# Filters

Filters zijn opdrachten die gepipede redirection en output wijzigen. Let op dat filtercommando's ook standaard Linux-commando's zijn die zonder pipes kunnen worden gebruikt.

* find - find retourneert bestanden met bestandsnamen die overeenkomen met het argument dat is doorgegeven om te zoeken.
* grep - grep retourneert tekst die overeenkomt met het tekenreekspatroon dat aan grep is doorgegeven.
* tee - tee leidt standaardinvoer om naar zowel standaarduitvoer als een of meer bestanden.m
* tr - tr zoekt en vervangt de ene string door een andere.
* wc - wc telt karakters, regels en woorden.
* cat - Wanneer tussen twee pipes, doet het cat-commando niets (behalve stdin op stdout zetten).
* cut -

# Voorbeelden

1. command > file

Dit patroon redirect de standaard output van een opdracht om naar een bestand.

ls ~ > root\_dir\_contents.txt

De bovenstaande opdracht geeft de inhoud van de hoofdmap van uw systeem door als standaardoutput en schrijft de output naar een bestand met de naam rootdircontents.txt. Het zal alle eerdere inhoud in het bestand verwijderen, omdat het een opdracht met één haakje is.

1. command > /dev/null

/dev/null is een speciaal bestand dat wordt gebruikt om alle gegevens die ernaar worden omgeleid, weg te gooien. Het wordt gebruikt om standaardoutput te verwijderen die niet nodig is en die anders de functionaliteit van een opdracht of een script zou kunnen verstoren. Elke uitvoer die naar /dev/null wordt gestuurd, wordt weggegooid.

**ls > /dev/null**

Dit commando negeert de standaard output stream die wordt geretourneerd door het commando ls door het door te geven aan /dev/null.

1. command 2> file

Dit patroon leidt de standaard error stream van een opdracht om naar een bestand, waarbij bestaande inhoud wordt overschreven.

**mkdir '' 2> mkdir\_log.txt**

Dit leidt de fout om die wordt veroorzaakt door de ongeldige mapnaam ", en schrijft deze naar log.txt. Merk op dat de fout nog steeds naar de terminal wordt verzonden en als tekst wordt weergegeven.

1. command >> file

Dit patroon leidt de standaard output van een opdracht om naar een bestand zonder de bestaande inhoud van het bestand te overschrijven.

**echo Written to a new file > data.txt**

**echo Appended content to an existing file >> data.txt**

Deze commando's redirecten eerst de tekst die door de gebruiker is ingevoerd via echo om naar een nieuw bestand. Vervolgens voegt het de tekst die is ontvangen door het tweede echo-commando toe aan het bestaande bestand, zonder de inhoud ervan te overschrijven.

# command 2>> file

Het bovenstaande patroon redirect de standaard error stream van een opdracht om naar een bestand zonder de bestaande inhoud van het bestand te overschrijven. Dit patroon is handig voor het maken van foutenlogboeken voor een programma of service, omdat de vorige inhoud van het logbestand niet wordt gewist telkens wanneer er naar het bestand wordt geschreven.

**find '' 2> stderr\_log.txt**

**wc '' 2>> stderr\_log.txt**

De bovenstaande opdracht leidt het foutbericht dat wordt veroorzaakt door een ongeldig zoekargument om naar een bestand met de naam stderr\_log.txt. Vervolgens voegt het de foutmelding veroorzaakt door een ongeldig wc-argument toe aan hetzelfde bestand.

# command | command

Leidt de standaard output van het eerste commando om naar de standaard input van het tweede commando.

**find /var lib | grep deb**

Dit commando zoekt door /var en zijn submappen voor bestandsnamen en extensies dat met de string *deb* overeenkomt. Daarna retourneert het de bestandspaden voor de bestanden, waarbij het overeenkomende gedeelte in elk pad rood gemarkeerd is.

# ****command | tee file****

Dit patroon (inclusief het tee-commando) leidt de standaard output van het commando om naar een bestand en overschrijft de inhoud ervan. Vervolgens wordt de redirected output in de terminal weergegeven. Het maakt een nieuw bestand aan als het bestand nog niet bestaat.

In de context van dit patroon wordt tee meestal gebruikt om de output van een programma te bekijken en tegelijkertijd op te slaan in een bestand.

**wc /etc/magic | tee magic\_count.txt**

Dit piped het aantal voor karakters, regels en woorden in het magic bestand (gebruikt door de Linux-shell om bestandstypes te bepalen) door naar het tee-commando, dat vervolgens de uitvoer van wc in twee richtingen splitst en het naar het terminaldisplay en de magic\_count.txt-bestand stuurt. Stel je voor het tee-commando de letter T voor. Het onderste deel van de letter zijn de initiële gegevens en het bovenste deel zijn de gegevens die in twee verschillende richtingen worden gesplitst (standaard output en de terminal).

Meerdere pipes kunnen worden gebruikt om de output over meerdere opdrachten en/of filters te redirecten.

# command | command | command >> file

Dit patroon stuurt de standaard output van de eerste opdracht vooraf en filtert deze door de volgende twee opdrachten. Vervolgens voegt hij het eindresultaat toe aan een bestand.

**ls ~ | grep \*tar | tr e E >> ls\_log.txt**

Dit begint met het uitvoeren van ls in je hoofdmap (~) en het resultaat door te sturen naar het grep-commando. In dit geval retourneert grep een lijst met bestanden die tar bevatten in hun bestandsnaam of extensie.

De resultaten van grep worden vervolgens redirect naar tr, dat de letter e vervangt door E, aangezien e wordt doorgegeven als het eerste argument (de tekenreeks waarnaar moet worden gezocht), en E wordt doorgegeven als het tweede argument (de tekenreeks die eventuele overeenkomsten voor het eerste argument). Dit eindresultaat wordt vervolgens toegevoegd aan het bestand ls\_log.txt, dat wordt aangemaakt als het nog niet bestaat).