



Aalto-yliopisto
Sähkötekniikan
korkeakoulu

Linuxin perusteet

ELEC-D0301 Protopaja
Juha Biström

Luennon aiheita

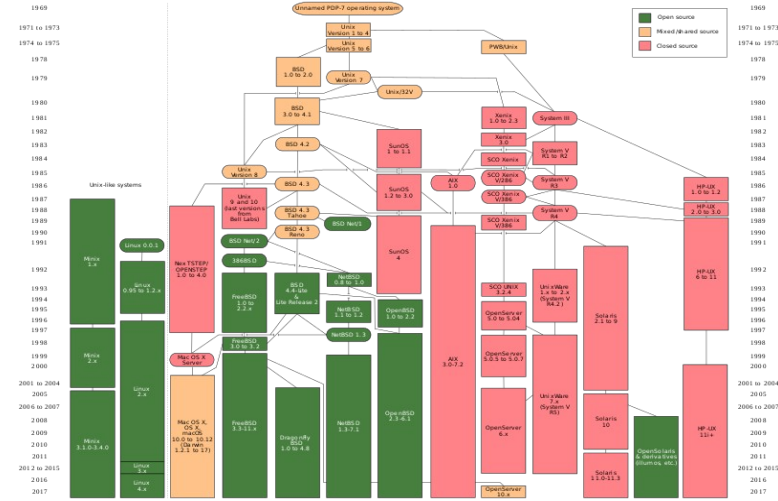
- Linuxin historiaa
- Yleistä rakenteesta
- Käyttäjät, ryhmät; tiedostojärjestelmä
- Sovellukset, pakettienhallinta
- Komentorivi, etäyhteys

Historiaa

- UNIX
- Richard Stallman (Harvard, MIT), GNU (GNU's Not Unix!) ja FSF (Free Software Foundation)
- Linus Torvalds ja Linux (1991 →)

UNIX

- Useita eri käyttöjärjestelmiä
- Luontaisesti modulaarisia moniajavia monen käyttäjän käyttöjärjestelmiä
- Kehittäminen alkanut 1970-luvulla Bell Labsilla
- Esim. BSD, AIX, Solaris, MacOS X



GNU

- Richard Stallman aloitti 1983, nykyään FSF:n hallinnoima
- Kyllästyi käyttämään huonoja työkaluja, joita ei ollut mahdollista parannella itse
- Aiempien sovellusten parantelu yksitellen, kunnes koko järjestelmä on valmis

Linuxin synty

- Linus Torvalds (Helsingin yliopisto Tietojenkäsittelyn laitos) halusi käyttää yliopistolta tuttua UNIX-järjestelmää myös kotona
- UNIXit olivat liian kalliita, joten teki oman toteutuksen
- Ensimmäinen versio syksyllä 1991, jaettu keskusteluryhmässä
- Hyödyntänyt GNU-projektin työkaluja aikaisesta vaiheesta

Avoin lähdekoodi

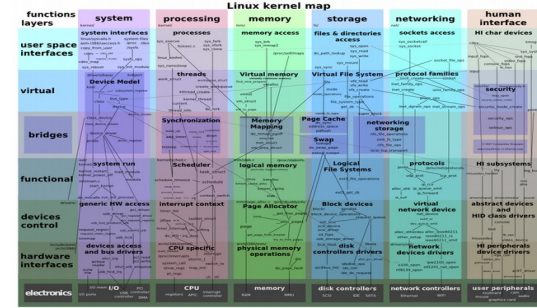
- Linuxin keskeisiä ja kantavia ajatuksia
- Mahdollistaa kaikille muutoksien tekemisen, esim. virheiden korjaamisen, ominaisuuksien parantelun
- Lisensointi asettaa tiettyjä ehtoja ja takaa oikeuksia edelleenlevittämiselle ja hyödyntämiselle
- Merkittäviä avoimen lähdekoodin lisenssejä (L)GPL, MIT, Apache, Mozilla, jnejne...

“Linuxin” rakenne

- Oikeastaan “Linux-käyttöjärjestelmä” on runsas valikoima erilaisia jakelupaketteja (distribution, distro), ikäänkuin eri valmistajien versioita samasta järjestelmästä, vrt. autovalmistajat
- Jakelupaketti on julkaisijansa näkemys toimivasta järjestelmästä
 - Optimoituja eri tarkoituksiin
 - Pitää sisällään ainakin järjestelmän ytimen, kernelin (kernel)
 - Järjestelmätyökaluja, “hyötyohjelmia”, graafisen työpöytäympäristön, ...



Ydin, kernel



- Varsinainen Linux, UNIX-tyyppinen, monoliittinen, modulaarinen ydin
- Vuorovaikuttaa laitteiston kanssa ja tarjoaa palveluita sovelluksille ja prosesseille
 - Laiteajurit, muistinhallinta, tallennusjärjestelmät, verkkotoiminnallisuus
 - API, ohjelmointirajapinta, POSIX-rajapinta, LSB-standardi, abstraktiotasoja (HAL, VFS, ...)
- Moduulien lisääminen ja poistaminen myös ajon aikana
- Tukee lukuisia laitteistoarkkitehtuureja (x86, x64, ARM, ...)
- Lähdekoodi jakelupaketin mukana tai kernel.org

Käyttäjät, ryhmät 1/2

- Pääkäyttäjä (super user, root) on järjestelmän jumala, jolla absoluuttinen valta tehdä kaikkea
 - Sisäänkirjautuminen nykyään usein estetty
 - Ei pidä käyttää normaalikäytössä lainkaan, vaan pelkästään ylläpitotoimiin, ja nekin mieluummin tavallisen käyttäjän korotetuilla oikeuksilla (sudo)
- Tavallinen käyttäjä normaalikäyttöön
 - Vähintään yksi luodaan asennuksen yhteydessä
 - Jos on lisätty ns. sudo-ryhmään, voi tehdä ylläpitotoimia korotetuilla oikeuksilla (sudo)
 - Sudo-oikeuksilla tiedostoa luodessa tulee omistajaksi root!

Käyttäjät, ryhmät 2/2

- Käyttäjärühmien oikeuksien helpompaan ja monipuolisempaan hallintaan
- Käyttäjä voi kuulua useampaan ryhmään, ryhmä voi kuulua ryhmään
 - Esim: juha, joel, mikko → assarit; kimmo, salu → opettajat; assarit, opettajat → henkilökunta
- Kotikäytössä ryhmät menevät asentaessa yleensä automaattisesti riittävän oikein
- Huomionarvoisia ryhmiä: sudo (tarvittaessa korotetut oikeudet), dialout (oikeus käyttää [USB-] sarjaporttia)

Tiedostojärjestelmä / hakemistopuu 1/3

- Puumainen rakenne
- Alin taso ns. juuri, root, /
- Tiedostojärjestelmiä (esim. USB-muisti) voidaan liittää puuhun
- Tiedostojen ja hakemistojen nimet merkkikokoriippuvaisia (case-sensitive) → abc ≠ ABC
- Tiedostoilla ja hakemistoilla omistajat ja käyttöoikeudet
 - User, Group, Other – käyttäjät, käyttäjäryhmät, muut
 - Oktaaliluku, 4 = lukuoikeus, 2 = kirjoitusoikeus, 1 = suoritusoikeus
 - Esim. 755, 640
 - `chmod <mode> tiedosto`, esim. `chmod 640 kuva.png`; `chmod o-w teksti.txt`

Tiedostojärjestelmä / hakemistopuu 2/3

- Piilotiedostojen/-hakemistojen nimi alkaa pisteellä, esim. `.salakuva.png`, `.salainen_hakemisto`
- Tiedostoihin voi luoda (symbolisia) linkkejä, `ln -s`
- Erikoishakemistoja/polkuja:
 - `.` → nykyinen hakemisto; `..` → “ylähakemisto”, parent directory; `~` → oma kotihakemisto

Tiedostojärjestelmä / hakemistopuu 3/3

- Juuressa olevia hakemistoja:

/bin, /sbin – matalan tason järjestelmäsovelluksia	/opt – perinteisesti binääreinä toimitetut sovellukset, nykyään merkitys vähentynyt
/boot – käynnistyksessä tarvittavia tiedostoja	/proc – järjestelmän ajonaikainen tila
/dev – laitteiston “rajapintatiedostot”, esim. äänikortin lähtö, kovalevyosiot	/root – pääkäyttäjän (ns. root-käyttäjä) kotihakemisto
/etc – konfiguraatiotiedostot	/run – interaktiokanava prosesseihin /sys – interaktiokanava laitteistoon
/home – käyttäjien kotihakemistot	/tmp – väliaikaistiedostot (voi olla lisäksi myös käyttäjän kotihakemistossa)
/lib – kernelin lähdekooditiedostot, kernelin moduulit yms.	/usr – ohjelmabinäärit, kirjastot, dokumentaatiot, kuvakkeet jnejne...
/media – käyttäjien liittämät tallennusmediat, esim. sd-kortit, usb-muistit, cd-/dvd-levyt /mnt – järjestelmän kiinteästi liitetyt tallennusosiot	/var – lokitiedostot, kovalevylle tallennettavat tilatiedostot, lukkotiedostot (lockfile), ...

/dev

- Kernelin laiteajurien tarjoamia rajapintoja laitteistoon
- Tärkeimpiä:
 - /dev/sdXY, missä X on kovalevyn “numero” ja Y on kovalevyn osion numero, esim. /dev/sda1
 - /dev/ttyS0, /dev/ttyS1 – perinteiset sarjaportit S0 ja S1 (COM1 ja COM2)
 - /dev/ttyUSB0 – normaali USB-sarjaportti (UART-over-USB)
 - /dev/ttyACM0 – USB CDC -laitteen sarjaportti, esim. Arduino Leonardo, aito Arduino UNO
- Näiden “laitetiedostojen” olemassaolosta voi päätellä esim. onko jokin laite kytketty ja tunnistunut oikein

Sovellukset

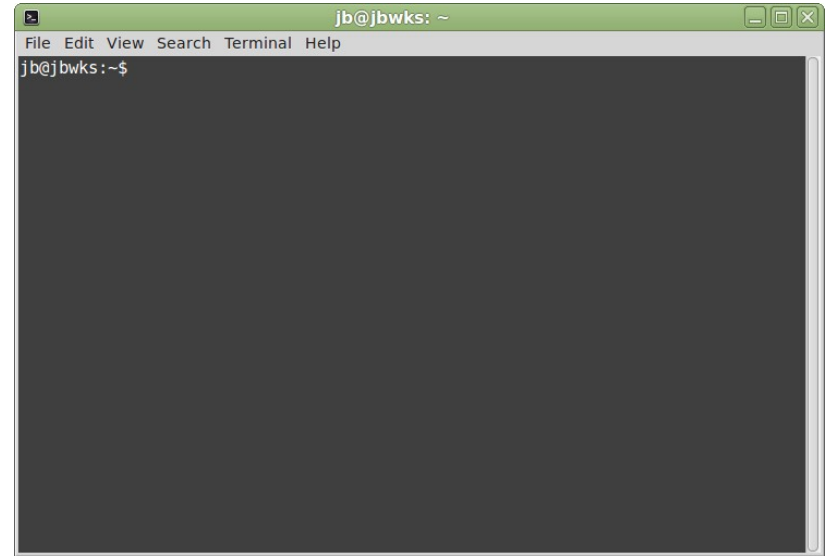
- Suunnattoman laaja valikoima eri sovelluksia
- Lähes kaikki sovellukset saatavilla lähdekoodipakettina
 - Täytyy itse kääntää suoritettavaksi binääritiedostoksi, yleensä melko suoraviivaista
- Varmistettava vaadittujen kirjastojen ja apuohjelmien olemassaolo
 - Muuten kääntäminen ei onnistu, dokumentaatio kertoo yleensä mitkä tarvitaan
 - Jokaisen sovelluksen mukaan on turha pakata samoja kirjastoja, kun ne voi jakaa sovellusten kesken
- Päivityksistä ja ristiriidoista huolehdittava itse

Pakettienhallinta

- Useimmissa jakelupaketeissa on keskitetty pakettienhallinta, joka mahdollistaa sovellusten ja kirjastojen helpon asentamisen ja päivittämisen
- Binääripaketit hyvin yleisiä, ei tarvitse kääntää itse
- Pitää huolen myös päivityksistä ja versioidenhallinnasta
 - Yleensä päivitykset tulevat melko nopeasti saataville
 - Päivitykset voi myös kytkeä täysin automaattisiksi
- Vahvasti suositeltu tapa sovellusten hallintaan

Komentorivi – Command line

- Tehokas lähestymistapa moniin toimenpiteisiin
- Kevyt, toimii hyvin myös etäyhteyden yli, ei vaadi erillisiä ajureita näytönohjaimelle
- Tavallisen käyttäjän komento-kehote päättyy \$-merkkiin
 - Pääkäyttäjällä #-merkkiin



Yleisiä komentorivikomentoja 1/2

- cd – change directory; pwd – print working directory; ls – list (directory contents); cp – copy (file); mv – move (file); rm – remove (file); mkdir – make directory; rmdir – remove directory
- cat – tulosta tiedoston sisältö; less – selaa tiedoston sisältöä; grep – tulosta tiedostosta ehtoon täsmäävät rivit
- chmod – change mode; chown – change owner
- ps – (list) processes; (h)top – interaktiivinen prosessilistaus

Yleisiä komentorivikomentoja 2/2

- nano – yksinkertainen tekstieditori; touch – luo tyhjän tiedoston;
- wget – lataa tiedosto netistä/etäkoneelta

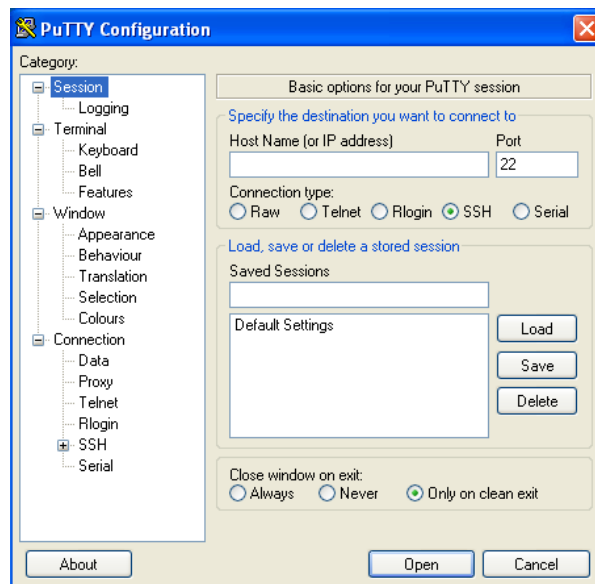
- man <komento> – sisäänrakennettu ohje

SSH – Secure Shell 1/2

- Salattu etäyhteys esim. palvelimelle
- Kätevä tapa käyttää ja hallita verkkoon liitettyjä laitteita, varsinkin jos ei ole fyysistä pääsyä
- Linuxin mukana on yleensä OpenSSH-asiakasohjelma
- Windowsille esim. PuTTY
 - <https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe>
- Tiedostojen kopiointiin SCP, SFTP ja WinSCP

SSH 2/2

- Linuxilla yhdistäminen: `ssh <user>@<host>`; tai `ssh <host> -l <user>`
 - <user> = etäkoneen käyttäjätunnus, <host> = etäkoneen osoite/ip
- PuTTY:lle annetaan käynnistysnäkyeseen etäkoneen osoite/ip, minkä jälkeen se kysyy käyttäjätunnuksen ja salasanan
- Esim. `kosh.aalto.fi`



screen

- Tekstitilan ikkunamanageri
- Kätevä jos esim. täytyy jättää sovellus/prosessi ajoon etäkoneella
 - Myös jos on tärkeää ettei etäpään prosessi kaadu, jos yhteys katkeaa
 - Irkin ajaminen etäkoneella

Tulosteen ohjaaminen

- Tarpeen jos halutaan tallentaa sovelluksen tuloste tiedostoon esim. myöhempää tarkastelua varten
- Myös jos halutaan estää terminaalin tulviminen sovelluksen tulosteesta
- `$ ls -la > lista.txt`; myös: `$ ls -la > /dev/ttyUSB0`
- `# ntpdate -u ntp.aalto.fi > /dev/null`

Komentojen “putkittaminen”

- Jos edellisen komennon tuloste halutaan ohjata suoraan seuraavan komennon syötteenä
- `$ ls -la |grep .pdf`
- `$ ls -la |grep .pdf |grep 2017`

Kernelin viestiloki

- Kernel kertoo virhetilanteista ja muista muutoksista viestilokissaan
- Voi olla suureksi avuksi esim. USB-laitteen toimintaa tutkiessa
- \$ dmesg |tail
- Useimmat lokit löytyvät /var/log -hakemistosta

- GNU Compiler Collection, yleensä viitataan C-kääntäjään
- Kääntää ja linkittää C-kieliset lähdekooditiedostot konekieliseksi suoritettavaksi ohjelmaksi
- `$ gcc -Wall -Werror -Wpedantic -std=c11 -o <ohjelman_nimi> <lista_ohjelman_lähdekooditiedostoista>`
 - Muista antaa valmiille ohjelmalle uusi nimi, ettei lähdekoodisi ylikirjoitu!
 - Käännetyt ohjelmatiedoston suorittaminen samassa hakemistossa: `./ohjelma`
 - Käännetyt binääritiedostot ovat alustariippuvia, eli esim. pöytäkoneella käännetty binääri ei toimi Raspilla, helpointa on kääntää koodi kohdealustalla

Lähteet

- Kuvat: Wikipedia.org / Wikimedia.org