

Ympäristömuuttujat

Ympäristömuuttujat ovat komentotulkin (shell) levittämiä muuttujia, joilla välitetään käyttäjän asetuksia. Tässä harjoitellaan bash-tulkin syntaksia.

1. Tulosta ympäristömuuttujasi: *printenv*
2. Tarkastele komentopolkuasi, joka löytyy muuttujasta *PATH*: *printenv PATH*
Komentopolku on lista hakemistoja, joista komentoriviltä ajettavia komentoja haetaan.
3. Tehdään esimerkiksi uusi muuttuja: *export ESIM=helloworld*
4. Ympäristömuuttujia voidaan käyttää komentorivillä: *echo \$ESIM* tai *echo \${ESIM}*
5. Muuttuja voi määritellä komentokohtaisesti myös: *VALIAIK="heipa hei" echo \$VALIAIK*
6. Tee kotihakemistoosi hakemisto *bin* ja lisää tämä komentopolulle: *export PATH=\${PATH}:/bin*

Komentotiedostot

Komentotiedosto (shell script, batch file) on suoritettava tiedosto, jossa on komentoja jotka komentotulki suorittaa peräkkäin. Komentotiedostot tunnistetaan laittamalla tiedoston alkuun ns. hashbang-tunniste (*#!/bin/sh*), joka kertoo millä tulkillla komentotiedosto tulee suorittaa. Lisäksi komentotiedosto tulee laittaa suoritettavaksi (*chmod +x tiedosto*). Jotkut nimeää komentotiedostot *.sh* päätteellä, mutta tämä ei ole pakollista.

Komentotiedostot ovat oma rajoittunut ohjelmointikielensä, mutta niiden käyttökelpoisuus tehtävien automatisoinnissa on kiistämätön. Käsitellään niiden rakennetta tässä lyhyesti parilla esimerkillä.

1. Luo bin-hakemistoosi tiedosto nimeltä *mun_ohjelmat.sh* ja syötä sinne komennot:

```
#!/bin/sh
# Risuaidalla voidaan tehdä kommentteja komentotiedostoihin
# Tallennetaan muuttujaan toisen komennon ulostulo:
MINAITSE='whoami'
# Toinen tapa tehdä sama asia:
# MINAITSE=$(whoami)
# Tulostetaan lista omista prosesseistani ja lasketaan rivien määrä
ps --no-headers -u $MINAITSE | wc -l
```

Edellä *whoami*-komennon kanssa käytetty merkki on ns. backtick, löytyy plus- ja backspace-nappien välistä näppäilyllä: *shift-‘ välilyönti*. Näiden kanssa tarkkana.

2. Lisää tiedostolle suoritusoikeus: *chmod +x mun_ohjelmat.sh*
3. Nyt voit suorittaa komennon *mun_ohjelmat.sh* ja se tulostaa montako prosessia sinulla on ajossa.

Ohjausrakenteet

Komentotiedostoissa on (komentotulkista riippuen; bash/tcsh/zsh) myös ohjausrakenteita kuten *if/then/else*-puut ja *for*-silmukkoja. Bashin manuaalisivu on rasittavan laaja, mutta yksi hyvä täsmäluettava on lista ehtolauseista luvussa "CONDITIONAL EXPRESSIONS".

1. Aloitetaan uuden komentotiedoston kirjoittaminen. Lisää siihen seuraavat komennot:

```
#!/bin/sh
# Tarkastetaan ettei komentotiedostoa saa ajaa root-käyttäjänä
if [ 'whoami' = 'root' ]
then
    echo 'Do NOT run as root, dummy!'
    exit 1
else
    echo "Welcome, 'whoami'."
    # Vihje: ero lainaus- (") ja heitto- (') merkkien välillä on että lainausmerkkien
    # välinen osa tulkitaan myös. Kokeile vaihtaa edeltävään komenttoon heittomerkit.
fi
```

2. Testaa ohjelmaa normaalikäyttäjänä ja root-käyttäjänä (esim. sudo-komennolla).
3. Yleinen tilanne on että komentoja joudutaan ajamaan useita kertoja ja tulokset halutaan eri tiedostoihin. Tehdään näin esimerkinomaisesti for-silmukalla ja *seq*-komennolla. Lisää edelliseen tiedostoon:

```
for $i in `seq -w 1 10`
do
    date > timestamp${i}.txt
    echo Sleeping $i seconds.
    sleep $i
done
```

4. Syntaksi on siis *for MUUTTUJA in ARVOT.. ; do komentoja ; done*. Jatketaan tiedostoa:

```
for $each in timestamp*.txt ; do
    cat $each
    rm $each
done
echo All done.
```

Komentoparametrit

Komentotiedostoille voi antaa myös parametreja kun niitä kutsutaan komentoriviltä. Näihin viitataan \$NUMERO -tyyliin. Nolla on suoritettavan komennon nimi, seuraavat siitä ylöspäin annetut parametrit. Esim:

```
#!/bin/sh
echo This script is 'basename $0'
# Onko
if [ "$1" = "" ] ; then
    echo Master, give me some parameters
    exit 0
fi
if [ -r "$1" ] ; then
    echo First parameter is $1 : `wc -l "$1"` lines
else
    echo There\'s no such file as $1
fi
```