Operační systémy 2

Implementace virtuální paměti a stránkování

Petr Krajča

Katedra informatiky
Univerzita Palackého v Olomouci

Úvodem

- procesory x86 se vyvíjeli několik desetiletí
- relikty minulosti (zpětná kompatibilita)
  - 8086/88 – segmentace použita k adresování (aby 16bitový procesor mohl adresovat 1 MB RAM, segmenty bloky o velikosti 64 kB pro kód, data, zásobník, ...)
  - 80286 – segmentace použita k ochraně paměti, implementaci virtuální paměti (omezení na velikost segmentu); stále 16b procesor
  - 80386 – přidána podpora stránkování, koexistuje se segmentací
  - AMD64 – stránkování primární způsob správy paměti (segmentace stále existuje, ale používá se jen ke zprávě oprávnění)

Segmentace

- paměť je rozdělena na několik segmentů (např. kód, data, zásobník)
- speciální případ přidělování souvislých bloků paměti
- stránkování je obvykle pro programátora nerozeznatelné
- naproti tomu segmentace umožňuje rozdělit program do logických celků (kód, data)
- logická adresu ve tvaru segment + offset (vzhledem k segmentu), ta se až poté převádí na fyzickou adresu pomocí stránkování
- ochrana paměti přes začátek a délku segmentu
- úroveň oprávnění segmetu s aktuálně prováděným kódem, určuje úroveň oprávnění procesoru

Úrovně oprávnění

- čtyři úrovně oprávnění (rings)
- nelze používat některé instrukce
- oddělují jádro a uživatelský prostor systému
- ring 0 – nejvyšší oprávnění (jádro)
- ring 3 – nejmenší oprávnění (uživatelské procesy)
- nejčastěji se používá kombinace ring 0 + ring 3 (privilegovaný a neprivilegovaný režim)
- ostatní původně určeny pro ovladače, příp. knihovny
- mikrokernel; virtualizace (XEN)
- přechod mezi úrovněmi přes brány (gates); nebo speciálními instrukcemi

Typy adres na procesorech i386

- **logická adresa** – pracuje s ní proces; 48 bitů (16 selektor segmentu; 32 offset); segment je často implicitní
  - instrukce implicitně brány z kódového segmentu (registr CS)
  - přístup do paměti přes ESP nebo EBP implicitně používá zásobníkový segment (registr SS)
  - ostatní instrukce využívají převážně datový segment (registr DS)
  - další segmenty např. pro lokální data vláken (thread-local storage)

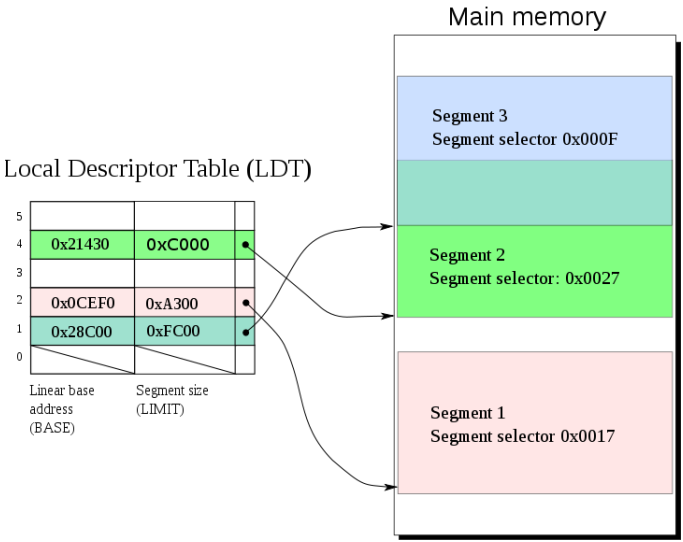
```
jmp 0xdeadbeef          ;; -->    jmp cs:0xdeadbeaf
mov eax, [esp + 4]      ;; -->    mov [ss:esp + 4]
mov eax, 0x12345678     ;; -->    mov [ds:0x12345678]
mov eax, [gs:ebx + 10]  ;; explicitni pouziti segmentu
```

- **lineární (virtuální) adresa** – v adresním prostoru procesu (32 bitů)
- **fyzická adresa** – ,,číslo bytu" přímo v primární paměti (32 bitů, s PAE 36); sdílená dalšími HW zařízeními; (nevyužité adresy mohou mít další použití, např. SWAP)

I386: Segmentace I.

- paměť je možné rozdělit na segmenty (kód, data, zásobník, etc.)
- pro každý segment lze nastavit oprávnění (ochrana paměti)
- segmenty jsou popsány pomocí deskriptorů (8 B záznam)
  - báze
  - limit (velikost segmentu)
  - požadovaná úroveň oprávnění (ring 0-3)
- deskriptory segmentů uloženy v
  - Global Descriptor Table (GDT) – sdílená všemi procesy
  - Local Descriptor Table (LDT) – každý proces má vlastní
- každá může mít až 8192 záznamů
- určené registry GDTR, LDTR
- první záznam v GDT ,,null" deskriptor
- granularita (stránky vs. byty)

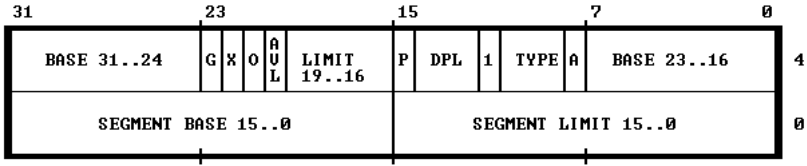
I386: Segmentace II.



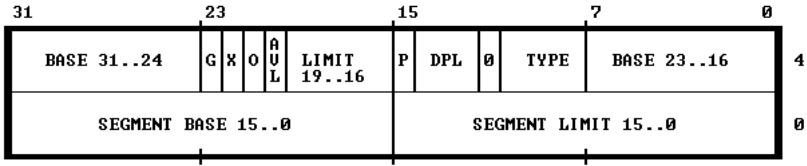
I386: Segmentace III.

Figure 5-3. General Segment-Descriptor Format

DESCRIPTORS USED FOR APPLICATIONS CODE AND DATA SEGMENTS



DESCRIPTORS USED FOR SPECIAL SYSTEM SEGMENTS



A - ACCESSED
AVL - AVAILABLE FOR USE BY SYSTEMS PROGRAMMERS
DPL - DESCRIPTOR PRIVILEGE LEVEL
G - GRANULARITY
P - SEGMENT PRESENT

I386: Segmentace IV.

- do segmentových registrů (CS, SS, DS, ES, FS, GS) se ukládá selector (16 bitů) – ukazatel do GDT nebo LDT

Figure 5-6. Format of a Selector

TI – TABLE INDICATOR
RPL – REQUESTOR'S PRIVILEGE LEVEL

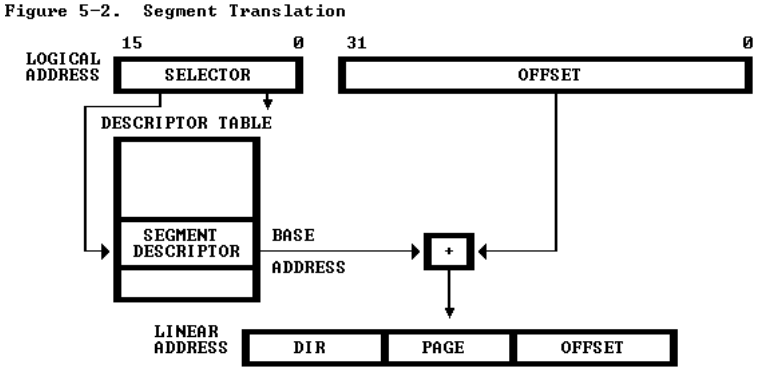
- při načtení se do seg. registru načte i deskriptor (ale nejde k němu explicitně přistupovat)

Figure 5-7. Segment Registers

	16-BIT VISIBLE SELECTOR	HIDDEN DESCRIPTOR
CS		
SS		
DS		
ES		
FS		
GS		

I386: Překlad adres I. (segmentace)

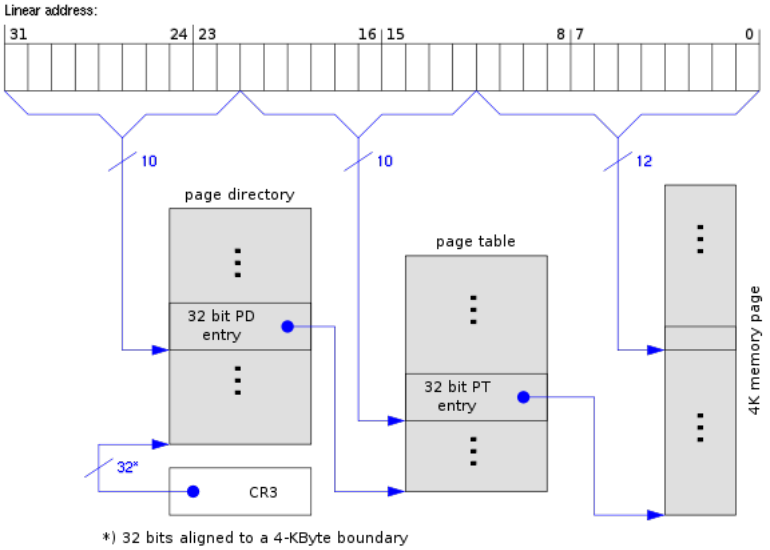
- logická adresa $\Longrightarrow$ linární adresa (segmentace)
- ověří se oprávnění a limit (přístup za hranici segmentu) $\Longrightarrow$ neoprávněný přístup
- báze segmentu je sečtena s offsetem $\Longrightarrow$ lineární adresa



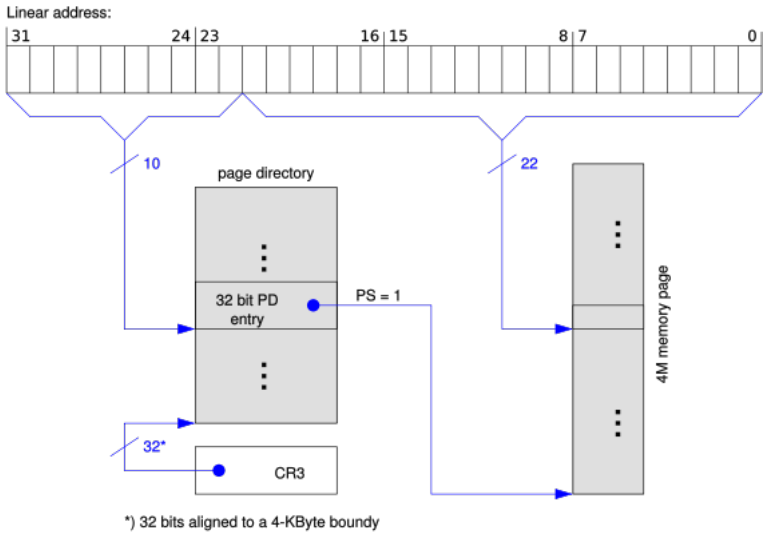
I386: Překlad adres II. (stránkování)

- linární adresa $\Longrightarrow$ fyzická adresa
- standardní stránka/rámec: 4 kB
- hierarchická struktura
  - adresář stránkovacích tabulek (Page Directory)
  - adresář stránek (Page Tables)
  - offset
- adresáře mají velikost jedné stránky, každá položka 4B $\Longrightarrow$ 1024 záznamů
- lineární adresa rozdělana na $10 + 10 + 12$ bitů (PDI + PTI + offset)
- maximální kapacita 4 GB
- adresa PDT v CR3 (zarovnaná na celé stránky)
- nastavením příznaku v PDT (pro adresu rámce se používá jen 20b), lze obejít přepočet přes PT a používat stránky velikosti 4MB (zbytek adresy je offset)
- velikost stránek lze kombinovat

I386: Překlad adres III. (stránkování – 4 kB stránka)

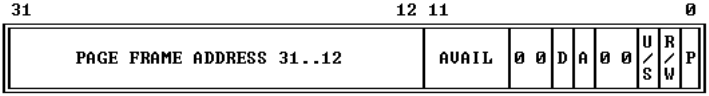


I386: Překlad adres IV. (stránkování – 4 MB stránka)



I386: Překlad adres V. (záznam v tabulce stránek)

Figure 5-10. Format of a Page Table Entry



P - PRESENT
R/W - READ/WRITE
U/S - USER/SUPERVISOR
D - DIRTY
AVAIL - AVAILABLE FOR SYSTEMS PROGRAMMER USE

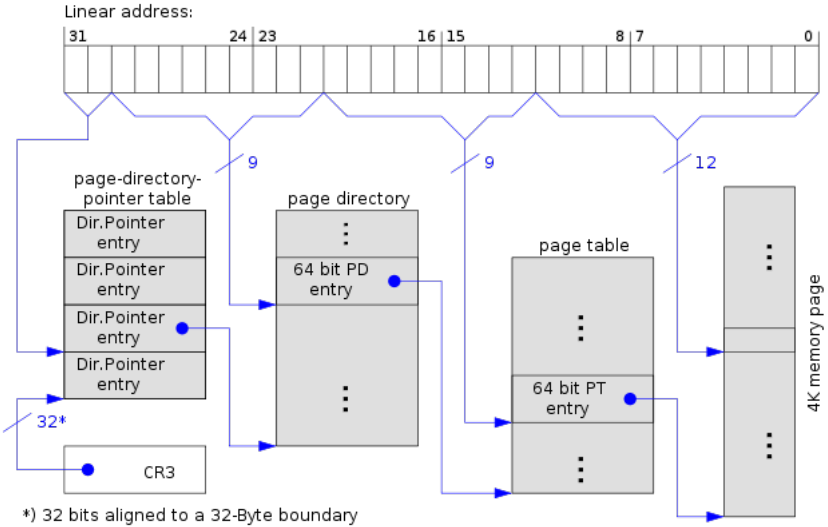
NOTE: 0 INDICATES INTEL RESERVED. DO NOT DEFINE.

I386: Překlad adres IV. (PAE)

Physical Address Extension

- umožňuje rozšířit využitelnou pamět RAM z 4GB na 64GB (Pentium Pro a novější)
- přesměruje část adresního prostoru do jiné části fyzické paměti
- změna formátu segmentových deskriptorů
- stránky 4 KB/2 MB
- změna na úrovni OS (případně ovladačů);
- bez úprav jednotlivé procesy stále omezeny na 4 GB (AWE)
- každá tabulka 4 KB, ale velikost záznamu 8 B $\Longrightarrow$ 512 záznamů
- stránkování trojúrovňové
- adresa rozdělena na $2 + 9 + 9 + 12$ bitů
  - 2b – ukazatel na adresář tabulek stránek (Page Directory Pointer Index)
  - 9b – ukazatel v adresáři tabulek stránek
  - 9b – ukazatel v tabulce stránek
  - 12b – offset
- velké stránky 2MB $\Longrightarrow$ offset 21 bitů
- potenciální rozšíření

I386: Překlad adres VI. (PAE – 4 kB stránka)



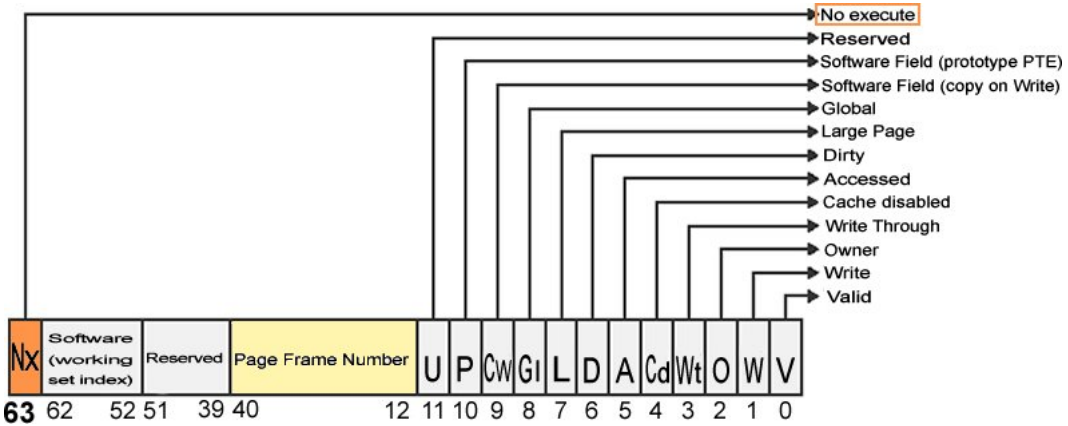
AMD64: Typy adres

- segmenty existují, ale nepoužívají se k adresaci, pouze ke kontrole oprávnění
- deskriptor kódového segmentu se používá k přechodu mezi 32- a 64bitovým režimem
- současné procesory AMD64:
  - 36 až 46bitové fyzické adresy (max. 52; způsob stránkování), tj. 64 GB až 64 TB
  - 48bitové logické adresy (max. 64; velikost registrů), tj. 256 TB (max. 16 EB)
- možnost rozšíření $\Longrightarrow$ kanonické adresy
- nejvyšší platný bit je okopírován do vyšších bitů
- dělí paměť na tři bloky (viz Keprt str. 119)

AMD64: Stránkování

- používá se režim podobný PAE
- zavedena čtyřúrovňová hierarchie stránkovací tabulky
- záznamy v tabulkách stránek mají 8 B
- při 4 kB stránkách $\Longrightarrow 4 \times 9 + 12 = 48$ adresovatelných bitů
- pro 2 MB stránky vynechaná jedna úroveň, možnost použít 4 rezervované bity $\Longrightarrow$ 52 bitů

AMD64: Formát záznamu v tabulce stránek



Ochrana paměti I. (segmenty)

- ochrana paměti na úrovni segmentů je zapnutá a nejde vypnout (ale jde nastavit, aby nebyla účinná)
- prováděné kontroly
  - kontrola typu segmentu (některé segmenty nebo segmentové registry můžou být použité jenom určitým způsobem)
  - kontrola velikosti segmentu (limitu), i.e., jestli program nepřistupuje za hranice segmentu
  - kontrola oprávnění
  - omezení adresovatelné domény (omezení přístupu jen k žádoucím segmentům)
  - omezení vstupních bodů procedur (brány)
  - omezení instrukční sady
- AMD64 v long mode neprovádí některé kontroly (báze a limit jsou ignorovány)

Ochrana paměti II. (stránkování)

- stránkování funguje souběžně se segmentací
- bit pro systémové stránky (zákaz přístupu z ring 3) $\Longrightarrow$ volání OS (přes bránu, nebo spec. instrukce)
- bit pro zákaz zápisu
- AMD64 + PAE mají NX bit (zákaz spouštění) $\Longrightarrow$ ochrana před útoky
- možnost nastavit bity na jednotlivých stránkách i adresářích $\Longrightarrow$ efektivnější