

# **581332-8**

# **Rinnakkaisohjelmistot**



**Liisa Marttinen**  
**Tietojenkäsittelytieteen laitos**  
**Helsingin yliopisto**  
**Kevät 2004**

# Rinnakkaisohjelmistot

- Asema opetuksessa
  - ci-oppimäärän pakollinen kurssi
- Esitiedot:
  - Tietokoneen toiminta (2 ov)
  - Käyttöjärjestelmät I (2 ov)

# Tavoitteet

## Ymmärtää

- rinnakkaisuudesta saatavat hyödyt
- rinnakkaisuuden hallinnan tarve
- keskeiset ongelmakohdat ja mekanismit niiden ratkaisemiseksi
- rinnakkaisten järjestelmien ohjelmoinnin peruskäsitteet

## Osata

- paikantaa kohdat, joissa tarvetta rinnakkaisuuden hallintaan
- soveltaa tavallisimpia ratkaisumenetelmiä algoritmitasolla

# Computing Curricula 2001: Concurrency [core]

## Topics to cover

KJ-I

- States and diagrams
- Structures (ready list, process control block, and so forth)
- Dispatching and context switching
- The role of interrupts

RIO

- Concurrent execution: advantages and disadvantages
- The "mutual exclusion" problem and some solutions
- Deadlock: causes, conditions, prevention
- Models and mechanisms (semaphores, monitors, condition variables, rendezvous)
- Producer-consumer problems and synchronization
- Multiprocessor issues (spin-locks, re-entrancy)

# Learning objectives

1. Describe the need for concurrency within the framework of an OS.
2. Demonstrate the potential run-time problems arising from the concurrent operation of many separate tasks.
3. Summarize the range of mechanisms that can be employed at the OS level to realize concurrent systems and describe the benefits of each.
4. Explain the different states that a task may pass through and the data structures needed to support the management of many tasks.

5. Summarize the various approaches to solving the problem of mutual exclusion in an OS.
6. Describe the reasons for using interrupts, dispatching, and context switching to support concurrency in an OS.
7. Create state and transition diagrams for simple problem domains.
8. Discuss the utility of data structures, such as stacks and queues, in managing concurrency.
9. Explain conditions that lead to deadlock.

***Source: ACM Computing Curricula 2001, Computer Science***

# Sisältö

## ① Johdanto

- rinnakkaisuus, samanaikaisuus
- laitteistoarkkitehtuureja
- prosessien välinen kommunikointi
  - yhteiskäyttöinen muisti, sanomanvälitys
- sovellusten luokittelua
- ohjelmointiparadigmat
- algoritmien ohjelmointi: määrittelyt ja notaatiot

# OSA I: Yhteiskäyttöiset muuttujat

= *Prosessit samassa koneessa, yhteinen muisti*

## ② Poissulkeminen ja synkronointi

- lukot, semaforit, synkronointi

## ③ Semaforit ja rinnakkaisuuden hallinta

- tuottajat - kuluttajat, resurssien hallinta
- lukijat - kirjoittajat

## ④ Lukkiutuminen

- ennaltaehkäisy, havaitseminen, välttäminen
- aterioivat filosofit, pankkiirin algoritmi

## ⑤ Monitorit

- käsite, synkronointitekniikoita



# Osa II: Ohjelmointi hajautetussa ympäristössä

= *Ei yhteistä muistia, prosessit eri koneissa*

## ⑥ Sanomanvälitys

- kanavat, kommunikointimallit (patterns), synkronointi
- asiakkaat - palvelija

## ⑦ Etäproseduurikutsu

- toteutus, virhesemantiikka

## ⑧ Rendezvous

- aktiivisten prosessien kohtaaminen

## ⑨ Yhteenvedoa

# Paljon esimerkkejä

- Tuottajat – kuluttajat (1-M puskuria, 1-N prosessia)
- Lukijat – kirjoittajat
- Puomisynkronointi
- Resurssien varaus ja jonotus (FCFS, SJN, ...)
- Nukkuva parturi
- Ruokailevat filosofit
- Ym. Ym.
  
- Emme käytä formaalia algoritmien oikeellisuuden todistamista - terveet sanalliset perustelut riittävät
  - Kurssi: Spesifioinnin ja verifioinnin perusteet

# Opetus ja opiskelu

- Kuuntele luennot! Kysy!
  - luentokalvot vain ”sisällysluettelo”
    - ’käsiteltävien asioiden muistilista’
    - lisää omat muistiinpanot
    - luentokalvot alkujaan Timo Alankon käsialaa, sitten Auvo Häkkisen täydentämät ja edelleenkehittämät
- Lue kirjasta! Kysy!
  - hanki omaksi, lainaa jostain, lue jossain...
- Tee harjoitukset! Kysy!

# Tiedon lähteet

- **Andrews G.R: Foundations of Multithreaded, Parallel, and Distributed Programming. Addison Wesley, 2000. (Luvut 1-8)**
- Stallings: Operating Systems, 4th ed. 2001. (Luvut 5-6)
- [Bacon: Concurrent Systems, 2nd ed. 1997. (Kuvia)]
- **Kurssisivu: <http://www.cs.helsinki.fi/>**
  - kalvokopiot, harjoitustehtävät
  - tarkistusksymyksiä, sanastoa, tiedotteita ym.
- Uutisryhmä: **[hy.opiskelu.tkl.fi](http://hy.opiskelu.tkl.fi/)**
- Kurssimappi A412: kalvokopiot, kuvia (osa vain mapissa!)

# RIO vs. muut kurssit

- Esitiedot: TITO, KJ-I
  - ajoaikainen pino, aktivointitietue
  - yhteiskäyttöinen muisti
  - prosessi/säie, prosessin tilat, prosessijonot
  - keskeytysmekanismi
  - vuorottamisen perusideat (FCFS, SJN, RR)
- Tämän jälkeen
  - Verkkosovellusten toteuttaminen
  - Hajautetut järjestelmät



**Tervetuloa**

**Työ se vaan on huviakin hauskeempaa!**