

# UNIX-käyttöjärjestelmä

## Luento 2

Timo Savola

`<timo.savola@evtek.fi>`

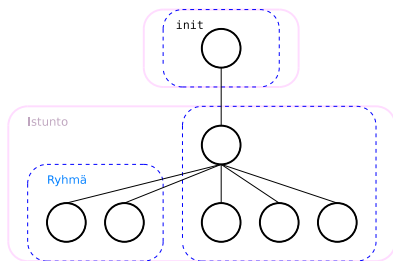
31. maaliskuuta 2006

Osa I

Prosessit

# Prosessien hierarkia

- ▶ Jokaisella prosessilla on numero (process ID, pid)
- ▶ Jokaisella prosessilla on vanhempi prosessi (parent process)
- ▶ `init`-prosessin numero on 1 ja se on kaikkien esi-isä
- ▶ Prosessit kuuluvat ryhmiin (process group)
  - ▶ Ryhmän numero on sen johtajaprosessin numero
- ▶ Prosessiryhmä voi olla istunto (session)
  - ▶ Istunnon numero on sen ryhmänjohtajan numero
  - ▶ Sisäänkirjautuminen aloittaa uuden istunnon

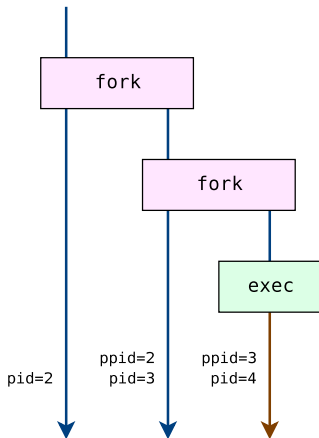


# Prosessin luonti

- ▶ Uusi prosessi luodaan kopiaamalla vanha prosessi (fork)
  - ▶ Prosessin omistamat resurssit kopioituvat (näennäisesti)
- ▶ Vanhasta prosessista tulee uuden prosessin parent-prosessi
- ▶ Molemmat prosessit jatkavat saman ohjelmakoodin suoritusta
  - ▶ Ohjelma voi tunnistaa kumpi prosessi suorittaa sitä

# Ohjelman suorittaminen

- ▶ Prosessi voi vaihtaa suoritettavaa ohjelmaa (exec)
- ▶ Vanha prosessi ei häviä, ainoastaan ohjelma muuttuu



# Proessin kuolema

- ▶ Mahdolliset syyt:
  - ▶ Ohjelma loppuu (päättää lopettaa suorituksen)
  - ▶ Ohjelma saa signaalin ja päättää lopettaa suorituksen
  - ▶ Ohjelma saa tappavan signaalin
- ▶ Parent-prosessi saa tiedon kuolemasta
  - ▶ Syy
  - ▶ Paluukoodi
- ▶ Jos kuolevalla prosessilla on lapsiprosesseja, niiden uudeksi parentiksi tulee `init`-prosessi

# Signaalit

- ▶ Yksinkertainen tapa viestiä prosessien välillä
- ▶ Mahdolliset kohteet:
  - ▶ Prosessi
  - ▶ Prosessiryhmä
  - ▶ Kaikki prosessit paitsi `init`
- ▶ Mahdolliset lähettäjät/aiheuttajat:
  - ▶ Prosessi
  - ▶ Kerneli
- ▶ Mahdolliset seuraamukset kohdeprosessille:
  - ▶ Ei huomioi signaalia
  - ▶ Suorittaa signaalinkäsittelyfunktion
  - ▶ Ohjelman suoritus keskeytyy toistaiseksi
  - ▶ Ohjelman suoritus jatkuu (jos keskeytetty)
  - ▶ Prosessi kuolee
  - ▶ Muistin sisältö kirjoitetaan tiedostoon ja prosessi kuolee

# Signaalit

- ▶ 28 standardia ennaltamääriteltyä signaalia

HUP Hangup

INT Interrupt

KILL Kill (ehdoton)

SEGV Segment violation

PIPE Broken pipe

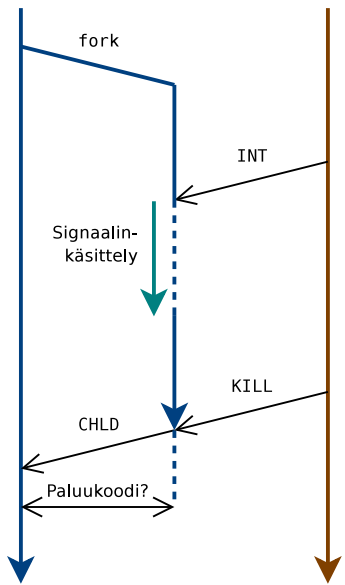
TERM Terminate

CHLD Child stopped or terminated

CONT Continue

STOP Stop (ehdoton)





Osa II

Shell

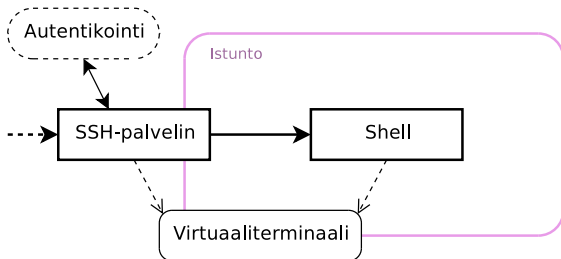
# Shell

- ▶ Shell on tavallinen ohjelma jolla on erityisasema
- ▶ UNIX-järjestelmän peruskäyttöliittymä:
  1. Lue suoritettava komento
  2. Suorita komento
  3. Näytä lopputulos
  4. Palaa kohtaan 1.

# Shellit

- ▶ Jokainen käyttäjä voi valita mieleisensä shell-ohjelman
- ▶ Yleisiä shellejä:
  - ▶ `bash` (Bourne Again Shell)
  - ▶ `zsh`
  - ▶ `tsh`
  - ▶ `ksh` (Korn Shell)
- ▶ POSIX määrittelee shellien perustoiminnallisuuden

# Kirjautuminen

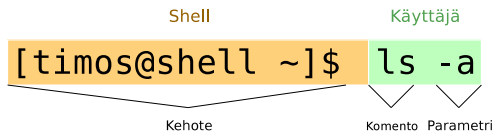


- ▶ Jokin ohjelma lukee käyttäjätunnuksen ja salasanan jostain:
  - ▶ Konsoli
  - ▶ Verkkoyhteys
- ▶ Selvitetään käyttäjän tiedot
  - ▶ Käyttöoikeudet
  - ▶ Shell-ohjelma
  - ▶ Kotihakemisto
- ▶ Shell suoritetaan uudessa istunnossa

# Terminaalit

- ▶ Mekanismi tiedon syöttämiseen ja tulostamiseen erilaisten resurssien avulla:
  - ▶ Konsoli (monitori, näppäimistö)
  - ▶ Sarjaportti
  - ▶ Virtuaaliterminaalit
- ▶ Shell-istunto käyttää jonkinlaista terminaalia
  - ▶ Konsolien tapauksessa shell käyttää sitä terminaalia miltä kirjautumisohjelma luki salasanan
  - ▶ SSH-palvelimen tapauksessa ei ole fyysistä resurssia joten luodaan virtuaaliterminaali
- ▶ Ohjelmat käyttävät terminaaleja kuten avoimia tiedostoja
  - ▶ Shellin käynnistämät ohjelmat perivät tiedostoviittaukset
  - ▶ Standardit tulostusfunktiot kirjoittavat terminaaliin jne.

# Komentorivi



- ▶ Shell tulostaa komentokehotteen ja odottaa komentoja
  - ▶ Kehote voi sisältää kaikenlaista tietoa:
    - Käyttäjän nimi (`timos`)
    - Koneen nimi (`shell`)
    - Työhakemisto (`~` tarkoittaa kotihakemistoa)
  - ▶ `$` viittaa joissain shelleissä normaalikäyttäjään
- ▶ Käyttäjä syöttää ohjelman nimen ja mahdolliset parametrit
  - ▶ Välilyönti toimii erottimena
- ▶ Shell luo lapsiprosessin ja suorittaa halutun ohjelman
- ▶ Shell tulostaa uuden komentokehotteen lapsiprosessin kuoltua

# Komennot

- ▶ Komento voi olla:
  - ▶ Nimi (`ls`)
  - ▶ Polku (`/bin/ls`)
- ▶ Jos annetaan pelkkä nimi, ohjelmatiedosto etsitään järjestelmän ohjelmahakemistoista
  - ▶ Ohjelman polun voi selvittää komennolla:  
`which nimi`



# Polut

- ▶ “/” on hakemistoerotin
  - ▶ Absoluuttiset polut alkavat /-merkillä
  - ▶ Suhteelliset polut muutetaan absoluuttisiksi yhdistämällä ne työhakemistoon
- ▶ Poluissa voidaan käyttää erikoishakemistoja:
  - “.” viittaa kyseiseen hakemistoon
  - “..” ylempään hakemistoon
- ▶ Esimerkkejä poluista (työhakemisto /users/hkunta/timos):

tiedosto ⇒ /users/hkunta/timos/tiedosto

.. ⇒ /users/hkunta

../../../../ ⇒ /

/bin/../../etc ⇒ /etc

/.. ⇒ /

/bin/. ⇒ /bin

./bin ⇒ /bin

- ▶ Työhakemistossa oleva ohjelmatiedosto suoritetaan käyttämällä suhteellista polkua:

./ohjelma

# Hakemistokomentoja

- ▶ `pwd` kertoo nykyisen työhakemiston (Print Working Directory)
- ▶ `ls` listaa hakemiston sisällön (LiSt)

```
ls
```

```
ls polku ...
```

- ▶ `cd` vaihtaa työhakemistoa (Change Directory)

```
cd polku
```

- ▶ `mkdir` luo uuden hakemiston (MaKe DIRectory)

```
mkdir polku ...
```

- ▶ `rmdir` poistaa tyhjän hakemiston (ReMove DIRectory)

```
rmdir polku ...
```

# Prosessien käsittely

- ▶ `ps`-komento listaa käynnissä olevia prosesseja
  - ▶ “`ps u`” näyttää kyseisen istunnon prosessit
  - ▶ “`ps ux`” näyttää kaikki käyttäjän prosessit
  - ▶ “`ps aux`” näyttää kaikki koneen prosessit
- ▶ `kill`-komento lähettää prosessille signaalin
  - ▶ Oletuksena lähetetään `TERM`-signaali
  - ▶ Kohdeprosessi määritellään kirjoittamalla sen numero (`pid`)
  - ▶ Signaalin voi valita kirjoittamalla sen symbolisen nimen tai numeron muodossa “*-signaali*”
  - ▶ Esimerkkejä:

```
kill 1000
kill -INT 1000
```
- ▶ `killall`-komento lähettää signaalin kaikille samannimisille prosesseille

```
killall -KILL bash
```

# Komentorivin toimintoja (bash)

## Navigointi

← → Liiku komentorivillä

Home Ctrl-A Rivin alkuun

End Ctrl-E Rivin loppuun

## Historia

↑ ↓ Selaa vanhoja komentorivejä

Ctrl-R Etsi vanhaa komentoriviä hakusanalla

## Täydennys

Tab Täydentää komennon nimen tai listaa vaihtoehdot

# Istunnon lopetus

- ▶ Shellistä poistutaan `exit`-komennolla tai painamalla `Ctrl-D` tyhjällä komentorivillä
- ▶ Istunto voidaan lopettaa myös `logout`-komennolla