

# SPRÁVA PAMĚTI V KONKRÉTNÍCH OS

Správa paměti je klíčovou součástí operačního systému, která zajišťuje přidělování, ochranu a efektivní využití paměti. Tato prezentace se zaměřuje na způsoby, jakými je paměť spravována v různých operačních systémech, včetně MS-DOS, Windows a UNIX/Linux. Zkoumáme adresování, sdílení paměti, segmentaci a stránkování, ochranu paměti, virtualizaci a architektury jako NUMA a endian. Prezentace rovněž poskytuje přehled o metodách optimalizace přístupu k paměti pro jednotlivé OS.

# SPRÁVA PAMĚTI V MS-DOS

## Základní segmentace

- Procesům je přiděleno několik segmentů paměti pro různé účely (kód, data, zásobník).

## Bez ochrany paměti

- Procesy mohou přistupovat k jakékoli části paměti včetně té, která je vyhrazena pro OS.

## Jednoprogramový systém

- MS-DOS může spustit vždy pouze jeden program.

## Rezidentní programy (TSR)

- Programy, které zůstávají v paměti pro zpracování přerušení a rozšíření funkcí OS.

# SPRÁVA PAMĚTI VE WINDOWS

## Adresace v chráněném režimu

- Deskriptory, selektory a LDT/GDT tabulky slouží k adresování segmentů paměti.

## Virtuální paměť

- Kombinace segmentace a stránkování na žádost; stránkovací soubor **pagefile.sys** na disku.

## Struktura adresového prostoru

- Na 32bitovém systému rozdělen na 2GB pro uživatele a 2GB pro systém; na 64bitovém až 16TB.

## Sdílení paměti a DLL

- Dynamicky linkované knihovny sdílené mezi procesy, copy-on-write při úpravě obsahu.

# NUMA ARCHITEKTURA VE WINDOWS

## Non-Uniform Memory Access (NUMA)

- Paměť rozdělena na uzly, každý připojen k specifickým procesorům pro rychlý přístup.

## Výhody NUMA

- Rychlejší přístup k lokální paměti uzlu, efektivní pro víceprocesorové servery.

## Přístup k paměti

- Procesor má rychlejší přístup ke své lokální paměti než k paměti jiného uzlu.

## Použití v praxi

- NUMA je preferována na výkonných serverech, kde je kritická rychlost přístupu k paměti.

# ENDIAN POŘADÍ (LITTLE VS BIG ENDIAN)

## Definice

- Způsob ukládání vícebajtových čísel do paměti (Little Endian – nižší bajt na nižší adrese).

## Little Endian

- Ukládá nižší bajty na nižších adresách (Intel, DEC Alpha).

## Big Endian

- Ukládá vyšší bajty na nižších adresách (Motorola, SPARC).

## Problematika compatibility

- Při komunikaci mezi systémy s různým Endian pořadím mohou nastat problémy.

## Příklad uložení čísla

- 1234ABCDh na adresu 400h

Little Endian	Big Endian
adresa 400h = CDh	adresa 400h = 12h
adresa 401h = ABh	adresa 401h = 34h
adresa 402h = 34h	adresa 402h = ABh
adresa 403h = 12h	adresa 403h = CDh
400 401 402 403	400 401 402 403
... CD AB 34 12 ...	... 12 34 AB CD ...

# SPRÁVA PAMĚTI V UNIX/LINUX

## Adresový prostor

- Rozdělen na segmenty: kód, globální data, zásobník uživatele a režimu jádra.

## Virtualizace paměti

- Stránkování na žádost nebo stránkování se segmentací; běžné jsou odkládací soubory.

## Sdílení paměti a mapování souborů

- Mechanismus mapování umožňuje přístup k souborům v paměti, sdílení kódu pro více procesů.

## Víceúrovňové stránkování

- Optimalizace zpracování stránek na základě hierarchických tabulek stránek.

# ROZDĚLENÍ ADRESOVÉHO PROSTORU V LINUXU

## Rozdělení na uživatelskou a systémovou část

- Na 32bitovém systému obvykle 3GB pro uživatelský prostor a 1GB pro režim jádra.

## 64bitová rozšíření

- Systémy mohou adresovat 256TB, typicky rozděleno na 128TB pro uživatele a systém.

## Správa segmentů a stránkování

- Procesy jsou segmentovány a stránky jsou spravovány víceúrovňově pro efektivní přístup.

## Optimalizace výběru obětí

- Používá se varianta hodinového algoritmu, kdy staré stránky mají nižší prioritu pro odložení.

# VIRTUALIZACE PAMĚTI A BEZPEČNOST V LINUXU

## Mechanismus copy-on-write

- Při úpravě sdílené paměti je vytvořena privátní kopie pro proces, který provádí změny.

## Paměťové ochranné bity (NX/XD)

- Ochrana před spuštěním kódu na datových stránkách (NX bit u AMD, XD bit u Intel).

## Sdílení paměti mezi procesy

- Sdílení kódu a mapování souborů; mapování souboru umožňuje pracovat s daty na disku jako s pamětí.

## Vícestrojová architektura a portabilita

- Linux běží na široké škále hardwarových platforem, což podporuje jeho flexibilitu.



# SHRNUTÍ

- Každý OS má unikátní přístup k segmentaci, stránkování a správě adresového prostoru.
- MS-DOS má omezenou ochranu paměti, kdežto Windows a Linux implementují pokročilé metody virtualizace a ochrany.
- NUMA architektura optimalizuje přístup k paměti na víceprocesorových serverech.
- Různé Endian formáty ovlivňují kompatibilitu dat mezi systémy.
- UNIX/Linux poskytuje pokročilé techniky pro sdílení a ochranu paměti.

# KONTROLNÍ OTÁZKY

1. Jak funguje rozdělení adresového prostoru v 32bitových a 64bitových systémech Windows?
2. Jaký je rozdíl mezi Little Endian a Big Endian a jaký je jejich dopad na správu paměti?
3. Jak NUMA architektura ovlivňuje výkon víceprocesorových systémů?
4. Co je mechanismus copy-on-write a jak funguje v Linuxu?
5. Jaké jsou hlavní rozdíly ve správě paměti mezi MS-DOS a moderními operačními systémy?

# DOPORUČENÁ LITERATURA

- 1. Silberschatz, A., Galvin, P. B., & Gagne, G.** - *Operating System Concepts* - Přehled správy paměti v OS.
- 2. Tanenbaum, A. S., & Bos, H.** - *Modern Operating Systems* - Pokročilé metody správy paměti.
- 3. Stallings, W.** - *Operating Systems: Internals and Design Principles* - Detailní principy alokace a ochrany paměti.
- 4. Maurice J. Bach** - *The Design of the UNIX Operating System* - Podrobný popis paměťových systémů UNIX.
- 5. Microsoft Documentation on Memory Management** - Oficiální dokumentace správy paměti v Windows OS.