

UNIX-käyttöjärjestelmä

Luento 3

Timo Savola

<timo.savola@evtek.fi>

7. huhtikuuta 2006

Osa I

Käyttöoikeudet

Käyttöoikeudet

- ▶ Monen käyttäjän järjestelmä
 - ▶ Prosessit kuuluvat eri käyttäjille
 - ▶ Tiedostot kuuluvat eri käyttäjille/käyttäjärhmille
- ▶ Käyttöoikeudet tarkastetaan jokaisen järjestelmäkutsun yhteydessä
 - ▶ Signaaleita voi lähettää vain saman käyttäjän prosesseille

Käyttäjät

- ▶ Järjestelmä erottaa käyttäjät toisistaan numerolla (user ID, uid)
- ▶ Kirjaututtaessa käyttäjän tiedot haetaan käyttäjätunnuksella
 - ▶ Tiedostosta /etc/passwd
 - ▶ LDAP-palvelimelta
- ▶ Shell-prosessin omistajaksi tulee tunnuksen liitetty uid

Käyttäjoryhmät

- ▶ Käyttäjä kuuluu yhteen tai useampaan ryhmään:
 - ▶ Ensisijainen ryhmä
 - ▶ Ylimääräiset ryhmät (supplementary groups)
- ▶ Ryhmillä on numerot (group ID, gid)
- ▶ `id` -komento tulostaa mm. käyttäjän ryhmät
 - `id`
 - `id tunnus`

Pääkäyttäjä














- ▶ Super-user
 - ▶ uid on 0
 - ▶ Käyttäjätunnus on "root"
- ▶ Voi tehdä (melkein) mitä tahansa

Tiedostojen omistajat ja suojaukset

- ▶ Omistajuus

- ▶ Käyttäjä
- ▶ Ryhmä

- ▶ Suojausbitit (mode)

	Luku	Kirjoitus	Suoritus	Set-ID	Sticky
Käyttäjä					
Ryhmä					
Muut					

- ▶ Hakemistojen yhteydessä *suoritus* tarkoittaa “voi käyttää”
- ▶ *Sticky* rajoittaa hakemiston sisällön muokkaamista

Ohjelmien suoritus

- ▶ Ohjelmatiedoston suoritus sallitaan vertaamalla sen ja prosessin tietoja keskenään
 1. Sama käyttäjä omistaa molemmat, ja tiedosto on käyttäjän suoritettavissa?
 2. Joku prosessin ryhmistä on sama kuin tiedoston ryhmä, ja tiedosto on ryhmän suoritettavissa?
 3. Tiedosto on muiden suoritettavissa?
- ▶ Jos tiedostolla on *set-uid* tai *set-gid* bitti, prosessin omistajuus muutetaan vastaamaan tiedoston omistajaa
 - ▶ Hallittu tapa sallia pääsy pääkäyttäjän toimintoihin

Komentoja

- ▶ `chown` muuttaa tiedoston omistajaa (CHange OWNership)

`chown käyttäjä:ryhmä polku`

- ▶ `chmod` muuttaa tiedoston suojausta (CHange MODe)

`chmod kohde=arvo polku`

`chmod kohde+arvo polku`

`chmod kohde-arvo polku`

kohde on yhdistelmä kirjaimista:

`u` käyttäjä

`g` ryhmä

`o` muut

`a` kaikki

arvo on yhdistelmä kirjaimista:

`r` luku

`w` kirjoitus

`x` suoritus

`s` set-uid tai set-gid

`t` sticky

Osa II

Tiedostot

Tiedostotyypit

- ▶ Tavallisten tiedostojen (regular file) sisältöön ei oteta kantaa
- ▶ Erikoistiedostojen sisältö on tarkkaan määritelty ja niitä käsitellään joskus erikoistapauksina
 - ▶ Symbolinen linkki (symlink)
 - ▶ Laitetiedosto (device)
 - ▶ Nimetty putki (FIFO, named pipe)
 - ▶ Socket

Linkit

- ▶ Linkki on viittaus tiedoston sisältöön
 - ▶ Käyttäjä näkee tiedostopolkuna
 - ▶ Joskus käytetään nimitystä “hard link”
- ▶ Symbolinen linkki on epäsuora viittaus tiedostoon
 - ▶ Erikoistiedosto, joka sisältää polun
 - ▶ Symlinkin avaaminen avaa kohdetiedoston
 - ▶ Kohdetiedostoa ei välttämättä ole olemassa (dangling link)
- ▶ Symlinkin suojauksia ei oteta huomioon
 - ▶ Symlinkin kohde määrittelee käyttöoikeudet

Piilotiedostot

- ▶ Pisteellä alkavat tiedostonimet ovat “piilotiedostoja”
 - ▶ Ei näytetä oletuksena tiedostolistauksissa
- ▶ Voidaan listata komennolla “ls -a”

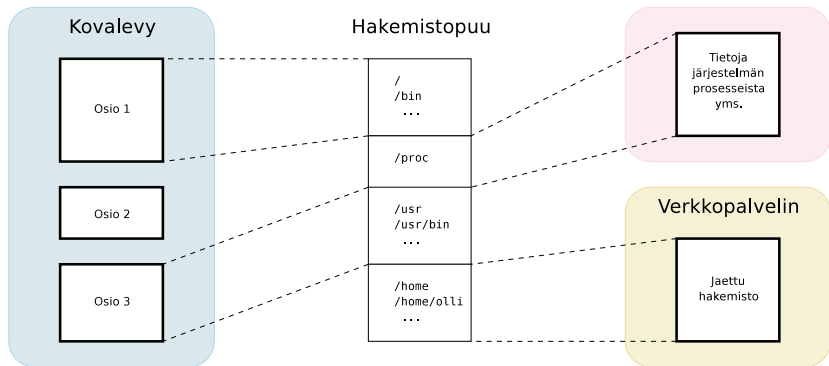
Laitetiedostot

- ▶ Nimityksiä:
 - ▶ Device
 - ▶ Special file
- ▶ Erikoistiedosto, joka viittaa järjestelmän resurssiin
 - ▶ Sisältää kaksi numeroa joilla laitteet erotetaan toisistaan
 - ▶ Major-numero
 - ▶ Minor-numero
 - ▶ Sisältää laitteen tyyppin
 - ▶ Puskuroimaton “merkkilaite” (character special file)
 - ▶ Puskuroitu “lohkolaite” (block special file)
- ▶ Laitetiedostot sijaitsevat yleensä /dev-hakemiston alla

Komentoja

- ▶ cp kopioi tiedoston (CoPy)
`cp vanha uusi`
- ▶ mv siirtää tiedoston uudelle nimelle (MoVe)
`mv vanha uusi`
- ▶ rm poistaa tiedoston (ReMove)
`rm polku`
- ▶ ln luo uusia linkkejä (LiNk)
`ln kohde linkki`
`ln -s kohde symlinkki`
- ▶ stat kertoo tietoja tiedostosta (STATus)
`stat polku`
`stat -L symlinkki`
- ▶ ls -l kertoo tietoja hakemistossa olevista tiedostoista
`ls -l`
`ls -la`

Tiedostojärjestelmät



- ▶ Tiedostot sijaitsevat tiedostojärjestelmissä
- ▶ Eri tiedostojärjestelmät palvelevat eri tarkoituksia
 - ▶ Normaalisti käytetään kovalevyn tms. kapasiteettia
 - ▶ Verkon yli jaetut hakemistot
 - ▶ Tilapäistiedostojen tallennus muistiin
 - ▶ Synteettiset/virtuaaliset tiedostot

Levyt

- ▶ Kovalevyt ja niiden osiot näkyvät laitetiedostoina
 - ▶ Block special file
- ▶ Esimerkkejä laitetiedostoista:
 - ▶ /dev/hda – Ensimmäinen IDE-levy
 - ▶ /dev/hda1 – Ensimmäisen IDE-levyn ensimmäinen osio
 - ▶ /dev/sdb3 – Toisen SCSI- tai SATA-levyn kolmas osio
- ▶ Joitain kovalevyillä käytettyjä tiedostojärjestelmiä:
 - ▶ ext3 (Linux)
 - ▶ reiserfs (Linux)
 - ▶ XFS (IRIX, Linux)
 - ▶ JFS (AIX, Linux)

Mount

- ▶ Tiedostojärjestelmät liitetään osaksi hakemistopuuta `mount`-komennolla
- ▶ Parametrit:
 - ▶ Tiedostojärjestelmän tyyppi
 - ▶ Tiedostojärjestelmän sijainti (yleensä laitetiedosto)
 - ▶ Kohdehakemisto
 - ▶ Mahdolliset optiot
- ▶ `/etc/fstab` -tiedostossa luetellut tiedostojärjestelmät mountataan käynnistysvaiheessa
- ▶ `df` -komento listaa kaikki hakemistot joihin on liitetty tiedostojärjestelmä (mount point)
 - ▶ "`df -h`" kertoo osioiden koot ihmisenluettavasti
 - ▶ "`df -a`" näyttää kaikki tiedostojärjestelmät

Osa III

Shell

Sisäänrakennetut komennot

- ▶ Shellissä on sisäänrakennettuja toimintoja joita käytetään samalla tavalla kuin ulkoisia ohjelmia
- ▶ Jos annettu komento ei sisällä polkua, etsitään sitä shellin toiminnoista ennen ohjelmahakemistoja
- ▶ Jotkut toiminnot ovat shellissä suorituskykyisiä:
 - ▶ `echo`
- ▶ Osa toiminnoista ei voisi toteuttaa shellin ulkopuolella:
 - ▶ `cd` (vaihtaa shell-prosessin työhakemistoa)
 - ▶ `exit` (lopettaa shell-prosessin)
- ▶ Sisäänrakennetun toiminnon voi ohittaa antamalla polun
 - ▶ `/bin/kill`

Parametrit ja optiot

- ▶ Ohjelma voi käyttää minkälaisia parametreja tahansa, mutta yleensä seurataan vakiintunutta käytäntöä:
 - ▶ Viivalla (“-”) alkavat argumentit tulkitaan optioiksi
 - ▶ Muut argumentit tulkitaan parametreiksi
- ▶ Optiosta voi olla lyhyt ja pitkä versio:
 - h
 - help
- ▶ Jotkut optiot tarvitsevat parametrin:
 - f *param*
 - f *param*
 - file=*param*
 - file *param*
- ▶ Parametrittomia lyhyitä optioita voidaan ryhmitellä:
 - a -b ⇒ -ab
- ▶ Kaksi viivaa (“--”) ilman muita merkkejä tarkoittaa, että loput argumentit tulkitaan parametreiksi:
 - ▶ “mv -foo bar” ei toimi
 - ▶ “mv -- -foo bar” toimii

Ympäristömuuttujat

- ▶ Ohjelmilla on *ympäristö* johon tehdyt muutokset periytyvät lapsiprosesseille ja käynnistetyille ohjelmille:
 - ▶ Lista muuttujista ja niiden arvoista
 - ▶ Muuttujien nimet on tapana kirjoittaa isolla
- ▶ Esimerkkejä:
 - ▶ PATH listaa ohjelmahakemistot
 - ▶ HOME sisältää käyttäjän kotihakemiston polun
 - ▶ SHELL sisältää käyttäjän shell-ohjelman polun
- ▶ Shellissä muuttujan arvoa voi käyttää kirjoittamalla `$NIMI`
 - ▶ “echo \$PATH” tulostaa muuttujan arvon
- ▶ `env` -komento listaa kaikki ympäristömuuttujat

Muuttujat

- ▶ Shellin muuttujat eivät oletuksena näy käynnistetyille ohjelmille
 - ▶ Asetetaan kirjoittamalla *NIMI=arvo*
 - ▶ Käyttö ei eroa ympäristömuuttujien käytöstä
 - ▶ Voidaan “korottaa” ympäristömuuttujaksi `export` -komennolla
- ▶ Muuttuja voidaan poistaa `unset` -komennolla
- ▶ Ympäristömuuttuja voidaan asettaa vain yhdelle ohjelmalle kirjoittamalla:

```
MUUTTUJA=arvo komento
```

- ▶ Esimerkkejä:

```
F00=xyz
```

```
export F00
```

```
unset F00
```

```
export PATH=$PATH:$HOME/bin
```

```
POSIXLY_CORRECT=1 /bin/df
```

Globbaus

- ▶ Erikoismerkkien avulla voidaan etsiä tiedostoja:
 - * vastaa tahansa merkkijonoa
 - ? vastaa mitä tahansa yhtä merkkiä
 - [...] vastaa mitä tahansa kirjainta mikä on lueteltu sulkeiden sisällä
- ▶ Tällaisia merkkijonoja kutsutaan nimellä glob pattern
- ▶ Esimerkkejä:
 - ▶ “*.txt” etsii kaikki .txt-päätteiset tiedostot
 - ▶ “unix[12].pdf” etsii tiedostoja unix1.pdf ja unix2.pdf
 - ▶ “abc?” etsii kaikkia abc-alkuisia tiedostoja joissa on 4 merkkiä
- ▶ Ohjelmien parametrit kirjoitetaan monesti glob-muodossa
 - ▶ Löytyneet polut annetaan ohjelmalle erillisinä parametreina

Suojaus

- ▶ Komento tai sen osa halutaan joskus "suojata" siten että shell ei muuta sitä
 - ▶ \$ -alkuiset merkkijonot tulkitaan muuttujiksi
 - ▶ * ja ? tulkitaan globbaukseksi
 - ▶ Välilyönti tulkitaan erottimeksi
- ▶ Suojausmerkit:
 - \ suojaa seuraavan merkin
 - '... ' suojaa välissä olevat merkit
 - "..." suojaa lukuunottamatta merkkejä \$, \ ja '
- ▶ Esimerkkejä:

```
ls "Mun hakemisto"  
echo \$ on dollarimerkki  
echo '*** VAROITUS ***'
```