

Käyttöjärjestelmät I

Luento 6: YKSINKERTAINEN SEGMENTOINTI JA SIVUTUS

Stallings, Luku 7

KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 1

Sisältöä (Luennot 5&6)

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)**

KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 2

Pikakertaus: a) b) c)

a) kiinteä partitointi

- muisti jaettu osiin etukäteen
- sisäinen pirstoutuminen (fragmentointi)

b) dynaaminen partitointi

- muistia varataan tarpeen mukaan
- ulkoinen pirstoutuminen (fragmentointi)
- tiivistäminen partitioita siirtämällä

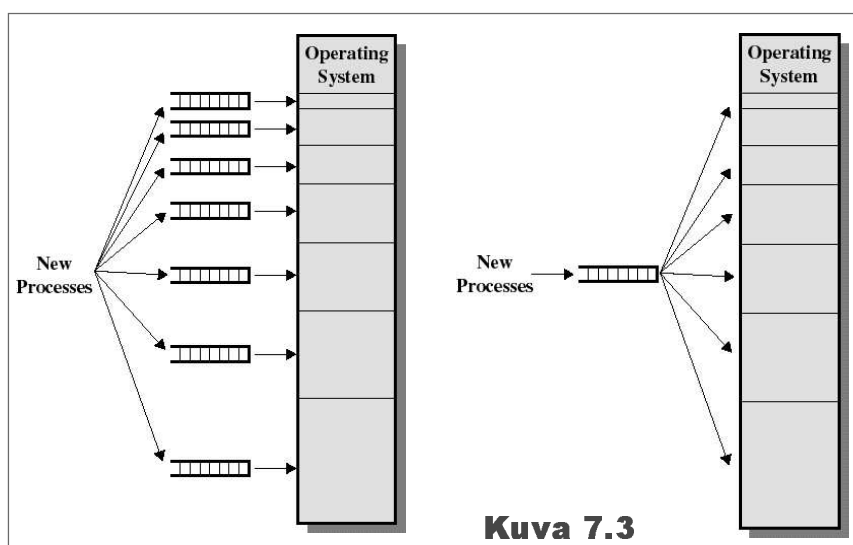
c) Buddy System

- dynaaminen, mutta etukäteen kiinnitetyt osiokoot, kukin koko 2^i , minimikoko asetettu
- hiukan sisäistä pirstoutumista, mutta ulkoinen hallittavissa vapaiden yhdistelyllä

KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 3

Kiinteät partitiot: Sijoitus



Kuva 7.3

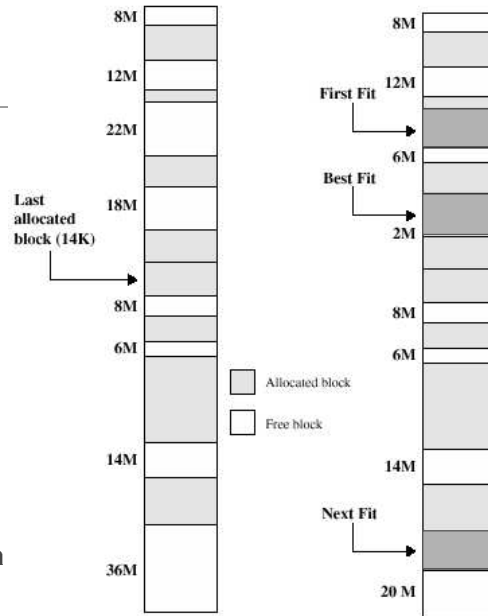
KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 4

Dyn. part.: sijoitus Kuva 7.5

Mistä kohtaa varataan?

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



Example Memory Configuration Before and After Allocation of 16 Mbyte Block

KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

Buddy System: esimerkki

| | | | | | | |
|---------------|-----------|----------|-----------|-----------|-----------|-------|
| 1 Mbyte block | 1 M | | | | | |
| Request 100 K | A = 128 K | 128 K | 256 K | 512 K | | |
| Request 240 K | A = 128 K | 128 K | B = 256 K | 512 K | | |
| Request 64 K | A = 128 K | C = 64 K | 64 K | B = 256 K | 512 K | |
| Request 256 K | A = 128 K | C = 64 K | 64 K | B = 256 K | D = 256 K | 256 K |
| Release B | A = 128 K | C = 64 K | 64 K | 256 K | D = 256 K | 256 K |
| Release A | 128 K | C = 64 K | 64 K | 256 K | D = 256 K | 256 K |
| Request 75 K | E = 128 K | C = 64 K | 64 K | 256 K | D = 256 K | 256 K |
| Release C | E = 128 K | 128 K | 256 K | D = 256 K | 256 K | |
| Release E | 512 K | | | D = 256 K | 256 K | |
| Release D | 1 M | | | | | |

Kuva 7.6

KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 6

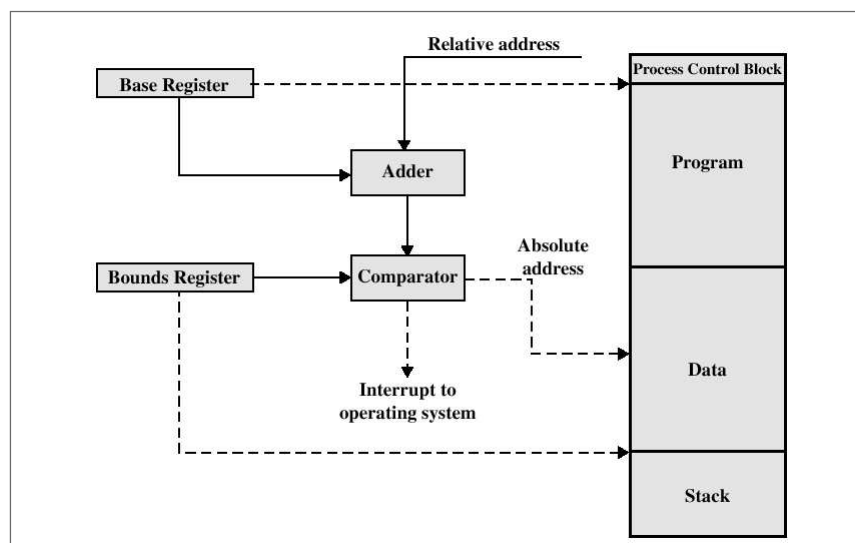
a) b) c) Osoitemuunnos

- **Koko prosessi yhdessä partitiossa**
- **Ohjelmassa loogiset osoitteet**
 - ◆ Suhteellisia ohjelman alun suhteen
 - ◆ Vapaa sijoitettavuus
- **Osoitemuunnos vasta käskyjä suoritettaessa**
- **MMU:n tehtävä**
 - ◆ Prosessorilta tulee käskyssä ollut looginen osoite
 - ◆ Prosessi yhdessä partitiossa
 - ◆ MMU muuntaa sen fyysiseksi osoitteeksi
 - ☞ Kanta ja rajarekisterit partitioihin viittaamiseksi
 - ☞ Base prosessin fyysinen alkuosoite
 - ☞ Bounds prosessin loppuosoite (tai pituus)

KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 7

a) b) c) Osoitemuunnos Kuva 7.8



KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 8

Käyttöjärjestelmät I

d) Yksinkertainen segmentointi

KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 9

Yksinkertainen segmentointi

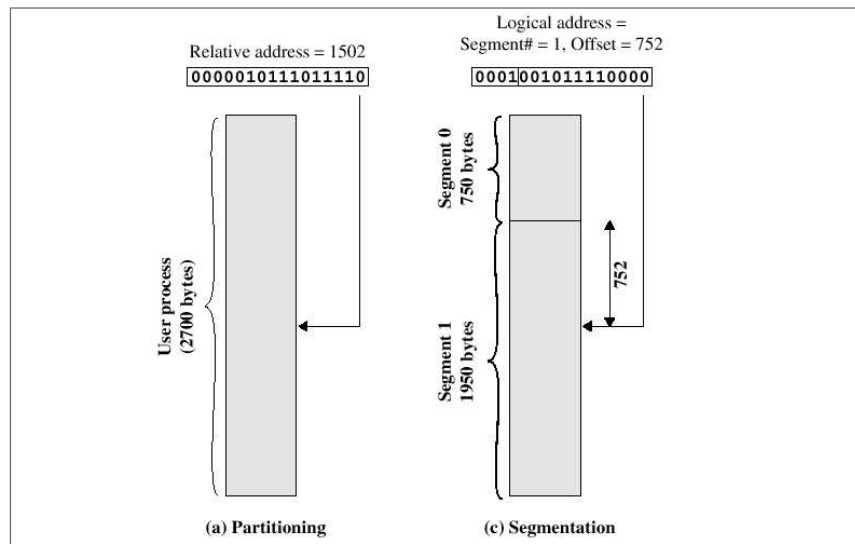
- **Ohjelmoija tai kääntäjä jakaa ohjelman kooltaan erilaisiin loogisiin kokonaisuuksiin, segmentteihin**
 - ◆ segmentti = esim. data-alue tai muutama aliohjelma
- **Kääntäjä tuottaa koodia, jossa segmentin sisäiset loogiset osoitteet**
 - ◆ osoite tavallaan muotoa (segmentti,siirtymä)
 - ☞ alkupään bitit kertoo segmenttinumeron
 - ☞ loppupään bitit kertoo siirtymän segm. sisällä
- **Järjestelmässä yleensä segmentin maksimikoko**

KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 10

Segmentointi

Kuva 7.11



KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 11

Yksinkertainen segmentointi

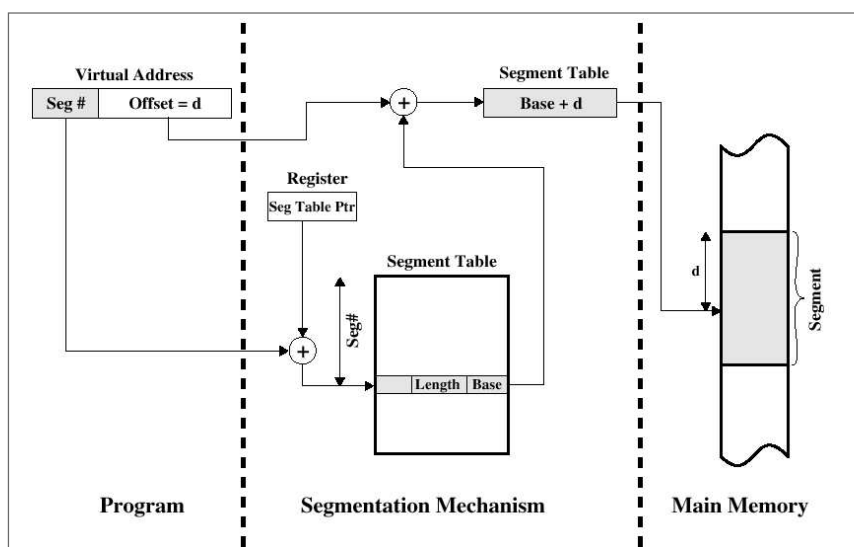
- **Kun KJ lataa prosessin muistiin, se voi sijoitella segmentit vapaasti muistiin**
 - ◆ kun käytössä yksinkertainen muistinhallinta, KJ tuo kaikki segmentit kerralla muistiin
 - ◆ muistia varataan aiemmin esitetyillä menetelmillä segmenttikohtaisesti
- **KJ ylläpitää prosessin segmenttitaulua**
 - ◆ PCB:ssä segmenttitaulun fyysinen alkuosoite
 - ☞ osoite MMU:hun, kun prosessi suoritukseen
 - ◆ alkioissa kunkin segmentin fyysinen alkuosoite (Base) ja pituus (Length)

KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 12

d) Osoitemuunnos

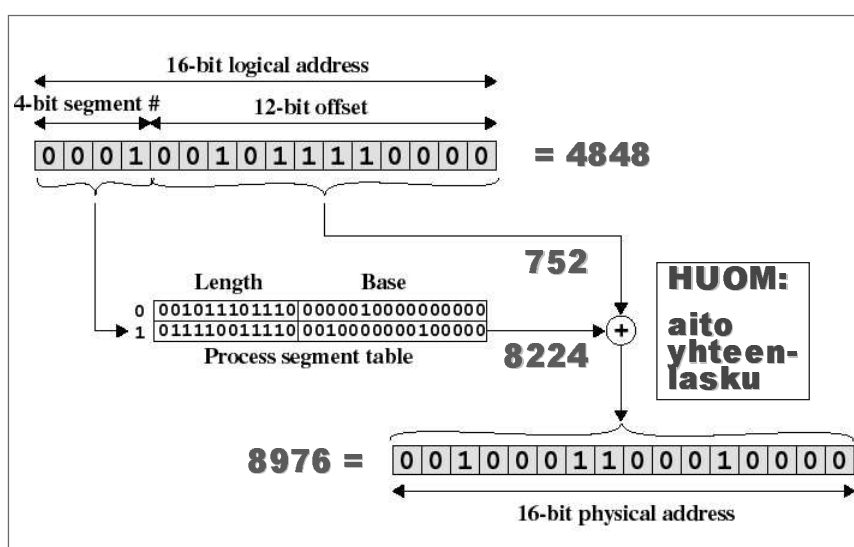
Kuva 8.12



KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 13

Looginen vs Fyysinen osoite



KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 14

Arviointia 1/2

- **Segmenttitaulun alkiossa alkuosoite ja pituus**
 - ◆ segmentin kokoa helppo kasvattaa/pienentää dynaamisesti
 - ☞ saattaa vaatia segmentin uudelleensijoittamista
 - ◆ osoitteen oikeellisuus tarkistettavissa MMU:ssa

- **Segmentti sopiva suojauksen yksikkö**
 - ◆ ohjelmoija määrittelee segmentit ja käyttöoikeudet
 - ◆ käytötapa kopioitu segmenttitaulun alkioon

Arviointia 2/2

- **Prosessille helpompi löytää paikka muistista**
 - ◆ varattava alue kooltaan jo selvästi pienempi
- **Ongelmana edelleen muistin pirstoutuminen**
 - ◆ segmentit eri kokoisia
 - ◆ muistin tiivistämistarve

- **Ratkaisu: sivutus**
 - ◆ jaa ohjelma aina vakiokokoisiihin sivuihin
 - ◆ varaa muistia aina sivunkokoisina palasina

Käyttöjärjestelmät I

e) Yksinkertainen sivutus

KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 17

Yksinkertainen sivutus

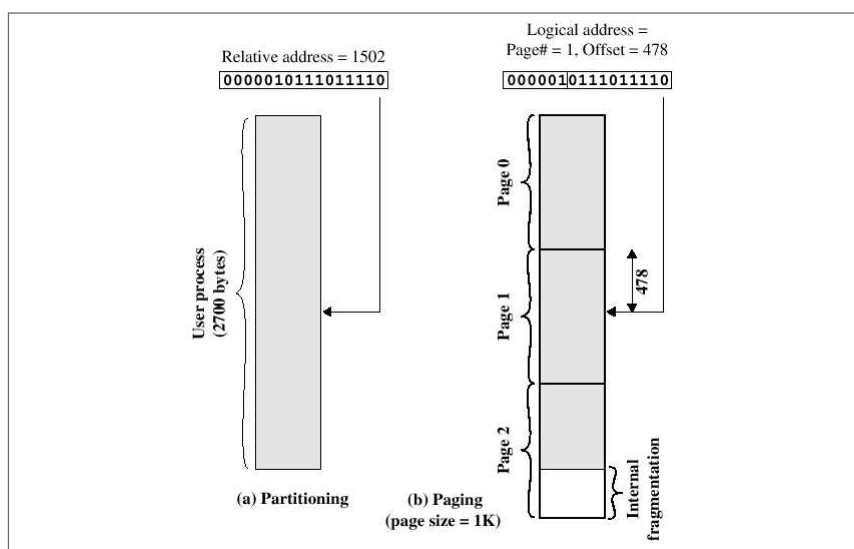
- **KJ varaa muistia sivutila kerrallaan** (page frame)
 - ◆ kaikki samankokoisia
 - ◆ 'suhteellisen pieniä', esim. 1KB tai 4 KB
 - ◆ koko aina joku 2:sen potenssi
 - ☞ käsittely helppoa laitetasolla
- **KJ käsittelee ohjelmaa sivuina** (page)
 - ◆ sivu ja sivutila samankokoisia
- **KJ sijoittaa prosessin sivut vapaisiin sivutiloihin**
 - ◆ Kun käytössä yksinkertainen muistinhallinta, KJ tuo kaikki sivut kerralla muistiin / vapauttaa

KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 18

Sivutus

Kuva 7.11



KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 19

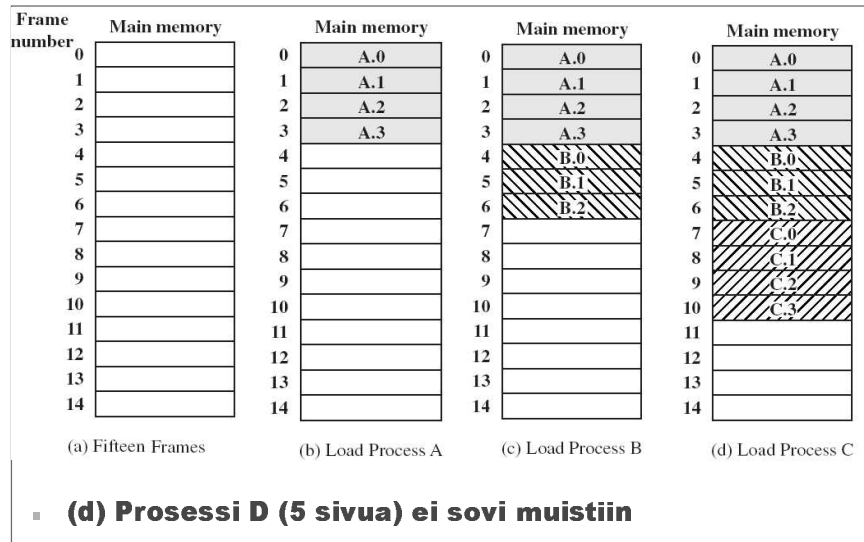
Yksinkertainen sivutus

- **KJ pitää prosessikohtaista sivutaulua** (page table)
 - ◆ PCB:ssä sivutaulun fyysinen alkuosoite
 - ☞ osoite MMU:hun, kun prosessi suoritukseen
 - ◆ alkiossa tieto sivutilasta, jossa ko. sivu majoituu
- **Looginen osoite muodostuu nyt parista (sivunumero, siirtymä)**
 - ☞ bittijonon alkupään bitit kertoo sivunumeron
 - ☞ bittijonon loppupään bitit siirtymän sivun sisällä
- **Vain viimeinen sivu voi aihe. pirstoutumista**

KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 20

Esimerkki muistin käytöstä



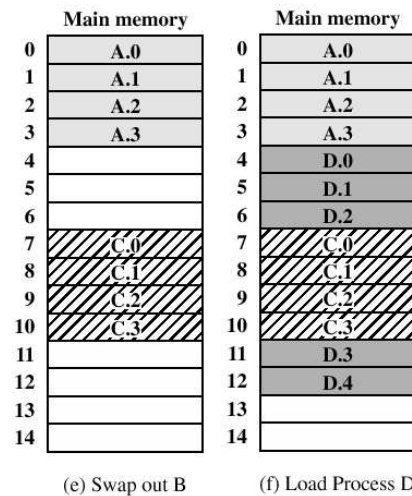
KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 21

Esimerkki (jatkuu)

Kuva 7.9

- **(e) KJ heittovaihtaa prosessin B levyille**
- **(f) KJ lataa prosessin D muistiin (5 sivua)**
- **Prosessin D sivut eivät sijaitse peräkkäin**
- **Ei ulk. pirstoutumista**
- **Sis. pirstoutumista keskim. 1/2 sivutilaa per prosessi**



KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 22

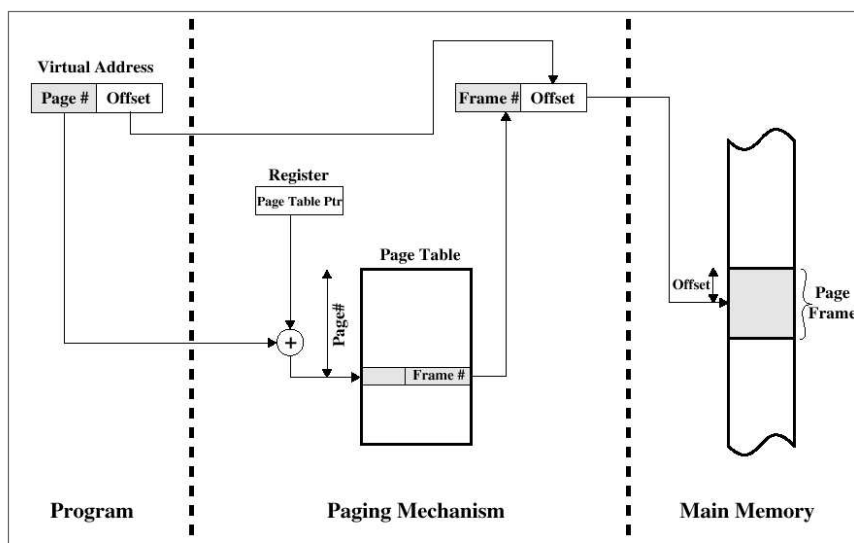
Esimerkin prosessien sivutaulut

| | | | | | | | | |
|----------------------|---|----------------------|---|----------------------|----|---|----|-----------------|
| 0 | 0 | 0 | — | 0 | 7 | 0 | 4 | Free frame list |
| 1 | 1 | 1 | — | 1 | 8 | 1 | 5 | |
| 2 | 2 | 2 | — | 2 | 9 | 2 | 6 | |
| 3 | 3 | Process B page table | | 3 | 10 | 3 | 11 | |
| Process A page table | | | | Process C page table | | 4 | 12 | |

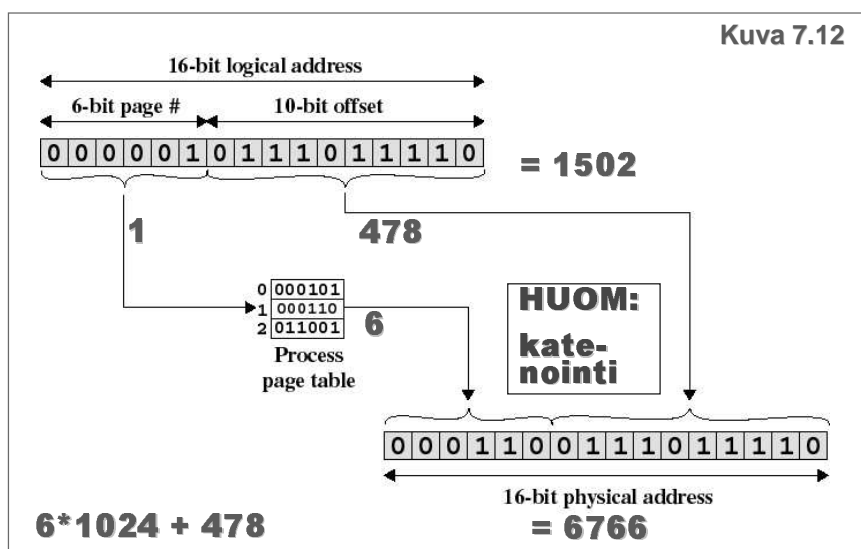
- **Jokaisella prosessilla ikioma sivutaulu**
 - ◆ Sivutaulun alkiossa sen sivutilan numero, jossa sivu sijaitsee
- **KJ:llä sivutilataulu (tai linkitetty lista), josta käy ilmi mitkä sivutilat vapaita**

Osoitemuunnos

Kuva 8.3



Looginen vs Fyysinen osoite



KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 25

Segmentointi vai Sivutus

- **Pirstoutuminen**
 - ◆ segmentointi aiheuttaa ulkoista, sivutus sisäistä
 - ◆ segmentoinnin yhteydessä muistin tiivistämistä
- **Osoitemuunnos MMU:ssa**
 - ◆ rekisteri, jossa segmentti-/sivutaulun fyys. osoite
 - ◆ segmentoinnissa yhteenlaskua (yht.laskulaitteistoa)
 - ◆ sivutuksessa katenointia (helppoa)

KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

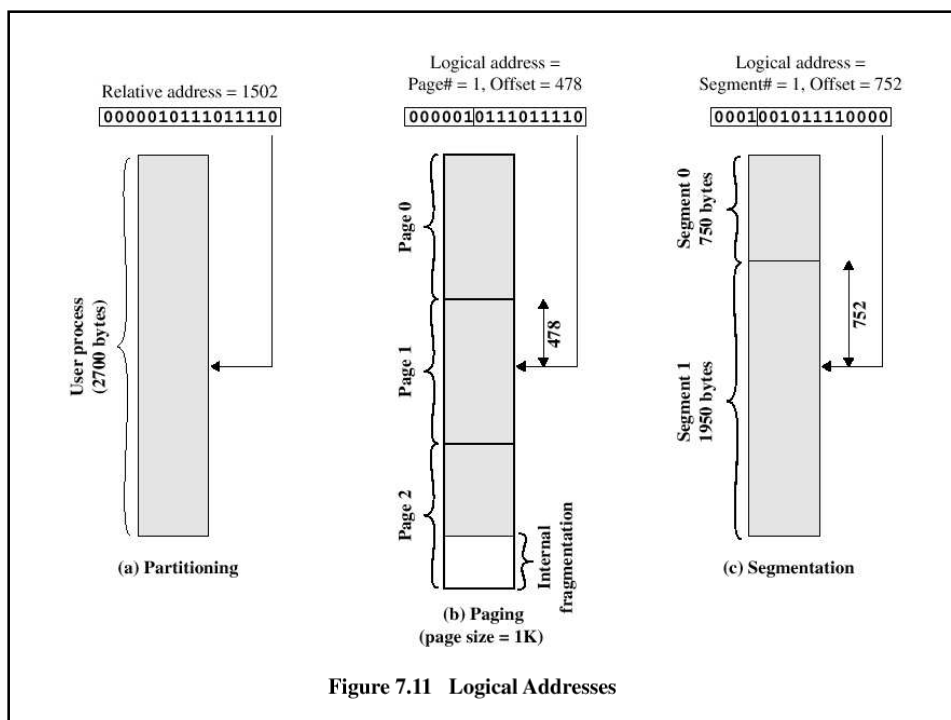
6 - 26

Segmentointi vai Sivutus

- **Sivutus ohjelmoijalle näkymätöntä, segmentointi ei**
- **Segmentti suojauksen kannalta loogisempi yksikkö kuin sivu**
- **Segmentointi sopii hyvin rutiinien dynaamiseen linkitykseen (yhteiskäyttö helpompi järjestää)**
- **Miksei molemmat: Yhdistettävissä!**

KJ-I S2004 / Tiina Niklander; kalvot Auvo Häkkinen

6 - 27



Yhteenveto

| Menetelmä | kuvaus | vahvuudet | heikkoudet |
|----------------------------|---|--|--|
| Kiinteä partitio | Muisti jaettu etukäteen osiin. Prosessin vain yhdessä osassa. | helppo toteutus | sisäinen pirstoutuminen maksimi prosessimäärä rajoitettu |
| Dynaaminen partitio | Muistia varataan tarpeen mukaan. Prosessin vain yhdessä osassa. | ei sis. pirst. par. muistin käyttöaste | ulkoinen pirstoutuminen, tiivistämistarve |
| Buddy System | Muistinvar. dyn., mutta kiinteänkokoisina osina. Prosessin vain yhdessä osassa. | ei juurikaan ulkoista pirstoutumista | vähäinen sisäinen pirstoutuminen |
| Yks. segmentointi | Prosessi jaettu segmentteihin. Segm. sijoitettavissa vapaasti. | ei sis. pirst. par. muistin käyttöaste | ulkoinen pirstoutuminen |
| Yks. sivutus | prosessi ja muisti jaettu sivuihin. Sij. vapaasti | ei ulk. pirst. | hyvin vähän sis. pirst. (vain viimeinen sivu) |