



HELSINGIN YLIOPISTO  
HELSINGFORS UNIVERSITET  
UNIVERSITY OF HELSINKI

# Käyttöjärjestelmät I

## Osio 2: Luennot 4-7

### Muistinhallinta

Tiina Niklander; kalvot © Auvo Häkkinen

Tietojenkäsittelytieteen laitos  
Helsingin yliopisto

# Käyttöjärjestelmät I

## YKSINKERTAINEN MUISTINHALLINTA

Stallings, Luku 7

# Sisältöä

Yleistä muistinhallinnasta (luku 7.1)

Yksinkertainen muistinhallinta

- a) kiinteät partitiokoot (luku 7.2)
- b) dynaamiset partitiokoot (luku 7.2)
- c) Buddy System (luku 7.2)
- d) yksinkertainen segmentointi (luku 7.4)
- e) yksinkertainen sivutus (luku 7.3)

Yksinkertainen =

prosessi aina kokonaan muistissa tai kokonaan levyllä

# Käyttöjärjestelmät I

YLEISTÄ  
MUISTINHALLINNASTA

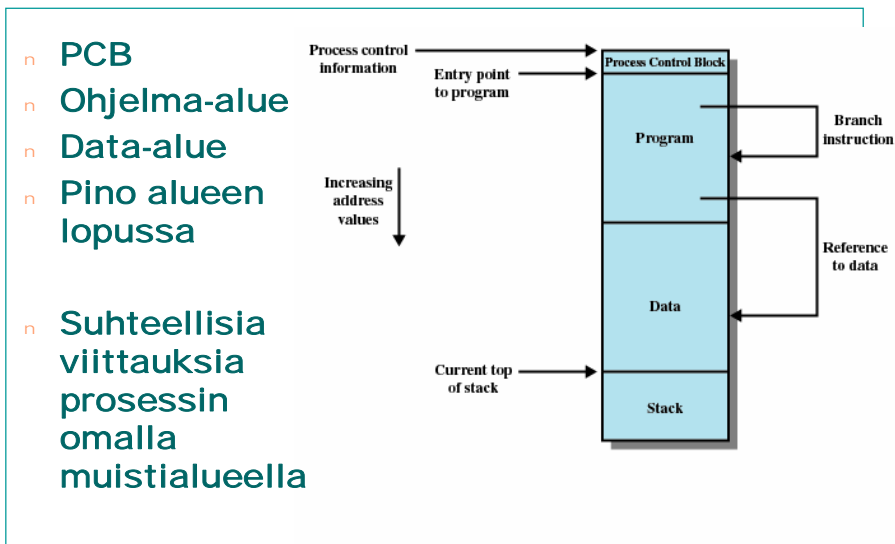
## Fyysinen muisti

- n **Fyysinen muisti, 'sitä voi potkaista'**
  - u muistiavaruus
  - u laitteisto (MMU, väylät) käyttää fyysisiä osoitteita
- n **Keskusmuisti**
  - u koodin + käsiteltävän datan suoritusaik. tallennus
  - u joukko peräkkäisiä tavuja
- n **Tukimuisti**
  - u tiedon (ohjelmat, data) pysyvä tallennus
  - u joukko peräkkäisiä lohkoja
- n **Siirto näiden välillä muistinhallinnan ja tiedostojärjestelmän leipätyötä**
  - u mahd. automaattisesti KJ:n toimesta

## Looginen muisti

- n **prosessin osoiteavaruus**
  - u loogiset osoitteet eli virtuaaliosoitteet
- n **voi olla suurempi kuin fyysinen muisti**
- n **kullakin prosessilla oma osoiteavaruus**
  - u osoitteet suhteellisia alun suhteen (0..MAX)
- n **sovellus jakaantuu loogisesti moduuleihin**
  - u ne voidaan tehdä eri aikoina, osoitteiden paikkaus linkityksessä
  - u erilaisia käyttöoikeuksia
    - F koodi R (vapaakäytisyys)
    - F data R / W / RW
  - u osa moduuleista tarkoitettu yhteiskäyttöön

## Prosessin rakenne



KJ-I S2005 / Tiina Niklander; kalvot Auvo Häkk

Figure 7.1 Addressing Requirements for a Process

## Yleistä muistinhallinnasta

- n Ytimelle voidaan sallia kiinteä paikka muistissa
  - u voi käyttää suoraan fyysisiä osoitteita
- n Loppu KJ:n muille osille ja sovelluksille
- n **KJ** huolehtii siitä, että muistiin mahtuu mahd. monta prosessia
  - u vapaan / varatun tilan hallinta
- n **Laitteisto** huolehtii siitä, etteivät prosessit sotke toisiaan
  - u MMU

KJ-I S2005 / Tiina Niklander; kalvot Auvo Häkkinen

4 - 8

## Muistinhallinta: vaatimuksia

### Vapaa sijoitettavuus

#### n Heittovaihto

- u KJ voi siirtää prosesseja välillä levyille
- u KJ voi ottaa suoritettavaksi useampia prosesseja
- u prosessin paikka voi vaihdella suorituksen aikana

#### n Ohjelmoija ei voi tietää minnepäin muistia sovellus sijoittuu suoritusaikana

- u suhteelliset osoitteet

#### n Viittaukset fyysiksi osoitteiksi viimeistään ennen muistinoutoa/talletusta

### è Ajonaikainen osoitemuunnos MMU:ssa

## Muistinhallinta: vaatimuksia

### Suojaus

#### n Luvatta ei saa käyttää toisen muistialueita

#### n Osoitetarkistus käännösaikana mahdotonta

- u sovelluksen moduulit voidaan kääntää eri aikoina
- u prosessien sijainti voi vaihtua suoritusaikana
- u käskykanta voi sallia osoitustapoja, joissa osoite lasketaan suoritusaikana

#### n KJ ei voi sitä tehdä!

### è Ajonaikainen laillisuustarkistus osittain MMU:ssa ja osittain KJ:ssa

# Muistinhallinta: vaatimuksia

## Yhteiskäyttö

- n Sallittava yhteisen koodin / datan käyttö
  - u suojauksista tinkimättä!
- n Koodi ei muutu suorituksen kuluessa
  - u vapaakäyntisyys (reentrancy)
  - u järkevämpää sallia koodin yhteiskäyttö kuin pitää muistissa useita kopioita
- n Monet prosessit tekevät yhteistyötä muiden kanssa, joten niillä yhteisiä tietorakenteita
  - u Esim. tuottajalla ja kuluttaja yhteinen puskuri

e käytä säikeitä, palvelupyynnöjä

# Käyttöjärjestelmät I

YKSINKERTAINEN  
MUISTINHALLINTA

## Yksink. muistinhallinta

= Prosessi kokonaan muistiin / levyille

n Jos ohjelma > fyysisen muistin koko, sitä ei välttämättä pystytä suorittamaan

n Menetelmät:

- u a) kiinteä partitointi
- u b) dynaaminen partitointi
- u c) Buddy System
- u d) yksinkertainen segmentointi
- u e) yksinkertainen sivutus

n Katso ns. katoava kansanperinne

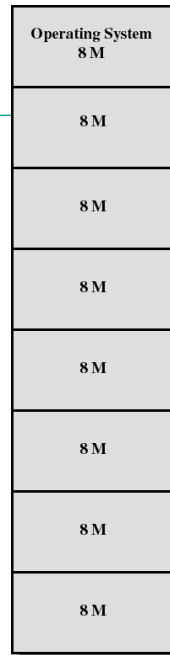
- u ei paljon käyttöä nykyisissä järjestelmissä
- u kuuluu kuitenkin KJ:n peruskäsitteistöön

## Käyttöjärjestelmät I

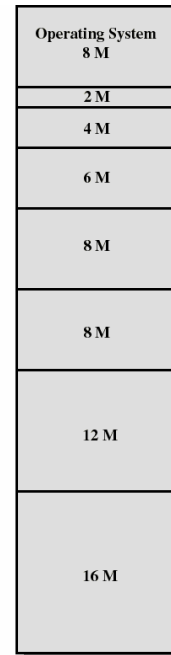
a) Kiinteät  
partitiot

## Kiinteät partitiot

- n KJ (operaattori) jakoi muistin kiinteänkokoisiin partitioihin
- n Varausalueet olivat kaikki yhtäsuuria tai niiden koot saattoivat vaihdella
- n Ohjelma, joka oli pienempi tai yhtäsuuri kuin partitio, voitiin ladata ja ajaa ko. partitiossa



Equal-size partitions



Unequal-size partitions ;

KJ-I S2005 / Tiina Niklander; kalvot Auvo Häkkinen

## Kiinteät partitiot

- n Jos ei riittävän suurta vapaata partitiota, KJ teki tilaa heittovaihdolla
  - u joku prosessi levyille
  - u PCB jäi muistiin
- n Jos ohjelma niin iso, ettei sopinut mihinkään partitioon, piti ohjelmoijan ratkaista tilanne
  - u kerrostus (overlying)
    - F "ohjelmoijan hoitama segmentointi"
  - u vain osa ohjelmasta muistissa
  - u piti koodata mitä osia (aliohjelmiä, 'segmenttejä') kullakin hetkellä muistissa ja missä kohdassa

KJ-I S2005 / Tiina Niklander; kalvot Auvo Häkkinen

4 - 16



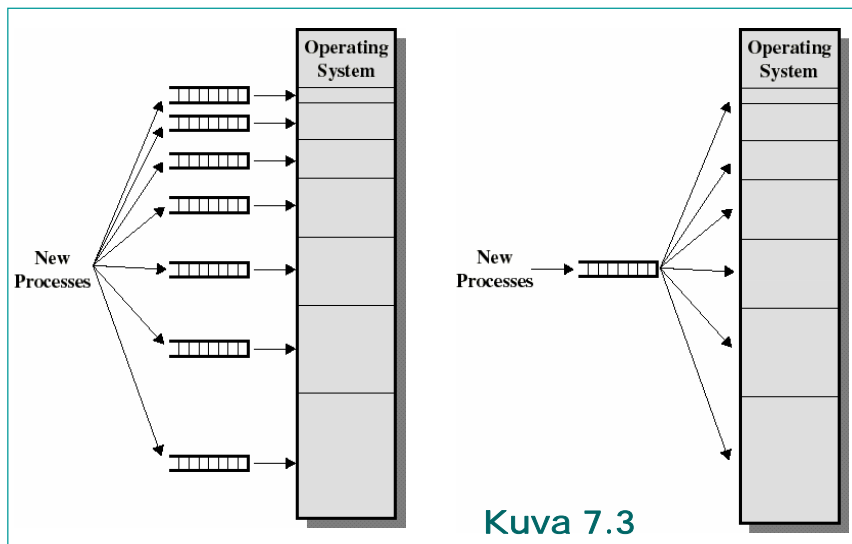
## Kiinteät partitiot

- n **Muistin käyttö melko tehotonta**
  - u varattiin aina kokonainen partitio, vaikka vähempikin olisi riittänyt
- n **Sisäinen pirstoutuminen (internal fragmentation)**
  - u partitioiden sisälle jäi tyhjää tilaa
  - u vapaa tila yhdessä olisi saattanut riittää uudelle prosessille, mutta se ei ollut yhtenäisellä alueella

## Kiinteät partitiot: Sijoitus

- n **Jos kaikki partitiot samankokoisia**
  - u jos vapaita, valitse joku niistä
  - u jos kaikki varattu, heittovaihda joku Blocked-prosessi levyille
    - F tilaa vapautuu aina saman verran
- n **Jos partitiot eri kokoisia**
  - u valitse pienin partitio, johon prosessi sopii
    - F yritti minimoida sisäistä pirstoutumista
  - u a) erikokoisille partitioille omat prosessijononsa
    - F partitiokoko yksi työn parametreista
  - u b) yksi jono, josta valittiin johonkin vapaaseen partitioon

## Kiinteät partitiot: Sijoitus



KJ-I S2005 / Tiina Niklander; kalvot Auvo Häkkinen

4 - 19

## Käyttöjärjestelmät I

### b) Dynaaminen partitiointi

KJ-I S2005 / Tiina Niklander; kalvot Auvo Häkkinen

4 - 20

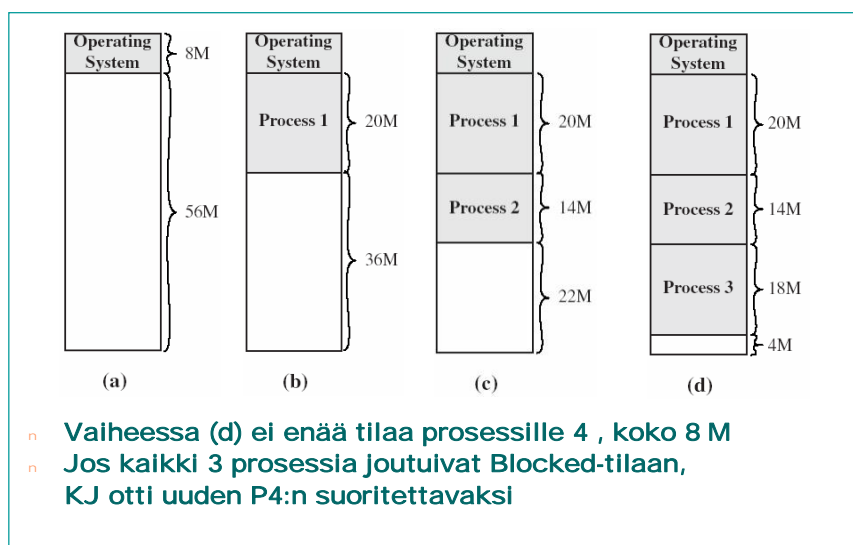
## Dynaaminen partitiointi

- n Ei etukäteen partitointia
- n Varausten koot ja lukumäärä vaihtelivat dynaamisesti prosessien tarpeiden mukaan
- n Prosessille muistia vain sen verran kuin tarvitsi
- n **Ulkoinen pirstoutuminen** (external fragmentation)
  - u varausten / vapautusten tuloksena väleihin jäi pieniä vapaita alueita
- n **KJ tiivistä muistia välillä** (compaction)
  - u prosesseja siirrettiin, jotta vapaa tila yhteen kohtaan
  - u yleisrasite

KJ-I S2005 / Tiina Niklander; kalvot Auvo Häkkinen

4 - 21

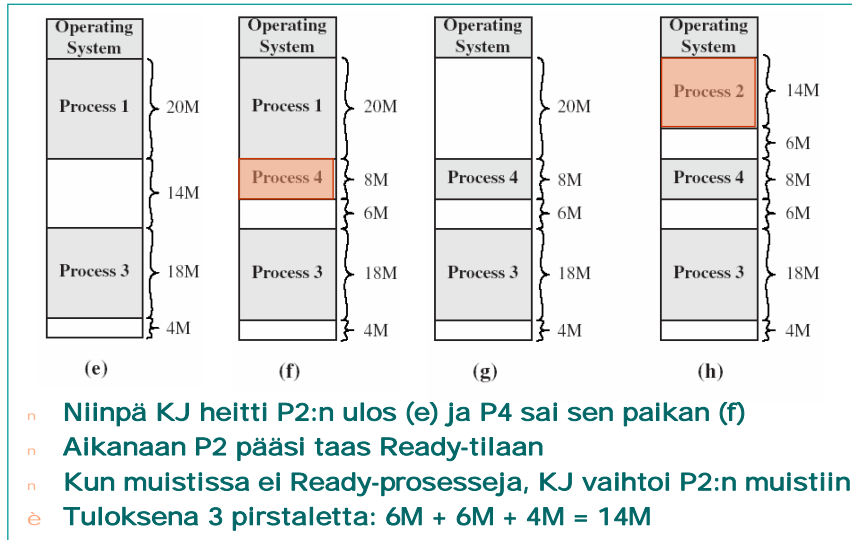
## Dynaaminen partitiointi Kuva 7.4



KJ-I S2005 / Tiina Niklander; kalvot Auvo Häkkinen

4 - 22

# Dynaaminen partitointi



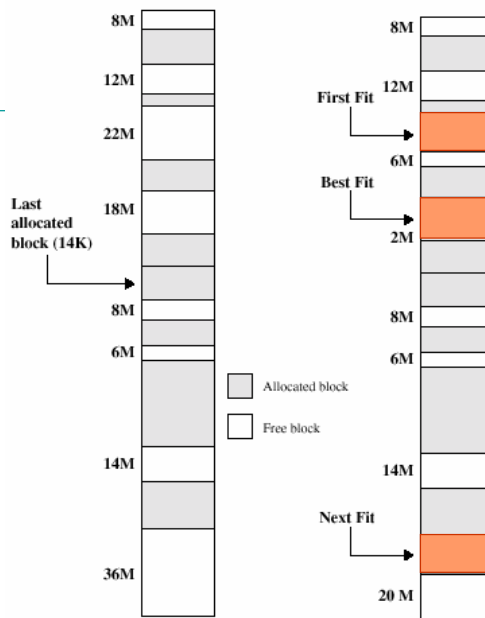
KJ-I S2005 / Tiina Niklander; kalvot Auvo Häkkinen

4 - 23

## Sijoitus Kuva 7.5

Mistä kohtaa varataan?

- n Tavoitteena vähäinen tiivistämistarve
- n **Best-fit** kooltaan sopivin
- n **First-fit** ens. kooltaan riittävän suuri
- n **Next-fit** jatka etsintää edellisestä kohdasta



Example Memory Configuration Before and After Allocation of 16 Mbyte Block

KJ-I S2005 / Tiina Niklander; kalvot Auvo Häkkinen

## Paras sijoitusalgoritmi?

- n **First fit**
  - u yksinkertainen
  - u helpoin toteuttaa tehokkaasti (paras)
  - u isosta aukosta jää jäljelle isohko aukko
- n **Best fit**
  - u hyvä nimi, mutta tuloksena mahd. pieniä aukkoja
  - u tuloksena nopeimmin pirstoutuminen
- n **Next fit**
  - u varaukset / aukot muistin loppuosaan
- n **Toteutus: linkitetty lista, jossa vapaat alueet koon tai osoitteen mukaisessa järjestyksessä**
  - u kumpi parempi?
  - u (osoite,pituus)à ... à (osoite,pituus)à

## Poistoalgoritmi

- n **Kun muistissa vain Blocked-prosesseja, kannatti ottaa uusi prosessi suoritukseen**
  - u ettei CPU jouten
  - u heittovaihda joku levyille
  - u poistettava prosessi Blocked-Suspend-tilaan
  - u uusi prosessi tilaan Ready-Suspend tai Ready
- n **Mikä pois, jotta saatiin sopivasti vapaata?**
  - u iso vs. pieni
  - u sellainen, jonka vieressä iso tyhjä alue
  - u pieni vs. suuri prioriteetti

# Käyttöjärjestelmät I

## c) Buddy System

KJ-I S2005 / Tiina Niklander; kalvot Auvo Häkkinen

4 - 27

## Buddy System

- n **Kompromissi kahdesta em. menetelmästä**
  - u kiinteät partitiokoot, mutta dynaaminen jako partitioihin
  - u ei haluta jättää pieniä tyhjiä tiloja
  - u varaa pikkuisen enemmänkin kuin tarve vaatii
- n **Varausyksikön koko ilmaistaavissa 2:n potenssissa**
  - u suurin mahdollinen yleensä koko muisti
  - u pienimmälle varausyksikölle joku minimikoko
    - F yksinkertaistaa vapaiden alueiden kirjanpitoa
- n **Varaukset ja vapaiden alueiden yhdistely toteutettavissa tehokkaasti**

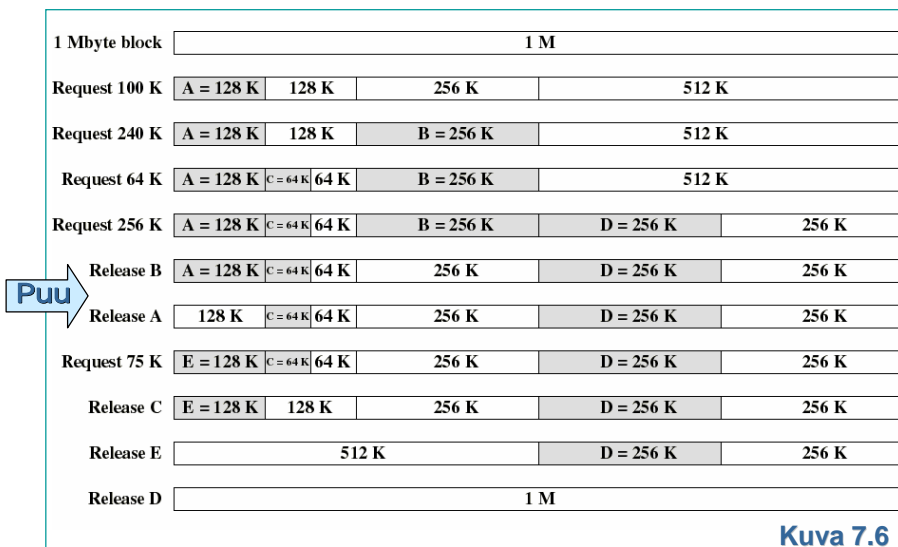
KJ-I S2005 / Tiina Niklander; kalvot Auvo Häkkinen

4 - 28

# Buddy System

- n **KJ ylläpitää** kustakin koosta omaa listaa
  - u lista osoitteiden mukaisessa järjestyksessä
  - u aluksi vain yksi suuri varausyksikkö
- n **Varaus**
  - u jos oikean kokoluokan listassa ei vapaata alkiota, jaa luokkaa suurempi alue kahdeksi pienemmäksi
  - u toista tarvittaessa
- n **Vapautus**
  - u kun listassa kaksi fyysisesti vierekkäistä aluetta, yhdistä ne kokoluokkaa suuremmaksi alueeksi
  - u toista tarvittaessa

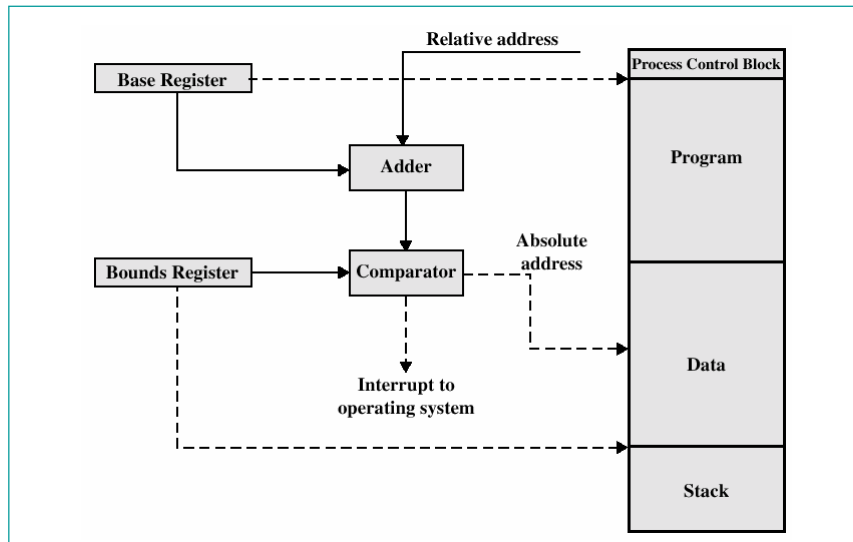
# Buddy System: esimerkki







## a) b) c) Osoitemuunnos Kuva 7.8



KJ-I S2005 / Tiina Niklander; kalvot Auvo Häkkinen

4 - 33

## Laitteistotuki

- n MMU:hun muunnosta ja suojaustarkistusta varten kaksi rekisteriä
  - u ns. kanta- ja rajarekisteri
  - u Base prosessin fyysinen alkuosoite
  - u Limit prosessin loppuosoite (tai pituus)
- n Kun prosessi suoritukseen, kopioidaan näille arvot PCB:stä
- n  $\text{Fyys.osoite} = \text{log.osoite} + \text{Base}$   
jos fyysinen osoite > Limit  
aiheuta poikkeus 'virh. muistiosoite'  
muuten  
 $\text{MAR} \leftarrow \text{fyysinen osoite}$

KJ-I S2005 / Tiina Niklander; kalvot Auvo Häkkinen

4 - 34