7 Muutamia Unix-työkaluja

Tässä luvussa esitellään muutamia yleisesti Unix:ssa käytettäviä työkaluohjelmia, joita voidaan käyttää sekä interaktiivisessa shellissä, että komentojonoissa. Komentojen kaikkia optioita ja argumentteja ei ole tässä lueteltu. Lähes kaikista komennoista saa lisätietoa kirjoittamalla `man` *komento* tai mikäli manuaalisivuja ei löydy voi kokeilla *komento -h* tai *komento –help*.

7.1 `At` komento

```
at [optiot] aika
```

`At`:lla voidaan suorittaa komentoja haluttuna ajankohtana (*aika*). `At` lukee suoritettavat komennot `stdin`:sta tai tiedostosta. Mikäli komentojen tulostusta ei uudelleensuunnata lähetetään tuloste sähköpostina käyttäjälle. `At` komennon ymmärtämä (*aika*) riippuu paljolti systeemistä (Unix-järjestelmästä).

Esimerkki.

at_komennot tiedosto

```
ls > at_outfile.txt
pwd >> at_outfile.txt
date >> at_outfile.txt
```

Suoritetaan tiedostossa *at_komennot* olevat komennot `at`:lla minuutin päästä tästä hetkestä.

```
at now + 1 minutes < at_komennot
```

tai

```
at -f at_komennot now + 1 minutes
```

7.2 `Compress` ja `uncompress` komennot

```
compress [optiot] [tiedosto(t)]
```

```
uncompress [optiot] [tiedosto(t)]
```

`Compress` komennolla voidaan pakata tiedostoja, jotta ne veisivät vähemmän levytilaa. `compress` ohjelma pakkaa annetun tiedoston *tiedosto* ja jos mahdollista muuttaa sen nimelle `tiedosto.Z`. Tiedoston omistaja, muutos- ja käyttöajat säilyvät entisellään. Ellei tiedostoa anneta pakataan `stdin` ja kirjoitetaan pakattu data `stdout`:iin. `Compress` komennolla voidaan pakata vain tavallisia tiedostoja, ei esimerkiksi symbolisia linkkejä.

Esimerkki.

```
ls -l *.tar
-rwx------   1 mjev mjev  51200 Oct  3 11:35 tiedosto.tar

compress tiedosto.tar
ls -l tiedosto.tar.Z
-rwx------   1 mjev mjev  21169 Oct  3 11:35 tiedosto.tar.Z
```

`Uncompress` ohjelmalla voidaan purkaa `compress` ohjelmalla pakattuja tiedostoja. `Uncompress` ohjelma lukee annetut tiedostot, joiden pääte on `.Z` ja joiden alussa on oikea "magic number". Ohjelma purkaa tiedostot ja kirjoittaa puretun tiedoston ilman `.Z` päätettä. Puretun tiedoston moodi, omistaja ja ajat ovat samat kuin pakatulla tiedostolla.

Esimerkki.

```
ls -l *.Z
-rwx------   1 mjev mjev  21169 Oct  3 11:35 tiedosto.tar.Z

uncompress tiedosto.tar.Z
ls -l tiedosto.tar
-rwx------   1 mjev mjev  51200 Oct  3 11:35 tiedosto.tar
```

7.3 Crontab

```
crontab [ -u user ] tiedosto
crontab [ -u user ] { -l | -r | -e }
```

`Crontab` komennolla ylläpidetään `cron` demonin suorittamia tehtävälistoja. `Cron` demoni suorittaa käyttäjien ajastamia tehtäviä, samaan tapaan kuin `at` demonikin. `Cron` demonia käytetään silloin, kun jokin tehtävä halutaan suorittaa säännöllisin väliajoin, esimerkiksi kerran viikossa. Halutut tehtävät asetetaan suoritukseen tai poistetaan `crontab` komennolla.

`Crontab` komennon optiot:

-l	Listaa nykyisen *crontab*in näytölle.
-r	Poistaa nykyisen *crontab*in.
-e	Editoidaan nykyistä *crontab*ia. Editoinnin jälkeen uusi työ asennetaan automaattisesti.

Myöskin tiedostoa, jossa `crontab` komennolla asetettava työ määritellään, kutsutaan *crontab*iksi.

Esimerkki *crontab* tiedostosta.

```
# My own crontab
5 2 * * * $HOME/cron/dailyjob >> $HOME/cron/dailyjob.log 2>&1
```

Tämä tarkoittaa, että tiedostossa `$HOME/cron/dailyjob` olevat komennot suoritetaan joka päivä kello 2.05.

Suoritettavien töiden ajankohta kerrotaan siis viidellä ensimmäisellä kentällä, jotka ovat järjestyksessä: *minuutit (0-59), tunnit (0-24), kuukaudenpäivä (0-31), kuukausi (0-12)* ja *viikonpäivä (0-7)*. Joissakin systeemeissä kuukausi ja viikonpäivä voidaan ilmoittaa myös nimellä. Jos johonkin kenttään laitetaan * tarkoittaa tämä kaikkia ajankohtia. Kuudes kenttä on sitten suoritetteva työ, joka normaalisti on tavallinen komentojono.

Ellei suoritettava työn tulosteita uudelleensuunnata lähetetään ne oletuksena käyttäjälle sähköpostina.

Mikäli edellä oleva esimerkki *crontab* tiedosto on talletettu nimelle `crontab.mjev` asetetaan se suoritettavaksi komennolla.

```
crontab crontab.mjev
```

7.4 `Diff` komento

```
diff [optiot] tiedosto1 tiedosto2
```

`diff` komennolla voidaan vertailla kahta tiedostoa riveittäin. Mahdolliset erot tulostetaan `stdout`:iin.

Esimerkki:
`diff /etc/passwd passwd`

7.5 `Exec` komento ja uudelleensuuntaus

`Exec` komennolla voidaan suorittaa komentoja nykyisen shellin paikalla, jolloin komennon suorituksen päätyttyä myös kyseisen shellin suoritus päättyy. Mikäli kyseessä on login shell kirjoittaudutaan järjestelmästä ulos kun komennon suoritus loppuu.

```
exec ls
```

Bourne ja Korn shelleissä voidaan `exec` komentoa käyttää tulostuksen ja syötteen uudelleensuuntaukseen. `Exec` komennolla voidaan myös avata tiedostoja lukemista tai kirjoittamista varten.

exec < tiedosto	Avataan tiedosto *tiedosto* syötteen lukemista varten. `stdin` otetaan tämän jälkeen tiedostosta *tiedosto*.
exec > tiedosto	Avataan tiedosto *tiedosto* tulostusta varten. `stdout` ohjataan tämän jälkeen tiedostoon *tiedosto*.
exec 3< data	Avataan tiedosto *data* tiedostokuvaajaan 3 syötteen lukemista varten.
`cat <&3`	Tulostetaan tiedosto *data* `cat` komennolla.
exec 4> tuloste	Avataan tiedosto *tuloste* kirjoitusta varten tiedostokuvaajaan 4.
`pwd >&4`	Ohjataan komennon `pwd` tulosteet tiedostoon *tuloste*.
exec 5<&4	Kopioidaan tiedostokuvaaja 4 tiedostokuvaajaan 5.
exec 3<&-	Suljetaan tiedostokuvaaja 3.

```
#!/bin/sh
#
# Ohjataan taman komentojono tuloste ja virhe tiedostoon out_file.
#
exec > out_file 2>&1

echo 'Terve maailma'
koe
ls -la
```

7.6 `Find` komento

```
find [polku] [optiot]
```

`Find` komennolla voidaan etsiä tiedostoja annetuista hakemistoista *polku*. Annetuista poluista tutkitaan oletuksena myös alihakemistot. Optioilla (`optiot`) voidaan asettaa erilaisia hakuehtoja. Hakuehto voi olla esimerkiksi: nimi, koko, omistaja, oikeudet, muutosaika jne.

Esimerkkejä:
`find . -name \*.tex`
Etsitään nykyhakemistosta (ja alihakemistoista) kaikki *.tex* päätteiset tiedostot.

`find /etc /var -type d`
Etsitään hakemistoista `/etc` ja `var` kaikki hakemistot (-type d).

`find ~ -size 0`
Etsitään käyttäjän kotihakemistosta kaikki tiedostot, joiden koko on nolla.

`find ~ -size 0 -exec rm {} \;`
Etsitään ja tuhotaan käyttäjän kotihakemistosta kaikki tiedostot, joiden koko on nolla.

`-exec` optiolla voidaan löytyneille tiedostoille suorittaa mikä tahansa Unix komento. Löytyneisiin tiedostoihin viitataan `{}` ja `-exec` optio päättyy aina `\;` merkkeihin.

7.7 `Grep`, `egrep` ja `fgrep` komennot

`grep [optiot] regexp [tiedosto(t)]`

`Grep` komennolla voidaan hakea tietyn merkkijonon (`regexp` regular expression) sisältäviä rivejä joukosta tiedostoja. Ne rivit, joihin `regexp` täsmää tulostetaan. `Egrep` komento (extented grep) käyttää laajennettuja `regexp` metamerkkejä. `Fgrep` komento (fast grep) tulkitsee kaikki `regexp` metamerkit sellaisenaan.

`Grep` komennon ymmärtämät `regexp` merkit:

`^`	Tarkoittaa rivin alkua.
`$`	Tarkoittaa rivin loppua.
`.`	Tarkoittaa yhtä merkkiä.
`[]`	Tarkoittaa yhtä hakasulkujen sisällä olevaa merkkiä.
`[^]`	Tarkoittaa yhtä merkkiä, joka ei ole hakasulkujen sisällä.
`\<`	Tarkoittaa sanan alkua.
`\>`	Tarkoittaa sanan loppua.
`\(...\)`	Merkkien sisällä olevaan tasmänneeseen rivin osaan voidaan viitata numerolla `\1` jne.

Lisäksi `egrep` komento tuntee seuraavat `regexp` merkit:

`+`	Tarkoittaa yhtä tai useampaa edeltävää merkkiä.
`?`	Tarkoittaa nollaa tai yhtä edeltävää merkkiä.
`x|y`	Tarkoittaa x:ää tai y:tä.
`()`	Merkkien ryhmittely.

Esimerkkejä:
`grep 'sh$' /etc/passwd`
Tulostetaan `/etc/passwd` tiedostosta ne rivit, jotka päättyvät merkkijonoon *sh*.

`grep [Tt]ar luento6.tex`
Tulostetaan `luento6.tex` tiedostosta ne rivit, joilla esiintyy sana *tar* tai *Tar*.

7.8 `Man` komento

`man [optiot] komento`

`Man` komentoa käytetään järjestelmästä löytyvien manuaalisivujen formatoimiseen ja lukemiseen. Unix-järjestelmässä lähes kaikki komennot (ja systeemikutsut) on dokumentoitu järjestelmästä löytyvien manuaalisivujen avulla. Vihjeenä voi sanoa, että mikäli haluaa oppia käyttämään Unix-järjestelmää tehokkaasti on opeteltava käyttämään komentojen manuaalisivuja.

Esimerkki:

`man man`
Näyttää käyttöohjeen `man` komennon käytöstä.

7.9 (GNU)`Nohup komento`

`nohup komento [argumentit]`

`Nohup` komennolla voidaan suorittaa muita Unix komentoja siten, että ajetut komennot eivät reagoi keskeytykseen (hangup). `Nohup` komentoa kannattaa käyttää erityisesti silloin kun halutaan jättää jokin komento suoritukseen uloskirjoittautumisen jälkeenkin.

Mikäli `stdout` on *tty* suunnataan se ja `stderr` tiedostoon `nohup.out`. Mikäli kyseistä tiedostoa ei voida luoda yritetään luoda `$HOME/nohup.out`. Ellei sekään onnistu ei komentoa suoriteta.

Oletuksena `nohup` ei laita komentoa taustasuoritukseen vaan komento on päätettävä & merkkiiin.

7.10 `Ps` komento

`ps [optiot]`

`Ps` komennolla saadaan tietoa aktiivisena olevista prosesseista. Ilman argumentteja `ps` tulostaa tietoa prosesseista jotka liittyvät kontrolliterminaaliin. `Ps` komennon optiot vaihtelevat Unix-järjestelmän mukaan.

Esimerkkejä:
`ps -u mjev`
Tulostaa tietoa käyttäjän *mjev* prosesseista.

`ps aux  # ucb Unix`
tai `ps -ef  # at&t Unix`
Tulostaa tiedot kaikista prosesseista.

7.11 `Sleep` komento

`sleep aika`

`Sleep` komennolla voidaan keskeyttää esimerkiksi komentojonon suoritus tietyksi ajaksi. Oletuksena *aika* on sekunteja, jotkin versiot hyväksyvät myös option, jolla kerrotaan mitä aikayksikköä käytetään *smhd*.

Esimerkki:

`( sleep 10; pwd; )&`
Suoritetaan komento `pwd` 10 sekunnin kuluttua.

7.12 `Sort` komento

```
sort [optiot] [tiedosto(t)]
```

`Sort` komento järjestää (ASCII) rivejä kaikista annetuista tiedostoista ja kirjoittaa tuloksen `stdout`:iin. Järjestäminen tehdään annetun tai annettujen avainten mukaan. Oletuksena käytetään koko riviä avaimena.

Esimerkkejä:
`sort /etc/passwd`
Järjestetään `/etc/passwd` tiedosto aakkosjärjestykseen.

`sort -t: -k4n /etc/passwd`
Järjestetään `/etc/passwd` tiedosto, kentästä neljä lähtien numeerisesti käyttäen kenttäerottimena ":" merkkiä.

7.13 `Strings` komento

```
strings [optiot] [tiedosto(t)]
```

`Strings` komento etsii ASCII merkkijonoja binaaritiedostoista ja tulostaa ne `stdout`:iin. Merkkijono on mikä tahansa vähintään neljän merkin jono, joka koostuu tulostettavista merkeistä ja päättyy joko rivinvaihtoon tai "null" merkkiin.

Esimerkki:
`strings /bin/tar`
Komento tulostaa merkkijonot, joita löytyy tiedostosta `/bin/tar`.

7.14 `(GNU)tar` komento

```
tar [optiot] tiedostot
```

Muutamia `tar` komennon optioita:

-c	Luodaan uusi tar-tiedosto.
-r	Lisätään tiedostoja tar-tiedostoon.
-t	Listataan tar-tiedosto.
-x	Puretaan tiedostot tar-tiedostosta.
-f	Tar-tiedosto, joka voi olla esimerkiksi tavallinen tiedosto tai nauha.

`Tar` (tape archiver) komentoa käytetään joko kokoamaan tai purkamaan tiedostoja "tar-tiedostoksi". Tiedostot voidaan koota suoraan esimerkiksi nauhalle tai normaaliksi tiedostoksi. Unix-järjestelmissä on yleensä omat versionsa `tar` komennosta, joten on syytä tutustua kyseisen järjestelmän `tar` komennon manuualisivuihin.

Esimerkkejä.

`tar -cf home.tar ~`
Luodaan käyttäjän kotihakemistosta alihakemistoineen "tar-tiedosto" nimelle `home.tar`.

`tar -tvf home.tar`
Listataan tiedoston `home.tar` sisältämät tiedostot.

`tar -cv . | (cd /tmp/test; tar -x;)`
Kopioidaan nykyisen hakemiston tiedostot ja alihakemistot hakemistoon `/tmp/test` (Huom! hakemiston `/tmp/test` pitää olla olemassa).

`tar -cv . | ssh mjev@jane 'cd test/; tar -x;'`
Kopioidaan nykyhakemisto alihakemistoineen koneeseen *jane* tunnuksen *mjev* kotihakemiston alihakemistoon `test`.
Huom. Tämä edellyttää, että tämän koneen ja *jane* koneen välille voidaan muodostaa `ssh` yhteys.

7.15 Touch komento

```
touch [optiot] tiedosto(t)
```

`Touch` komennolla voidaan muuttaa argumentteina annettujen tiedostojen käyttö- ja muutosaikoja. Mikäli annettua tiedostoa ei ole olemassa se luodaan. Mikäli aikaa ei anneta käytetään nykyistä aikaa.

Esimerkkejä:

```
touch abc

stat abc
```

```
  File: "abc"
  Size: 0            Filetype: Regular File
  Mode: (0664/-rw-rw-r--)  Uid: (21056/ mjev)  Gid: (21056/ mjev)
Device:  3,0   Inode: 135340    Links: 1
Access: Sun Oct  3 19:14:52 1999(00000.00:00:08)
Modify: Sun Oct  3 19:14:52 1999(00000.00:00:08)
Change: Sun Oct  3 19:14:52 1999(00000.00:00:08)
```

Laitetaan tiedoston *abc* käyttö- ja muutosaika tähän hetkeen. Jos tiedostoa *abc* ei ole olemassa se luodaan.

```
touch 05052354 abc

stat abc

  File: "abc"
  Size: 0            Filetype: Regular File
  Mode: (0664/-rw-rw-r--)  Uid: (21056/ mjev)  Gid: (21056/ mjev)
Device:  3,0   Inode: 135340    Links: 1
Access: Wed May  5 23:54:00 1999(00150.19:23:39)
Modify: Wed May  5 23:54:00 1999(00150.19:23:39)
Change: Sun Oct  3 19:17:25 1999(00000.00:00:14)
```

Laitetaan tiedoston *abc* käyttö- ja muutosajoiksi 5.5 kello 23:54.

7.16 `Wc` komento

```
wc [optiot] [tiedosto(t)]
```

`Wc` komennolla tulostetaaan argumenttina annetun tiedoston rivit, sanat ja merkit.

Esimerkki:
`wc /etc/passwd`