

Hoofstuk 9 - Binding

- Program word gelaai en vorm 'n proses
- Prosesse beslaan enige beskikbare geheue
- Hoe word adresse genereer?
 1. Vertaaltyd
 2. Looptyd
 3. Uitvoertyd

1

Hoofstuk 9 - Binding

- Dinamiese laai van roetines
- Dinamiese binding
- *Overlays*

2

Hoofstuk 9 - Virtuele en Fisiese Adresse

- SVE genereer logiese adresse
- Geheue-eenheid genereer fisiese adresse
- Logiese adresse → fisiese adresse

3

Hoofstuk 9 - Ruiling (*Swapping*)

- Proses moet in geheue wees om uit te voer
- Proses kan tydelik na sekondêre geheue beweeg word
- Wat beïnvloed die uitruilingsproses?
 1. Binding
 2. Hoeveelheid geheue
 3. Onvoltooide stelsel opdragte (*system calls*)

4

Hoofstuk 9 - Geheue Allokasie

- Enkel vs Veelvoudige partisies
- Strategieë vir allokasie: *First-fit*, *Best-fit* en *Worst-fit*
- Interne en eksterne fragmentasie
- Kompaktering van geheue

5

- Moontlike oplossing vir fragmentasie
- Fisiese geheue verdeel in rame (*frames*)
- Logiese geheue verdeel in bladsye (*pages*)
- Logiese adresse verdeel in bladsynommer (*p*) en verplasing (*d*)
- *p* word gebruik as indeks in 'n bladsytabel (*page table*)
- Tabel bevat beginadres (*base address*) van elke raam in fisiese geheue
- Beginadres en verplasing gekombineer, vorm fisiese adres en oorgedra na geheue-eenheid

6

Hoofstuk 9 - *Paging*

- Grootte van rame deur hardware bepaal
- Grootte meestal uitgedruk in magte van 2
- Voorbeeld: Logiese ruimte van 2^m en bladsy grootte van 2^n . Boonste $m - n$ bise van logiese address verskaf p terwyl onderste n bise die verplasing verskaf
- Interne fragmentasie kom steeds voor
- Tabelle vir elke proses: Konteksverandering duurder

7

Hoofstuk 9 - Hardware

- Tabel as versameling registers implementeer maar nie effektief vir groot tabelle
- Oplossing: Tabel in geheue gestoor en register wys na begin van tabel (PTBR)
- Nadeel: Elke geheue verwysing verg 2 opsoeke
- Oplossing: TLB
- Bladsynommer deur TLB vertaal na raamverwysing
- Elke konteksverandering maak TLB skoon om verkeerde vertalings te voorkom
- Berekening van effektiewe toegangstyd

8

Hoofstuk 9 - Beskerming

- Elke raam bevat beskermingsbisse
- Beheer uitgeoefen: Lees, skryf, lees/skryf
- Ongeldige operasie deur hardeware en stelsel onderskep
- Geldigheidsbis: Bepaal of proses na 'n spesifieke bladsy mag verwys

9

- Logiese adresruimte te groot: 2^{32}
- Voorbeeld: 4K bladsy grootte binne 32-bis ruimte benodig 2^{20} inskrywings. Elke inskrywing is 4 grepe. Totaal van 2^{22} grepe geheue benodig vir die tabel
- Oplossing: Gebruik tabel om bladsy tabelle te vind deur bladsynommer te verdeel
- Nadeel: Elke vlak van indireksie verg addisionele geheue toegang.
- Watter impak het dit op konteksverandering?

10

Hoofstuk 9 - Segmentasie

- Programmeerder sien geheue anders as bedryfstelsel
- Programme het logiese uitleg t.o.v. data en kode
- Segmentasie ondersteun bg. uitkyk
- Logiese adresse verdeel in segmente en verplasinge

11

Hoofstuk 9 - Segmentasie Hardeware

- Logiese adres beeld steeds af op fisiese adres
- Segmenttabel: Inskrywings beskryf eienskappe van segmente
- Eienskappe: Basis, Limiet en Toegangsregte
- Segmenttabel-basisregister (STBR)
- Segmenttabel-lengteregister (STLR)

12