

Segmentovanie a stránkovanie v procesoroch Intel Pentium

- Adresovanie v reálnom režime
- Adresovanie v 32-bitovom chránenom režime
 - segmentovanie
 - stránkovanie
- Adresovanie v 64-bitovom chránenom režime

Autor: Peter Tomcsányi

Niektoré práva vyhradené v zmysle licencie Creative Commons

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Použité obrázky z učebnice:

Andrew. S. Tanenbaum, Structured Computer Organization

<http://www.cs.vu.nl/~ast/books/>

Adresovanie v reálnom režime

- Procesor 8086 mal 20 bitovú adresu ale len 16-bitové registre.
- Riešením tohoto rozporu bolo zavedenie pojmu segment aj bez zavedenia virtuálnej pamäte
- Segment je časť pamäte maximálnej veľkosti 64KB (teda v rámci segmentu môžeme adresovať 16-bitovou adresou).
- Program vždy pracuje len s niekoľkými segmentami a ich počiatočné adresy má uložené v štyroch špeciálnych segmentových registroch, ktorých význam je takýto:
 - CS (code segment) - inštrukčný segment
 - DS (Data segment) - dátový segment
 - SS (Stack segment) - zásobníkový segment
 - ES (Extra segment) - voľný segmentový register pre prístup k dátam v ďalšom segmente.
- Aby mohli byť aj segmentové registre len 16-bitové, dovoľíme segmentom začínať len na adresách deliteľných 16, teda adresy začiatkov segmentov končia vždy 4 nulami a tie si nemusíme pamätať v segmentových registroch.
- Používa sa dvojrozmerné adresovanie segment:offset (teda adresa segmentu a adresa v rámci segmentu).

Adresovanie v reálnom režime (2)

IP ukazuje na nasledujúcu inštrukciu.

Fyzická adresa sa vypočíta ako

CS:IP teda ako $16 * CS + IP =$

$40230 + 103A = 4126A$

IP: **103A**

MOV AX,[06C2]

Pri adresovaní dát sa implicitne použije register DS. Teda táto inštrukcia preniesie do AX dva bajty z adresy

$636C0 + 06C2 = 63D82$

MOV AX,ES:[06C2]

Segmentový register môžeme určiť aj explicitne. Táto inštrukcia preniesie do AX dva bajty z adresy

$72F30 + 06C2 = 735F2$

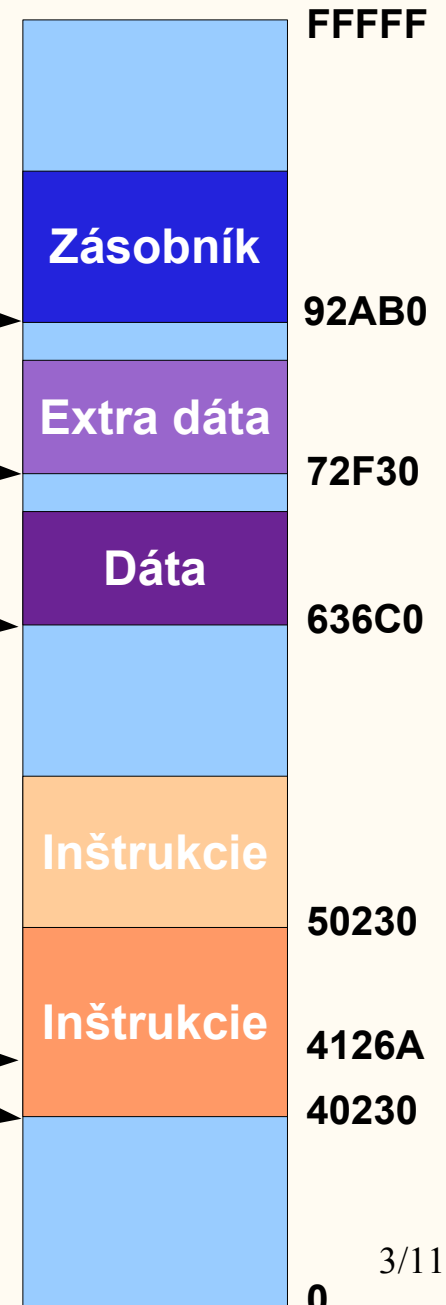
JMP 23FA

Blízky skok obsahuje len offset adresy. Pri takomto skoku sa zmení len IP, CS ostane bez zmeny. Teda skáče v rámci segmentu

JMP 5023:1756

Ďaleký skok obsahuje segment aj offset adresy. Pri takomto skoku sa zmení IP aj CS. Skáče do iného segmentu

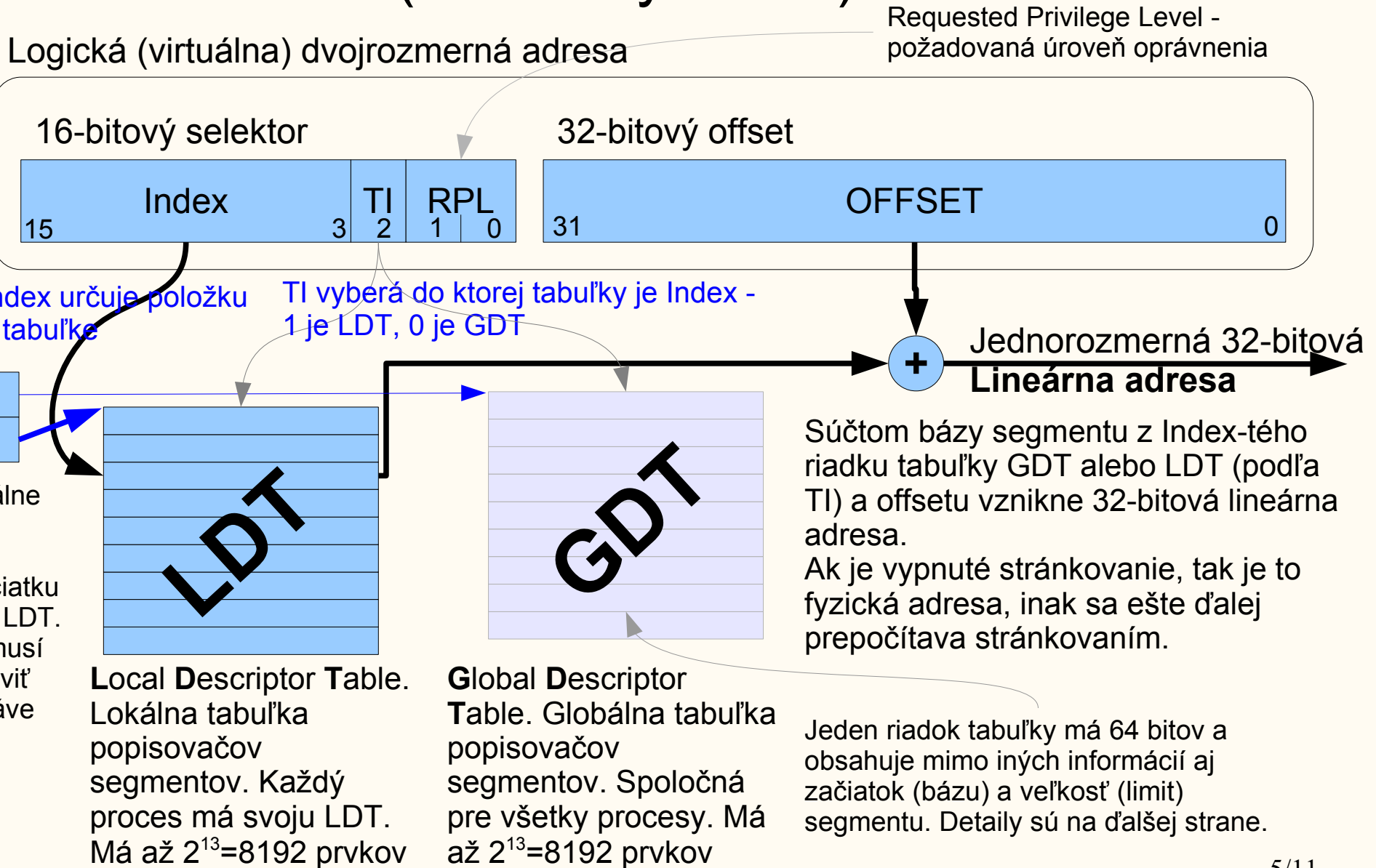
SS:	92AB
ES:	72F3
DS:	636C
CS:	4023



Adresovanie v chránenom režime (32-bitový režim)

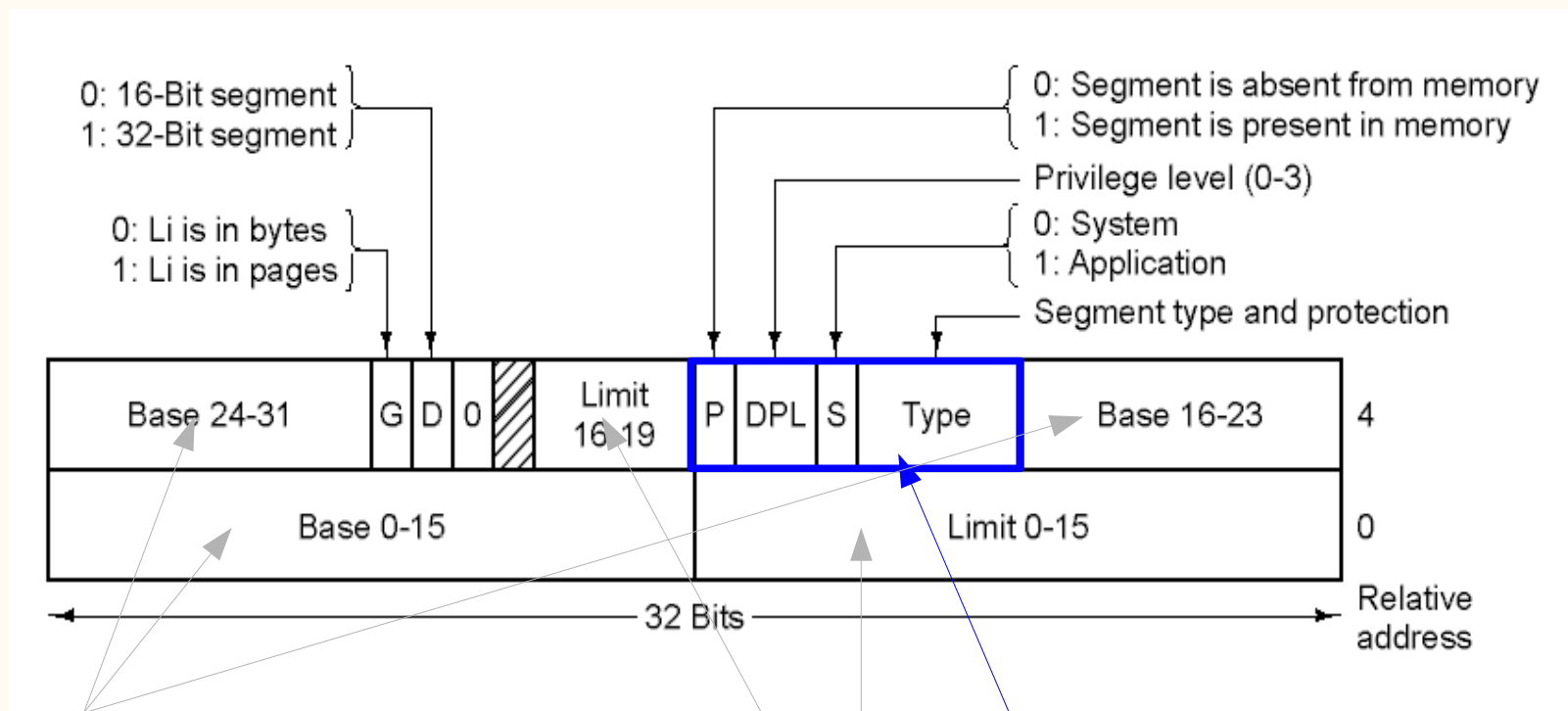
- Chránený režim podporuje virtuálnu pamäť
- Podporuje segmentovanie a stránkovanie zároveň
- Stránkovanie sa dá vypnúť, segmentovanie ale nie
- Pri segmentovaní sa používajú rovnaké segmentové registre ako v reálnom režime
- Segmentový register už neobsahuje fyzickú adresu začiatku segmentu, ale je to selektor, ktorý určuje index (poradové číslo) segmentu vo virtuálnom adresnom priestore procesu, fyzická adresa sa musí nájsť v tabuľke
- Pribudli dva segmentové registre FS a GS (od procesora 386)
- Offset (adresa v rámci segmentu) je 32-bitový

Segmentovanie v chránenom režime (32-bitový režim)



Popisovač segmentu

Jeden riadok segmentovej tabuľky LDT resp. GDT



Base (32 bitov) je počiatočná adresa segmentu v pamäti - je

Prístupové práva - detaily na ďalšej strane

Limit (20 bitov) Určuje veľkosť segmentu. Ak je bit G=0, tak je limit v bajtoch (0-1MB), ak je G=1, tak je limit v 4KB stránkach (0-4GB). Segment teda môže mať až 4GB, ale nemôže mať úplne ľubovoľnú veľkosť

Prístupové práva

bajt určujúci typ segmentu a pôsob prístupu

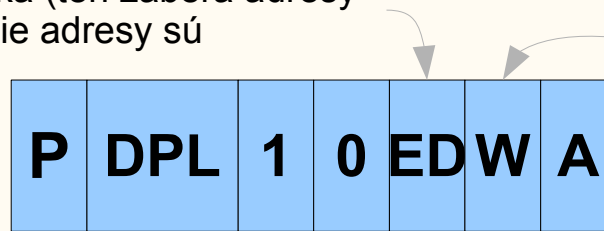
Extend Down - 0 je pre bežný dátový segment, 1 je pre segment zásobníka (ten zaberá adresy nad hodnotou Limit a nižšie adresy sú nepoužité)

Dátový segment:
obsahuje dáta alebo zásobník

Present - či je v pamäti alebo na disku

Inštrukčný segment:
obsahuje program

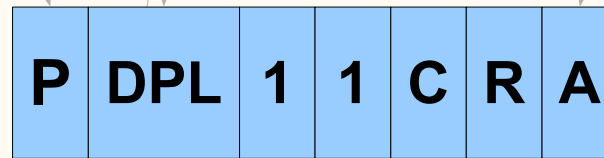
Systémový segment:
obsahuje systémové informácie ako napr. GDT a LDT alebo je to brána



Descriptor Privilege Level - úroveň oprávnenia popisovača

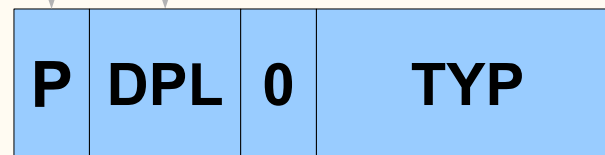
Writable - ak je 1 tak sa smie zapisovať, ak je 0, tak sa smie len čítať

Accessed - procesor ho nastaví na 1 keď použije tento popisovač pri adresovaní. Je to implementácia bitu **R** z teórie



Readable - ak je 1 tak sa smie čítať, ak je 0, tak sa smie len vykonať

Conforming



TYP má 4 bity, rozlišuje 16 typov systémových segmentov

Urýchľovanie výpočtu adresy pri segmentovaní

- Tabuľky GDT a LDT sú v pamäti, teda pre prepočet každej adresy treba jeden prístup do pamäti navyše.
- Pri adresovaní je selektorom takmer vždy obsah niektorého zo **segmentových registrov** (výnimkou sú ďaleký JMP a CALL).
- Ak pri každej zmene obsahu segmentového registra prenesie procesor informáciu z príslušného riadku tabuľky do programátorovi neprístupnej časti segmentového registra, tak pri použití registra v adresovaní už netreba pristupovať do pamäti.

Dôsledky:

- Do segmentového registra nemožno uložiť ľubovoľnú hodnotu (v reálnom režime to bolo možné)
- Aby bolo možné zobrazit' NULL (adresu "nikam"), je povolené do segmentového registra uložit' nulu (označuje NULL), chyba vznikne až pri pokuse použiť nulový segmentový register v adresovaní. Teda 0-tý prvok segmentovej tabuľky GDT sa nepoužíva.
- Po zmene v LDT alebo GDT treba znova naplnit' segmentové registre.

segmentovanie

Stránkovanie

32-bitová lineárna adresa

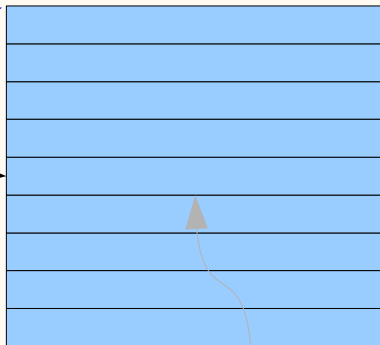
Číslo stránky (20 bitov, 10+10)

Offset (12-bitov)

Vyšších 10 bitov indexuje do stránkového adresára

Nižších 10 bitov indexuje do stránkovej tabuľky, ktorá sa našla v stránkovom adresári

Stránkový adresár

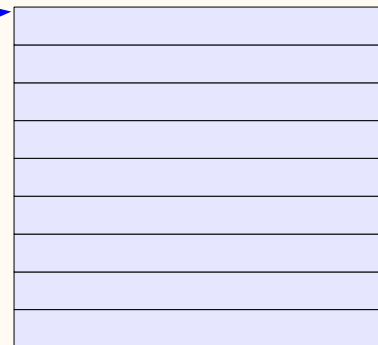


CR3

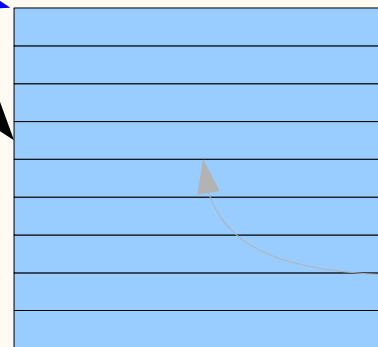
Register CR3 obsahuje adresu rámca, v ktorom je stránkový adresár

Položka stránkového adresára obsahuje číslo rámca, v ktorom je stránková tabuľka pre daný interval stránok

Stránková tabuľka



Stránková tabuľka



32-bitová fyzická adresa

20-bitové číslo rámca

12-bitový offset

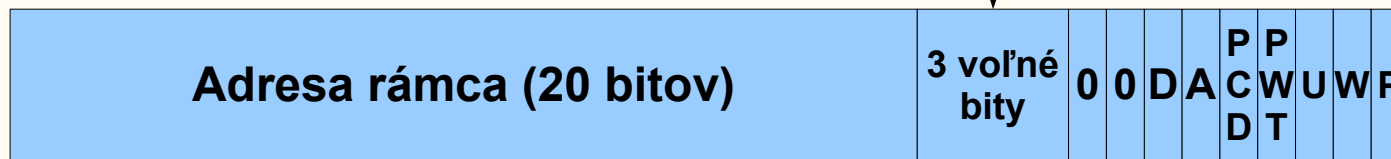
Zložením čísla rámca nájdeného v stránkovej tabuľke a offsetu z pôvodnej adresy vznikne 32-bitová adresa do fyzickej pamäte

Položka stránkovej tabuľky obsahuje číslo rámca, v ktorom je daná stránka

Položka stránkového adresára a stránkovej tabuľky

Adresa rámca v položke stránkového adresára ukazuje na rámec so stránkovou tabuľkou, adresa rámca v položke stránkovej tabuľky ukazuje na rámec, kde sa nachádza daná stránka

3 voľné bity môže použiť operačný systém na uloženie ľubovoľnej informácie



Dirty - procesor nastaví na 1 keď sa do stránky zapíše. Je to implementácia bitu **M** z teórie

Accessed - procesor ho nastaví na 1 keď použije táto položka pri výpočte adresy. Je to implementácia bitu **R** z teórie

Page Cache Disable - ak je 1 tak sa pre prístup do tejto stránky nepoužíva cache

Page Write Trough - ak je 1 tak sa pri zápise do tejto stránke vždy zapíše do cache aj do pamäti

Present - či je stránka v pamäti alebo na disku

Writable - ak je 1 tak sa smie zapisovať, ak je 0, tak sa smie len čítať

User Accessible - ak je 1 tak túto stránku môže používať aj proces s najnižšou úrovňou oprávnenia (bežnú užívateľský proces), ak je 0 tak do nej majú prístup len systémové procesy

Chránený režim v 64-bitových procesoroch Intel

- Segmentovanie je v podstate vypnuté, je k dispozícii len lineárny virtuálny adresný priestor.
- Súčasné procesory používajú len najviac 48 bitov virtuálnej adresy, (teda virtuálny adresný priestor má najviac 256 terabajtov) a 52 bitov fyzickej adresy (4 petabajty).
- Má niekoľko režimov stránkovania
 - **32-bitové:** 32-bitová virtuálna adresa so 4KB stránkami (32-bitová fyzická adresa) alebo so 4MB stránkami (40-bitová fyzická adresa, 1 terabajt)
 - **PAE (Physical address extension):** 32 bitová virtuálna adresa, 52-bitová fyzická adresa, stránky 4KB a 3 úrovne tabuliek (2,9 a 9 bitov) alebo stránky 2MB a dve úrovne tabuliek (2 a 9 bitov).
 - **IA-32e:** 48-bitová virtuálna adresa, 52-bitová fyzická adresa, stránky 4KB (4 úrovne tabuliek po 9 bitov), 2MB (3 úrovne tabuliek) alebo 1GB (2 úrovne tabuliek).
- Detaily a obrázky nájdete napríklad tu:
<https://www.cs.rutgers.edu/~pxk/416/notes/10-paging.html>