

Tomáš Oberhuber

Faculty of Nuclear Sciences and Physical Engineering
Czech Technical University in Prague

REAL /
AUXILIARY /
VIRTUAL
storage

Způsoby
adresování
podle
architektury

Adresový
prostor

Mapa
adresového
prostoru

Datové
prostory

Stránkování v
adresových
prostorech

Dynamický
překlad adres

- 1 REAL / AUXILIARY / VIRTUAL storage
- 2 Způsoby adresování podle architektury
- 3 Adresový prostor
- 4 Mapa adresového prostoru
- 5 Datové prostory
- 6 Stránkování v adresových prostorech
- 7 Dynamický překlad adres

REAL /
AUXILIARY /
VIRTUAL
storage

Způsoby
adresování
podle
architektury

Adresový
prostor

Mapa
adresového
prostoru

Datové
prostory

Stránkování v
adresových
prostorech

Dynamický
překlad adres

Video na Youtube

REAL / AUXILIARY / VIRTUAL storage

REAL /
AUXILIARY /
VIRTUAL
storage

Způsoby
adresování
podle
architektury

Adresový
prostor

Mapa
adresového
prostoru

Datové
prostory

Stránkování v
adresových
prostorech

Dynamický
překlad adres

Data se ukládají do:

- **REAL STORAGE** = "rychlá" paměť např. RAM
- **AUXILIARY STORAGE** = diskové úložiště - tzv. **DASD** (Direct Acces Storage Device)

Uživatel pracuje s tzv. **VIRTUAL STORAGE**

- jde o vrstvu vytvořenou operačním systémem
- momentálně používaná data jsou uložena v real storage
- momentálně nepotřebná data jsou v souborech zvaných **PAGING DATA SETs** na DASD

Způsoby adresování podle architektury

Historie mainframe:

- 1970 - System/370 s operačním systémem MVS/370
 - používal 24. bitové adresování
 - maximální adresovatelné množství paměti = 16 MB
 - tato hranice se nazývá **THE LINE**
- 1983 - System/370 XA s operačním systémem MVS/XA (XA = extended architecture)
 - používal 31. bitové adresování
 - 32. bit = příznak pro určení, zda se adresuje pomocí 24 bitů nebo 31 bitů
 - maximální adresovatelné množství paměti = 2 GB
 - tato hranice se nazývá **THE BAR**
- 2000 - zSeries s operačním systémem z/OS
 - používá 64. bitové adresování
 - maximální adresovatelné množství paměti = 16 EB (exabytů) $1.6 \cdot 10^{18}$

Způsoby adresování podle architektury

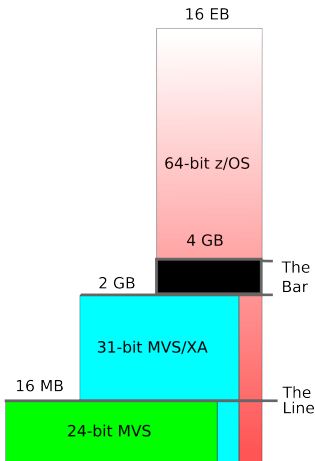


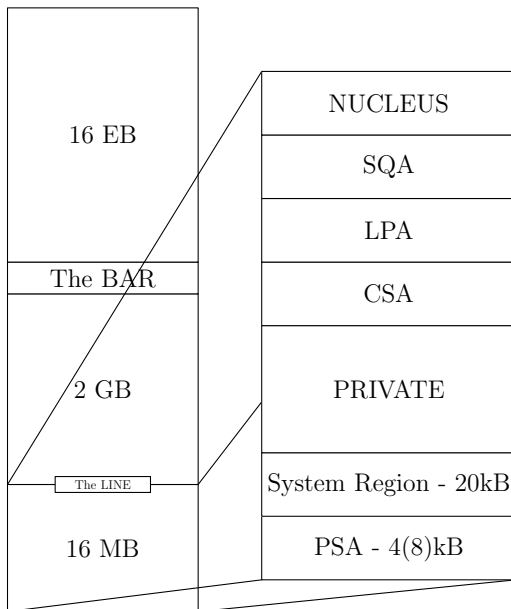
Figure: Zpětná kompatibilita zajišťuje, že dnešní z/OS podporuje 24. i 31. bitové adresování.

ADDRESS SPACE

- souvislý interval virtuálních adres ve VIRTUAL STORAGE
- začíná 0 a končí nejvyšší možnou adresou dané architektury
 - na počátku má však každý a.s jen 2 GB
 - pro alokaci nad tuto hranici je nutné použít instrukce pro 64. bitové adresování - IARV64
- a.s. obsahuje data, instrukce programu a systémový kód
- první a.s. se vytváří během **IPL** = Initial Program Load (odpovídá bootování u Unixu nebo Windows)
- další a.s. vznikají při spuštění nové úlohy nebo při přihlášení uživatele

- a.s. je identifikován pomocí tzv. **ADDRESS SPACE ID - ASID**
- každý uživatel má svůj a.s. - to zabraňuje konfliktům mezi uživateli (důležité pro bezpečnost)
- z/OS umožňuje rychlou komunikaci mezi různými a.s.
- v každém a.s. lze spouštět více úloh, ty jsou pak popsány pomocí TASK CONTROL BLOCK - TCB
- analogie s Unixem
 - ASID = Unix process ID
 - TCB = vlákno

Mapa adresového prostoru



REAL /
AUXILIARY /
VIRTUAL
storage

Způsoby
adresování
podle
architektury

Adresový
prostor

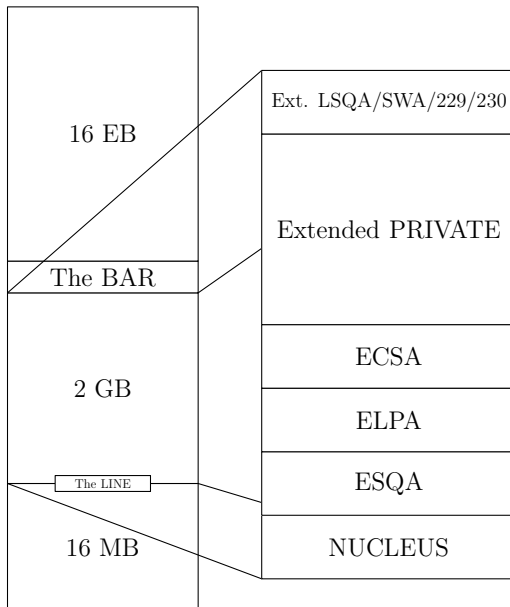
Mapa
adresového
prostoru

Datové
prostory

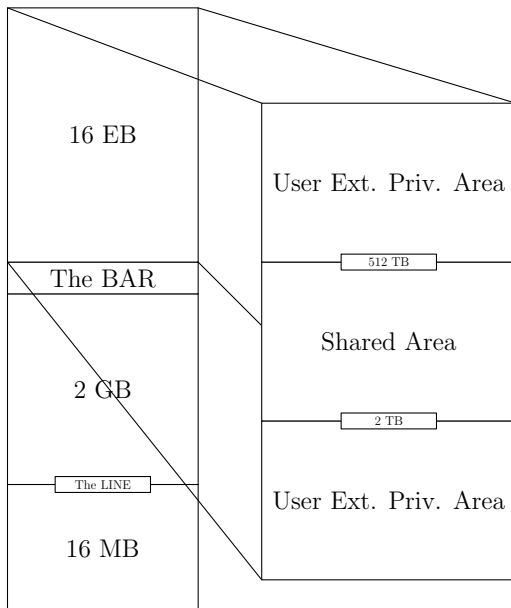
Stránkování v
adresových
prostorech

Dynamický
překlad adres

Mapa adresového prostoru



Mapa adresového prostoru



Mapa adresového prostoru

REAL /
AUXILIARY /
VIRTUAL
storage

Způsoby
adresování
podle
architektury

Adresový
prostor

Mapa
adresového
prostoru

Datové
prostory

Stránkování v
adresových
prostorech

Dynamický
překlad adres

- **PRIVATE** = oblast pro aplikace a data
- **NUCLEUS** = oblast pro jádro
 - jádro se načítá při IPL, je vždy v REAL STORAGE
 - může obsahovat rozšiřující moduly obsažené v SYS1.NUCLEUS
 - konfigurace se provádí pomocí SYS1.PARMLIB(NUCLSTxx)
- **SQA** = system queue area
 - obsahuje tabulky a fronty potřebné pro systém (závisí na konfiguraci)
 - velikost se určuje v SYS1.PARMLIB
 - nejprve se používá ESQA
 - pokud se ESQA vyčerpá, alokuje se v ECSA, dále SQA, CSA -> spadnutí systému
- **CSA** = common service area
 - obsahuje data adresovatelná všemi a.s.
 - používá se při komunikaci mezi různými a.s. ???
 - velikost se určuje v SYS1.PARMLIB

Mapa adresového prostoru

REAL /
AUXILIARY /
VIRTUAL
storage

Způsoby
adresování
podle
architektury

Adresový
prostor

Mapa
adresového
prostoru

Datové
prostory

Stránkování v
adresových
prostorech

Dynamický
překlad adres

- **LPA** = link pack area
 - obsahuje moduly systémové programy, které jsou read-only
 - tato oblast je stejná pro všechny A.S
 - dělí se na:
 - PLPA = pageable LPA - obsahuje moduly, které mohou být odswapovány
 - FLPA = fixed LPA - musí být stále v paměti
 - MPLA = modified LPA - používá se při IPL
- **LSQA** = local system queue area
 - obsahuje tabulky a fronty určené pro daný A.S.
- **SWA** = scheduler work area
 - obsahuje CB = control blocks

REAL /
AUXILIARY /
VIRTUAL
storage

Způsoby
adresování
podle
architektury

Adresový
prostor

Mapa
adresového
prostoru

Datové
prostory

Stránkování v
adresových
prostorech

Dynamický
překlad adres

DATA SPACE = jako A.S., ale je určený jen pro data, nelze z něj spouštět programy

HIPERSPACE = DATA SPACE umístěný v REAL STORAGE

Výhody:

- více místa pro data
- izolace dat od jiných úloh běžících v daném A.S., zlepšuje bezpečnost
- umožňuje sdílení dat mezi programy z různých A.S. - místo CSA

Stránkování v adresových prostorech

REAL /
AUXILIARY /
VIRTUAL
storage

Způsoby
adresování
podle
architektury

Adresový
prostor

Mapa
adresového
prostoru

Datové
prostory

Stránkování v
adresových
prostorech

Dynamický
překlad adres

Veškerá paměť se dělí na stránky o velikosti 4 kB.

- **FRAMES** v případě REAL STORAGE
- **SLOTS** v případě AUXILIARY STORAGE
- **PAGES** v případě VIRTUAL STORAGE

Stránkování v adresových prostorech

REAL /
AUXILIARY /
VIRTUAL
storage

Způsoby
adresování
podle
architektury

Adresový
prostor

Mapa
adresového
prostoru

Datové
prostory

Stránkování v
adresových
prostorech

Dynamický
překlad adres

VIRTUAL STORAGE

PAGES			
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

REAL STORAGE

FRAMES			
D		L	J
	A		
	O	I	
K			E

AUXILIARY STORAGE

SLOTS			
H	B	P	
	F		N
G		C	
	M		

Stránkování v adresových prostorech

REAL /
AUXILIARY /
VIRTUAL
storage

Způsoby
adresování
podle
architektury

Adresový
prostor

Mapa
adresového
prostoru

Datové
prostory

Stránkování v
adresových
prostorech

Dynamický
překlad adres

- **PAGING** = přesun stránky mezi REAL STORAGE a AUXILIARY STORAGE.
Do AUXILIARY STORAGE se dávají nejméně využívané stránky.
- **SWAPING** = přesun více stránek mezi REAL STORAGE a AUXILIARY STORAGE
- **PAGE STEALING** = pokud je málo FRAMEů v REAL STORAGE, může se sebrat FRAME od jiného uživatele, který není tolik využíván.

Dynamický překlad adres

REAL /
AUXILIARY /
VIRTUAL
storage

Způsoby
adresování
podle
architektury

Adresový
prostor

Mapa
adresového
prostoru

Datové
prostory

Stránkování v
adresových
prostorech

Dynamický
překlad adres

DAT = dynamic address translation

Jde o určení fyzického umístění stránky podle její virtuální adresy, tj. musíme

- zjistit, zda je stránka v paměti nebo odswapovaná na disk
- najít adresu počátku stránky
- přičíst offset adresy uvnitř stránky

Tento proces musí být velice rychlý, protože se provádí při každém přístupu do paměti.

Nabízí se vytvořit jednu velkou tabulku s informacemi o každé stránce adresového prostoru.

Dynamický překlad adres

REAL /
AUXILIARY /
VIRTUAL
storage

Způsoby
adresování
podle
architektury

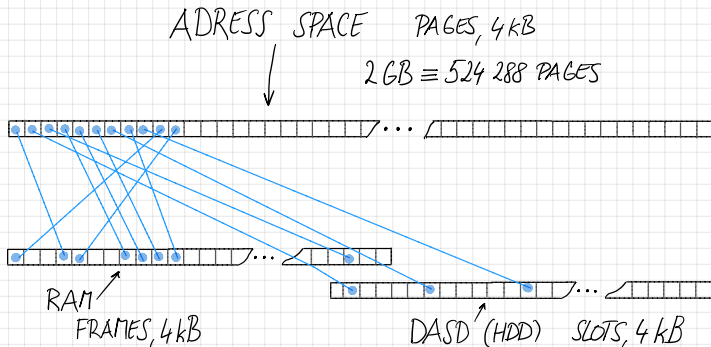
Adresový
prostor

Mapa
adresového
prostoru

Datové
prostory

Stránkování v
adresových
prostorech

Dynamický
překlad adres



Dynamický překlad adres

REAL /
AUXILIARY /
VIRTUAL
storage

Způsoby
adresování
podle
architektury

Adresový
prostor

Mapa
adresového
prostoru

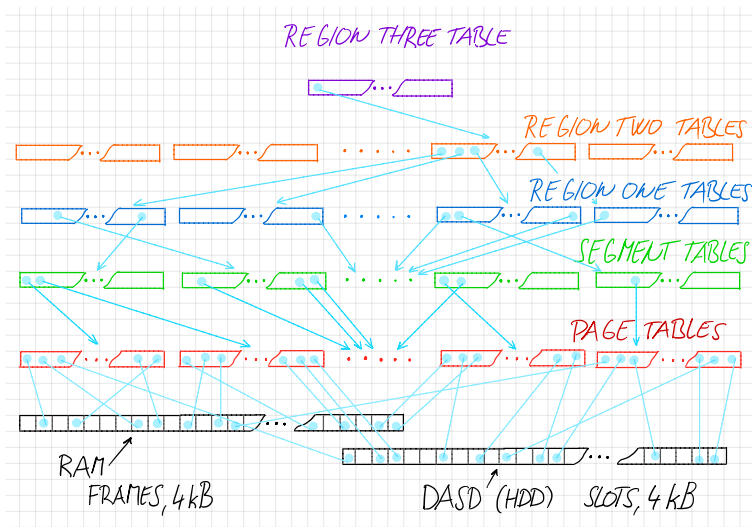
Datové
prostory

Stránkování v
adresových
prostorech

Dynamický
překlad adres

- taková tabulka by byla příliš velká
- navíc ne každý adresový prostor zabírá celé 2 GB
- tabulku nahradíme strukturou podobnou B-stromu

Dynamický překlad adres



Dynamický překlad adres

REAL /
AUXILIARY /
VIRTUAL
storage

Způsoby
adresování
podle
architektury

Adresový
prostor

Mapa
adresového
prostoru

Datové
prostory

Stránkování v
adresových
prostorech

Dynamický
překlad adres

- **PGT** = page table - 256 položek -> pokrývá 1 MB = 1 SEGMENT
- **SGT** = segment table - 2048 položek -> pokrývá 2 GB = 1 REGION
- **R1T** = region first table - 2048 položek -> pokrývá 4 TB
- **R2T** = region second table - 2048 položek -> pokrývá 8 PB
- **R3T** = region third table - 2048 položek -> pokrývá 16 EB

Dynamický překlad adres

REAL /
AUXILIARY /
VIRTUAL
storage

Způsoby
adresování
podle
architektury

Adresový
prostor

Mapa
adresového
prostoru

Datové
prostory

Stránkování v
adresových
prostorech

Dynamický
překlad adres

- **PGT** = page table - 256 položek -> pokrývá 1 MB = 1 SEGMENT
- **SGT** = segment table - 2048 položek -> pokrývá 2 GB = 1 REGION
- **R1T** = region first table - 2048 položek -> pokrývá 4 TB
- **R2T** = region second table - 2048 položek -> pokrývá 8 PB
- **R3T** = region third table - 2048 položek -> pokrývá 16 EB

Položka PGT obsahuje fyzickou adresu stránky. Pokud není stránka v REAL STORAGE, vznikne hardwarové přerušení - page fault interrupt - a stránka je přenesena z AUXILIARY STORAGE.

Rozdělení virtuální adresy

REAL /
AUXILIARY /
VIRTUAL
storage

Způsoby
adresování
podle
architektury

Adresový
prostor

Mapa
adresového
prostoru

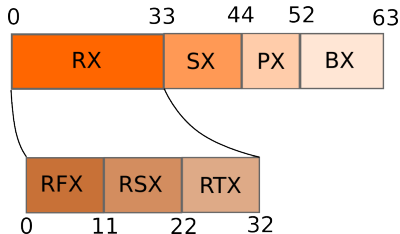
Datové
prostory

Stránkování v
adresových
prostorech

Dynamický
překlad adres

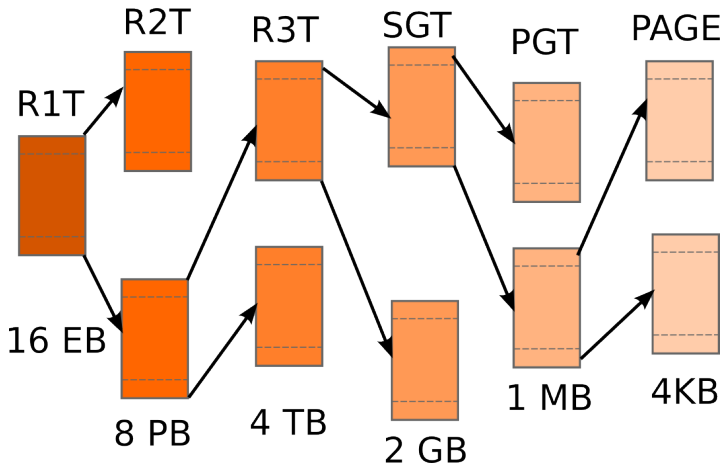
- **SEGMENT** = souvislá skupina stránek v adresovém prostoru o velikosti 1 MB
- **REGION** = souvislá skupina segmentů o velikosti 2-8 GB

Rozdělení virtuální adresy



- **BX** = byte index (52-63) 12 bitů $\equiv 4 \text{ kB}$
- **PX** = page index (44-51) 8 bitů $\equiv 256 \cdot 4 \text{ kB} = 1 \text{ MB}$
- **SX** = segment index (33-43) 11 bitů $\equiv 2048 \cdot 1 \text{ MB} = 2 \text{ GB}$
- **RX** = region index (0-32) 33 bitů $\equiv 8589934592 \cdot 2 \text{ GB} = 16 \text{ EB}$
 - **RFX** = region first index (0-10)
 - **RSX** = region second index (11-21)
 - **RTX** = region third index (22-32)

Dynamický překlad adres



REAL /
AUXILIARY /
VIRTUAL
storage

Způsoby
adresování
podle
architektury

Adresový
prostor

Mapa
adresového
prostoru

Datové
prostory

Stránkování v
adresových
prostorech

Dynamický
překlad adres