

# RCS: REVISIE CONTROL SYSTEEM

## Versiebeheer (deel 1)

Als Linux gebruiker speel je regelmatig met configuratiebestanden. Versiebeheer zorgt dat je altijd terug kunt naar een eerdere versie. Leer werken met RCS. Dit is een gemakkelijk te gebruiken versiebeheersysteem dat heel geschikt is voor losse configuratiebestanden. **Matto Fransen**

**W**anneer je een tekstbestand op je computer aanpast en opslaat, overschrijf dit bestand en daarmee gaat de oorspronkelijke inhoud verloren. Achteraf kan dit erg onhandig zijn. Werk je met meer mensen aan hetzelfde project, dan wordt dit een nog veel groter probleem. Begin jaren '70 is men daarom versiebeheersystemen gaan ontwikkelen.

Versiebeheer heeft zich bewezen als uiterst nuttig en wie er ervaring mee heeft, wil niet meer zonder. Versiebeheer is een belangrijke voorwaarde voor goede samenwerking tussen ontwikkelaars. Het geeft een historisch overzicht van wijzigingen en daarmee inzicht in de evolutie van de code. Het maakt het mogelijk dat mensen tegelijk aan hetzelfde project kunnen werken. Ook wanneer je in je eentje werkt, biedt versiebeheer veel voordelen.

Hieronder bespreken we eerst de algemene functies die elk versiebeheer systeem biedt. Daarna gaan we aan de slag met RCS.

### DE VIER W'S

Versiebeheer richt zich op de wijzigingen in bestanden. Centraal daarbij staan de vier W's: Wat, Waarom, Wanneer en Wie.

Het "Wat" gaat over de feitelijke wijziging zelf. In versiebeheersystemen is de tekstregel de kleinste eenheid van wijziging. Hierbij zijn drie mogelijkheden: een regel is toegevoegd, een regel is verwijderd of de tekst in een regel is gewijzigd. Bij het opvragen van de wijzigingen, toont het systeem in dat laatste geval de oorspronkelijke en de nieuwe regel.

Het "Waarom" geeft de verklaring van de wijziging. Bij het opslaan van wijzigingen voer je steeds een verklarende boodschap in. Je kunt daarbij meerdere regels tekst invoeren. Sommige versiebeheersystemen bieden de mogelijkheid een verkorte versie van het wijzigingslog te tonen, met alleen de bovenste regel van deze verklarende boodschappen. Probeer daarom in de eerste paar woorden uit te leggen waarom je wijziging hebt aangebracht.

"Wanneer" en "Wie" spreken voor zich. Bij het opslaan van de wijziging slaat het versiebeheersysteem een timestamp op en de naam van degene die dat gedaan heeft. Het is natuurlijk altijd fijn wanneer je iemand anders de schuld kunt geven, maar het is ook handig om later iemand een toelichting te vragen.

### SUPER UNDO

Het versiebeheersysteem slaat wijzigingen op als opeenvolgende versies. Hierbij bepaalt het systeem het verschil ten opzichte van de voorgaande versie en bewaart dit verschil.

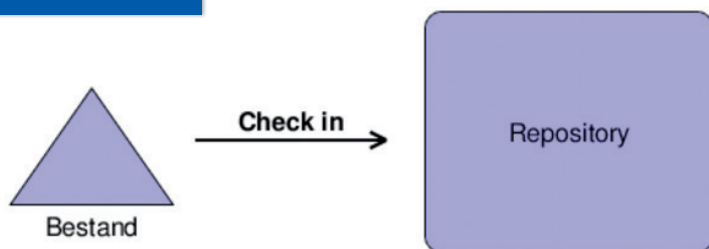
Alle tussenstappen tussen de allereerste versie van het bestand tot en met de allernieuwste worden bewaard. Je kunt daardoor later de verschillen tussen twee willekeurige versies opvragen, maar ook terug gaan naar een eerdere versie. Het maakt dan niet uit of die versie een paar minuten of enkele jaren oud is. Een versiebeheersysteem geeft je daarmee een super undo functie.

De repository is de centrale plek waar het versiebeheersysteem alles opslaat. Werken met de bestanden doe je op basis van een werkkopie. Je vraagt de bestanden uit de repository op en later check je hier de wijzigingen weer in. Het is gebruikelijk dat je alleen handgemaakte bestanden in het versiebeheersysteem incheckt. Bestanden die gegenereerd of gecompileerd worden, neem je niet in de repository op. Die genereer je opnieuw.

### RCS

RCS ("Revision Control System") is een goede manier om kennis te maken met versiebe-

Een check-in plaatst het bestand in de repository



heer. RCS stamt uit 1982, is gemakkelijk in gebruik en is een eerste generatie versiebeheersysteem. De repository bestaat uit platte tekstbestanden, waardoor je desgewenst kunt volgen wat onder de motorkap gebeurt. Sommige applicaties, zoals wiki's, gebruiken RCS voor versiebeheer van de content. RCS is gemaakt voor source code, maar is geschikt voor alle tekstbestanden. Dit artikel is bijvoorbeeld onder RCS-beheer geschreven.

RCS werkt op bestandsniveau en een lokale repository. Meestal is dat een nadeel. De tweede en derde generatie versiebeheersystemen werken daarom op directoryniveau en met een centrale repository op het netwerk. Dat RCS lokaal en op bestandsniveau werkt, is juist een voordeel bij het beheer van configuratiebestanden. Het is uitstekend geschikt voor bijvoorbeeld je `apache.conf`, `php.ini`, of de dotfiles in je home-directory.

Controleer met het commando `rcs -V` of RCS op je systeem staat. Maak een oefendirectory aan en plaats daarin een eenvoudig tekstbestandje van zo'n vijf regels met de naam "tekstbestand". Maak in je oefendirectory een subdirectory "RCS" aan. Met behulp van het commando `ci -u tekstbestand` checken we het bestand in, dat wil zeggen: opnemen in de repository. RCS vraagt om een boodschap. Type hier iets in en sluit af met een regel met uitsluitend een punt. Het tekstbestand is nu read-only gemaakt (controleer dit met `ls -l`) en de RCS directory bevat een nieuw bestand "tekstbestand,v". We gaan nu het onderhoud van dit bestand nabootsen en maken een paar extra opeenvolgende versies.

## CHECK OUT

Check het bestand uit. Dit is het opvragen uit de repository, met `co -l tekstbestand`. Voeg een extra regel aan het bestand toe, sla het op en check dit weer in met `ci -u tekstbestand`. Zo maken we nog wat aanpassingen. Check het bestand uit, verwijder een regel en check het weer in. Check het hierna weer uit, wijzig een regel en check het opnieuw in.

Met "rlog tekstbestand" vraag je vervolgens de geschiedenis op. Je ziet precies welke versies wanneer en door wie zijn gemaakt en met welke boodschap die zijn ingecheckt.

Wijzigingen tussen de verschillende versies vraag je op met `rcsdiff`. Het commando `"rcsdiff -r1.2 -r1.4 tekstbestand"` toont de verschillen tussen versie 1.2 en versie 1.4. Check het bestand nog een keer uit, en wijzig het, maar doe geen check in. Het commando `"rcsdiff tekstbestand"` toont nu de verschillen tussen je huidige versie en de actuele versie in de repository. Het is een goede gewoonte om dit voorafgaand aan iedere check-in te doen. Dit helpt bij het opstellen van de verklarende boodschap.

## OUDE VERSIE OPVRAGEN

Verwijder het bestand met `"rm tekstbestand"`. Wanneer je nu een check-out doet, is je bestand weer terug. RCS beschermt je zo een klein beetje tegen het per ongeluk weggooiën van bestanden. Met `"co -l -r1.3 tekstbestand"` vraag je versie 1.3 van het bestand op. Op deze manier ga je terug naar een oude versie.

Wanneer je meerdere bestanden in je oefendirectory aanmaakt en deze incheckt, zul je zien dat voor elk bestand in de RCS-directory een, v-bestand wordt aangemaakt. Ontbreekt de RCS-directory is, dan plaatst RCS deze, v-bestanden lokaal, naast de werkbestanden.

## SAVE-POINTS

Iedere keer wanneer je een check-in doet, maak je een "save-point" aan, waar je later naar terug kunt. Bij het editten van grotere

bestanden, zul je niet elke gewijzigde letter inchecken. Door echter regelmatig in te checken, maak je het later wel makkelijker om terug te zien wat er allemaal is gebeurd. Ook heb je meer tussenfasen waar je later eventueel op kan terugvallen. Dit kan bijvoorbeeld bij het debuggen erg fijn zijn. Maak dus bewust gebruik van deze save-points.

Om met RCS te werken, hoef je eigenlijk maar vier commando's te kennen: `ci` (check in), `co` (check out), `rlog` en `rcsdiff`. Wil je een bestand in bijvoorbeeld `/etc` aanpassen, maak dan eerst een RCS-directory aan en doe een check-in. Nu kun je naar hartenlust het bestand weer uitchecken, wijzigen, testen, inchecken, eventueel weer uitchecken, aanpassen en inchecken, enzovoorts. Via `rlog` kun je later terugzien waarom wat gewijzigd is en wanneer iets misgaat, heb je een vangnet. Je kan altijd terug naar een eerdere versie.

## TOT SLOT

RCS is niet moeilijk en vereist geen centrale server. Het geeft inzicht in de veranderingen en de zekerheid dat je later wijzigingen ongedaan kunt maken. Maak er een gewoonte van om elk configuratiebestand en elk belangrijk tekstbestand voor het editten eerst even in te checken. Er komt een dag dat je daar achteraf heel blij mee bent!

Een check-out vraagt het bestand op uit de repository

