SHELL-SKRIPTAUS KÄYTÄNNÖSSÄ

UNIX-kerho 26.03.2019 (UNIXTIME 1553623200)

https://github.com/unixkerho/shell-scripts

Maks Turtiainen

POSIX-STANDARDI

- Ennen shelleihin menoa olisi hyvä tietää on olemassa standardi joka määrittelee miten UNIX-tyyppisten käyttöjärjestelmien kuuluisi suunnilleen toimia
- POSIX IEEE-standardi on tällainen
- Tämän vuoksi samat shell-skriptit yleensä toimivat Linuxeilla, Darwin-pohjaisilla(Mac OS, iOS), *BSD:llä, Solariksella, Androidilla, HP-UX:lla, IBM AIX:lla jne.
- POSIX Windowsilla: Cygwin, MinGW, WSL (Windows Subsystem for Linux)

UNIX-SHELL

- UNIX-shell on komentorivitulkki UNIX-tyyppisiin käyttöjärjestelmiin
- Voi käyttää joko interaktiivisesti suoraan komentoriviltä tai suorittaa ennalta määrätyt komennot tiedostosta (skriptistä)
- Skriptien käyttökohteita: DevOps, asennusskriptit, automatisointi, tekstin parsiminen, simppelit ohjelmat

ERI SHELLIT

- Eri shellejä on paljon. Osa enemmän ja osa vähemmän POSIX-yhteensopivia. Useimmissa lisäominaisuuksia joita POSIX ei määrittele.
- POSIX-yhteensopivia: Bash, ZSH, DASH, KornShell (ksh, mksh), FreeBSD sh
- Muita: C shell (csh, tcsh), Fish
- Keskitymme tässä lähinnä Bashiin, koska se on aika pitkälti suosituin
- Käytössä olevan shellisi voit tarkistaa komennolla:

```
echo $SHELL
```

BASICS

- Ensimmäinen rivi määrittelee käytettävän shellin. ns. "shebang-line"
- Muuttujien määrittely `muuttuja=jotain`, viittaus `$muuttuja`
- `if` päättyy `fi`:n ja `case` päättyy `esac`:n
- Vertailuoperaattorit: `=`, `!=`, `-gt`, `-eq` (man test)
- Komentoriviparametrit: `$1`, `$2`, `$3` ...
- Kommentiksi tulkitaan kaikki `"#"` jälkeen
- Putkitus | ja redirectaus >
- https://en.wikipedia.org/wiki/List_of_Unix_commands

ENSIMMÄINEN SKRIPTI

```
touch ekaskripti.sh # luo tiedoston
chmod +x ekaskripti.sh # antaa tiedostolle suoritusoikeuden
$EDITOR ekaskripti.sh # avaa tiedoston oletuseditorilla
```

```
#!/usr/bin/env bash
echo "Moi!" # tulostus
echo "Tässä tämän hakemiston tiedostot:"
ls # listaa tiedostot tällä hakemistossa
```

```
./ekaskripti.sh # suorittaa skriptin
```

ENSIMMÄINEN SKRIPTI

```
#!/usr/bin/env bash
echo "Moi!"
printf "Haluatko listata tiedostot? (k/e) " # tulostus
read -r VASTAUS # inputti muuttujaan
if [ "$VASTAUS" = "k" ]; then
    echo "Listataan tiedostot"
    ls
fi
```

```
./ekaskripti.sh
```

PARAMETRIEN/INPUTIN PARSIMINEN OIKEIN

```
#!/usr/bin/env bash
# Esimerkki parametrien ja user inputin parsimiseen
# Tehty UNIX-kerhoa varten

# Funktio apujen tulostamiseen
apua() {
    echo "1 parametri on pakollinen. Parametrit:"
    echo
    echo "--hakemisto       Suorittaa hakemistotyökalun"
    echo "--levytila        Näyttää levytilan tilanteen"
    echo "--help            Näyttää tämän aputekstin"
    echo
}

# Funktio hakemistotyökalulle
hakemisto () {
```

```
./parametrit.sh --help
```

KÄYTÄNNÖLLISIÄ ONELINEREITA

```
# Korvaa kaikista tiedostonimistä tässä hakemistossa " ":n "_":lla
for f in * *; do mv "$f" "${f// /_}"; done
```

```
# Sama mutta rekursiivisesti
find . -name "* *" | awk '{ print length, $0 }' | sort -nr -s | cut -d" " -f2- | w
```

```
# Poista ylimääräiset tiedostot reposta
git status --short | cut -d " " -f 3 | xargs rm
```

```
# Random commit viesti
git commit -m "$(w3m whatthecommit.com | head -n 1)"
```

```
# Muunna kaikkien kuvien kokoa hakemistossa (imagemagick)
for file in *.png; do convert $file -resize 32x32! 32-$file; done
```

```
# Missä tiedostossa lukikaan "jotain"
grep -r "jotain" .
```

```
# Search-replace kaikista tiedostoista tästä hakemistosta ja alihakemistoista
find /home/mjt/projekti -type f -print0 | xargs -0 sed -i 's/vasen/oikea/g'
```

```
# Oikeudet turvallisemmaksi tässä ja alihakemistoissa
find . -perm 777 -exec chmod 755 {} \;
```

AUTOMATISOINTIA JA MUUTA

- Jotta toimisi perus komentoina skriptit oltava `$PATH`-muuttujan joss

```
mkdir ~/sh; echo 'export PATH=$PATH:~/sh' >> ~/.bashrc
```

- Muita vaihtoehtoja: aliakset ja funktiot bashrc:ssä (`~/.bashrc`)

```
# Vimistä PDF vieweri? (tämä .bashrc:n)
function vimpdf () {
  pdftotext $1 temp.txt && vim temp.txt
}
```

```
# Lataa youtubesta mp3:na Tätä voi käyttää vaikka selaimen user skriptinä
function joutube () {
  cd ~/musat
  youtube-dl --metadata-from-title "%(artist)s - %(title)s" --extract-audio --audi
}
```

```
# Tulostaa kaikki terminaalivärit
alias colors='for code ({000..255}) print -P -- "$code: %F{$code}COLORS%f"'
```

```
#!/usr/bin/env bash
# Notifikaatio SSH:n yli
ssh mjt@tp.turtia.org "tmux splitw 'zsh -c '\'' notify-send \"irssi\" \"Joku lähet
```

```
# Tällä kertaa ei tarvitse edes vaivautua avaamaan .bashrc:tä
# Wikipedia komentorivillä
echo 'function wiki() { \n elinks -dump "https://fi.wikipedia.org/w/index.php?searc
```

EXTRACT.SH

Ei enää `tar --help`

```
#!/usr/bin/env bash
if [[ -f "$1" ]] ; then
   case $1 in
      *.tar.bz2)  tar          xjf   "$1";;
      *.tar.gz)   tar          xzf   "$1";;
      *.bz2)      bunzip2            "$1";;
      *.rar)      rar          x     "$1";;
      *.gz)       gunzip             "$1";;
      *.tar)      tar          xf    "$1";;
      *.tbz2)     tar          xjf   "$1";;
      *.tgz)      tar          xzf   "$1";;
      *.zip)      unzip              "$1";;
      *.Z)        uncompress         "$1";;
      *.tar.xz)   tar          xJf   "$1";;
      *.xz)       xz           -d    "$1";;
      *.7z)       p7zip        e     "$1";;
```

BACKUP-TO-GDRIVE.SH

```
#!/usr/bin/env bash
# Backuppaa koko home-kansiosi Google Driveen
REMOTENAME="gdrive"
FILENAME=$(date +%Y-%m-%d)-home-backup.tar.gz;
# Pakataan koko home
tar cvzf "$FILENAME" "$HOME"
# Kryptataan välissä
gpg -r user-id -e "$FILENAME"
FILENAME=$FILENAME.gpg
# Lähetetään gdriveen
rclone copy "$FILENAME" "$REMOTENAME":backups
# Poistetaan ylimääräinen pakkaus
rm "$FILENAME"
echo "Backupattu $HOME Google Driveen ($REMOTENAME:/backups/$FILENAME)"
```

GIT-APURI.SH

```
#!/usr/bin/env bash
# Nopeat commitit ja pushaus
git add .
git commit -m "$1"
git push origin master
```

LAAJENNETTAVUUS

Lisää toiminnallisuutta muilla kielillä

```
#!/usr/bin/env bash
BASHVAR="Moi Bash"
echo $BASHVAR
python - << EOF
pythonVar = "Moi Python"
print pythonVar
EOF
echo "Moi taas Bash"
```

Lisää toiminnallisuutta "käyttöliittymäkirjastoilla"

```
#!/usr/bin/env bash
OPTIONS="Lock\nReboot\nShutdown"
LAUNCHER="rofi -width 30 -dmenu -i -p rofi-power:"
option=`echo -e $OPTIONS | $LAUNCHER | tr -d '\r\n'`
if [ ${#option} -gt 0 ]; then
    case $option in
    Lock) i3lock ;;
    Reboot) reboot ;;
    Shutdown) systemctl poweroff ;;
    *) ;;
   esac
fi
```

MUUTA

- Multithreadaava Bash-skripti?
- https://github.com/mjturt/dotfiles/tree/master/scripts
- Statusskriptit https://github.com/x70b1/polybar-scripts

Isompia skriptejä

- https://www.passwordstore.org/
- https://github.com/mjturt/dotfiles/blob/master/install.sh

Skriptit kuntoon (esim. Vim integraatio mahdollista)

- https://www.shellcheck.net/

???

pacman -Qi | sed '/^Name/{ s/ //; s/^. //; H;N;d}; /^URL/,/^Build Date/d; /^Install Reason/,/^Description/d; /^ /d;x; s/^.: ... //; s/Jan/01/; s/Feb/02/; s/Mar/03/; s/Apr/04/; s/May/05/; s/Jun/06/; s/Jul/07/; s/Aug/08/; s/Sep/09/; s/Oct/10/; s/Nov/11/; s/Dec/12/; / [1-9]{1} /{ s/[[:digit:]]{1}/0&/3 }; s/(^[[:digit:]][[:digit:]]) ([[:digit:]][[:digit:]]) (.) (....)/\4-\1-\2 \3/' | sed ' /^[[:alnum:]]. $/ N; s/\n/ /; s/(^[[:graph:]]) (.$)/\2 \1/'

KITOS