**Versionhallinta, Git**

**TK0802-3001 tehtävät**

**TRTKNU15**

Elias Koskinen

SISÄLLYS

SANASTO 1

1 Taso 1 1

1.1 Asenna Git koneellesi 1

1.2 Konfiguroi Git ensimmäistä käyttökertaa varten 1

1.3 Luo uusi paikallinen tietovarasto ja ensimmäisen pysyvän muutoksen tekeminen 1

1.4 Luo paikallinen tietovarasto olemassa olevasta etätietovarastosta 1

1.5 Tulevien muutosten katselu ennen pysyvän muutokset tekemistä 1

1.6 Millä komennolla saat lisättyä nykyisestä työkansiosta kaikki muutokset tulevaan pysyvään muutokseen, jos tiedostoja on useampi kuin yksi? 1

1.7 Mitä commit komento tekee? 1

1.8 Miten sivuutan tiedostot, joiden muutoksia en halua jäljitettävän? 2

1.9 Tulevien pysyvien muutosten peruuttaminen 2

1.10 Mitä tekee git mv –komento? 2

1.11 Mitä tekee git rm –komento? 2

1.12 Luo oma etätietovarasto ja työnnä sinne muutoksesi 2

2 TASO 2 2

2.1 Etätietovaraston viittauksien lisääminen ja poistaminen 2

2.2 Etätietovaraston kehityshaaran hakeminen paikalliseen tietovarastoon 3

2.3 Etätietovaraston kehityshaaran hakeminen paikalliseen tietovarastoon vetämällä muutokset 3

2.4 Paikallisten tietovaraston master -kehityshaaran työntäminen etätietovarastoon 3

2.5 Paikallisen tietovaraston kehityshaaran tai tagin työntäminen etätietovarastoon 3

3 TASO 3 3

3.1 Luo uusi kehityshaara paikalliseen tietovarastoon 3

3.2 Kehityshaarojen tarkastelu 4

3.3 Kehityshaarojen yhdistäminen 4

3.4 Kehityshaaran uudelleenkirjoittaminen (rebase) 4

3.5 Kehityshaarojen poistaminen 4

3.6 Tag –merkinnät 4

4 TASO 4 4

4.1 Git login tarkastelu 4

4.2 Muutosten tarkastelu eri versioiden välillä 4

4.3 Konfliktien hallinta ja selviytyminen 5

4.4 Muutosten piilottaminen väliaikaisesti 5

4.5 Tutustu graafiseen Git –työkaluun 5

5 TASO 5 5

5.1 Gitin perusteet tositoimissa 5

SANASTO

Kerää sanastoon Git -työkalun käytössä vastaan tulleita termejä. Dokumentissä useat termit on käännetty suomeksi, joten yritä löytää ainakin niiden englanninkieliset tai tekniset alkuperäiset termit. Sanaston on tarkoitus auttaa sinua ymmärtämään paremmin konteksti, jossa joudut toimimaan ja kommunikoimaan käyttäen oikeita termejä.

1. Taso 1
   1. Asenna Git koneellesi

Asentamisen jälkeen anna komento **git –version** ja ota kuvakaappaus.

* 1. Konfiguroi Git ensimmäistä käyttökertaa varten

Konfiguroinnin jälkeen anna komento **git config –list** ja ota kuvakaappaus.

* 1. Luo uusi paikallinen tietovarasto ja ensimmäisen pysyvän muutoksen tekeminen

Tietovaraston luonnin jälkeen lisää muutama tiedosto versionhallinnan jäljitettäväksi ja sen jälkeen tee muutoksista pysyviä. Anna tämän jälkeen komento **git log –stat** ja ota kuvakaappaus.

* 1. Luo paikallinen tietovarasto olemassa olevasta etätietovarastosta

Luo paikallinen kopio seuraavasta julkisesta tietovarastosta:

<https://github.com/nyluntu/hacksummit-forecast>

Tämän jälkeen anna komento **git remote –v** ja ota kuvakaappaus.

* 1. Tulevien muutosten katselu ennen pysyvän muutokset tekemistä

Mieti millä komennolla näet mitkä tiedostot ovat uusia, poistettuja tai lisätty versionhallinnan jäljitettäväksi, joissa on muutoksia. Tämän jälkeen anna kyseinen Git komento ja ota kuvakaappaus. Kuvakaappauksessa tulisi vähintään näkyä uusi lisätty tiedosto ja yksi tulevaan pysyvään muutokseen lisätty muutos tai jäljitettävä tiedosto.

* 1. Millä komennolla saat lisättyä nykyisestä työkansiosta kaikki muutokset tulevaan pysyvään muutokseen, jos tiedostoja on useampi kuin yksi?

Kerro tähän vaadittavat komennot tai anna muunlainen selitys mitä tulisi tehdä.

* 1. Mitä commit komento tekee?

Kerro tähän vapaamuotoisesti mitä commit –komento tekee käytännössä?

* 1. Miten sivuutan tiedostot, joiden muutoksia en halua jäljitettävän?

Kerro miten sivuutus tapahtuu ja miksi käyttäisit tällaista ominaisuutta? Tarvittaessa ota kuvakaappaus perustelun tueksi.

* 1. Tulevien pysyvien muutosten peruuttaminen

Millä komennolla voit peruuttaa jo valmistellut tiedostot tulevaan pysyvään muutokseen? Eli olet esimerkiksi muuttanut tiedostoa tai lisännyt uuden jäljitettäväksi mutta huomaat, ettet haluakaan ottaa uutta tiedostoa mukaan seuraavaan pysyvään muutokseen. Muutos on siis jo tässä kohdin *staged* tilassa. Miten siis peruuttaisit tulevan muutokset, että saat osan tiedostoista pois *staged* tilasta? Kerro tarvittavat komennot ja ota myös **git status** –komentoa käyttäen pari kuvakaappausta ennen ja jälkeen tilanteen.

* 1. Mitä tekee git mv –komento?

Kerro mitä otsikossa mainittu komento tekee ja miksi käyttäisit tai et käyttäisi sitä?

* 1. Mitä tekee git rm –komento?

Kerro mitä otsikossa mainittu komento tekee ja miksi käyttäisit tai et käyttäisi sitä?

* 1. Luo oma etätietovarasto ja työnnä sinne muutoksesi

Voit luoda etätietovaraston haluamaasi palveluun. Kun olet muutokset työntänyt tietovarastoon, ota kuva palvelun Git historia -näkymästä sekä sinun paikallisesta historiasta komennolla **git log -10 –oneline**

1. TASO 2
   1. Etätietovaraston viittauksien lisääminen ja poistaminen

Miten voit lisätä ja poistaa viittauksia etätietovarastoihin? Paikallinen tietovarasto osoittaa useimmiten myös yhteen etätietovarastooon, jos lähdekoodit jaetaan muiden kehittäjien kanssa. Millä komennoilla näiden osoitteiden ja viittausten hallinta onnistuu, kun ei käytetä **clone** –komentoa? Anna lopuksi komento **git remote –v** ja ota kuvakaappaus, josta näkyy vähintään kaksi eri viittausta etätietovarastoon.

* 1. Etätietovaraston kehityshaaran hakeminen paikalliseen tietovarastoon

Millä komennoilla saat haettua etätietovarastosta erillisen kehityshaaran paikalliseen tietovarastoon? Kehityshaara ei saa olla siis **master** –kehityshaara vaan jokin muu. Kehityshaara tulee hakea omaan paikalliseen kehityshaaraan ilman, että tapahtuu ns. **merge**. Kerro vaadittavat komennot ja kuvaile mitä ne tekevät.

* 1. Etätietovaraston kehityshaaran hakeminen paikalliseen tietovarastoon vetämällä muutokset

Kysymys on sama kuin edellinen mutta nyt **merge** –tapahtuma saa tapahtua samalla. Kehityshaaran on oltava jokin muu kuin **master** –kehityshaara. Kerro vaadittavat komennot ja kuvaile mitä ne tekevät.

* 1. Paikallisten tietovaraston master -kehityshaaran työntäminen etätietovarastoon

Kerro millä komennoilla saat työnnettyä paikallisessa tietovarastossa olevan **master** -kehityshaaran etätietovarastoon.

* 1. Paikallisen tietovaraston kehityshaaran tai tagin työntäminen etätietovarastoon

Sama idea kuin edellisessä mutta kehityshaaran pitää olla jokin muu kuin **master** –kehityshaara. Ota kuvakaappaus etätietovarastosta, jossa tämä kehityshaara näkyy. Ota kuvakaappaus etätietovarastosta, josta näkyy sinne merkitty tag –merkintä.

1. TASO 3
   1. Luo uusi kehityshaara paikalliseen tietovarastoon

Kerro millä eri tavoin voit luoda haaroja paikallisessa tietovarastossa? Kerro tarvittavat komennot ja anna lopuksi seuraava komento

**git log --graph --pretty=oneline --abbrev-commit**

Ota komennon antamasta tulosteesta kuva, jossa näkyy luotu kehityshaara. Huomioi, että sinun pitää tehdä kehityshaaraan vähintään yksi pysyvä muutos.

* 1. Kehityshaarojen tarkastelu

Kerro millä komennoilla voit nähdä paikallisen tietovaraston tai etätietovarastojen kehityshaaroja? Ota lopuksi kuvakaappaus paikallisessa tietovarastossa olevista kehityshaaroista. Kehityshaaroja pitää olla useampi kuin kaksi.

* 1. Kehityshaarojen yhdistäminen

Kerro millä komennoilla voit yhdistää kehityshaaroja paikallisessa tietovarastossa? Ota lopuksi kuvakaappaus käyttäen seuraavaa komentoa

**git log --graph --pretty=oneline --abbrev-commit**

* 1. Kehityshaaran uudelleenkirjoittaminen (rebase)

Kerro mitä **git rebase** komennolla voi tehdä?

* 1. Kehityshaarojen poistaminen

Kerro millä komennoilla voit poistaa kehityshaaroja? Ota kuvakaappaus ennen ja jälkeen toimenpiteen kehityshaarojen listauksesta, josta näkyy tarvittavat muutokset tietovarastossa.

* 1. Tag –merkinnät

Kerro mitä ovat tag –merkinnät tietovarastossa ja missä tilanteessa käyttäisit niitä?

1. TASO 4
   1. Git login tarkastelu

Gitissä on tehokas **log** komento. Kerro miten saat komennon avulla listattua edellisen 1 viikon aikana tehdyt pysyvät muutokset? **Log** komentoa on käytetty jo aikaisemmin, joten tutustu sen parametreihin ja myös muihin vaihtoehtoihin.

* 1. Muutosten tarkastelu eri versioiden välillä

Kerro millä komennolla voit nähdä yhden tiedoston muutokset verrattuna sen edelliseen versioon? Tai millä komennolla voit nähdä muutokset edelliseen pysyvään merkintään verrattuna tietovarastossa?

* 1. Konfliktien hallinta ja selviytyminen

Yritä saada aikaiseksi konflikti. Kerro mitä konflikti tarkoittaa ja miten siitä voi selvitä? Miten Git –merkitsee konfliktitilanteen?

* 1. Muutosten piilottaminen väliaikaisesti

Millä komennolla voit piilottaa nykyiset muutokset väliaikaisesti ja palata jatkamaan myöhemmin? Tämä tulee tehdä ilman **commit** –komentoa. Esimerkiksi sinulla voi olla tilanne, jossa olet tehnyt muutoksia **master** –kehityshaaraan ja olet vetämässä etätietovarastosta muutoksia. Et tahdo tehdä turhaa pysyvää muutosta mutta tahdot väliaikaisesti piilottaa muutokset, vetää päivitetyn version ja sen jälkeen tuoda takaisin muutokset siihen tilaan missä alun perin olit?

* 1. Tutustu graafiseen Git –työkaluun

Git –versionhallintaa ei tarvitse aina käyttää komentojen avulla. Tutustu yhteen monista graafisista työkaluista ja tee sillä muutamia aikaisemmin läpikäytyjä komentoja. Työkalu voi olla myös integroitu IDE –ohjelmistoon. Ota lopuksi pari kuvakaappausta todisteeksi, että näin on tapahtunut ja olet tutustunut vähintään yhteen työkaluun.

1. TASO 5
   1. Gitin perusteet tositoimissa

Aikaisemmin olet käynyt läpi useita erilaisia komentoja. Tällä tasolla ei ole enää yksittäisiä tehtäviä vaan nyt sinun pitää näyttää osaaminen. Moduulin aikana on erilaisia projekteja ja harjoitustehtäviä, joita voit käyttää tämän tukena. Voit myös käyttää omaa henkilökohtaista projektia, jos haluat. Tärkeintä on kuitenkin, että hyödynnät versionhallintaa mahdollisimman paljon ja teet useita pysyviä muutoksia tietovarastoon, josta jää muutoshistoria näkyville. Tämän jälkeen palauta tähän linkki etätietovarastoon, jonka haluat arvosteltavan. Huomioi, että etätietovarasto n pitää olla julkinen tai muuten käyttöoikeudet kunnossa, että siihen pääsee teeman ohjaaja käsiksi.