

I. VIRTUELLE SPEICHERVERWALTUNG

Notwendigkeit

Immer größere Programme

Immer mehr Programme „gleichzeitig“

↪ verfügbarer Arbeitsspeicher schnell zu klein

Lösung: Nur gerade benötigte Teile der aktiven Programme im Arbeitsspeicher, Rest bei Bedarf aus Hintergrundspeicher nachladen (*swapping, paging*)

Umsetzung: **MMU** (*memory management unit*) setzt virtuelle Adressen in physikalische um

Virtueller Speicher

Speicherkapazität größer als effektive Hauptspeicherkapazität

Betriebssystem lagert nach Bedarf Speicherbereiche ein/aus

MMU-Adressberechnung hardwaremäßig eindeutig

Abbildungsinformation in Übersetzungstabellen gespeichert

↪ Abbildungsinformation für zusammenhängende Adressbereiche, um Übersetzungstabellen klein zu halten

Virtueller Speicher – Verwaltung (Segmentierung)

Virtueller Adressraum wird in Segmente verschiedener Länge zerlegt

Mehrere Segmente pro Programm (zB für Programmcode, Daten)

Segmente enthalten logisch zusammenhängende Informationen, relativ groß

Vorteile:

- spiegelt logische Programmstruktur wieder
- große Segmente ↪ relativ seltener Datentransfer

Nachteile:

- Datentransfer umfangreich falls notwendig
- Programm aus nur einem Code- und Datensegment
- ↪ muss vollständig eingelagert werden

Virtueller Speicher – Verwaltung (Seiten)

logischer und physikalischer Adressraum in Teile fester Länge (Pages) zerteilt

Pages relativ klein (256-4k Byte)

Viele Seiten pro Prozess, keine logischen Zusammenhänge

Vorteile:

- kleine Seiten ↪ nur wirklich benötigter Programmteil wird eingelagert
- geringerer Verwaltungsaufwand als Segmentierung

Nachteile:

- häufigerer Datentransfer als bei Segmentierung