HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Johdatus tietojenkäsittelytieteeseen - tietojenkäsittelyn mekaniikat

Matemaattis-luonnontieteellinen tiedekunta

Tietojenkäsittelytieteen laitos

Kurssin sisältö

Lähde: Peter J. Denning: Great Principles of Computing (Communications of the ACM, 46, 11, marraskuu 2003, sivut 15-20).

Luku 1: Historiaa
Luku 2: Kokonaiskuva
Luku 3: Eettiset perusteet
Luku 7: COMPUTING PRACTICES

programming
engineering systems
modeling
innovating
applying

Luku 4:
CORE TECHNOLOGIES

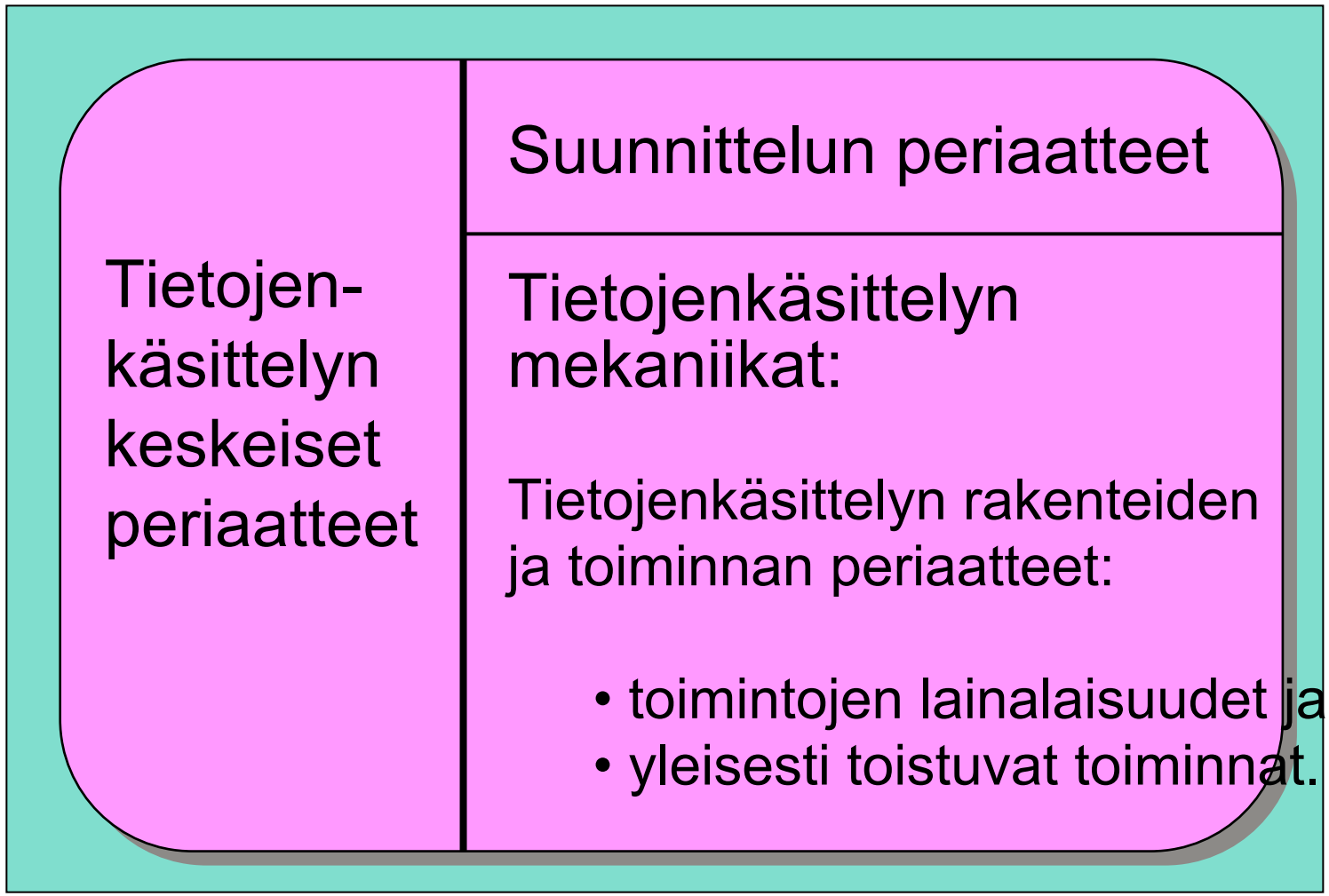
architectures
artificial intelligence
databases
operating systems
algorithms
networks
ETC. ...

GREAT PRINCIPLES OF COMPUTING

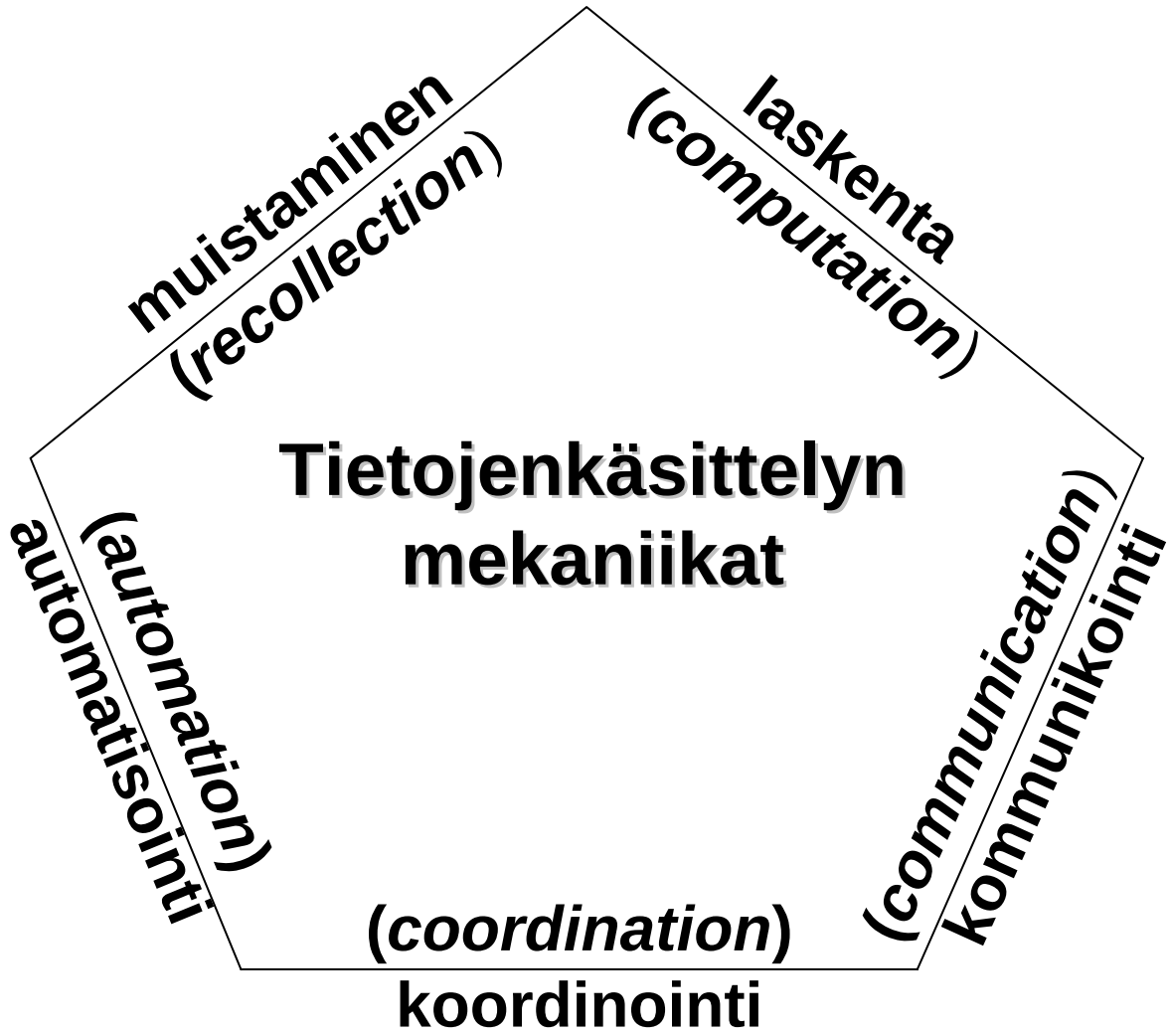
Luku 6: DESIGN
simplicity, performance, reliability, evolvability, security

Luku 5: MECHANICS
computation, communication, coordination, automation, recollection

Tietojenkäsittelyn mekaniikat (*mechanics*)



Viisi näkymää tietojenkäsittelyn mekaniikkoihin



Näkymät lyhyesti

1. Laskenta.
   - Mitä voidaan laskea – laskennan rajat.
2. Kommunikointi.
   - Sanoman tai viestin lähettäminen paikasta toiseen.
3. Koordinointi.
   - Vähintään kaksi toimijaa ja yhteinen tavoite.
4. Automatisointi.
   - Tietokoneella suoritettavat kognitiiviset tehtävät.
5. Muistaminen.
   - Tiedon tallettaminen ja hakeminen.

Näkymät tietojenkäsittelyn mekaniikkoihin koostuvat lukuisista tarinoista

1. Laskennan tarinoita.
   - Algoritmit (*algorithms*)
   - Ohjausrakenteet (*control structures*)
   - Tietorakenteet (*data structures*)
   - Automaatit (*automata*)
   - Turingin koneet (*Turing machines*)
   - Turingin kompleksisuus (*Turing complexity*)
   - Kolmogorovin kompleksisuus (*Kolmogorov complexity*)
   - Predikaattilogiikka (*predicate logic*)
   - Likimääräismenetelmät (*approximations*)
   - Heuristiikat (*heuristics*)
   - Muunnokset (*translations*)

Näkymät tietojenkäsittelyn mekaniikkoihin koostuvat lukuisista tarinoista

2. Kommunikoinnin tarinoita.
   - Tiedonsiirto (*data transmission*)
   - Shannonin entropia (*Shannon entropy*)
   - Tiedon fyysinen esittäminen (*encoding to medium*)
   - Kanavan kapasiteetti (*channel capacity*)
   - Kohinan poisto (*noise suppression*)
   - Tiedon tiivistäminen (*file compression*)
   - Salakirjoitus (*cryptography*)
   - Pakettiverkko (*reconfigurable packet network*)
   - Virheiden havaitseminen ja korjaaminen (*error detection and correction*)

Näkymät tietojenkäsittelyn mekaniikkoihin koostuvat lukuisista tarinoista

3. Koordinoinnin tarinoita.
   - Ihmisten välinen (*human-to-human*)
   - Ihmisen ja tietokoneen välinen (*human-computer*)
   - Tietokoneiden välinen (*computer-computer*)
     - Synkronointi (*synchronization*)
     - Kilpatilanteet (*race*)
     - Lukkiutuminen (*deadlock*)
     - Sarjallistuvuus (*serializability*)
     - Atomiset toimenpiteet (*atomic actions*)

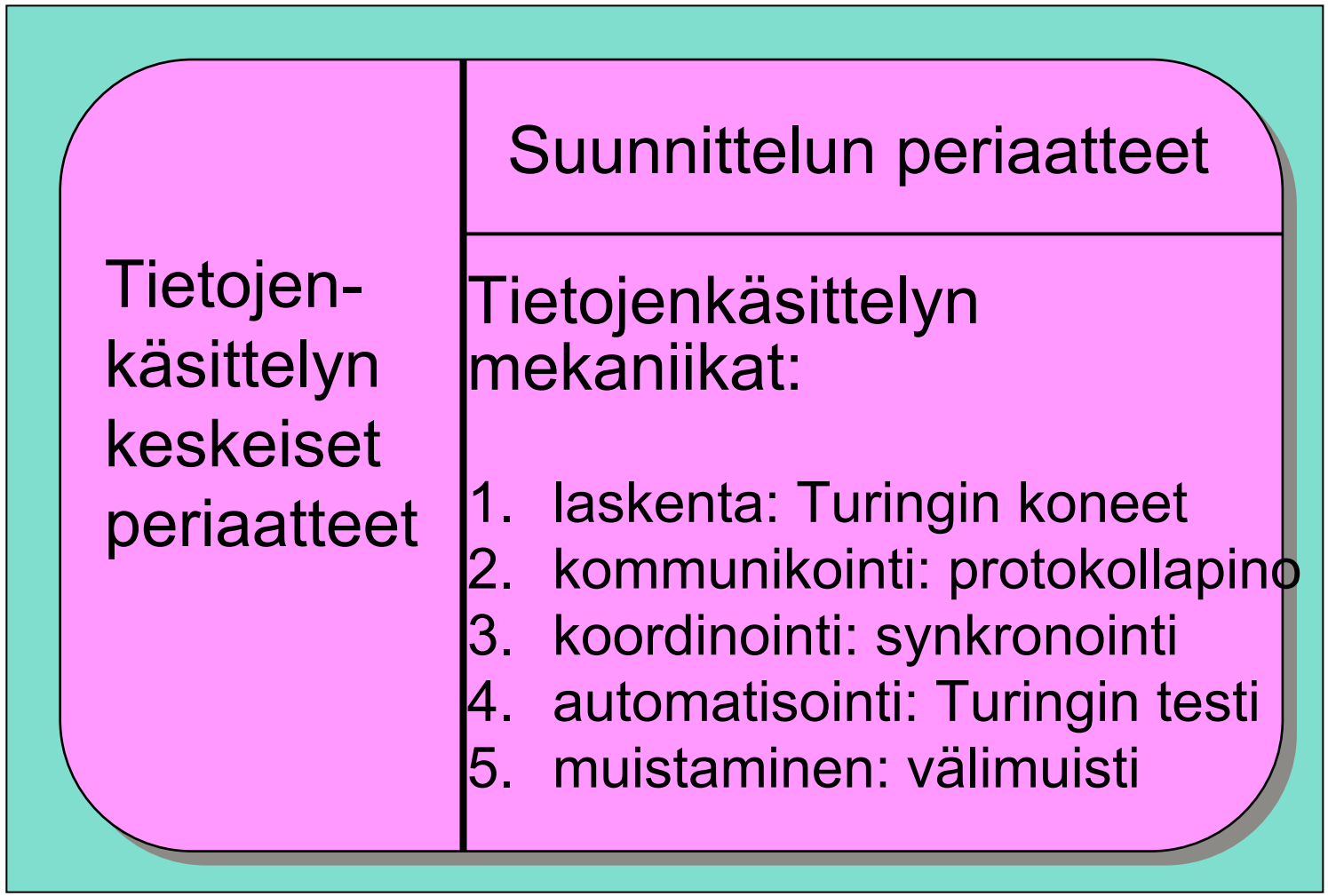
Näkymät tietojenkäsittelyn mekaniikkoihin koostuvat lukuisista tarinoista

4. Automatisoinnin tarinoita.
   - Kognitiivisten tehtävien simulointi (*simulation of cognitive tasks*)
   - Automatisoinnin filosofia (*philosophical distinctions about automation*)
   - Asiantuntemus ja asiantuntijajärjestelmät (*expertise and expert systems*)
   - Älykkyyden lisääminen (*enhancement of intelligence*)
   - Turingin testit (*Turing tests*)
   - Koneoppiminen ja tunnistaminen (*machine learning and recognition*)
   - Bioniikka (*bionics*)

Näkymät tietojenkäsittelyn mekaniikkoihin koostuvat lukuisista tarinoista

5. Muistamisen tarinoita.
   - Muistihierarkiat (*hierarchies of storage*)
   - Viittausten paikallisuus (*locality of reference*)
   - Välimuistit (*caching*)
   - Osoiteavaruudet ja niiden kuvaukset (*address space and mapping*)
   - Nimeäminen (*naming*)
   - Yhteiskäyttö (*sharing*)
   - Haku nimen perusteella (*retrieval by name*)
   - Haku sisällön perusteella (*retrieval by content*)

Viisi tarinaa tietojenkäsittelyn mekaniikoista



HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Johdatus tietojenkäsittelytieteeseen - tietojenkäsittelyn mekaniikat: laskenta: Turingin koneista

Matemaattis-luonnontieteellinen tiedekunta

Tietojenkäsittelytieteen laitos

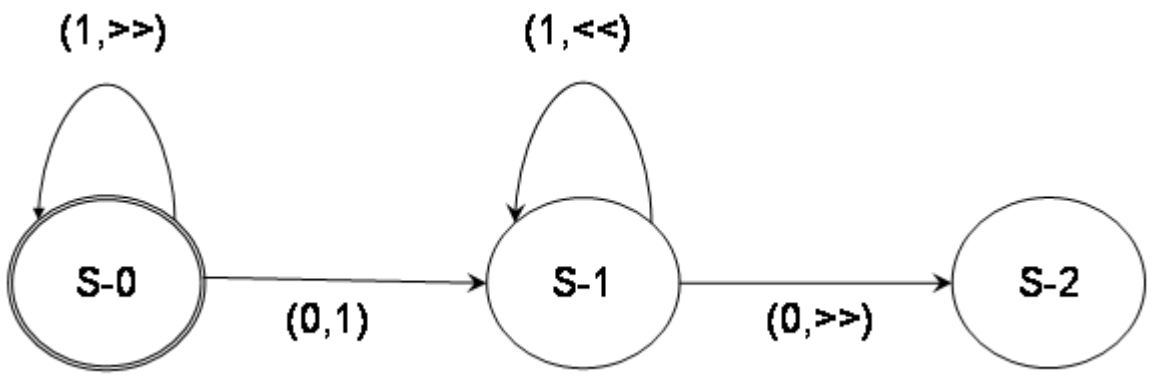
Turingin koneet

- Turingin kone on tietokoneen toiminnan teoreettinen malli.
  - Englantilainen matemaatikko Alan Turing.
  - Ajalta ennen tietokoneita.
  - Taustalla Gödelin epätäydellisyysteoreema vuodelta 1931.
    - Matematiikassa on olemassa lauseita, jotka ovat tosia, mutta niitä ei voi todistaa.
  - Tarkat ohjeet laskentatehtävän mekaaniseksi suorittamiseksi.
- Laskennan rajojen tutkimiseksi.
  - Mitä voidaan algoritmisesti ratkaista.

Turingin koneet

- Turingin kone on tilakone: se on aina yhdessä tiloistaan.
  - Tiloja (*state*) on äärellinen joukko.
  - Peräkkäismuisti on yksiulotteinen (nauha ilman loppua).
  - Aakkosto on äärellinen (usein vain 0 ja 1).
  - toimenpiteet: talletus, luku- ja kirjoituspään siirto (>> tai <<).
- Tilasiirtymät: <nykytila, aakkonen, uusitila, toimenpide>

- Turingin kone: ”Lisää yksi”:



Turingin koneiden laskennallinen voima

- Churchin - Turingin teesi.
  - Ei ole olemassa ongelmaa, joka voitaisiin ratkaista tietokoneella, mutta ei Turingin koneilla.
    - Ei ole pystytty todistamaan.
    - Ei ole kumottu eli ei tunneta vastaesimerkkejä.

Turingin koneet

- Laskettavuuden teoriasta.
  - Turingin koneiden avulla on todistettu, että pysähtymisongelma (*halting problem*) on laskennallisesti ratkeamaton.
    - Ei ole olemassa ohjelmaa, joka pystyisi päättelemään päättyykö vai ei minkä tahansa toisen ohjelman suoritus millä tahansa syötteellä.
  - Turingin todistus perustuu vastaesimerkkiin.

Universaalit Turingin koneet (*universal Turing machines, UTM*)

- Jokainen Turingin kone laskee yhden tietyn laskettavissa olevan funktion arvon.
- Turing osoitti, että on olemassa universaali Turingin kone, joka pystyy simuloimaan minkä tahansa Turingin koneen toiminnan.
  - Universaalia Turingin konetta voi pitää ohjelmoitavana tietokoneena.
- Universaalit Turingin koneet ovat yllättävän ”pieniä”.
  - Pienimmät tunnetut ovat
    - 2 x 18: 2 tilaa 18 aakkosta,
    - 3x10, 4x6, 5x5, 7x4, 10x3, 22x2.

Laskettavuuden rajoja etsimässä

- Mitkä (millaiset) ongelmat ovat todistettavasti algoritmisesti ratkeamattomia?
  - Esiintyy esim. muodollisen päättelyn alueella, mikä on vaikuttanut mm. tekoälyn kehittymiseen.

- Mitkä (millaiset) ongelmat voidaan periaatteessa ratkaista algoritmisesti, mutta laskenta tuloksen saamiseksi kestää niin kauan, että ratkaisu on valmistuttuaan käytännössä hyödytön.
  - Tällaisia hankalia eli NP-täydellisiä (*NP-complete, intractable*) ongelmia on runsaasti esim. tilanteissa, joissa halutaan löytää paras mahdollinen ratkaisu.

Laskennallinen vaativuus

- n on algoritmille annettavan syötteen koko.
- Algoritmin tarvitsema operaatioiden määrä eli aikavaatimus tuloksen laskemiseksi voi olla esimerkiksi O(n log n) eli verrannollinen operaatioiden lukumäärään n log n.
- Jos algoritmin aikavaativuus on $O(e^n)$, niin algoritmin
  - sanotaan olevan skaalautumaton ja
  - laskenta-aika kasvaa eksponentiaalisesti syötteen koon kasvaessa.

Laskennallisen vaativuuden tuntemisen merkityksestä

- Ei kannata tuhlata aikaa algoritmin kirjoittamiseen, jos on todistettu ettei tavoiteltua algoritmia ole olemassa.
- Jos tulee luvanneeksi kirjoittaa pysähtymisongelman ratkaisevan algoritmin, niin jossakin vaiheessa joutuu tunnustamaan ettei osaa.
  - Ammattilainen ei olisi tullut luvanneeksi.
- Tietojenkäsittelyn ammattilaisen tietoihin kuuluu laskennan teorian perusteiden ja perustulosten hallinta ja taitoihin kuuluu laskennan vaativuuden arviointi.

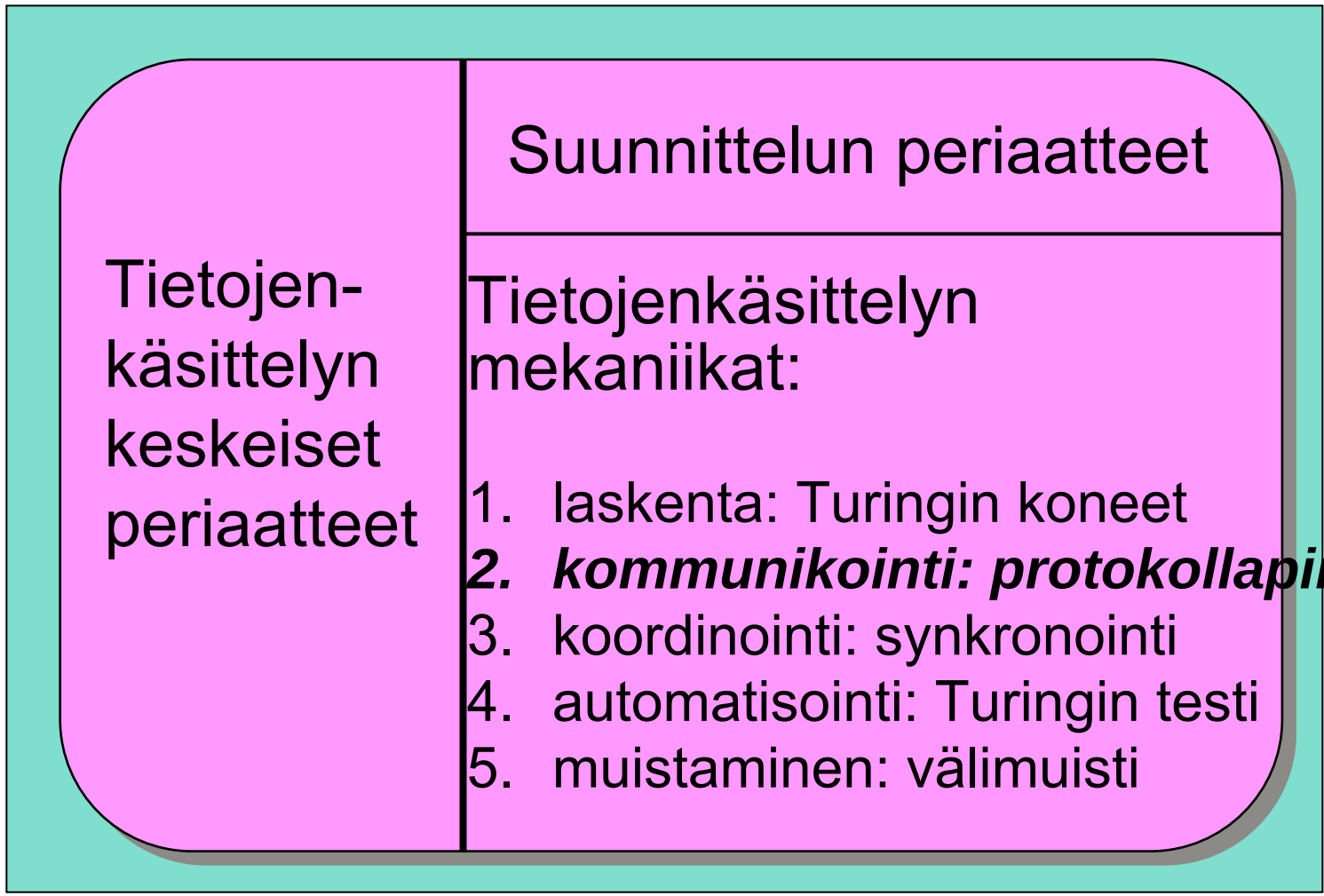
HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Johdatus tietojenkäsittelytieteeseen - tietojenkäsittelyn mekaniikat: kommunikointi: protokollapino

Matemaattis-luonnontieteellinen tiedekunta

Tietojenkäsittelytieteen laitos

Viisi tarinaa tietojenkäsittelyn mekaniikoista



ISO:n (*International Standardization Organization*) OSI-malli (*Open Systems Interconnection Reference Model*)

OSI-mallin periaatteita

- Ylempi kerros on lähempänä käyttäjää kuin alempi.
- Kukin kerros käyttää vain välittömästi alemman kerroksen toimintoja ja tarjoaa toimintojaan vain välittömästi ylemmälle kerrokselle. Rajapinnat on täsmällisesti määritelty.

Sovelluskerros

- Sovelluksen vuoropuhelu verkossa.
  - Määritellään viestit: niiden rakenne ja merkitys.
- Sovellustason protokollia ovat mm.
  - sähköposti,
  - uutisryhmät ja
  - Web.

Esitystapakerros

- Sanoman sisällön esitystapa.
- Internet-maailmassa perinteisesti ollut lähes olematon.
  - Jätetty sovelluksen sisäiseksi asiaksi.
- W3C:n (*World Wide Web Consortium*) XML (*eXtensible Markup Language*) on yleistynyt sanoman sisällön esitystapana.

Istuntokerros

- Perustetaan, hallitaan ja lopetetaan yhteys paikallisen ja toisaalla olevan sovelluksen välillä.
- Ei ole tarjottu Internetissä.
- Istuntokerroksen puuttuminen korvattu evästimillä (*cookie*).

Kuljetuskerros

- Sanoman siirtäminen päätepisteiden välillä.
- Internetin keskeiset kuljetusprotokollat ovat
  - TCP (Transmission Control Protocol), joka takaa luotettavan tietovuon ja
  - UDP (User Datagram Protocol), joka on epäluotettava tietosähke.

Verkkokerros

- Sanomien reititys lähettäjältä vastaanottajalle.
  - Ruuhkanhallinta.
- Internetissä verkkokerroksen keskeisin protokolla on IP (*Internet Protocol*).
  - Ruuhkanhallinta ratkaistu suoraviivaisesti: jos sanomia on liikaa, niin jotkut niistä tuhotaan.

Linkkikerros

- Toiminnat ja menettelytavat tiedon siirtämiseksi verkon kahden pisteen välillä.
  - Virheiden havaitseminen ja mahdollinen korjaus.
  - Muuttumattomat kehykset kuitataan vastaanotetuksi.
  - Rikkoontuneet pyydetään lähettämään uudelleen.

Fyysinen kerros

- Bittien lähettäminen tiedonsiirtokanavaa pitkin.
- Tiedonsiirtolaitteiden sähköiset fyysiset ominaisuudet.
  - Määritellään volttiarvo kummallekin bitille (0 ja 1).
  - Bitin kesto.
  - yms.

DoD-malli (*Department of Defence*) on TCP/IP-pinon perusta

DoD- ja OSI-mallien erot

- DoD-mallista puuttuvat esitystapa- ja istuntokerrokset.
  - Toiminnallisuus toteutetaan jokaisessa sovellustason protokollassa erikseen.
- DoD-mallissa fyysinen ja linkkikerros on yhdistetty verkkoonpääsykerrokseksi.
  - Käytännössä erolla ei ole suurta merkitystä.
  - Toiminnallisuus on yleensä verkkokortilla.

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Johdatus tietojenkäsittelytieteeseen
- tietojenkäsittelyn mekaniikat: koordinointi: synkronointi

Matemaattis-luonnontieteellinen tiedekunta

Tietojenkäsittelytieteen laitos

Koordinoinnin tarinoita

- Ihmisten välinen (*human-to-human*)
- Ihmisen ja tietokoneen välinen (*human-computer*)
- Tietokoneiden välinen (*computer-computer*)
  - Synkronointi (*synchronization*)
  - Kilpatilanteet (*race*)
  - Lukkiutuminen (*deadlock*)
  - Sarjallistuvuus (*serializability*)
  - Atomiset toimenpiteet (*atomic actions*)

Synkronointi

- Synkronointi eli tahdistus on aikaan liittyvää koordinointia.
- Esimerkiksi monikanavaviestinnässä eri kanavien tietovirtojen tulee edetä samaa tahtia.
- Toisissa tilanteissa synkronoinilla varmistetaan, että itsenäisten toimijoiden (väli)tulokset valmistuvat oikeassa järjestyksessä.

Kellojen synkronointi

- Eräs hajautetun tietojenkäsittelyn perusongelma.
- GPS – *Global Positioning System*.
- NTP – *Network Time Protocol*.

Synkronoinnin keskeiset ongelmat

- Liittyvät usein yhteiskäyttöisten resurssien hallintaan.
- Ratkaisuiden on estettävä sekä nälkiintyminen että lukkiutuminen.
  - Nälkiintyminen (*starvation*): joku joutuu odottamaan vuoroaan ikuisesti.
  - Lukkiutuminen (*deadlock*): kaikki odottavat, että joku toinen tekisi jotain.

Tuottaja – kuluttaja ongelma

- Tuottajan tuotettava ennen kuin kuluttaja voi kuluttaa.
- Välivaraston (puskurin) täyttyessä on tuottajan odotettava, että kuluttaja kuluttaa.
- Kun tuottajia ja kuluttajia on useita, niin on myös huolehdittava poissulkemisesta.
  - Tuottajien tuotokset on saatava välivarastossa eri paikkoihin.
  - Kaksi kuluttajaa ei saa saada samaa tuotosta.
  - Samanaikaiset lisäykset ja poistot eivät saa sotkea varastokirjanpitoa.

Lukija – kirjoittaja ongelma

- Kaikki voivat lukea samanaikaisesti, mutta vain yksi kerrallaan voi kirjoittaa.
- Kirjoittajan nälkiintyminen on estettävä.
  - Jos kirjoittaja ei saa kirjoitusvuoroa, niin kaikki lukijat lukevat vanhentunutta tietoa.

Kilpatilanne (*race condition*)

- Virhe järjestelmän suunnittelussa.
- Kaksi signaalia kilpailee siitä, kumpi pääsee ensin vaikuttamaan tulokseen.
  - Kaksi tai useampi ohjelma pääsee synkronoimattomasti käsiksi yhteiskäyttöiseen resurssiin samanaikaisesti.

```
global integer A = 0;
task Received()
    { A = A + 1; }
task Timeout()
// Print only theeven numbers
    { if (A is divisible by 2)
          { print A; } }
```

- Received aktivoituu aina, kun sarjaportti aiheuttaa keskeytyksen eli vastaanottaa dataa.
- Timeout aktivoituu kerran sekunnissa (kellolaitekeskeytys).

Mahdollinen suoritusjärjestys

```
        global integer A = 0;
3.      task Received()
4.              { A = A + 1; }
1.      task Timeout()
        // Print only theeven numbers
2.              { if (A is divisible by 2)
5.                      { print A; } }
```

- Timeout tulostaakin parittoman A:n arvon, jos askeleessa 2 oli parillinen A.

Aterioivat filosofit

http://www.doc.ic.ac.uk/~jnm/concurrency/classes/Diners/Diners.html

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Johdatus tietojenkäsittelytieteeseen - tietojenkäsittelyn mekaniikat: automatisointi: Turingin testi

Matemaattis-luonnontieteellinen tiedekunta

Tietojenkäsittelytieteen laitos

Automatisoinnin tarinoita

- Kognitiivisten tehtävien simulointi (*simulation of cognitive tasks*)
- Automatisoinnin filosofia (*philosophical distinctions about automation*)
- Asiantuntemus ja asiantuntijajärjestelmät (*expertise and expert systems*)
- Älykkyyden lisääminen (*enhancement of intelligence*)
- Turingin testit (*Turing tests*)
- Koneoppiminen ja tunnistaminen (*machine learning and recognition*)
- Bioniikka (*bionics*)

Turingin testi

- Ihminen kirjoittaa kahdelle testattavalle (ihminen ja tietokone) kysymyksiä.
- Saamiensa kirjallisten vastausten perusteella hän yrittää päätellä, kumpi vastaajista on tietokone.
- Perustuu englantilaiseen seurapiirileikkiin *Imitation game*.
- Turing ehdotti testiään älykkyyden määritelmäksi 1950.
  - Keskustelua käytiin värikkäästi tietokoneen kyvyistä.
    - Ajatteleva kone (*thinking machine*).
    - Sähköaivot.
- Nykyisin Turingin testiä pidetään marginaali-ilmiönä tekoäly-yhteisössä.

Turingin testin arvostelua

- Toimii kuin ihminen ei ole järkevä tavoite.
  - Ihminen tekee virheitä.
- Lentäminen perustuu aerodynamiikkaan.
  - Linnun hyväkään matkiminen ei riitä.
- John Searlen kiinalainen huone.
- Älykkyys ei ole matkimista.
  - Tietoisuus.
  - Tavoitteellisuus.

Turingin testin läpäisemisen edellytyksiä

1. Luonnollisen kielen käsittely.
2. Tietämyksen esittäminen.
3. Automaattinen päättely.
4. Koneoppiminen.

- Kaikki tietojenkäsittelyn, erityisesti tekoälyn, keskeisiä ongelmia.

Osaongelmien tunnuspiirteitä

1. Luonnollisen kielen käsittely.
   - Järjestelmän on ymmärrettävä sille esitetyt kysymykset ja muotoiltava vastaukset.
   - Eräs ongelma: yksi sana – monta eri asiayhteydestä riippuvaa merkitystä.

2. Tietämyksen esittäminen.
   - Järjestelmän on talletettava tietämänsä ja kuulemansa.
   - Tietämyksen laajetessa talletuksen ja etsinnän tehokkuudet voivat olla ongelmallisia.

Osaongelmien tunnuspiirteitä

3. Automaattinen päättely.
   - Järjestelmän on talletetun informaation perusteella osattava tehdä päätelmiä, joiden perusteella vastaus voidaan muotoilla.
   - Loogisen päättelyn jotkut ongelmat ovat todistetusti ratkeamattomia, joten järjestelmän on myös osattava päätellä, mitä se ei pysty päättelemään.

4. Koneoppiminen.
   - Järjestelmän on sopeuduttava uusiin tilanteisiin.
   - Järjestelmän ”maailmankuva” muuttuu uuden datan perusteella.
     - Maailmankuvalla tarkoitetaan järjestelmän käytössä olevaa mallia kiinnostavista asioista.

Turingin testin läpäisemisestä

- Yleisesti pidetään mahdottomana.
- ”Oikea vastaus väärään kysymykseen”.
  - Yleensä hyödytön.
  - Turingin testin tapauksessa hyödyllinen: osaongelmien ratkaisut olisivat hyödyllisiä.
    - Luonnollisen kielen automaattinen käsittely (ihmisen ja koneen vuorovaikutus).
    - Tietämyksen automaattinen esittäminen ja käyttäminen.
    - Automaattinen päättely (mitä käyttäjä seuraavaksi haluaa).
    - Koneoppiminen.
- Järjestelmän tulee toimia siten kuin ihminen HALUAA.

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Johdatus tietojenkäsittelytieteeseen - tietojenkäsittelyn mekaniikat: muistaminen: välimuisti

Matemaattis-luonnontieteellinen tiedekunta

Tietojenkäsittelytieteen laitos

Muistamisen tarinoita

- Muistihierarkiat (*hierarchies of storage*)
- Viittausten paikallisuus (*locality of reference*)
- Välimuistit (*caching*)
- Osoiteavaruudet ja niiden kuvaukset (*address space and mapping*)
- Nimeäminen (*naming*)
- Yhteiskäyttö (*sharing*)
- Haku nimen perusteella (*retrieval by name*)
- Haku sisällön perusteella (*retrieval by content*)

Välimuistit

- Miten saadaan dataa nopeasti käsiteltäväksi.
  - Prosessori ↔ keskusmuisti.
  - Keskusmuisti ↔ massamuisti.
  - Internetin siirtoviiveet.
- Data talletetaan väliaikaisesti käsittelypaikan lähelle.
  - Prosessorin välimuisti.
  - Tiedostovälimuisti.
  - Webin välimuisti.

Välimuistit

- Välimuistit ovat osoittautuneet keskeisiksi tietokoneiden suoritusnopeuden kasvattamisessa.

- Datan käsittely on usein paikallista (*locality of reference*).
  - Tietoalkiota käsitellään useita kertoja ajallisesti lähekkäin.
  - Lähekkäin olevia tietoalkioita käsitellään usein ajallisesti lähekkäin.
    - Tietorakenteen alkiot.
    - Tiedoston läpikäynti.

Välimuistin hallinta

- Välimuisti on pieni verrattuna varsinaiseen muistiin.
- Välimuistista poisto (*replacement policy*).
  - LRU: *least recently used.*
- Kirjoitus varsinaiseen muistiin, kun välimuistiin on kirjoitettu (*write policy*).
  - Läpikirjoitus (*write-through cache*).
  - Viivästetty kirjoitus (*write-back cache*).
- Eheysprotokolla (*coherency protocol*).
  - Sama tieto välimuistissa ja varsinaisessa muistissa.
  - Vanhentuminen (*stale*).

Prosessorin välimuisti

- Prosessorit nopeutuneet enemmän kuin muistipiirit ja väylät.
- Prosessorilla pieni ja nopea välimuisti.
  - Laitteistototeutus.
- Nykyisin useita erikoistuneita välimuisteja.
- Tavoite: saantiviiveen (*access latency*) lyhentäminen.

Muistien nopeusongelmia

- 1970-luku: supertietokoneiden ongelma.
- 1980-luku: graafisten työasemien ongelma.
- 1990-luku: pöytäkoneiden ongelma.
- 2000-luku: prosessori suorittaa satoja käskyjä yhden keskusmuistista noudon aikana.

Virtuaalimuisti

- Jokaisella suoritettavalla ohjelmalla on illuusio, että se on yksin tietokoneessa ja voi käyttää omaa osoiteavaruutta (*address space*).
- Prosessori muuttaa virtuaaliosoitteet todellisiksi muistiosoitteiksi.
- Prosessoreissa on yleensä osoitteenmuunnokset tekevä laitteisto.
  - Muistinhallintayksikkö (*memory management unit, MMU*).
  - MMU:lla on yleensä oma välimuisti.
    - Osoitteenmuunnospuskuri (*translation lookaside buffer, TLB*).

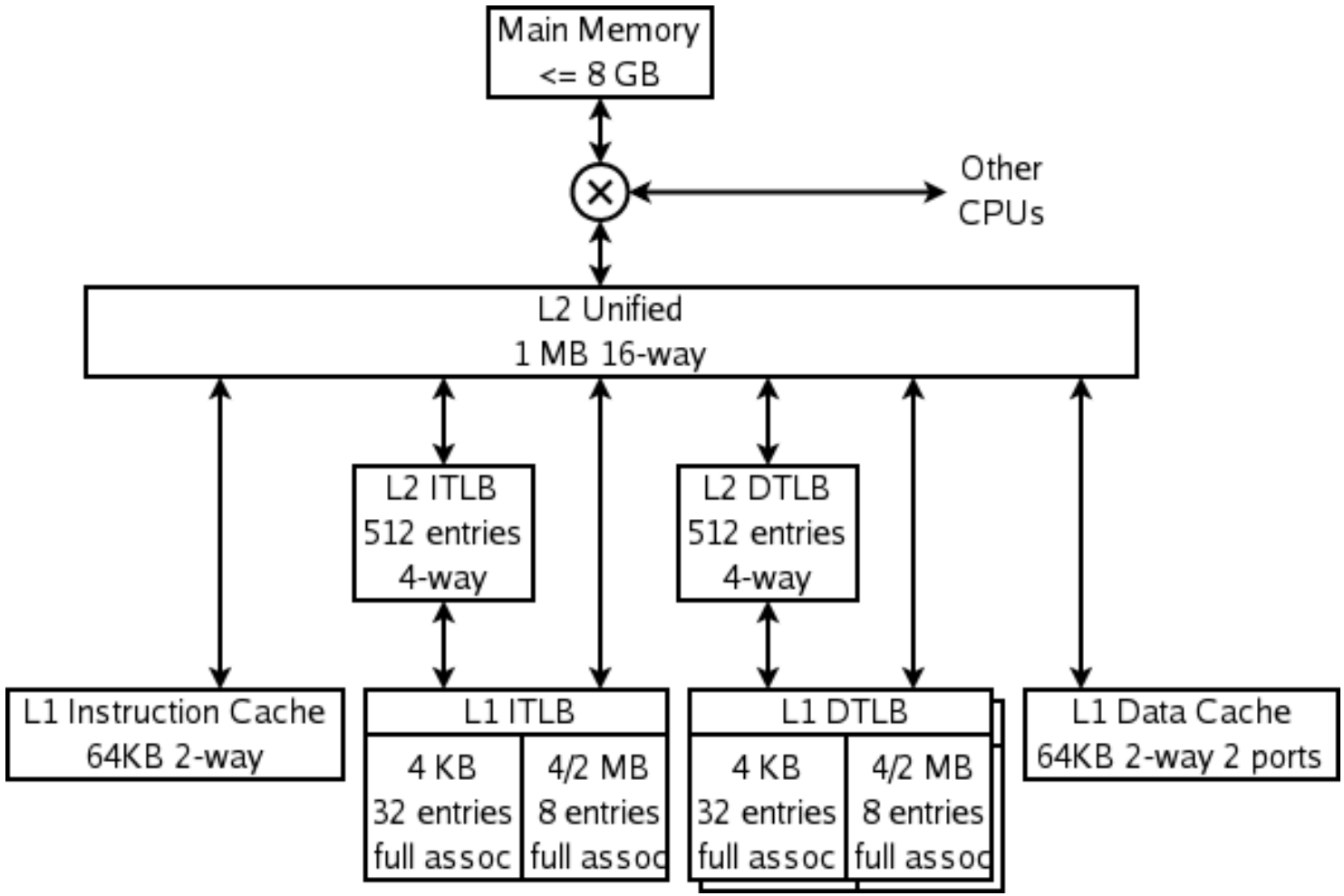
Prosessorin käskynkäsittely

- Liukuhihnoitus (*pipelining*) nopeuttaa prosessoreja.
- Idea: käskyn suoritus voidaan jakaa vaiheisiin:
  - Käskyn nouto (*instruction fetch*).
  - Osoitteenmuunnos (*address translation*).
  - Tietoalkion nouto (*data fetch*).
- Muistiin on päästävä käsiksi jokaisessa vaiheessa.
  - Jokaisella vaiheella oma välimuisti.
  - Laitteistotasolla ei tarvitse varautua kilpatilanteeseen.

Välimuistien hierarkia

- Nykyisissä prosessoreissa on yleensä hierarkinen välimuistijärjestelmä.
- Suunnittelussa joudutaan tekemään kompromisseja (*tradeoff*).
  - Mitä laajempi välimuisti sitä hitaampi, mutta osumistodennäköisyys (*hit ratio*) suurempi eli noudettava tieto on välimuistissa useammin.
  - Nykyisissä prosessoreissa on yleensä ainakin kaksi välimuistitasoa.

AMD Athlon 64 prosessorin välimuistit



HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Johdatus tietojenkäsittelytieteeseen - suunnittelu

Matemaattis-luonnontieteellinen tiedekunta

Tietojenkäsittelytieteen laitos

Kurssin sisältö

Lähde: Peter J. Denning: Great Principles of Computing (Communications of the ACM, 46, 11, marraskuu 2003, sivut 15-20).

Luku 1: Historiaa
Luku 2: Kokonaiskuva
Luku 3: Eettiset perusteet
Luku 7: COMPUTING PRACTICES

programming
engineering systems
modeling
innovating
applying

Luku 4:
CORE TECHNOLOGIES

architectures
artificial intelligence
databases
operating systems
algorithms
networks
ETC. ...

GREAT PRINCIPLES OF COMPUTING

Luku 6: DESIGN
simplicity, performance, reliability, evolvability, security

Luku 5: MECHANICS
computation, communication, coordination, automation, recollection

Suunnittelu

Suunnittelun vakiintuneita käytäntöjä

- Abstrahointi (*abstraction*).
  - Epäolennaisten yksityiskohtien häivyttäminen.
- Informaation piilottaminen (*information hiding*).
  - Ohjelmiston osan (moduulin) sisäisten tietojen ja tietorakenteiden piilottaminen muilta ohjelmiston moduuleilta.
- Moduulit (*modules*).
  - Ohjelmiston jakaminen osiin siten, että osien väliset vuorovaikutukset ja rajapinnat ovat hyvin määriteltyjä.
- Erikseen kääntäminen (*separate compilation*).
  - Moduulien kääntäminen erikseen ja linkittäminen myöhemmin suorituskelpoiseksi kokonaisuudeksi.

Suunnittelun vakiintuneita käytäntöjä

- Pakkaukset (*packages*).
  - Jakelu- ja asennusyksikkö, johon on koottu ohjelmisto(je)n osat ja dokumentaatiot.
- Versionhallinta (*version control*).
  - Menetelmät, joilla hallitaan ohjelmiston kehitystyötä ja sen aikana syntyviä ohjelmaversioita.
- Hajota ja hallitse (*divide-and-conquer*).
  - Suuret kokonaisuudet jaetaan pienempiin ja helpommin hallittaviin osakokonaisuuksiin.
- Toimintatasot (*functional levels*).
  - Toiminnallisuuden ryhmittely eri tasoisiksi toimenpiteiksi.

Suunnittelun vakiintuneita käytäntöjä

- Kerrosajattelu (*layering*).
  - Toimintakokonaisuuden jakaminen kerroksiin siten, että kukin kerros tarjoaa joitakin (muutamia) palveluja ylemmälle kerrokselle ja käyttää alemman kerroksen palveluja.
- Hierarkiat (*hierarcy*).
  - Suunnittelussa hierarkiat liittyvät yleensä olio-ohjelmoinnin luokkarakenteeseen.
- Ongelmien eriyttäminen (*separation of concerns*).
  - Periaate, jonka mukaan keskitytään kerrallaan yhteen, hyvin määriteltyyn (osa)tehtävään.
- Uudelleenkäyttö (*reuse*).
  - Olemassa olevien moduulien, määritysten, suunnitelmien jne käyttäminen sen sijaan, että tehtäisiin uudestaan.

Suunnittelun vakiintuneita käytäntöjä

- Rajapinta (*interface*).
  - Rajapinnan välityksellä ohjelmisto- tai laitteistokomponentti tarjoaa toiminnallisuutensa muiden komponenttien käyttöön.
- Virtuaalikone (*virtual machine*).
  - Ohjelmisto, joka toteuttaa määritellyn abstraktin koneen toiminnallisuuden.

Suunnittelun viisi tavoitetta

1. Yksinkertaisuus (*simplicity*).
   - Abstraktiot ja rakenteet, joilla hallitaan sovelluksen luontaista monimutkaisuutta.
2. Suorituskyky (*performance*).
   - Suoritustehon (*throughput*) ja vasteajan (*response time*) ennustaminen, pullonkaulojen (*bottlenecks*) paikallistaminen sekä kapasiteetin suunnittelu (*capacity planning*).
3. Luotettavuus (*reliability*).
   - Päällekkäisyys (*redundancy*), toipuminen (*recovery*), varmistaminen (*checkpointing*), eheys (*integrity*) ja luottamus (*trust*).

Suunnittelun viisi tavoitetta

4. Kehitettävyys (*evolvability*).
   - Varautuminen toiminnallisuuden ja käytön laajuuden muutoksiin.
5. Tietoturva (*security*).
   - Pääsynvalvonta (*access control*), salassapito (*secrecy*), yksityisyys (*privacy*) autentikointi (*autenthication*) ja turvallisuus (*safety*).

Suunnittelun rajoitteet

- Kustannukset.
- Aikataulut.
- Yhteensopivuus (*compatibility*).
- Käytettävyys (*usability*).

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Johdatus tietojenkäsittelytieteeseen - suunnittelu: yksinkertaisuus

Matemaattis-luonnontieteellinen tiedekunta

Tietojenkäsittelytieteen laitos