

Hvordan holde styr på arbeidet ditt med Mercurial

av Åsmund Ervik

Sammendrag

Mercurial holder styr på filene dine, og sørger for backup når du bruker det på `login.stud.ntnu.no`.

Når du gjør en stor endring i programmet ditt gjør du:

add

commit - husk å skrive gode kommentarer!

På slutten av dagen gjør du:

push

Introduksjon

•Hva er Mercurial?

Mercurial er et system for å holde styr på kildekode, f.eks. Python eller LaTeX-filer (eller begge deler). Mercurial gjør det enkelt å gå tilbake til tidligere versjoner dersom en feil oppstår, eller du sletter noe du ikke burde slettet. I motsetning til `angre`-funksjonen i en teksteditor, angreer man mellom forskjellige lagrede versjoner, noe som er nyttig når man hele tiden lagrer og kompilerer programmet sitt.

•Er dette noe som brukes i den virkelige verden?

Ja! Alle store prosjekter bruker en eller annen form for versjons-system. Akkurat Mercurial brukes bl.a. av de som lager Firefox, OpenOffice og Octave.

•Hva trenger jeg for å bruke Mercurial?

På Windows trenger du programmet TortoiseHg, som kan lastes ned fra <http://tortoisehg.bitbucket.com>. På Linux kan du installere programmet `mercurial` som kan brukes fra kommandolinjen (der det heter `hg`)¹, eller bruke TortoiseHg hvis du foretrekker knapper å trykke på. Har du Mac, anbefales fortsatt å bruke kommandolinjevarianten. Dersom du har en uforklarlig fobi for kommandolinjer kan f.eks. `MacHg`, <http://jasonfharris.com/machg/> brukes, men da må du Google deg frem til et oppsett selv (eller spørre Orakel).²

Dersom du ikke har skjønt hvorfor alle programmene heter noe med Hg, så er det på tide å friske opp det periodiske systemet.

Hvordan bruker jeg Mercurial?

Det enkleste for deg er å sette opp din egen Mercurial-server på NTNU sin tjener `login.stud.ntnu.no` - det tar bare noen minutter. Når du har satt opp dette, har du et system som gjør at du kan ha bedre kontroll på filene dine, men du nyter også godene av et veldig bra backup-system som virker selv om PC-en din dør.³ Dette er svært nyttig når du f.eks. skal skrive masteroppgave, og da har du allerede oppsettet i boks hvis du gjør dette nå!

Oppsett på `login.stud.ntnu.no`

For det første trenger du en ssh-klient, dvs. Putty dersom du er på Windows, og ssh (i terminalen) på Mac og Linux. Logg deg inn på serveren med Putty ved å skrive inn adressen [login.stud.](http://login.stud.ntnu.no)

¹linux-personer liker korte navn, mercurial tar alt for lang tid å skrive!

²Det arbeides med å porte TortoiseHg til Mac, en (foreløpig) litt ustabil versjon kan finnes vha. Google

³eller i tilfelle kjernefysisk angrep

`ntnu.no`, eller med ssh ved å skrive `ssh login.stud.ntnu.no` i en terminal. Nå er du på en Linux-maskin, og filene du jobber med er hjemmeområdet ditt - kommandoen `ls` burde vise deg kjente filer. Lag en mappe som heter “hg” - `mkdir hg`. Åpne denne mappen med `cd hg`. Lag en ny mappe til det nåværende prosjektet ditt, f.eks. “elmag”, og åpne denne. Her skriver du nå den magiske kommandoen `hg init`. Når du har et nytt prosjekt, lager du en ny mappe og kjører `hg init` der. Denne kommandoen oppretter en mappe som heter heter `.hg`. **Aldri** tukle med denne mappen!

Dersom du vil sjekke at alt virker, kan du nå lage en tekstfil i denne mappen. Med kommandoen `hg add` ber du Mercurial om å se etter nye filer, og legge dem til i arkivet. Så må du gjøre dette til en revisjon med kommandoen `hg commit -m “Min kommentar”`. Da er du i boks på Linux-serveren! Dvs., det er en ting til du godt kan gjøre, men det er ikke obligatorisk, og det er...

`.hgignore` - en fantastisk fil! Du må lage den selv, men når du har gjort det kan du få Mercurial til å ignorere enkelte filer - f.eks. unyttige `.log`-filer eller `.out`-filer som Latex-kompilatoren din lager. Lag selv en fil med navnet “`.hgignore`” i mappen “`elmag`”. På første linje skriver du *syntax*: `glob`⁴ og på de neste linjene skriver du f.eks. `*.log`, `*.out` osv. på hver sin linje.

Nå må du selvfølgelig legge til filen `.hgignore` (`hg add`), og committe den!

Oppsett på egen maskin

Det antas at du har klart å installere det riktige programmet selv. Da er steg 1 å fortelle programmet hvem du er. Du bør unngå å bruke norske bokstaver.

- *hg* på *Linux/Mac*: rediger filen `.hgrc` som ligger i home-mappa di. Skriv inn:
[ui]
username = Bjarne Pettersen <bjarnep@stud.ntnu.no>
- *TortoiseHg*: Høyreklikk på skrivebordet, velg TortoiseHg og så “Global Settings”. Velg “Commit” i listen til venstre, og endre feltet “username”, f.eks.: “Bjarne Pettersen <bjarnep@stud.ntnu.no>”. Resten av instillingene trenger du ikke tenke på, så bare trykk “OK” nederst.

Det var det du trengte av oppsett!

Begynn å bruke Mercurial!

Da er du klar! Det første vi skal gjøre er å “klone” det arkivet du lagde på Linux-serveren. Da må du først gå til den mappen du vil jobbe i. I denne mappen blir det nå lagd en ny mappe som heter “`elmag`”, der alle filene dine skal bo. Det gjør du slik:

- *hg*: `hg clone ssh://bjarnep@login.stud.ntnu.no/hg/elmag`
- *TortoiseHg*: Høyreklikk i mappen, velg menyen TortoiseHg, velg “clone”. Rediger “source path” til `ssh://bjarnep@login.stud.ntnu.no/hg/elmag`

I begge tilfeller må du skrive inn ditt NTNU-passord.

Da har det blitt opprettet en mappe som heter `elmag`, med de filene du lagde på Linux-serveren. Nå kan du starte arbeidet som vanlig!

Forresten, hvis du har Windows Vista eller 7, bør du be antivirus-programmet ditt (og indekseringsprogrammet) om å ignorere de mappene der du bruker TortoiseHg. Hvis ikke kan PC-en din henge med 100% CPU-bruk. Dette skyldes en feil i Windows 7⁵ som vil bli rettet i Service Pack 1.

⁴du kan også bruk *syntax*: `regexp` hvis du liker regulære uttrykk

⁵denne feilen kan best beskrives med det noe fargerike uttrykket “clusterfuck”

Vanlig arbeid

Når du skal til å gjøre en stor endring, f.eks. slette en stor bolk med kode, er det lurt å gjøre en `add+commit` først. Bruker du `hg` på Linux eller Mac, vet du allerede hvordan dette foregår. På TortoiseHg: høyreklikk i mappen -> TortoiseHg -> `add`, og så høyreklikk -> `Hg Commit`.

På slutten av arbeidsøkten bør du så “pushe” endringene over til Linux-serveren. Bruker du `hg`, skriver du da `hg push ssh://bjarnep@login.stud.ntnu.no/hg/elmag`. Bruker du TortoiseHg, bør du fyre opp Tortoise Repository Explorer (høyreklikk). Her velger du “Synchronize”, og så “Push”. Som alltid, oppgi brukernavn og passord når det trengs.

Hvis du er flink til å committe, og flink til å skrive gode kommentarer når du gjør det, vil Mercurial antakelig redde nakken din en gang eller tre!

Krise!!!

Du har gjort en tabbe, og slettet den mest geniale koden ever. Hva gjør du???

- Har du vært flink til å committe ofte? Hvis ikke, synd for deg!!!
- Hvis du har vært flink, så vil du rulle tilbake til en tidligere versjon. Det gjøres med handlingen “update”. Først vil du antakelig sjekke hvilken versjon du skal gå tilbake til. . .
- Med `hg` på Linux/Mac: kan du skrive `hg log` for en liste over alle versjoner (nyeste øverst). Har du skrevet gode kommentarer når du har committet, ser du forhåpentligvis hva som er riktig versjon. Der det står `changeset: 6:00cb6d40913e` indikerer dette versjon nr. 6 (kolon etter 6). Vil du dobbeltsjekke at du har funnet riktig versjon, kan du skrive `hg cat -r 6 viktigFil.py` for å se hvordan `viktigFil.py` så ut ved versjon 6. For å sammenlikne en fil ved revisjon 3 og 6, kan du skrive `hg cat -r 3:6 viktigFil.py`.

Når du er sikkert på hvilken utgave du vil gå til, bruker du kommandoen `hg update -r 6` for å gå til versjon 6.

- Med TortoiseHg: åpne Repository Explorer. Her får du opp et tre over tidligere og nåværende versjoner. Når du klikker på en versjon, vises filene som er endret fra forrige versjon til denne versjonen nederst til venstre. Når du klikker på en fil her forhåndsvises den nederst til høyre. Ignorer de øverste linjene i filen, det er Mercurial-data. Nye linjer i filen er markert med +, slettede linjer er markert med - (relativt til forrige versjon).

Let deg frem til riktig versjon, høyreklikk, og velg “update”.

Noen små kommentarer

Dersom du noen gang går inn i mappen “`hg/elmag`” på `login.stud`-serveren (eller på hjemmeområdet ditt) vil du kanskje bli forferdet: hvor er filene mine???. Slapp helt av, de er der. Det er nemlig sånn at hvis du vil se den nyeste versjonen på `login.stud`-serveren, må du kjøre kommandoen `hg update`⁶ først. Da vil du havne på nyeste versjon, og alle filene dine er der! Dette virker kanskje litt merkelig, men slik er verden.

En annen advarsel: hvis du noen gang gjør endringer direkte inne på `login.stud`-serveren (selvfølgelig etter å ha gjort `hg update`) står du overfor en ny problemstilling. Når du kommer tilbake til maskinen du jobber på til vanlig, **må** du huske å gjøre operasjonen `pull`⁷ etterfulgt av `update`. Hvis du glemmer dette, må du gjøre noe som heter “merging”. Det er ikke så vanskelig,

⁶hvis du ikke spesifiserer en versjon med f.eks. `-r 314`, vil `update` gå til den nyeste versjonen tilgjengelig

⁷på Linux: `hg pull ssh://blabla`, på TortoiseHg: Repository Explorer -> Synchronize -> Pull

men tar litt tid, og er unødvendig så lenge du husker “pull+update”. Vi diskuterer ikke merging her, men Google er din venn.

Hvis du må jobbe på en annen maskin enn den du bruker til vanlig, kan du simpelthen kjøre “clone”-operasjonen der, og jobbe som vanlig (og selvfølgelig “pushe” når du er ferdig)! Da må du igjen huske å gjøre en “pull+update” når du kommer tilbake til din vanlige maskin. Hvis du glemmer dette, må du merge.

Takk for meg!

Da har du forhåpentligvis lært nok til å komme igang. Hvis du lurer på f.eks. hvordan du kan slippe å skrive inn passord hver gang du push-er, er Google din beste venn (tips: RSA SSH). Det er også ganske enkelt å slippe å skrive inn `ssh://bjarnep@login.stud.ntnu.no/hg/elmag` hver gang, du finner instruksjer for dette på nettet hvis du bruker Mercurial mye.