

Introduktion til Pacman



BigLinux

Forfatter:: Carl AndersenIntroduktion til Pacman
Version: 0.0.6
Dated: 14. March 2026
Licens: Dette dokument er licenseret under Non
Commercial licens

Ansvarsfraskrivelse

Oplysningerne i dette dokument er baseret på offentligt tilgængelig dokumentation og forfatterens personlige erfaring baseret på **Arch Linux, BigLinux, Manjaro og Mabox Linux Pacman**. Forfatteren er under ingen omstændigheder ansvarlig for nogen direkte, indirekte, følgeskader, strafbare, særlige eller tilfældige skader (herunder, uden begrænsning, erstatning for tab af fortjeneste, forretningsafbrydelse eller tab af information), der opstår som følge af brugen eller manglende evne til at bruge dette dokument, selvom forfatteren er blevet informeret om muligheden for sådanne skader. Forfatteren giver ingen erklæringer eller garantier med hensyn til nøjagtigheden eller fuldstændigheden af indholdet af dette dokument og forbeholder sig retten til at foretage ændringer i dette dokument til enhver tid uden varsel. Forfatteren forpligter sig ikke til at opdatere oplysningerne i dette dokument.

Titel	Introduktion til Pacman
Forfatter	Carl Andersen
Email	Se på hjemmesiden
Version	0.0.6
Document Date	14. Mar. 2026

Indholdsfortegnelse

1. Pacman Manual	side 4
2. Operationer	side 4
3. Muligheder	side 6
4. Transaktionsmuligheder	side 8
5. Opgraderingsmuligheder (GÆLDER FOR -S OG -U)	side 9
6. Forsørgelsesmuligheder (GÆLDER FOR -Q)	side 10
7. Fjern muligheder (GÆLDER FOR -R)	side 12
8. Synkronisering Muligheder (GÆLDER FOR -S)	side 13
9. Database Muligheder (GÆLDER TIL -D)	side 14
10. Filmuligheder (GÆLDER FOR -F)	side 15
11. Håndtering af config filer	side 16
12. Pakkehåndtering Arch Linux – Manjaro – Mabox Linux	side 17
13. Grundlæggende operationer	side 17
14. Forespørgsel på specifikke pakker	side 20
15. Forespørgsel på pakkens afhængigheder	side 20
16. Administration af installationskilder	side 21
17. Tilsidesætte	side 21
18. Verifikation og reparation	side 21
19. Søgning på pakker	side 23
20. Installation af pakker	side 24
21. Fjernelse af pakker	side 25
22. Visning og fjernelse af forældreløse pakker	side 26
23. Downloade pakker uden at installere dem	side 26
24. Bestem hvilke pakker der ejer en fil	side 26
25. Rensning af cachen	side 26
26. Konfigurationsfilen pacman.conf	side 27

1. Pacman Manual

Pacman er et pakkehåndteringsværktøj, der sporer installerede pakker på et Linux system. Den har afhængighedssupport, pakkegrupper, kan installere og afinstallere scripts, og muligheden for at synkronisere din lokale maskine med et fjernlager til automatisk at opgradere pakker. Pacman pakker er et zippet tar format.

Siden version 3.0.0 har pacman været front-end til **libalpm** (**man libalpm**), "Arch Linux Package Management" bibliotek. Dette bibliotek tillader alternative frontends, der skal skrives (for eksempel en GUI-frontend som også findes i dag).

At bruge pacman involverer at specificere en operation med ethvert potentiale muligheder og mål at operere efter. Et **mål** er normalt et pakkenavn, filnavn, URL eller en søgestreng. Mål kan leveres som kommando linjeargumenter. Derudover, hvis stdin ikke er fra en terminal og en enkelt bindestreg (-) sendes som et argument, mål vil blive læst fra stdin.

2. Operationer

-D, --database

Brug af pakkedatabasen. Denne operation giver dig mulighed for ændre visse attributter for de installerede pakker i pacman's database. Det giver dig også mulighed for at kontrollere databaserne for interne konsistens.

-Q, --query

Forespørg pakkedatabasen. Denne handling giver dig mulighed for at se installerede pakker og deres filer, samt meta-information om individuelle pakker (afhængigheder, konflikter, installationsdato, byggedato, størrelse). Dette kan køres mod den lokale pakke database eller kan bruges på individuelle pakkefiler. Først og fremmest tilfælde, hvis ingen pakkenavne er angivet på kommandolinjen, alle installerede pakker vil blive forespurgt. Derudover diverse filtre kan anvendes på pakkelisten.

-R, --remove

Fjern pakke(r) fra systemet. Grupper kan også angives til fjernes, i hvilket tilfælde hver pakke i den gruppe vil blive det fjernet. Filer, der tilhører den angivne pakke vil blive slettet, og databasen vil blive opdateret. De fleste konfigurationsfiler vil være gemt med en **.pacsave** udvidelse, medmindre **-nosave** indstillingen bruges. Se indstillinger .

-S, --sync

Synkroniser pakker. Pakker installeres direkte fra fjerndepoter, inklusive alle afhængigheder, der kræves for at køre pakker. For eksempel vil `pacman -S qt` downloade og installere qt og alle de pakker det afhænger af. Hvis der findes et pakkenavn i mere end ét depot, kan depotet eksplicit specificeret for at tydeliggøre pakken, der skal installeres: `pacman -S testing/qt`. Du kan også angive versionskrav: `pacman -S "bash>=3.2"`. Citater er nødvendige, ellers fortolker shell ">" som omdirigering til en fil.

Ud over pakker kan grupper også angives. Til for eksempel, hvis gnome er en defineret pakkegruppe, så `pacman -S gnome` vil give en prompt, der giver dig mulighed for at vælge, hvilke pakker du vil installere fra en nummereret liste. Pakkevalget er specificeret ved hjælp af en mellemrums og/eller kommasepareret liste over pakkenumre. Sekventielle pakker kan vælges ved at angive den første og sidste pakkenumre adskilt af en bindestreg (-). Eksklusiv pakker opnås ved at sætte et indryk foran et tal eller række af tal (^).

Pakker, der leverer andre pakker, håndteres også. For eksempel, `pacman -S foo` vil først lede efter en foo pakke. Hvis foo ikke er fundet, vil pakker, der giver den samme funktionalitet som foo være som du søgte efter. Hvis der findes en pakke, vil den blive installeret. En valgprompt fremkommer, hvis flere pakker fremkommer på pakken foo.

Du kan også bruge `pacman -Su` til at opgradere alle pakker, der er ude-af-dato. Se Synkroniseringsindstillinger nedenfor. Ved opgradering, vil `pacman` udfører versionsammenligning for at bestemme, hvilke pakker der skal bruge opgradering. Denne adfærd fungerer som følger:

Alfanumerisk: $1.0a < 1.0b < 1.0beta < 1.0p < 1.0pre < 1.0rc < 1.0 < 1.0.a < 1.0.1$

Numerisk: $1 < 1,0 < 1,1 < 1,1,1 < 1,2 < 2,0 < 3,0,0$

Derudover kan versionsstrengene have en **epokeværdi** defineret vil tilsidesætte enhver versionsammenligning, medmindre epokeværdierne er det lige. Dette er angivet i et `epoch:version-rel` format. Til f.eks. er `2:1.0-1` altid større end `1:3.6-1`.

-T, --deptest

Tjek afhængigheder; dette er nyttigt i scripts såsom `makepkg` to tjek installerede pakker. Denne operation vil kontrollere hver afhængighed angivet og returnere en liste over afhængigheder, der ikke er i øjeblikket tilfreds med systemet. Denne operation accepterer ingen andre muligheder. Eksempel på brug: `pacman -T qt "bash>=3.2"`.

-U, --upgrade

Opgrader eller tilføj pakke(r) til systemet og installer de nødvendige afhængigheder fra synkroniseringslagre. Enten en URL eller filsti kan være specificeret. Dette er en "fjern-og-tilføj"-proces. Se Opgradering Valgmuligheder nedenfor; se også Håndtering af konfigurationsfiler for en forklaring på hvordan pacman tager sig af konfigurationsfiler.

-F, --files

Forespørg filedatabasen. Denne operation giver dig mulighed for at lede efter pakker, der ejer bestemte filer eller viser filer, der ejes af visse pakker. Det er kun pakker, der er en del af dine synkroniseringsdatabaser søgte.

-V, --version

Vis version og afslut.

-h, --help

Vis syntaks for den givne operation. Hvis ingen operation blev leveret, så vises den generelle syntaks.

3. Muligheder

-b, --dbpath <sti>

Angiv en alternativ databaseplacering (standarden er /var/lib/pacman). Dette bør ikke bruges, medmindre du ved, hvad du laver. **BEMÆRK:** Hvis det er angivet, er dette en absolut sti, og rodstien er ikke automatisk sat foran.

-r, --root <sti>

Angiv en alternativ installations root (standard er /). Dette bør ikke bruges som en måde at installere software i /usr/local i stedet for /usr. **BEMÆRK:** Hvis databasestien eller logfilen ikke er det angivet på enten kommandolinjen eller i pacman.conf (man pacman.conf), deres standardplaceringen vil være inden for denne root sti. **BEMÆRK:** Denne mulighed er ikke egnet til at udføre operationer på en monteret gæst system. Se i stedet **--sysroot**.

-v, --verbose

Outputstier såsom root konf.-fil, DB-sti, cache-dirs osv.

--arch <arch>

Angiv en alternativ arkitektur.

--cachedir <dir>

Angiv en alternativ pakkecache placering (standarden er /var/cache/pacman/pkg). Flere cache-mapper kan være specificeret, og de prøves i den rækkefølge, de videregives til pacman. **BEMÆRK:** Dette er en absolut sti, og root stien er ikke automatisk foranstillet.

--color <når>

Angiv, hvornår farvning skal aktiveres. Gyldige muligheder er , **always**, **never**, eller **auto**. tvinger **always** farver på; tvinger **never** farver på; og **auto** aktiverer kun automatisk farver, når der udskrives på en tty.

--config <fil>

Angiv en alternativ konfigurationsfil.

--debug

Vis fejlretningsmeddelelser. Når du rapporterer fejl, anbefales denne mulighed at blive brugt.

--gpgdir <dir>

Angiv en mappe med filer, der bruges af GnuPG til at bekræfte pakke signaturer (standarden er /etc/pacman.d/gnupg). Denne mappe skal indeholde to filer: pubring.gpg og trustdb.gpg. pubring.gpg besidder de offentlige nøgler for alle pakkere. trustdb.gpg indeholder en såkaldt tillidsdatabase, som specificerer, at nøglerne er autentisk og pålidelig. **BEMÆRK:** Dette er en absolut vej, og root stien er ikke automatisk sat foran.

--hookdir <dir>

Angiv en alternativ mappe, der indeholder hook filer (standard er /etc/pacman.d/hooks). Der kan angives flere hook mapper med kroge i senere mapper, der har forrang over hooks i tidligere mapper. **BEMÆRK:** Dette er en absolut sti, og root stien er ikke automatisk indsat.

--logfile <fil>

Angiv en alternativ logfil. Dette er en absolut vej, uanset af installationens root indstilling.

--noconfirm

Omgå alle "Are you sure?" beskeder. Det er ikke en god idé at gøre dette, medmindre du vil køre pacman fra et script.

--confirm

Annulerer virkningerne af en tidligere **--noconfirm**.

--disable-download-timeout

Deaktiver standardindstillinger for lav hastighedsgrænse og timeout ved downloads. Brug dette, hvis du har problemer med at downloade filer med proxy og/eller sikkerhedsgateway.

--sysroot <dir>

Angiv en alternativ system root. Pacman vil chroot og chdir ind i system root, inden du kører. Dette tillader monteret gæst systemer, der skal betjenes korrekt. Alle andre veje, der gives, vil være tolkes som i forhold til system root. Kræver root privilegier.

4. Transaktionsmuligheder (GÆLDER FOR -S, -R OG -U)

d, --nodeps

Springer kontrol af afhængighedsversion over. Pakkenavne er stadig kontrolleret. Normalt vil pacman altid tjekke en pakkes afhængighedsfelter for at sikre, at alle afhængigheder er installeret, og at der ikke er nogen pakke konflikter i systemet. Angiv denne indstilling to gange for at springe over alle afhængighedstjek.

--assume-installed <pakke=version>

Tilføj en virtuel pakke "pakke" med version "version" til transaktion for at tilfredsstille afhængigheder. Dette gør det muligt at deaktivere specifikke afhængighedstjek uden at påvirke alle afhængighedstjek. For at deaktivere al afhængighedskontrol, se **--nodeps** indstillingen.

--dbonly

Tilføjer/fjerner kun databaseposten, og efterlader alle filer på plads.

--noprogessbar

Vis ikke en statuslinje, når du downloader filer. Dette kan være nyttigt til scripts, der kalder pacman og fanger outputtet.

--noscriptlet

Hvis der findes en installationsscriptlet, skal du ikke udføre den. Brug ikke dette medmindre du ved hvad du laver.

-p, --print

Udskriv kun målene i stedet for at udføre den faktiske operation (synkronisere, fjerne eller opgradere). Brug **--print-format** til at angive hvordan mål vises. Standardformatstrengen er "%l", hvilket viser URL'er med **-S**, filnavne med **-U** og pkgname-pkgver med **-R**.

--print-format <format>

Angiv et printf-lignende format for at kontrollere outputtet af **--print** operation. De mulige attributter er: "%n" for pakkenavn, "%v" for pkgver, "%l" for placering, "%r" for repository og "%s" for størrelse. Indebærer **--print**.

5. Opgraderingsmuligheder (GÆLDER FOR -S OG -U)

-w, --downloadonly

Hent alle pakker fra serveren, men installer/opgrader ikke hvad som helst.

--asdeps

Installer pakker ikke-eksplicit; med andre ord, fake deres installation til at blive installeret som en afhængighed. Dette er nyttigt til makepkg og andre bygge-fragment-kildeværktøjer, der skal installere afhængigheder før du bygger pakken.

--asexplicit

Installer pakker eksplicit; med andre ord, fake deres installation til at være eksplicit installeret. Dette er nyttigt, hvis du vil markere en afhængighed som eksplicit installeret, så den ikke fjernes ved --rekursiv fjernoperation.

--ignore <pakke>

Instruerer pacman til at ignorere opgraderinger af pakken, selvom der er en ledig. Flere pakker kan specificeres ved at adskille dem med et komma.

--ignoregroup <gruppe>

Instruerer pacman til at ignorere opgraderinger af alle pakker i **gruppen**, selvom der er en ledig. Flere grupper kan specificeres ved adskille dem med et komma.

--needed

Geninstaller ikke de mål, der allerede er opdaterede.

--overwrite <glob>

Omgå filkonfliktkontrol og overskriv modstridende filer. Hvis pakke, der er ved at blive installeret, indeholder filer, der er allerede installeret og matcher **glob**, vil denne mulighed forårsage alle disse filer, der skal overskrives. Brug af **--overwrite** vil ikke tillade overskrive en mappe med en fil eller installere pakker med modstridende filer og mapper. Flere mønstre kan være angivet ved at adskille dem med et komma. Kan specificeres flere gange. Mønstre kan negeres, således at filer matcher dem vil ikke blive overskrevet ved at sætte et udråbstegn foran dem. Efterfølgende matcher vil tilsidesætte tidligere matcher. Et førende bogstavligt udråbstegn eller omvendt skråstreg skal undgås.

6. Forsørgelsesmuligheder (GÆLDER FOR -Q)

-c, --changelog

Se ændringsloggen for en pakke, hvis den findes.

-d, --deps

Begræns eller filtrer output til pakker installeret som afhængigheder. Denne mulighed kan kombineres med **-t** for at angive ægte forældrelose pakker, der blev installeret som afhængigheder, men som ikke længere er det kræves af enhver installeret pakke.

-e, --explicit

Begræns eller filtrer output til eksplicit installerede pakker. Dette mulighed kan kombineres med **-t** for at liste eksplicit installeret pakker, der ikke kræves af nogen anden pakke.

-g, --groups

Vis alle pakker, der er medlemmer af en navngivet gruppe. Hvis et navn er ikke angivet, skal du angive alle grupperede pakker.

-i, --info

Vis information om en given pakke. Muligheden **-p** kan bruges hvis du forespørger efter en pakkefil i stedet for den lokale database. Bestået to **--info** eller **-i** flag vil også vise listen over backupfiler og deres modifikationstilstande.

-k, --check

Kontroller, at alle filer, der ejes af den eller de givne pakker, er til stede på systemet. Hvis pakker ikke er angivet, eller filterflag ikke er forudsat, skal du kontrollere alle installerede pakker. Angivelse af denne mulighed to gange vil udføre mere detaljeret filkontrol (inklusive tilladelser, filstørrelser og ændringstider) for pakker, der indeholder den nødvendige mtree fil.

-l, --list

Liste over alle filer, der ejes af en given pakke. Flere pakker kan være angivet på kommandolinjen.

-m, --foreign

Begræns eller filtrer output til pakker, der ikke blev fundet i synkronisere database(r). Disse er typisk pakker, der blev downloadet manuelt og installeret med **–upgrade**.

-n, --native

Begræns eller filtrer output til pakker, der findes i synkroniseringen database(r). Dette er det omvendte filter af **--foreign**.

-o, --owns <fil>

Søg efter pakker, der ejer den eller de angivne fil(er). Stien kan være relativ eller absolut, og en eller flere filer kan angives.

-p, --file

Angiver, at pakken, der leveres på kommandolinjen, er en fil og ikke en post i databasen. Filen vil blive dekomprimeret og forespurgt. Dette er nyttigt i kombination med **--info** og **--list**.

-q, --quiet

Vis færre oplysninger for visse forespørgselshandlinger. Dette er nyttigt når pacmans output behandles i et script. Søgning vil kun vis pakkenavne og ikke version, gruppe og beskrivelse Information; owns vil kun vise pakkenavne i stedet for "file is owns of pkg" beskeder; gruppen vil kun vise pakkenavne og udelade gruppenavne; listen vil kun vise filer og udelade pakkenavne; check vil kun vise par af pakkenavne og manglende filer; -en forespørgsel vil kun vise pakkenavne i stedet for navne og versioner.

-s, --search <regexp>

Søg efter navne eller beskrivelser i hver lokalt installeret pakke der matcher regexp. Kun når du inkluderer flere søgetermer pakker med beskrivelser, der matcher ALLE disse termer.

-t, --unrequired

Begræns eller filtrer output til kun at udskrive pakker, hverken påkrævet heller ikke valgfrit påkrævet af nogen aktuelt installeret pakke. Angiv denne mulighed to gange for at inkludere pakker, som er valgfrit, men ikke direkte, krævet af en anden pakke.

-u, --upgrades

Begræns eller filtrer output til pakker, der er forældede på lokale system. Kun pakkeversioner bruges til at finde forældede pakker; udskiftninger er ikke markeret her. Denne mulighed fungerer bedst hvis synkroniseringsdatabasen opdateres med **-Sy**.

7. Fjern muligheder (GÆLDER FOR -R)

-c, --cascade

Fjern alle målpakker, såvel som alle pakker, der er afhængige af en eller flere målpakker. Denne operation er rekursiv og skal bruges med forsigtighed, da det kan fjerne mange potentielt nødvendige pakker.

-n, --nosave

Instruerer pacman til at ignorere betegnelser for sikkerhedskopiering af filer. Normalt hvornår en fil fjernes fra systemet, så kontrolleres databasen for at se hvis filen skal omdøbes med filtypenavnet **.pacsave**.

-s, --recursive

Fjern hvert specificeret mål inklusive alle deres afhængigheder, forudsat at (A) de ikke kræves af andre pakker; og (B) de blev ikke eksplicit installeret af brugeren. Denne operation er rekursiv og analog med en baglæns **--sync**, og det hjælper med at holde et rent system uden forældreløse pakker. Hvis du vil undlade betingelse (B), GENTAG denne mulighed to gange.

-u, --unneeded

Fjerner mål, der ikke kræves af andre pakker. Dette er mest nyttigt, når du fjerner en gruppe uden at bruge **-c** indstillingen, for at undgå at bryde eventuelle afhængigheder.

8. Synkronisering Muligheder (GÆLDER FOR -S)

-c, --clean

Fjern også pakker, der ikke længere er installeret, fra cachen som i øjeblikket ubrugte synkroniseringsdatabaser for at frigøre diskplads. Når pacman downloader pakker, det gemmer dem i en cache-mappe. Derudover gemmes databaser for hver synkroniserings database, du downloader og slettes ikke, selvom de fjernes fra konfigurationen filen **pacman.conf** (man **pacman.conf**). Brug **-clean** til kun at fjerne pakker der ikke længere er installeret; brug **to** til at fjerne alle filer fra cache. I begge tilfælde vil du have en ja eller nej mulighed for at fjerne pakker og/eller ubrugte downloadede databaser.

Hvis du bruger en netværksdelt cache, se indstillingen **CleanMethod** i **pacman.conf** (man **pacman.conf**)

-g, --groups

Vis alle medlemmerne for hver specificeret pakkegruppe. Hvis nej gruppenavne er angivet, vil alle grupper blive vist; brug flaget **to** gange for at se alle grupper og deres medlemmer.

-i, --info

Vis information om en given synkroniseringsdatabasepakke. Indtast **to --info** eller **-i** flag for at vise disse pakker i alle mål akiverne, der afhænger af denne pakke.

-l, --list

Liste alle pakker i de angivne lagre. Mange rarkiver kan angives på kommandolinjen.

-q, --quiet

Vis færre oplysninger for visse synkroniseringshandlinger. Dette er nyttigt når pacmans output behandles i et script. Søgning vil kun vis pakkenavne og ikke lager, version, gruppe og beskrivelse information; listen vil kun vise pakkenavne og udelade databaser og versioner; gruppe vil kun vise pakkenavne og udelade gruppenavne.

-s, --search <regexp>

Dette vil søge i hver pakke i synkroniseringsdatabaserne efter navne eller beskrivelser, der matcher regexp. Når du inkluderer flere søgninger, vil kun pakker med beskrivelser, der matcher ALLE disse vilkår blive returneret.

-u, --sysupgrade

Opgraderer alle pakker, der er forældede. Hver aktuelt installeret pakke vil blive undersøgt og opgraderet, hvis en nyere pakke findes. En rapport over alle pakker, der skal opgraderes, vil være præsenteret, og handlingen vil ikke fortsætte uden bruger bekræftelse. Afhængigheder løses automatisk på dette niveau og vil blive installeret/opgraderet om nødvendigt.

Brug denne mulighed to gange for at aktivere pakkenedgraderinger; I dette tilfælde, vil pacman vælge synkroniseringspakker, hvis versioner ikke matcher de lokale versioner. Dette kan være nyttigt, når brugeren skifter fra et testlager til et stabilt.

Yderligere mål kan også specificeres manuelt, så **-Su foo** vil lave en systemopgradering og installere/opgradere "foo"-pakken i samme operation.

-y, --refresh

Download en ny kopi af masterpakkedatabasen fra server(e) defineret i **pacman.conf** (man pacman.conf). Dette skal typisk bruges hver gang du bruger **--sysupgrade** eller **-u**. Ved brug af to **--refresh** eller **-y** flag vil tvinge en opdatering af alle pakkedatabaser, selvom de synes at være opdateret.

9. Database Muligheder (GÆLDER TIL -D)

--asdeps <pakke>

Marker en pakke som ikke eksplicit installeret; med andre ord sæt deres installationsgrund til at blive installeret som en afhængighed.

--asexplicit <pakke>

Marker en pakke som eksplicit installeret; med andre ord, sæt deres installationsgrund til at være eksplicit installeret. Dette er nyttigt, hvis du ønsker at beholde en pakke installeret, selv da den var oprindeligt installeret som en afhængighed af en anden pakke.

-k, --check

Tjek, at den lokale pakke database er internt konsistent. Dette vil kontrollere, at alle nødvendige filer er til stede, og at de er installerede pakker har de nødvendige afhængigheder, ikke konflikter og ikke flere pakker ejer ikke den samme fil. Gentagelse af denne mulighed to gange vil udføre en kontrol af synkroniseringsdatabaserne for at sikre alle specificerede afhængigheder er tilgængelige.

q, --quiet

Undertryk meddelelser ved vellykket gennemførelse af databaseoperationer.

10. Filmuligheder (GÆLDER FOR -F)

-y, --refresh

Download friske pakke databaser fra serveren. Brug to gange til tvinge en opdatering, selvom databaser er opdaterede.

-l, --list

List de filer, der ejes af den forespurgt pakke.

-x, --regex

Fortolk hver forespørgsel som et regulært udtryk.

-q, --quiet

Vis mindre information for visse filhandlinger. Dette er nyttigt når pacmans output behandles i et script, kan du dog evt ønsker at bruge **--machinereadable** i stedet for.

--machinereadable

Udskriv hver kamp i et maskinlæsbart outputformat. Formatet er **repository\0pkgname\0pkgver\0path\n** med **\0** som NULL tegn og **\n** for en linjefeed.

11. Håndtering af config filer

Pacman bruger den samme logik som rpm til at bestemme handling mod filer der er udpeget til at blive sikkerhedskopieret. Under en opgradering, tre MD5 hashes bruges til hver backup fil for at bestemme den nødvendige handling: en for den originale installerede fil, en for den nye fil, der handler om skal installeres, og en til den faktiske fil, der findes på fil system. Efter at have sammenlignet disse tre hashes, kan følgende scenarier resultat:

original=X, nuværende=X, ny=X

Alle tre filer er ens, så overskrivning er ikke et problem. Installer den nye fil.

original=X, nuværende=X, ny=Y

Den aktuelle fil er den samme som den originale, men den nye adskiller sig. Da brugeren aldrig ændrede filen, og den nye en kan indeholde forbedringer eller fejlrettelser, installerer den nye fil.

original=X, nuværende=Y, ny=X

Begge pakkeversioner indeholder nøjagtig den samme fil, men den ene på filsystemet er blevet ændret. Lad den aktuelle fil være på plads.

original=X, nuværende=Y, ny=Y

Den nye fil er identisk med den nuværende fil. Installer den nye fil.

original=X, nuværende=Y, ny=Z

Alle tre filer er forskellige, så installer den nye fil med en **.pacnew** udvidelsen og advare brugeren. Brugeren skal derefter manuelt flette eventuelle nødvendige ændringer ind i den originale fil.

original=NULL, nuværende=Y, ny=Z

Pakken var ikke tidligere installeret, og filen allerede findes på filsystemet. Installer den nye fil med en **.pacnew** udvidelse og advare brugeren. Brugeren skal så manuelt flette evt nødvendige ændringer i den originale fil.

12. Pakkehåndtering Arch Linux – Manjaro – Mabox Linux

13. Grundlæggende operationer

Installer en eller flere pakker efter navn

```
pacman -S
```

Søg efter pakker ved at søge udtrykket i navn, beskrivelse, kort beskrivelse. Hvilke nøjagtige felter, der søges som standard, varierer i hvert værktøj.

```
pacman -Ss
```

Opgrader pakker - nstaller pakker, der allerede har en ældre version installeret

```
pacman -Syu
```

Opgrader pakker – En anden form for opdateringskommandoen, som kan udføre mere komplekse opdateringer

- f.eks. Distributionsopgraderin ger. Når den sædvanlige opdateringskommando vil udelade pakkeopdateringer. som inkluderer ændringer i afhængigheder. Det downloader en ny kopi af masterpakke-databasen fra server(e). Dette skal typisk bruges hver gang du bruger systemopgradering. Det vil tvinge en opdatering af alle pakke-databaser, selvom de synes at være opdateret.

```
pacman -Syyu
```

Ryd op i alle lokale caches. Valgmuligheder kan begrænse, hvad der rent faktisk rengøres.

```
pacman -Sc
```

Eller

pacman -Scc

Fjern afhængigheder, der ikke længere er nødvendige, fordi f.eks. pakken, der havde brug for afhængighederne, blev fjernet.

pacman -Qdtq | pacman -Rs

Fjern pakker, der ikke længere er inkluderet i nogen arkiver.

pacman -Qmq | pacman -Rs

Marker en pakke, der tidligere er installeret, asexplicit som en afhængighed som udtrykkeligt påkrævet.

pacman -D - -asexplicit

Installer pakker som afhængighed / uden markering som udtrykkeligt påkrævet.

pacman -S --asdeps

Downloader kun den eller de givne pakkeruden at pakke dem ud eller installere dem.

pacman -Sw

Vis en log over handlinger, som softwareadministrationen har foretaget.

Læs /var/log/pacman.log

Få en dump af hele systemoplysningerne - Udskriver, gemmer eller lignende den aktuelle tilstand i

pakkehåndteringssystemet. Foretrukket output er tekst eller XML. (Bemærk: Hvorforen- eller her?

Intet værktøj giver mulighed for at vælge outputformat.)

Se `/var/lib/pacman/local`

14. Forespørgsel på specifikke pakker

Søg efter pakker ved at søge udtrykket i navn.

`pacman -Ss`

Liste pakker med en tilgængelig opdatering.

`pacman -Qu`

Vis en liste over alle pakker i alle installationskilder, der håndteres af pakkehåndteringen.

`pacman -Sl`

Genererer en liste over installerede pakker sammen med version nummer.

`pacman -Q`

Liste over pakker, der er installeret, men ikke er tilgængelige i nogen installationskilde (længere).

`pacman -Qm`

Søg lokalt installeret pakke efter navne eller beskrivelser.

pacman -Qs

Liste over pakker, der ikke kræves af nogen anden pakke.

pacman -Qtt

Liste pakker installeret eksplicit (ikke som afhængigheder).

pacman -Qe

Liste pakker installeret automatisk (somafhængigheder).

pacman -Qd

15. Forespørgsel på pakkens afhængigheder

Vis pakker, der kræver, at X skal installeres, også kendt som at vise omvendte afhængigheder.

pacman -Sii

Angiv alle pakker, der er nødvendige for den givne pakke, også kendt pacman **-Si** eller som afhængigheder.

pacman -Si

Eller

pacman -Qi

16. Administration af installationskilder

Administration af installationskilder.

Rediger /etc/pacman.conf

Udskriv en liste over alle installationskilder, herunder vigtige oplysninger som URI, alias osv.

cat /etc/pacman.d/mirrorlist

Liste over alle pakker fra et bestemt repo.

paclist <repo>

17. Tilsidesætte

Tilføj en pakkelåsregel for at forhindre, at dens nuværende tilstand ændres.

Rediger /etc/pacman.conf

ændring af

IgnorePkg array

Slet en pakkelåsregel.

Rediger /etc/pacman.conf

fjernelse af pakke fra IgnorePkg linie

Vis en liste over alle låseregler.

cat /etc/pacman.conf

Angiv prioriteten for den givne pakke for at undgå opgradering, tvinge nedgradering eller overskrive enhver standardadfærd. Kan også bruges til at foretrække en pakkeversion fra en bestemt installationskilde.

Rediger /etc/pacman.conf

ændring af

HoldPkg

og/eller

IgnorePkg arrays

18. Verifikation og reparation

Verificer alle pakker.

pacman -Qk

Kan tilføjes endnu en k for at få yderligere output.

pacman -Qkk

Geninstaller den givne pakke; dette vil geninstallere den givne pakke uden afhængighedsbesvær.

pacman -S

Kontroller afhængigheder af det komplette system; bruges, hvis installationsprocessen blev kraftigt ødelagt

pacman -Dk

Brug noget magi til at løse ødelagte afhængigheder i et system.

For pacman -afhængighedsniveau, brug:

```
pacman -Dk
```

For delt biblioteksniveau, brug:

```
findbrokenpkgsAUR
```

Eller

```
lddd
```

(fra devtools)

19. Søgning på pakker

For at søge i Manjaro og Mabox arkiverne efter tilgængelige pakker kan du bruge kommandoen **pacman -Ss nøgleord**. Det vil søge både i pakkenavnet og beskrivelsen efter søgeordet. For at søge efter pakker, der indeholder søgeordet smplayer, kan du f.eks. bruge:

```
pacman -Ss smplayer
```

Når du har fundet en pakke, kan du bruge **pacman -Qi** til at få mere information om en installeret pakke

```
pacman -Qi smplayer
```

Til sidst, for at få en liste over alle installerede pakker på dit system, skal du indtaste følgende kommando:

```
pacman -Qi smplayer
```

20. Installation af pakker

ADVARSEL:

Installer aldrig en pakke uden først at opdatere systemet. På en rullende udgivelse kan dette føre til et system, der ikke kan startes

For at installere en softwarepakke er den grundlæggende syntaks **pacman -S** pakkenavn. Men at installere en pakke uden at opdatere systemet vil føre til en delvis opgradering situation, så alle eksemplerne her vil bruge pacman **-Syu** pakkenavn, som vil installere pakken og sikre, at systemet er opdateret. For at installere smplayer er kommandoen for eksempel:

```
sudo pacman -Syu smplayer
```

Du vil derefter blive præsenteret for en liste over software, der skal installeres. Du bemærker muligvis, at denne liste har flere pakker, end du anmodede om. Dette skyldes, at mange pakker også har afhængigheder, som er pakker, der skal installeres, for at den software, du valgte, kan fungere korrekt.

Pacman kan også installere pakker direkte fra det lokale system eller en placering på internettet. Formatet på den kommando er pacman **-U** pakkeplacering. For at installere en kopi af din pakkecache kan du for eksempel gøre noget som:

```
sudo pacman -U /var/cache/pacman/smplayer-22.2.0-1-x86_64.pkg.tar.zst
```

Alternativt kan du få det direkte fra et af Manjaros spejle:

```
sudo pacman -U https://mirrors.dotsrc.org/manjaro/stable/community/x86_64/smplayer-22.2.0-1-x86_64.pkg.tar.zst
```

ADVARSEL:

Når du bruger **pacman -U**, er det op til dig at sikre, at den pakke, du installerer, er fuldt ud kompatibel med dit system.

21. Fjernelse af pakker

ADVARSEL:

Gennemgå altid pakkelisten, før du bekræfter, når du fjerner pakker. Hvis du ikke er forsigtig, kan du nemt fjerne hele dit skrivebord på grund af afhængigheder.

For at fjerne en softwarepakke er den grundlæggende syntaks `sudo pacman -R` pakkenavn. Vi kunne fjerne `smplayer` pakken med:

```
sudo pacman -R smplayer
```

Dette vil fjerne pakken, men vil efterlade alle afhængigheder. Hvis du også vil fjerne de unødvendige afhængigheder, kan du bruge `pacman -Rsu` pakkenavn som vist i dette eksempel:

```
sudo pacman -Rsu smplayer
```

Den mest nukleare mulighed er `pacman -Rcs` pakkenavn. Dette vil fjerne alt, der afhænger af pakkenavn og fortsætte med at gøre det på dets afhængigheder. Dette bør egentlig kun bruges under ekstraordinære omstændigheder, såsom når du fjerner et helt skrivebordsmiljø og prøver ikke at efterlade noget.

Pacman opretter normalt også backup konfigurationsfiler, når pakker slettes. For at fjerne dem kan du tilføje `n` til et hvilket som helst af eksemplerne ovenfor. For eksempel:

```
sudo pacman -Rn smplayer
```

```
sudo pacman -Rsun smplayer
```

```
sudo pacman -Rcn smplayer
```

22. Visning og fjernelse af forældreløse pakker

Liste alle forældreløse, installerede pakker, der ikke bruges af andet, og som ikke længere burde være nødvendige:

```
pacman -Qdt
```

Sådan fjerner du alle forældreløse pakker:

```
sudo pacman -Rs $(pacman -Qdtq)
```

23. Downloade pakker uden at installere dem

I nogle tilfælde kan det være nyttigt at downloade en pakke uden at installere. For eksempel at installere på et andet system, der ikke er forbundet til internettet. Dette kan gøres med **pacman -Sw** pakkenavn. For eksempel:

```
sudo pacman -Sw smplayer
```

Pakken og eventuelle påkrævede afhængigheder vil blive downloadet til din pacman cache til [/var/cache/pacman/pkg](#)

24. Bestem hvilke pakker der ejer en fil

Det er ofte nyttigt at forstå, hvilken pakke der installerede en fil på dit system. Dette er nemt at gøre med pacman ved at bruge **pacman -Qo /sti/til/filnavn**. For eksempel:

```
pacman -Qo /usr/bin/smplayer
```

25. Rensning af cachen

Når pacman installerer pakker, gemmer den en kopi af alle de gamle pakker, du har downloadet. Denne cache kan være meget nyttig, hvis du skal installere ældre pakker i en nødsituation. Men uden markering vil denne cache vokse sig meget stor over tid. Systemer, der kører [Pacman](#), vil allerede have adgang til dens automatiske pacman cache rensningsfunktioner. Det er også muligt at rense dem manuelt ved hjælp af pacman.

For at rydde cachen for pakker, der ikke længere er installeret, skal du indtaste følgende kommando:

```
sudo pacman -Sc
```

Ellers skal du indtaste følgende kommando for at rydde cachen fuldstændigt (brug den med forsigtighed):

```
sudo pacman -Scc
```

En mere sikker måde at fjerne gamle pakkecache filer på er at fjerne alle pakker undtagen de seneste tre pakkeversioner ved hjælp af **paccache**:

```
paccache -rvk3
```

26. Konfigurationsfilen pacman.conf

Pacmans indstillinger er placeret i **/etc/pacman.conf**. Denne fil ejes af **root**. Se venligst denne [vejledning](#), hvis du har brug for mere information om, hvordan du redigerer denne fil. En fuldstændig reference til disse muligheder kan findes i [Arch Wiki](#).

Bemærk **pacman.conf** indstillingerne skelner mellem store og små bogstaver.

I det hele taget er [Arch Wiki](#) en god vejledning. Arch Linux overholder KISS-princippet ("Keep It Simple, Stupid")

Linu

Linux Philosophy

FSF

"On a UNIX/Linux system, everything is a file; if something is not a file, it is a process."

Utilities used for system administration (and other root-only commands) are stored in `/sbin`, `/usr/sbin`, and `/usr/local/sbin`. `/sbin` contains binaries essential for booting, restoring, recovering, and/or repairing the system in addition to the binaries in `/bin`. Programs executed after `/usr` is known to be mounted (when there are no problems) are generally placed into `/usr/sbin`. Locally-installed system administration programs should be placed into `/usr/local/sbin`.

`/var` contains variable data files. This includes spool directories and files, administrative and logging data, and transient and temporary files. Some portions of `/var` are not shareable between different systems. For instance, `/var/log`, `/var/lock`, and `/var/run`. Other portions may be shared, notably `/var/mail`,

`/var/cache/man`, `/var/cache/fonts`, and `/var/spool/news`. `/var` is specified here in order to make it possible to mount `/usr` read-only. Everything that once went into `/usr` that is written to during system operation (as opposed to installation and software maintenance) must be in `/var`. If `/var` cannot be made a separate partition, it is often preferable to move `/var` out of the root partition and into the `/usr` partition. (This is sometimes done to reduce the size of the root partition or when space runs low in the root partition.) However, `/var` must not be linked to `/usr` because this makes separation of `/usr` and `/var` more difficult and is likely to create a naming conflict. Instead, link `/var` to `/usr/var`.

Applications must generally not add directories to the top level of `/var`. Such directories should only be added if they have some system-wide implication, and in consultation with the FHS mailing list.

1. Linux Basics

Linux/Unix Philosophy.....	17
Login.....	17
root and sudo	17
The desktop environment.....	17
X Window System.....	17
Window Manager.....	18
Shells.....	18
Linux commands.....	18
Redirection.....	19
Shell Scripting.....	20
How to write shell scripting.....	20
Components.....	20
Variables.....	20
Naming convention for Variables.....	21



Be careful when doing administrative tasks, you might damage your system!

Shell Name	Developed By	Where	Remarks
BASH (Bourne-Again SHell)	Brian Fox and Chet Ramey	Free Software Foundation	Most common shell in Linux. It's Freeware shell.
CSH (C SHell)	Bill Joy	University of California (For BSD)	The C shell's syntax and usage are very similar to the C programming language.
KSH (Korn SHell)	David Korn	AT & T Bell Labs	
TCSH	Ken Greer, Paul Placeway, Christos Zoulas, etc	Carnegie Mellon University	TCSH is an enhanced but completely compatible version of the Berkeley UNIX C shell (CSH).

Linux commands

- Copy Files and Directories
- Move Files and Directories
- Removing files and Directories
- List Files and Directories
- Make Directory
- Display commands
 - Clear Screen
 - cat
 - less
 - head
 - tail
- Search commands
 - less (/<contentToSearch> flag)
 - grep
 - mc
- Size
 - df
 - du

1. redirect stdout to a file
2. redirect stderr to a file
3. redirect stdout to a stderr
4. redirect stderr to a stdout
5. redirect stderr and stdout to a file
6. redirect stderr and stdout to stdout
7. redirect stderr and stdout to stderr

Also 1 'represents' stdout and 2 stderr.

Samples:

- output of first command to be written to another file

```
ls -l > out.txt
```

- Saves the standard output (exclude error message) from first command to another file and it is same as first command

```
ls -l /nonexistent 1> stdout.txt
```

- Saves the error message from first command to another file

```
ls -l /nonexistent 2> errout.txt
```

- **chmod +x <filename>**
- **chmod 755 <filename>**

Sample shell script

```
#!/bin/bash
echo Hello World
echo "Hello World"
echo \"Hello World\"
```

Components

- **Shell keywords** such as if..else, do..while.
- **Shell commands** such as pwd, test, echo, continue, type.
- **Linux binary commands** such as w, who, free etc..
- **Text processing utilities** such as grep, awk, cut.
- **Functions** - add frequent actions together via functions. For example, /etc/init.d/functions file contains functions to be used by most or all system shell scripts in the /etc/init.d directory.

-

1. Spaces are not allowed on either side of the equal sign when assigning values to variable.

2. Null variable can be defined as

```
null=  
null=""
```

3. Variables are case-sensitive

4. Special characters such as ?. * etc can not be used in variable names. Only exception is underscore.

Defining Variable