FILE; но архивирайте директорията.

--exclude-vcs Изключете поддирекциите за контрол на версиите. Тази опция разпознава файловете и директориите, използвани от много широко използвани VCS системи. Изключените файлове са:

CVS / и всичко под него

RCS / и всичко под него

SCCS

cpio копира файлове в и от архивите.

описание

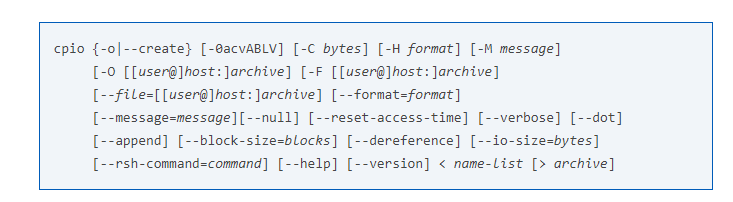
GNU cpio е инструмент за създаване и извличане на архиви или копиране на файлове от едно място на друго. Той обработва редица cpio формати, както и четене и писане на tar файлове.

Поддържат се следните архивни формати: двоичен, стар ASCII, нов ASCII, CRC, HPUX двоичен, HPUX стар ASCII, стар катран и POSIX.1 tar. Форматът на tar е предвиден за съвместимост с програмата tar. По подразбиране cpio създава архиви за двоичен формат, за съвместимост с по-старите програми на cpio. При извличане от архиви, cpio автоматично разпознава какъв вид архив чете и може да чете архиви, създадени на машини с различен байт-ред.

Синтаксис

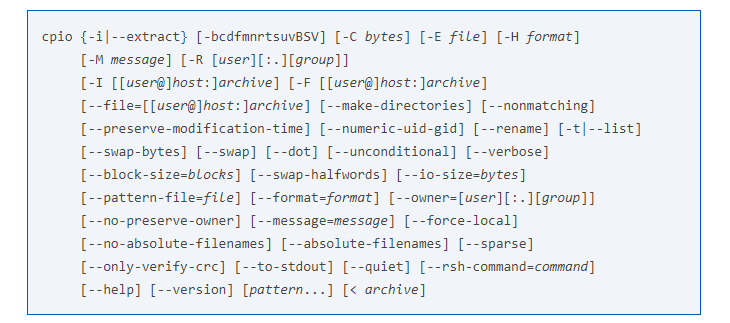
Синтаксис на режим на копиране:

В режим на копиране cpio копира файлове в архив. Той чете списък с имена на файлове, по един на ред, на стандартния вход и записва архива на стандартния изход. Типичен начин за генериране на списъка с имена на файлове е командата find; трябва да дадете опцията -depth, за да минимизирате проблемите с разрешения за директории, които не могат да се четат. Синтаксис на режима за копиране:



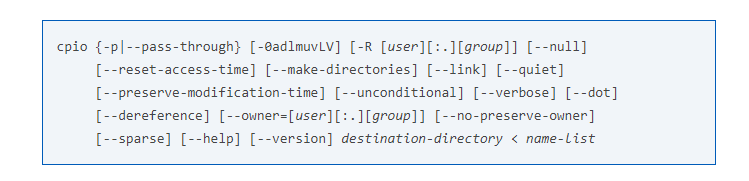
Синтаксис на режим на копиране:

В режим на копиране, cpio копира файлове от архив или изброява съдържанието на архива. Той чете архива от стандартния вход. Всички аргументи на командния ред, които не са опция, са шаблони на обвивката на обвивката; от архива се копират само файловете в архива, чиито имена съответстват на един или повече от тези шаблони. За разлика от черупката, първоначално '.' в име на файл съвпада със заместващ знак в началото на шаблон, а '/' в име на файл може да съответства на заместващи символи. Ако не са дадени шаблони, всички файлове се извличат. Синтаксис на режима за копиране:



**Copy-Pass Mode Syntax:**

In copy-pass mode, **cpio** copies files from one directory tree to another, combining the copy-out and copy-in steps without actually using an archive. It reads the list of files to copy from the standard input; the directory into which it will copy them is given as a non-option argument. Copy-Pass mode syntax:



Настроики

-0, --null Прочетете списък с имена на файлове, прекратени с нулев знак, вместо нов ред, така че файловете, чиито имена съдържат нови линии, могат да бъдат архивирани. GNU find е един от начините да се създаде списък с нулево прекъснати имена на файлове. Тази опция може да се използва при копиране и копиране.

-a, --reset-access-time Нулирайте времето за достъп на файловете, след като ги прочетете, така че да не изглежда, че току-що са били прочетени.

-A, - добавяне Добавяне към съществуващ архив. Работи само в режим на копиране. Архивът трябва да бъде диск, посочен с опцията -O или -F (-file).

-b, --swap Прехвърлете и двете думи в думите и байта на полуслов в данните. Еквивалент на -sS. Тази опция може да се използва в режим на копиране. Използвайте тази опция, за да конвертирате 32-битови цели числа между големи и малки машини.

-B Задайте размера на I / O блока до 5120 байта. Първоначално размерът на блока е 512 байта.

--block-size = BLOCK-SIZE Задайте размера на I / O блока на BLOCK-SIZE \* 512 байта.

-c Идентично на '-H newc'; използва новия (SVR4) преносим формат. Ако искате стария преносим (ASCII) архивен формат, вместо това използвайте '-H odc'.

-C IO-SIZE, --io-size = IO-SIZE Задайте размера на I / O блока на IO-SIZE байта.

-d, --make-directories Създаване на водещи директории, където е необходимо.

-E FILE, --pattern-file = FILE Прочетете допълнителни шаблони, указващи имената на файлове за извличане или списък от FILE. Редовете на ФАЙЛ се третират като че ли са аргументи без опция за cpio. Тази опция се използва в режим на копиране.

-f, --nonmatching Копирайте само файлове, които не съвпадат с нито един от дадените модели.

-F, --file = архив Името на файла за архивиране вместо стандартния вход или изход. За да използвате лентово устройство на друга машина като архива, използвайте име на файл, което започва с „HOSTNAME:“. Името на хоста може да бъде предшествано от потребителско име и '@' за достъп до отдалеченото устройство за лента като този потребител, ако имате разрешение за това (обикновено запис в файла на потребителя '/ .rhosts').

--force-local С -F, -I или -O, използвайте името на архивния файл като локален файл, дори ако съдържа двоеточие, което обикновено означава име на отдалечен хост.

-H FORMAT, --format = FORMAT Използвайте FORMAT във формат на архив. Валидните формати са изброени по-долу; същите имена се признават и в all-caps. По подразбиране в режим на копиране е автоматично разпознаване на архивния формат, а в режим на копиране е 'bin'.

bin: Остарял двоичен формат.

odc: Старият (POSIX .1) преносим формат.

newc: Новият (SVR4) преносим формат, който поддържа файлови системи с повече от 65536 inodes.

crc: Новият (SVR4) преносим формат с добавена контролна сума.

tar: Старият формат на tar.

ustar: POSIX .1 катранен формат. Също така, разпознава GNU tar архиви, които са подобни, но не са идентични.

hpbin: Остарял двоичен формат, използван от cpio на HPUX (който различно съхранява файловете на устройствата).

hpodc: Преносим формат, използван от cpio на HPUX (който различно съхранява файловете на устройствата).

-i, --extract Изпълни в режим на копиране. (вижте „Режим на копиране“).

-I архивиране на името на файла на архива вместо стандартния вход. За да използвате лентово устройство на друга машина като архива, използвайте име на файл, което започва с „HOSTNAME:“. Името на хоста може да бъде предшествано от потребителско име и '@' за достъп до отдалеченото устройство за лента като този потребител, ако имате разрешение за това (обикновено запис в файла на потребителя '/ .rhosts').

-k пренебрегнато; за съвместимост с други версии на cpio.

-l, --link Свързва файлове, вместо да ги копира, когато е възможно.

-L, --dereference Копирайте файла, към който сочи символна връзка, а не самата символна връзка.

-m, --preserve-modification-time Запазвайте предишните модификации на файловете при създаване на файлове.

-M MESSAGE, --message = MESSAGE Отпечатайте MESSAGE, когато се стигне до края на том на носителя за архивиране (като например лента или дискета), за да накарате потребителя да вмъкне нов том. Ако MESSAGE съдържа низ '% d', той се заменя с текущия номер на тома (започващ от 1).

-n, --numeric-uid-gid Показват числови UID и GID вместо да ги превеждат в имена, когато използвате опцията "--verbose".

--no-absolute-filenames Създаване на всички файлове по отношение на текущата директория в режим на копиране, дори ако те имат абсолютно име на файл в архива.

--absolute-filenames Това е по подразбиране: кажете на cpio да не изважда водещите компоненти на файловото име, които съдържат '..' и водещи наклонени черти от имена на файлове в режим на копиране.

--no-preserve-owner Не променяйте собствеността на файловете; оставете ги собственост на потребителя, който ги извлича. Това е по подразбиране за не-root потребители, така че потребителите в System V не случайно раздават файлове. Тази опция може да се използва в режим на копиране и копиране.

-o, --създайте Run в режим на копиране. (вижте „Режим за копиране“).

-O Архив Името на файла за архивиране вместо стандартния изход. За да използвате лентово устройство на друга машина като архива, използвайте име на файл, което започва с „HOSTNAME:“. Името на хоста може да бъде предшествано от потребителско име и '@' за достъп до отдалеченото устройство за лента като този потребител, ако имате разрешение за това (обикновено запис в файла на потребителя '/ .rhosts').

--only-verify-crc Проверете CRC на всеки файл в архива, когато четете архив във формат CRC. Всъщност не извличайте файловете.

-p, - pass-through Работи в режим копиране. (вижте „Режим на копиране“).

--quiet Не отпечатвайте броя на копираните блокове.

-r, --rename Интерактивно преименуване на файлове.

-R [потребител] [:.] [Група], - собственик [потребител] [:.] [Група] Задаване на собственост на всички файлове, създадени към посочения потребител и / или група в режими за копиране и копиране , Трябва да присъстват или потребителят, или групата, или и двете. Ако групата е пропусната, а „:“ или „.“ разделител е даден, използвайте групата за вход в дадения потребител. Само суперпотребителят може да променя собствеността на файловете.

--rsh-command = COMMAND Известява cpio, че трябва да използва COMMAND за комуникация с отдалечени устройства.

-s, --swap-bytes Разменете байтовете на всяко полуслово (двойка байта) във файловете. Тази опция може да се използва в режим на копиране.

-S, --swap-halfwords Разменете полудушовете на всяка дума (4 байта) във файловете. Тази опция може да се използва в режим на копиране.

--sparse Записване на файлове с големи блокове от нули като разредени файлове. Тази опция се използва при копиране и копиране.

-t, --list Отпечатва съдържанието на входа.

--to-stdout Извличане на файлове към стандартен изход. Тази опция може да се използва в режим на копиране.

-u, --unconditional Сменете всички файлове, без да питате дали да замените съществуващите по-нови файлове с по-стари файлове.

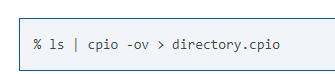
-v, --verbose Избройте обработените файлове, или с '-t', въведете 'ls -l' списък със съдържание на таблицата със съдържание. В подробна таблица на съдържанието на един архив на потребителя, имената на потребителите и групите в архива, които не съществуват в локалната система, се заменят с имена, които съответстват локално на числовите UID и GID, съхранени в архива.

-V, --dot Печат a. за всеки обработен файл.

--version Отпечатайте номера на версията на програмата и излезте.

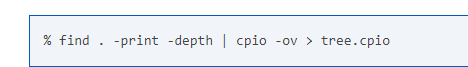
Примери

Когато създавате архив, cpio заема списъка с файлове, които ще се обработват от стандартния вход, след което изпраща архива към стандартния изход или към устройството, дефинирано от опцията '-F'. Обикновено намирането или ls се използва за предоставяне на този списък на стандартния вход. В следващия пример можете да видите възможностите за архивиране на съдържанието на една директория:



Опцията '-o' създава архива, а опцията '-v' отпечатва имената на архивираните файлове, както са добавени. Забележете, че опциите могат да бъдат поставени заедно след единичен "-" или могат да бъдат поставени отделно в командния ред. '>' Пренасочва изхода cpio към файла 'directory.cpio'.

Ако искате да архивирате цялото дърво на директориите, командата find може да предостави списъка с файлове на cpio:

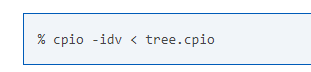


Това ще вземе всички файлове в текущата директория, директориите по-долу и ще ги постави в архива tree.cpio. Отново '-o' създава архив, а опцията '-v' ви показва името на файловете, когато са архивирани (вижте 'Режим за копиране'). Използвайки '.' в оператора find ще ви даде по-голяма гъвкавост при възстановяването, тъй като ще запази имената на файловете с относителна пътека по твърд, абсолютен път. Опцията '-depth' принуждава 'find' да отпечата записите в директорията, преди да отпечата самата директория. Това ограничава ефектите на ограничителните разрешения за директории чрез отпечатване на записите в директорията в директорията преди самото име на директория.

Извличането на архив изисква малко повече мисъл, защото cpio няма да създава директории по подразбиране. Друга характеристика е, че няма да презапише съществуващи файлове, освен ако не е посочено.

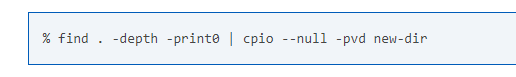


Това ще извлече архивираните файлове във файла directory.cpio и ще ги постави в настоящата директория. Опцията '-i' извлича архива и '-v' показва имената на файловете, които са извлечени. Ако се занимавате с архивирано дърво на директории, трябва да използвате опцията '-d', за да създавате директории, ако е необходимо, като:



Това ще вземе съдържанието на архива tree.cpio и ще го извлече в текущата директория. Ако се опитате да извлечете файловете върху файловете със същото име, които вече съществуват (и имат същото или по-късно време за модификация), cpio няма да извлече файла, освен ако не е казано да го направи чрез опцията -u (вижте 'Копиране режим ").

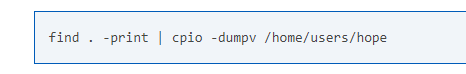
В режим копиране, cpio копира файлове от едно дърво на директория в друго, комбинирайки стъпките за копиране и копиране, без действително да използва архив. Той чете списъка с файлове, които да се копират от стандартния вход; директорията, в която ще ги копира, се дава като аргумент без опция (вижте „Режим на копиране“).



Примерът показва копиране на файловете от настоящата директория и поддиректориите в нова директория, наречена new-dir. Някои нови опции са '-print0' с GNU find, комбинирани с опцията '- null' на cpio. Тези две опции действат заедно, за да изпращат имена на файлове между find и cpio, дори ако в имената на файловете са вградени специални символи. Друг е '-p', който казва на cpio да предава файловете, които намира, в директорията 'new-dir'.



По-горе, с помощта на командата find ще бъдат изброени всички файлове и директории, а с помощта на командата cpio ще бъдат копирани файловете, изброени в дискета.



В горния пример командата find ще изведе всички файлове и поддиректории на текущата директория и ще ги прехвърли в командата cpio, която копира тези файлове в потребителския акаунт на надеждата.



Gzip Command в Linux

Командата gzip компресира файлове. Всеки един файл се компресира в един файл. Компресираният файл се състои от GNU zip хедър и дефлирани данни.

Ако даден файл като аргумент, gzip компресира файла, добавя суфикс “.gz” и изтрива оригиналния файл. Без аргументи gzip компресира стандартния вход и записва компресирания файл на стандартен изход.

Разлика между Gzip и zip командата в Unix и кога да се използва коя команда

ZIP и GZIP са два много популярни метода за компресиране на файлове, за да се спести място, или да се намали времето, необходимо за предаване на файловете в мрежата или интернет.

Като цяло, GZIP е много по-добър в сравнение със ZIP, по отношение на компресия, особено при компресиране на огромен брой файлове.

Обичайната практика при GZIP е архивирането на всички файлове в един тарбол преди компресиране. В ZIP файлове отделните файлове се компресират и след това се добавят към архива.

Когато искате да изтеглите един файл от ZIP, той просто се извлича, след което се декомпресира. С GZIP целият файл трябва да бъде декомпресиран, преди да можете да извлечете желания файл от архива.

Когато изтегляте 1MB файл от 10GB архив, е съвсем ясно, че в GZIP ще отнеме много повече време, отколкото в ZIP.

GZIP в това, как тя функционира, също е отговорен за предимството на GZIP. Тъй като алгоритъмът за компресиране в GZIP компресира един голям файл вместо няколко по-малки, той може да се възползва от излишъка във файловете, за да намали още повече размера на файла.

Ако архивирате и компресирате 10 идентични файла с ZIP и GZIP, ZIP файлът ще бъде над 10 пъти по-голям от получения GZIP файл.

Синтаксис :



Пример:



Тази команда ще създаде компресиран файл на mydoc.txt, наречен mydoc.txt.gz и ще изтрие оригиналния файл.

Настроики :

-f опция: Понякога файлът не може да бъде компресиран. Може би се опитвате да компресирате файл, наречен "myfile1", но вече има файл, наречен "myfile1.gz". В този случай командата „gzip“ обикновено няма да работи.

За да принудите командата "gzip" да прави нещата си просто използвайте опцията -f:



Това ще насилва компресиране на файл с име myfile.txt, дори ако вече съществува файл с име myfile.txt.gz

-k опция: По подразбиране, когато компресирате файл с помощта на командата "gzip", ще получите нов файл с разширение ".gz". Ако искате да компресирате файла и да запазите оригиналния файл, трябва да стартирате gzip команда с опция -k:



Горната команда ще завърши с файл, наречен "mydoc.txt.gz" и "mydoc.txt".

-L опция: Тази опция показва лиценза на gzip.



-r option: Тази опция може да компресира всеки файл в папка и нейните подпапки. Тази опция не създава файл, наречен foldername.gz. Вместо това тя прелиства структурата на директориите и компресира всеки файл в тази структура на папките.



Това ще компресира всички файлове, налични в тестовата папка.

- [1-9] опция: Тя позволява да се промени нивото на компресия. Файлът може да бъде компресиран по различни начини. Например, можете да отидете за по-малка компресия, която ще работи по-бързо или можете да отидете за максимална компресия, която има компромис отнема повече време, за да се изпълни.



Това ще постигне максимална компресия при най-бавната скорост



За да получите минимална компресия с най-висока скорост

-v option: Тази опция показва името и процентното намаление за всеки компресиран или декомпресиран файл.



-d опция: Тази опция позволява декомпресиране на файл с помощта на командата “gzip”.



Тази команда ще разархивира компресирания файл с име mydoc.txt.gz.

bzip2 команда в Linux с примери

Командата bzip2 в Linux се използва за компресиране и декомпресиране на файловете, т.е. помага при свързването на файловете в един файл, който заема по-малко място за съхранение, както използва оригиналния файл. Той има по-бавно време за декомпресия и по-голяма памет. Той използва алгоритъма за компресиране на текстове за сортиране на Бъроуз-Уилър и кодиране на Huffman. Всеки файл е заменен от компресирана версия на себе си, с името оригинално име на файла, последвано от разширение bz2.



Настроики:

-z: Тази опция принуждава компресията. Това е противоположна команда за декомпресия, т.е. -d Опция.

Забележка: Тази опция изтрива и оригиналния файл.

-k: Тази опция прави компресия, но не изтрива оригиналния файл.

-d: Тази опция се използва за декомпресиране на компресирани файлове.

-t: Тази опция проверява целостта на файла и не декомпресира файла. Това ни дава представа, че файлът е повреден или не.

-V: Големият режим показва коефициента на компресия за всеки обработен файл. Той също така увеличава степента на подробност, извлича много информация, която е от първостепенна важност за диагностични цели.

-h –help: За да покажете помощното съобщение и да излезете.

-L –license -V –version: Използва се за показване на версията на софтуера, лицензионните условия и условията.

-q –quiet: Потиска несъществените предупредителни съобщения. Съобщенията, свързани с I / O грешки и други критични събития, няма да бъдат потискани.

-f –force: Ще принуди презаписването на изходните файлове.

Команда ZIP в Linux с примери

ZIP е програма за компресиране и пакетиране на файлове за Unix. Всеки файл се съхранява в единичен файл .zip {.zip-filename} с разширение .zip.

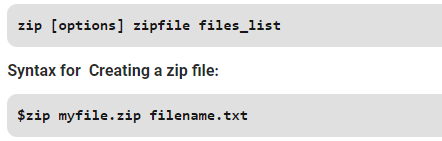
zip се използва за компресиране на файловете, за да се намали размера на файла, а също и като помощна програма за пакети. zip е наличен в много операционни системи като unix, linux, windows и т.н.

Ако имате ограничена пропускателна способност между два сървъра и искате да прехвърлите файловете по-бързо, тогава затворете файловете и прехвърлете.

Zip програмата поставя един или повече компресирани файлове в един zip архив, заедно с информация за файловете (име, път, дата, време на последната модификация, защита и проверка на информацията, за да се провери целостта на файла). Цялата структура на директориите може да бъде опакована в zip архив с една команда.

Коефициентите на компресия от 2: 1 до 3: 1 са общи за текстовите файлове. zip има един метод за компресиране (дефлация) и може също да съхранява файлове без компресия. zip автоматично избира по-доброто от двете за компресиране на всеки файл.

Програмата е полезна за опаковане на набор от файлове за разпространение; за архивиране на файлове; и за спестяване на дисково пространство чрез временно компресиране на неизползвани файлове или директории.



Извличане на файлове от zip файл

Разархивирайте ще изброи, тества или извлече файлове от ZIP архив, често срещан в Unix системите. Поведението по подразбиране (без опции) е да извлечете във текущата директория (и под-директориите под него) всички файлове от указания ZIP архив.



Настроики :

1. -d Опция: Премахва файла от zip архива. След създаването на zip файл, можете да премахнете файл от архива, като използвате опцията -d.

Да предположим, че следните файлове в текущата ми директория са изброени по-долу:

hello1.c

hello2.c

hello3.c

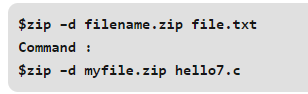
hello4.c

hello5.c

hello6.c

hello7.c

hello8.c



2.-u Опция: Актуализира файла в zip архива. Тази опция може да се използва за актуализиране на посочения списък от файлове или добавяне на нови файлове към съществуващия zip файл. Актуализирайте съществуващ запис в zip архива само ако е бил променен по-скоро от версията вече в zip архива.

Синтаксис:

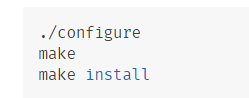


3. -m Опция: Изтрива оригиналните файлове след zipping. Всъщност преместете посочените файлове в zip архива, това изтрива целевите директории / файлове, след като направите зададения zip архив. Ако директорията стане празна след премахване на файловете, директорията също се премахва. Не се правят изтривания, докато zip не е създал архива без грешки. Това е полезно за запазване на дисковото пространство, но е потенциално опасно премахване на всички входни файлове.

Инсталиране от сорс код

Магията зад конфигуриране, създаване, инсталиране (configure, make files)

Ако сте използвали някакъв вкус на Unix за разработване, вероятно сте инсталирали софтуер от източник с това магическо заклинание:



Знам, че съм го написал много, но в ранните си дни, използвайки Linux, аз наистина не разбрах какво означава това, просто знаех, че ако искам да инсталирам софтуер, това е заклинанието да рецитирам.

Наскоро изграждах свои собствени Unix инструменти и исках да използвам този стандартен процес на инсталиране; Той не само е познат на много потребители на Unix, но и е чудесна отправна точка за създаване на пакет за Homebrew и различните пакети на Linux и BSD мениджъри. Беше време да се промъкнем в Unix Grimoire и да разберем какво прави заклинанието.

Какво прави всичко това

В този процес има три отделни стъпки:

Конфигурирайте софтуера

Конфигуриращият скрипт е отговорен за подготовката за изграждане на софтуера на вашата конкретна система. Той гарантира, че всички зависимости за останалата част от процеса на изграждане и инсталиране са достъпни, и открива каквото трябва да знае, за да използва тези зависимости.

Програмите Unix често се пишат на C, така че ние обикновено се нуждаем от C компилатор, за да ги изградим. В тези случаи конфигуриращият скрипт ще установи, че вашата система наистина има компилатор C, и ще открие как се нарича и къде да го намери.

Изградете софтуера

След като конфигурирането е свършило работата си, ние можем да призовем make за изграждане на софтуера. Това изпълнява серия от задачи, дефинирани в Makefile, за да се изгради завършената програма от нейния изходен код.

Изтеглената от вас архива обикновено не включва завършен Makefile. Вместо това той идва с шаблон, наречен Makefile.in и configure script произвежда персонализиран Makefile, специфичен за вашата система.

Инсталирайте софтуера

Сега, когато софтуерът е изграден и готов за работа, файловете могат да бъдат копирани в крайните им местоназначения. Командата make install ще копира изградената програма и нейните библиотеки и документация на правилните местоположения.

Това обикновено означава, че двоичният файл на програмата ще бъде копиран в директория на PATH, страницата с ръководството на програмата ще бъде копирана в директория на MANPATH, а всички други файлове, от които зависи, ще бъдат съхранени на подходящо място.

Тъй като инсталационната стъпка също е дефинирана в Makefile, където софтуерът е инсталиран, може да се промени на базата на опции, прехвърлени към конфигуриращия скрипт, или неща, които скриптът за конфигуриране е открил за вашата система.

В зависимост от мястото, където се инсталира софтуерът, може да се наложи да имате ескалирани разрешения за тази стъпка, така че да можете да копирате файлове в системни директории. Използването на sudo често ще свърши работа.

Откъде идват тези скриптове

Всичко това работи, защото скриптът за конфигуриране изследва вашата система и използва информацията, която открива, за да преобразува шаблон на Makefile.in в Makefile, но откъде идват конфигурационният скрипт и шаблонът Makefile.in?

Ако някога сте отваряли скрипт за конфигуриране или свързан Makefile.in, ще видите, че те са хиляди редове с плътен скрипт. Понякога тези поддържащи скриптове са по-дълги от изходния код на програмата, която инсталират.

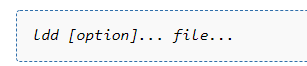
Дори да се започне от съществуващ скрипт за конфигуриране, би било много трудно да се създаде ръчно. Не се притеснявайте обаче: тези скриптове не са построени на ръка.

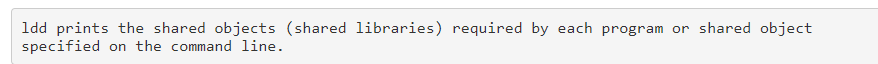
Програмите, които са построени по този начин, обикновено са били пакетирани, използвайки набор от програми, общо наричани autotools. Този пакет включва autoconf, automake и много други програми, всички от които работят заедно, за да направят живота на поддържащия софтуер значително по-лесен. Крайният потребител не вижда тези инструменти, но те отнемат болката от инсталирането на процес на инсталиране, който ще се изпълнява последователно на много различни вкусове на Unix.

Команда Linux ldd

Ако вашата работа включва задълбочени познания за изпълними и споделени библиотеки в Linux, има няколко инструмента за командния ред, които трябва да знаете. Един от тях е ldd, който можете да използвате за достъп до общите зависимости на обекта. В този урок ще обсъдим основите на тази програма, като използваме някои лесни за разбиране примери.

Както вече споменахме в началото, командата ldd отпечатва зависимости на общите обекти. Следва синтаксисът на командата:





LDCONFIG

ldconfig - конфигуриране на динамични връзки за изпълнение на линкер

ldconfig създава необходимите връзки и кеш към най-новите

       споделени библиотеки, намерени в указаните в командата директории

       във файла /etc/ld.so.conf и в доверените директории,

       / lib и / usr / lib (на някои 64-битови архитектури като x86-64, / lib

       и / usr / lib са доверени директории за 32-битови библиотеки, докато

       / lib64 и / usr / lib64 се използват за 64-битови библиотеки).

       Кешът се използва от изпълнителния линкер, ld.so или ld-linux.so.

       ldconfig проверява заглавието и имената на библиотеките

       среща при определянето кои версии трябва да имат своите връзки

       актуализиран.

       ldconfig ще се опита да изведе типа библиотеки ELF (т.е.

       libc5 или libc6 / glibc) на базата на библиотеките, ако има такива

       е свързан с.

       Някои съществуващи библиотеки не съдържат достатъчно информация за разрешаване

       приспадането на техния вид. Следователно файлът /etc/ld.so.conf

       формат позволява спецификацията на очакван тип. Това се използва

       само за онези библиотеки на ELF, които не можем да разработим. Форматът

       е "dirname = TYPE", където TYPE може да бъде libc4, libc5 или libc6. (Това

       синтаксисът работи и в командния ред.) Пространствата не са разрешени.

       Вижте също опцията -p. ldconfig обикновено трябва да се управлява от

       суперпотребител, тъй като може да изисква разрешение за писане на някои собственици на root

Инсталиране на софтуеар с RPM, YUM, DPKG, APT, Aptitude

Инсталирането на софтуеар в линукс става с една от горните команди като ще ги разделиме за редхат дистрибуции ( RPM YUM DNF ) и Дебиан базирани (apt aptitude dpkg )

Примери за командите RPM за заявка, инсталиране, премахване и надграждане на пакети

Целият софтуер на Linux системата е разделен на пакети, които могат да бъдат инсталирани, деинсталирани, обновени, запитвани и проверени. CentOS / RHEL използва Red Hat Package Manager (RPM), за да улесни инсталирането, надграждането и премахването на софтуерните пакети.

Помощната програма rpm предоставя много полезни опции за заявки и проверка на пакети, както и за инсталиране, надграждане и премахване на пакети. По-долу са дадени примери за тези опции.

Пакети с заявки

1. Списък на всички инсталирани пакети

За да изброите всички инсталирани пакети, използвайте следната команда:



Форматът на имената на RPM пакети е name-version-release.architecture. Примерът показва пакетите за версия 7 на EnterPrise Linux (el7) с архитектури на:

x86\_64: Всички AMD64 или Intel 64 процесори

noarch: Всяка CPU архитектура

i686: 32-битова операционна система

2. Показване на информация за пакета

За да се покаже подробна информация за пакета (например на bash пакета), въведете:



3. Намерете пакета от файл

За да извършите обратно търсене, т.е. да определите към кой пакет принадлежи даден файл (например / etc / hosts), въведете:



4. Намерете конфигурационни файлове на пакета

За да изброите конфигурационни файлове, свързани с пакета (напр. Bash пакет), въведете:



Инсталиране и актуализиране на пакети

1. Инсталиране или надграждане на пакети

Използването на командата rpm -U package\_name обновява инсталираните пакети, както и инсталира

нови пакети. Например, за да инсталирате или надстроите пакета rsync:



Опцията -v (подробна) показва повече информация

Опцията -h (hash) показва напредъка.

2. Инсталиране на ново ядро

Когато инсталирате ново ядро, използвайте опцията –i, за да не надстроите текущото ядро, например:



Премахване на пакети

За да премахнете пакет (например пакетът rsync), въведете:

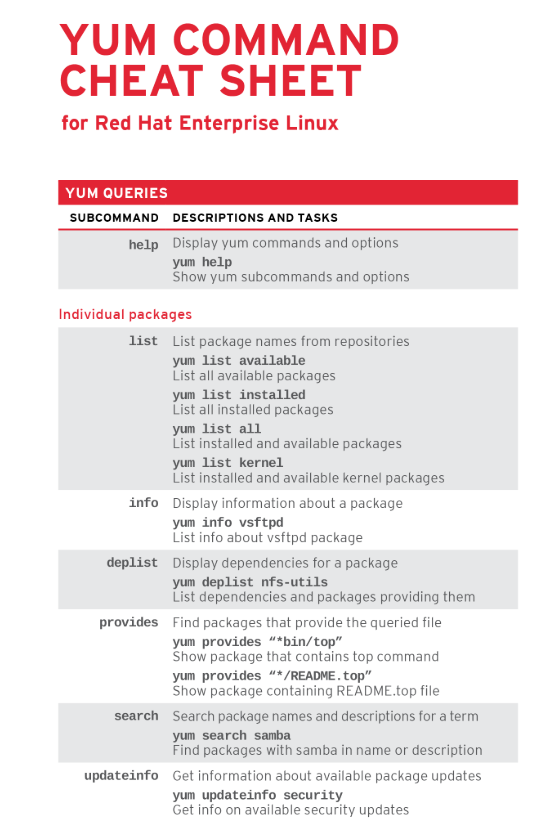


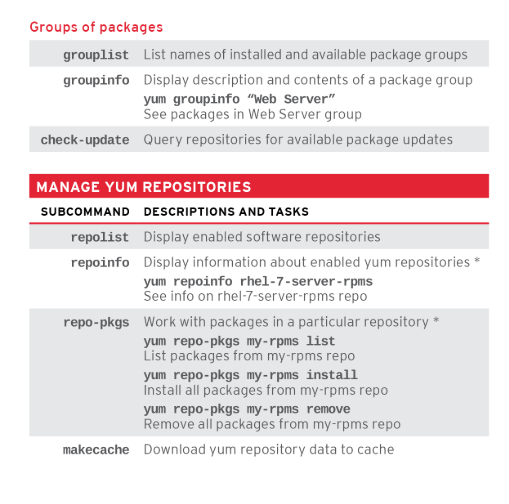
Какво е yum?

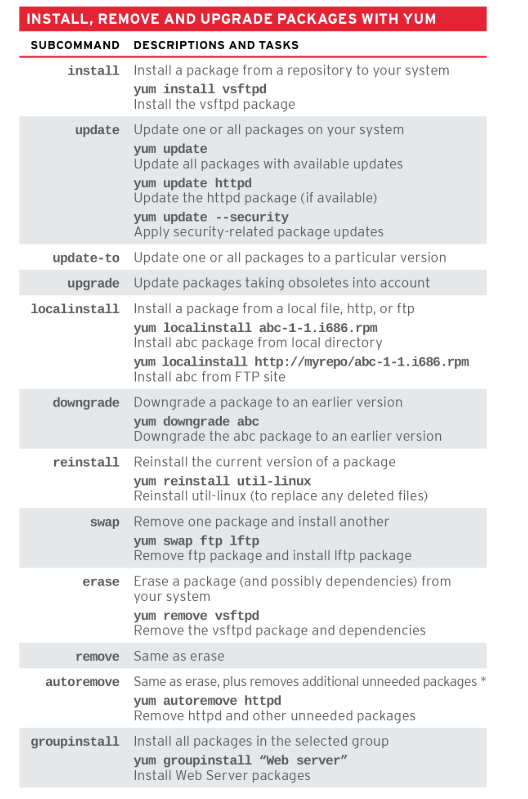
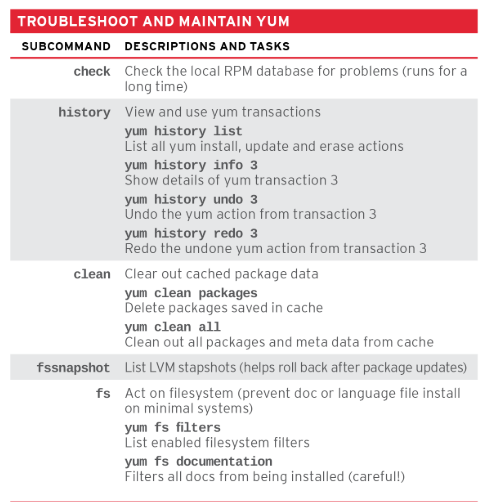
yum е основният инструмент за получаване, инсталиране, изтриване, запитване и управление на софтуерните пакети на Red Hat Enterprise Linux RPM от официалните репозиториуми на Red Hat, както и от други хранилища на трети страни. yum се използва в Red Hat Enterprise Linux версии 5 и по-нови. Версии на Red Hat Enterprise Linux 4 и по-рано използвани up2date.

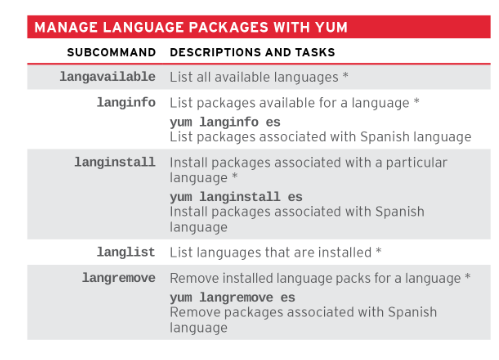
Yum Command Cheat Sheet за [Red Hat Enterprise Linux](https://access.redhat.com/articles/yum-cheat-sheet)

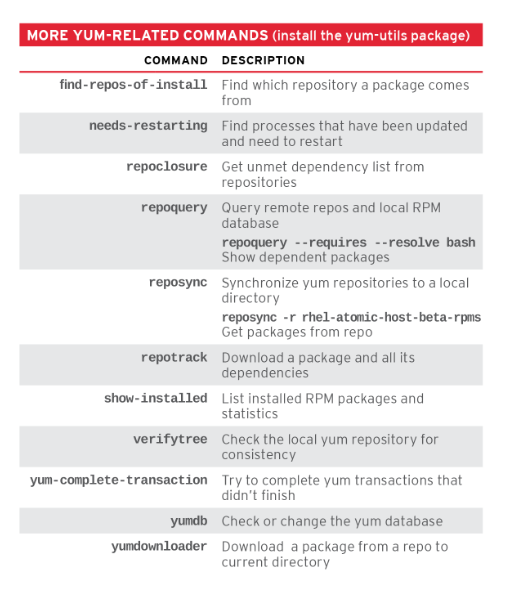
Командата yum е основният инструмент за получаване, инсталиране, изтриване, запитване и по друг начин управление на софтуерните пакети на Red Hat Enterprise Linux RPM от официалните софтуерни хранилища на Red Hat, както и от други хранилища на трети страни. Изтеглете приложената референтна карта и я използвайте като бърза справка за yum команди, опции, задачи и примерни командни линии.

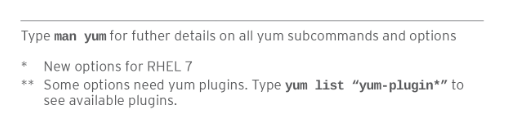






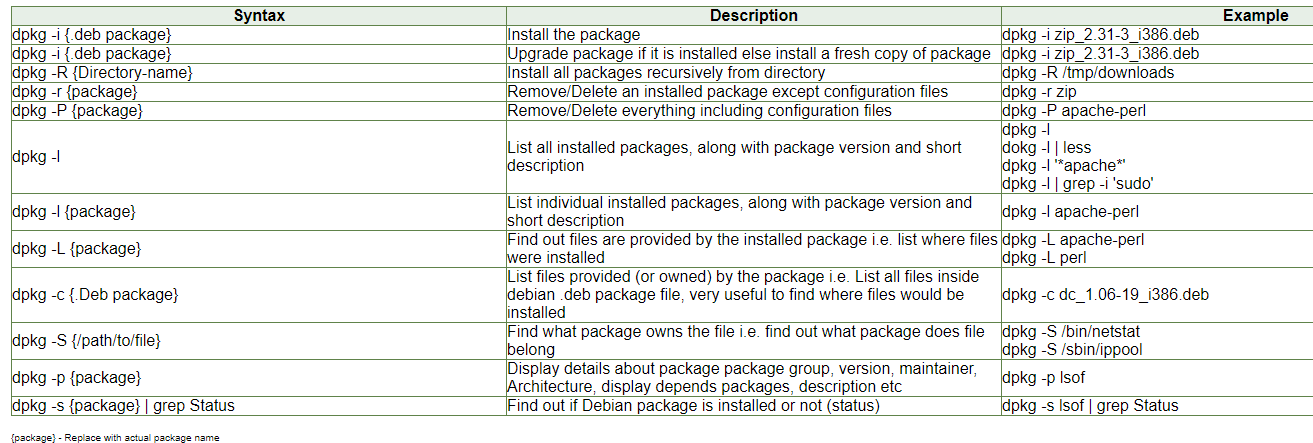






dpkg командния лист за Debian и Ubuntu Linux

dpkg е пакет мениджър за Debian, Ubuntu и много други дистрибуции на Linux. Използва се за инсталиране / управление на отделни пакети. Ето един бърз лист за измама, който ще ви бъде удобен, когато използвате dpkg в командния ред на shell:



Командата APT (подходящ)

Командата apt е мощен инструмент за командния ред, който работи с Ubuntu Advanced Packaging Tool (APT), изпълняващ такива функции като инсталиране на нови софтуерни пакети, надграждане на съществуващи софтуерни пакети, актуализиране на индекса на списъка с пакети и дори надграждане на цялата Ubuntu. система.

Като прост инструмент за командния ред, apt има много предимства пред другите инструменти за управление на пакети, налични в Ubuntu за сървърни администратори. Някои от тези предимства включват лекота на използване над прости терминални връзки (SSH) и възможността да бъдат използвани в системните административни скриптове, които от своя страна могат да бъдат автоматизирани от помощната програма за планиране на cron.

Някои примери за популярни приложения за помощната програма:

Инсталиране на пакет: Инсталирането на пакети с помощта на apt tool е доста просто. Например, за да инсталирате мрежовия скенер nmap, въведете следното:



Премахване на пакет: Премахването на пакета (или пакетите) също е лесно. За да премахнете инсталирания пакет в предишния пример, въведете следното:



Множество пакети: Можете да посочите няколко пакета, които да бъдат инсталирани или премахнати, разделени с интервали.

Също така, добавянето на опцията --purge към apt remove ще премахне и конфигурационните файлове на пакета. Това може или не може да бъде желания ефект, така че използвайте с повишено внимание.

Актуализиране на индекса на пакета: Индексът на APT пакет е по същество база данни от налични пакети от хранилищата, дефинирани в файла /etc/apt/sources.list и в директорията /etc/apt/sources.list.d. За да актуализирате индекса на местния пакет с последните промени, направени в хранилищата, въведете следното:



Пакети за надстройка: С течение на времето актуализираните версии на пакети, инсталирани понастоящем на компютъра ви, могат да станат достъпни от хранилищата на пакети (например актуализации на защитата). За да надстроите системата си, първо актуализирайте индекса на пакета си, както е посочено по-горе, и след това въведете:



Linux [aptitude](https://www.debian.org/doc/manuals/aptitude/rn01re01.en.html) command

aptitude е текстово базиран интерфейс за системата за управление на APT пакети. Тя позволява на потребителя да види списъка с пакети и да изпълнява задачи за управление на пакети, като инсталиране, надстройка и премахване на пакети. Действията могат да се изпълняват от визуален интерфейс или от командния ред.

Синтаксиса е :

