



Počítačové sítě a operační systémy

I/O systém Vnější paměti

Jaromír Plhák
xplhak@fi.muni.cz

Hardware (1)

- HW pro I/O je značně rozmanitý
- Existují však určité běžně používané prvky
 - Port
 - Sběrnice (bus)
 - Řadič (host adapter, controller)
- I/O zařízení jsou řízena I/O instrukcemi (IN, OUT)

Hardware (2)

- Adresy I/O zařízení
 - Uváděné přímo v I/O instrukcích (např. IN AL, DX : DX port, AL získaný bajt)
- I/O se mapuje na přístup k paměti (např. grafická karta, videopaměť)
- Základní způsoby ovládání I/O
 - Polling, programované I/O operace
 - Aktivní čekání na konec operace
 - Přerušení
 - DMA

Rozmístění I/O portů v PC

I/O address range (hexadecimal)	device
000–00F	DMA controller
020–021	interrupt controller
040–043	timer
200–20F	game controller
2F8–2FF	serial port (secondary)
320–32F	hard-disk controller
378–37F	parallel port
3D0–3DF	graphics controller
3F0–3F7	diskette-drive controller
3F8–3FF	serial port (primary)

Techniky provádění I/O

- Programovaný I/O (busy-waiting)
 - Opakovaně se ptám na stav zařízení
 - Připraven / Pracuje / Chyba
- I/O řízený přerušením
 - Zahájení I/O pomocí I/O příkazu
 - Paralelní běh I/O s během procesoru
 - I/O modul oznamuje přerušením konec přenosu
- Direct Memory Access (DMA)
 - Kopírování bloků mezi pamětí a I/O zařízením na principu kradení cyklů paměti
 - Přerušlení po přenosu bloku (indikace konce)

Přerušení

- Přerušení obsluhuje ovladač přerušení (kód OS)
- Maskováním lze některá přerušení ignorovat nebo oddálit jejich obsluhu
- Patřičný ovladač přerušení se vybírá přerušovacím vektorem
 - Některá přerušení nelze maskovat
 - Přerušení mohou být uspořádána podle priorit
- Přerušení se používá i pro řešení výjimek (nejsou asynchronní)

DMA

- Přímý přístup do paměti (Direct Memory Access - DMA)
 - Nahrazuje programovaný I/O při velkých přesunech dat
 - Vyžaduje speciální DMA řadič
 - Při přenosu dat se obchází procesor, přístup do paměti zajišťuje přímo DMA řadič
 - Procesor a DMA soutěží o přístup k paměti

Aplikační rozhraní I/O

- Jádro OS se snaží skrýt rozdíly mezi I/O zařízeními a programátorům poskytuje jednotné rozhraní
- Dále vrstva ovladačů ukrývá rozdílnost chování I/O řadičů i před některými částmi jádra
- Některé vlastnosti I/O zařízení
 - Mód přenosu dat: znakové (terminál) / blokové (disk)
 - Způsob přístupu: sekvenční (modem) / přímý (disk)
 - Sdílené/dedikované: klávesnice / páska
 - Rychlost přenosu: vystavení, přenos, ...
 - Read-write, read only, write only

Bloková a znaková zařízení

- Bloková zařízení – typicky disk
 - Příkazy: read, write, seek
 - Logický způsob přístupu: obecný I/O nebo souborový systém
 - Možný přístup formou souboru mapovaného do paměti
- Znaková – klávesnice, myš, sériový port
 - Příkazy: get, put
 - Nad nimi knihovní podprogramy pro další možnosti (např. řádková editace)

Síťová zařízení

- Přístup k nim se značně liší jak od znakových, tak od blokových zařízení
 - Proto mívají samostatné rozhraní OS
- Unix i Windows obsahující rozhraní nazývané „sockets“
 - Separují síťové protokoly od síťových operací
 - Přístup jako k souborům (včetně funkce select)
- Existuje celá řada přístupů k síťovým službám
 - Pipes (roury), FIFOs, streams, queues, mailboxes

Blokující a neblokující I/O (1)

- Blokující
 - Z hlediska procesu synchronní
 - Proces čeká na ukončení I/O
 - Snadné použití (programování), snadné porozumění (po provedení operace je hotovo to co jsem požadoval)
 - Někdy však není dostačující (z důvodu efektivity)
- Neblokující
 - Řízení se procesu vrací co nejdříve po zadání požadavku
 - Vhodné pro uživatelské rozhraní, bufferovaný I/O
 - Bývá implementováno pomocí vláken
 - Okamžitě vrací počet načtených či zapsaných znaků



Blokující a neblokující I/O (2)

- Asynchronní
 - Proces běží souběžně s I/O
 - Konec I/O je procesu hlášen signály
 - Obtížné na programování, složité používání, ale v případě vhodně promyšleného programu velice efektivní

I/O subsystém v jádru (1)

- Plánování
 - Některé I/O operace požadují řazení do front na zařízení
 - Některé OS se snaží o „spravedlnost“
- Vyrovnání (vyrovnávací paměti), buffering
 - Ukládání dat v paměti v době přenosu k/ze zařízení
 - Řeší rozdílnoř rychlosti
 - Řeší rozdílnoř velikosti datových jednotek

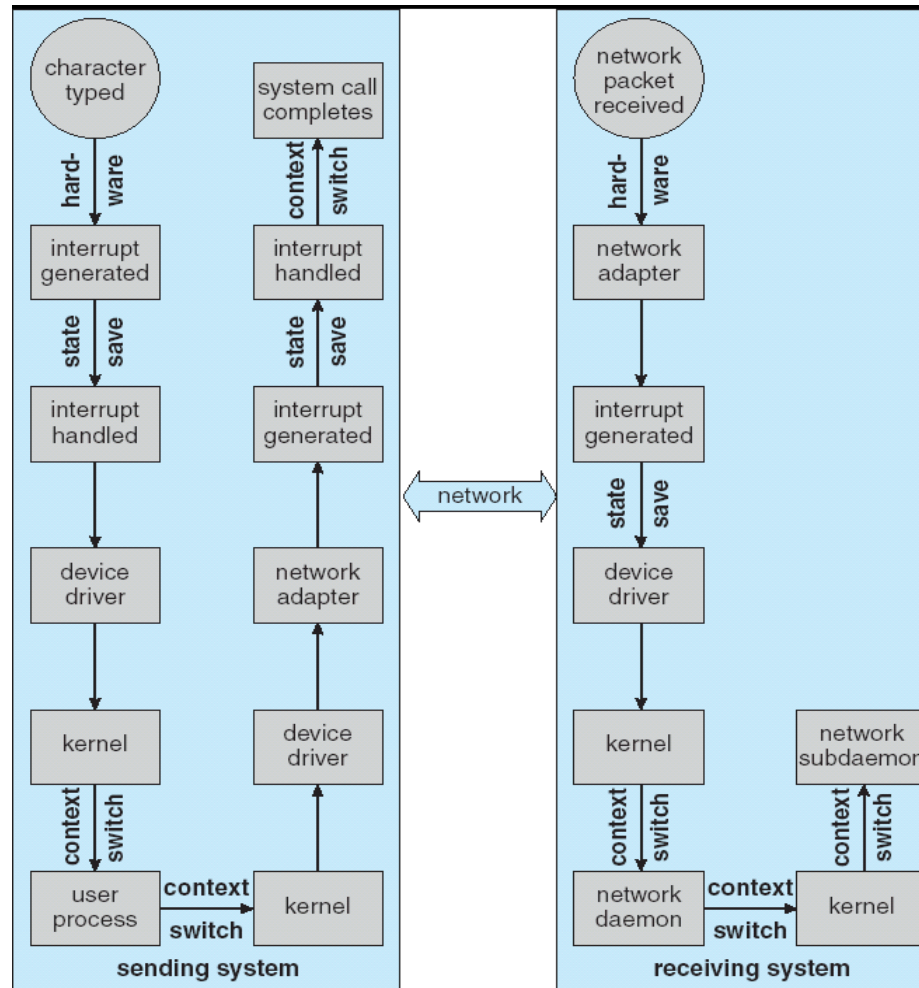
I/O subsystém v jádru (2)

- Caching
 - Rychlá paměť udržuje kopii dat
 - Vždy pouze kopii
 - Caching je klíčem k dosažení vysokého výkonu
- Spooling
 - Udržování fronty dat určených k výpis na zařízení
 - Pokud zařízení může vyřizovat požadavky pouze sekvenčně
 - Typicky tiskárna
- Rezervace zařízení
 - Exkluzivita přístupu k zařízení pro proces
 - Rezervace / uvolnění – volání systému
 - Pozor na uváznutí (deadlock)

Výkon

- I/O je nejvýznamnějším faktorem výkonu celého systému
 - CPU musí provádět ovladače a programy I/O části jádra
 - Při přerušení se přepíná kontext
 - Provádí se kopírování dat
 - Zvláště významný je síťový provoz

Příklad: Síťová aplikace

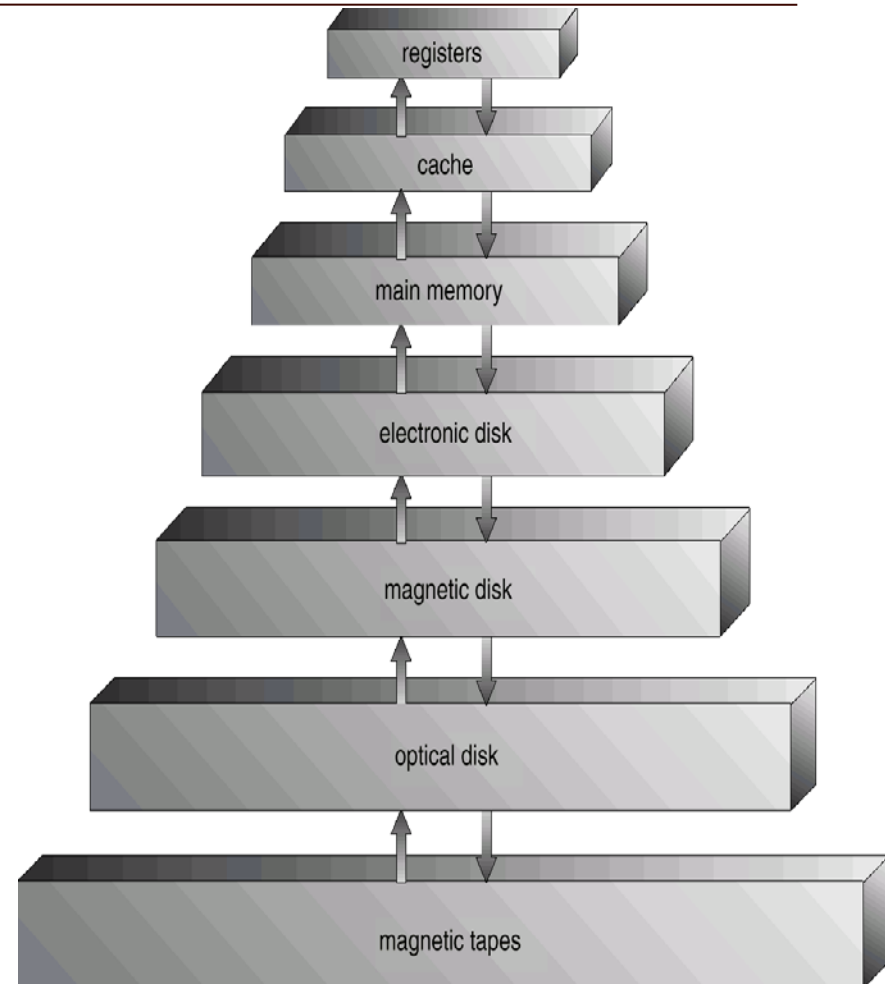


Zvyšování výkonu

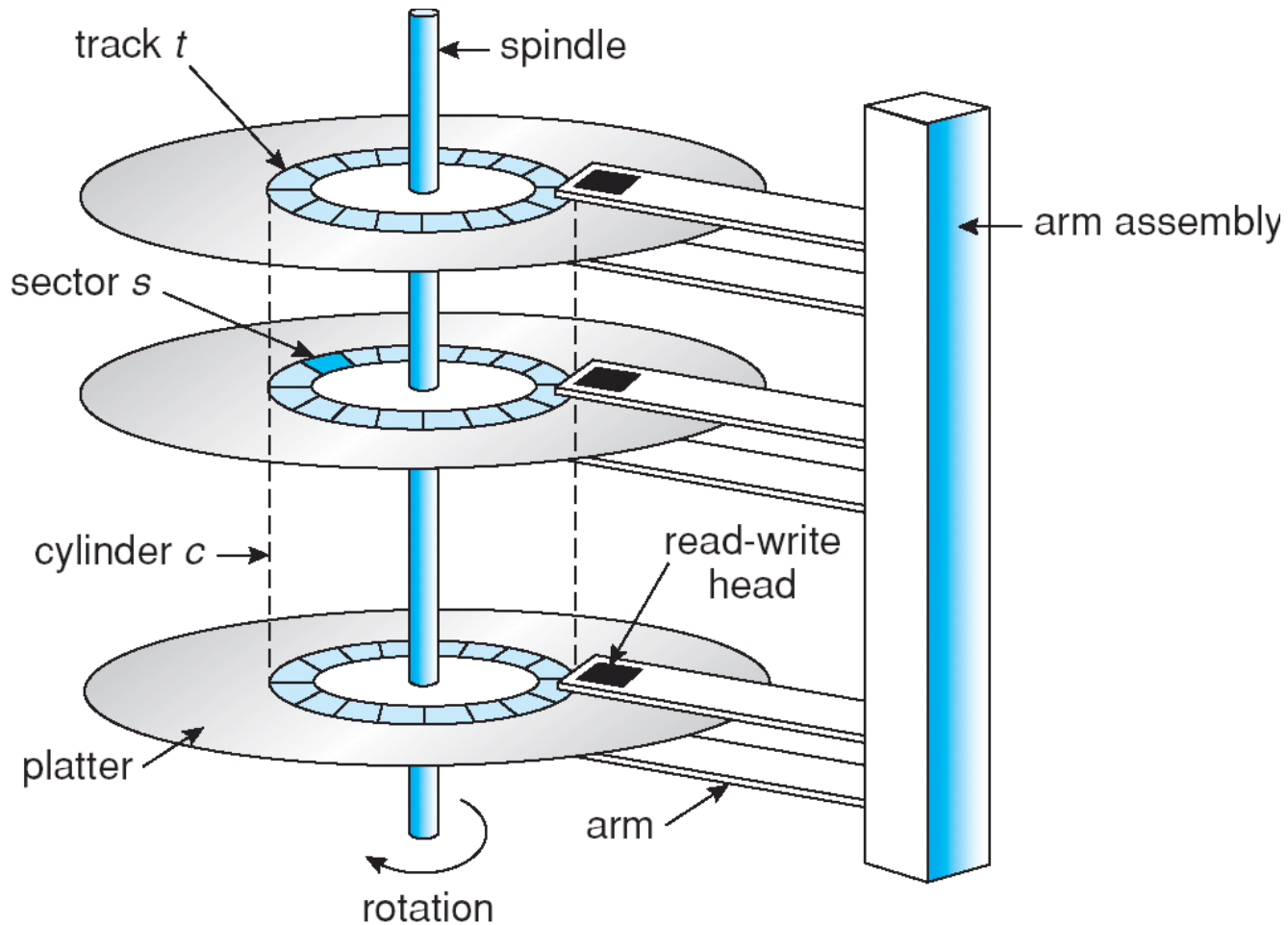
- Omezujeme počet přepnutí kontextu
- Omezujeme zbytečné kopírování dat
- Omezujeme počet přerušení tím, že přenášíme delší bloky
- Využíváme všech výhod (funkcí) moderních řadičů
- Používáme co nejvíce DMA
- Všechny komponenty kombinujeme s cílem dosažení co nejvyšší propustnosti
 - CPU, paměť, sběrnice, I/O zařízení

Paměťová hierarchie

- Primární paměti
 - Nejrychlejší
 - Energeticky závislé
 - Cache, hlavní (operační) paměť
- Sekundární paměti
 - Středně rychlé
 - Energeticky nezávislé
 - Také nazývané „on-line storage“
 - Flash disky, magnetické disky
- Terciální paměti
 - Levná typicky vyměnitelná média
 - Pomalé
 - Energeticky nezávislé
 - Také nazývané „off-line storage“
 - Floppy disky, magnetické pásky, optické disky



Magnetické disky



Struktura disku

- Diskové mechanismy se adresují jako velká 1-dimensionální pole logických bloků
 - Logické bloky jsou nejmenší jednotkou přenosu dat
- 1-dimensionální pole logických bloků je zobrazováno do sektorů disku sekvenčně
 - Sektor 0
 - První sektor na první stopě vnějšího válce
 - Zobrazování pokračuje po této stopě, potom po ostatních stopách tohoto válce, a potom po válcích směrem ke středu

Plánování disku (1)

- OS je odpovědný za efektivní používání hardware
 - Pro disky: co nejrychlejší přístup a co největší šířka pásma
- Doba přístupu (access time) je dána:
 - Dobou vystavení (seek time) – na válec se stopou s adresovaným sektorem
 - Dobou rotačního zpoždění – dodatečná doba do průchodu adresovaného sektoru pod čtecí/zápisovou hlavou
- Minimalizace doby vystavení
 - Doba vystavení \approx vystavovací vzdálenosti
 - Řeší plánování činnosti disku

Plánování disku (2)

- Rotační zpoždění
 - Shora omezeno konstantou
- Šířka pásma
 - Počet přenesených bytů / doba od zadání skupiny požadavků do jejich ukončení
 - Převzatý pojem z telekomunikací

Plánování disku (3)

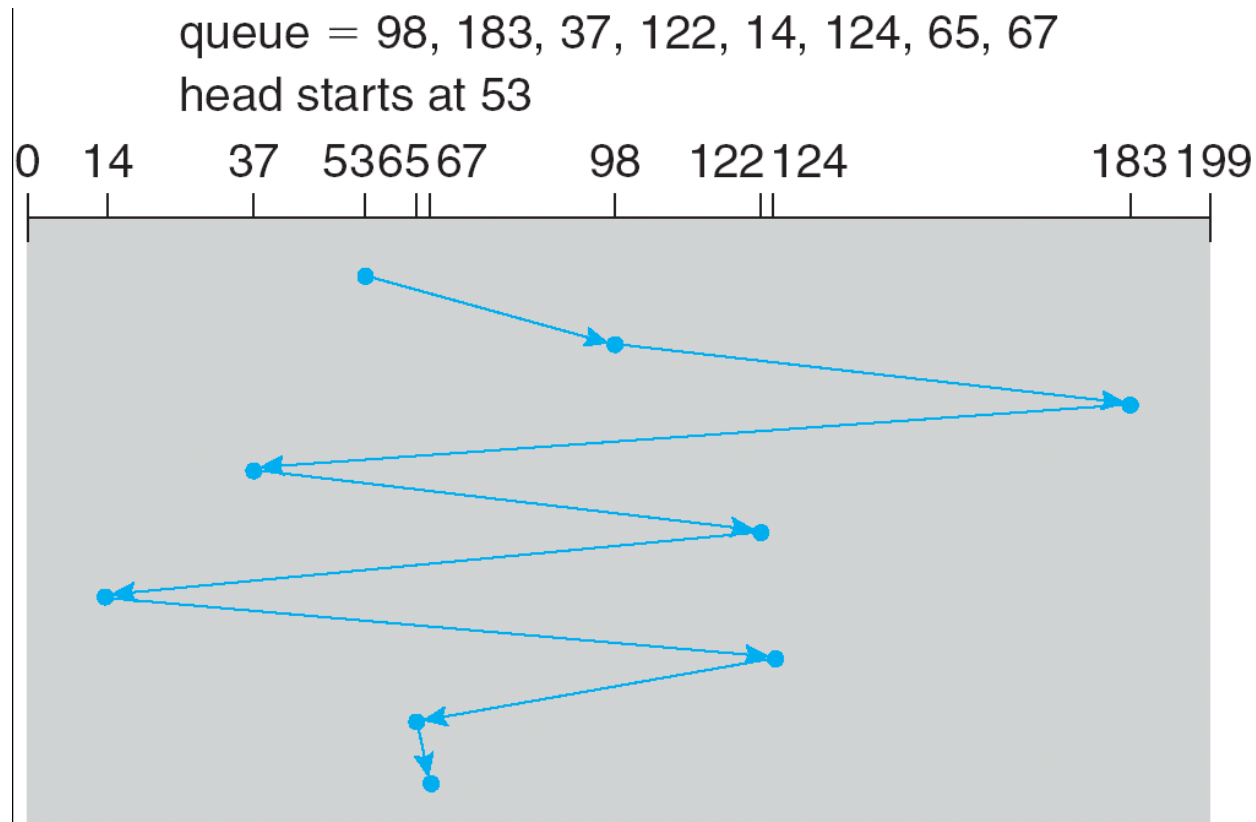
- Existuje celá řada algoritmů pro plánování přístupu na disk
- Příklad: vzorová fronta požadavků na přístup k disku (máme válce 0-199)

98, 183, 37, 122, 14, 124, 65, 67

- Hlavička disku vystavena na pozici 53

FCFS

- Celkem přesun o 640 válců

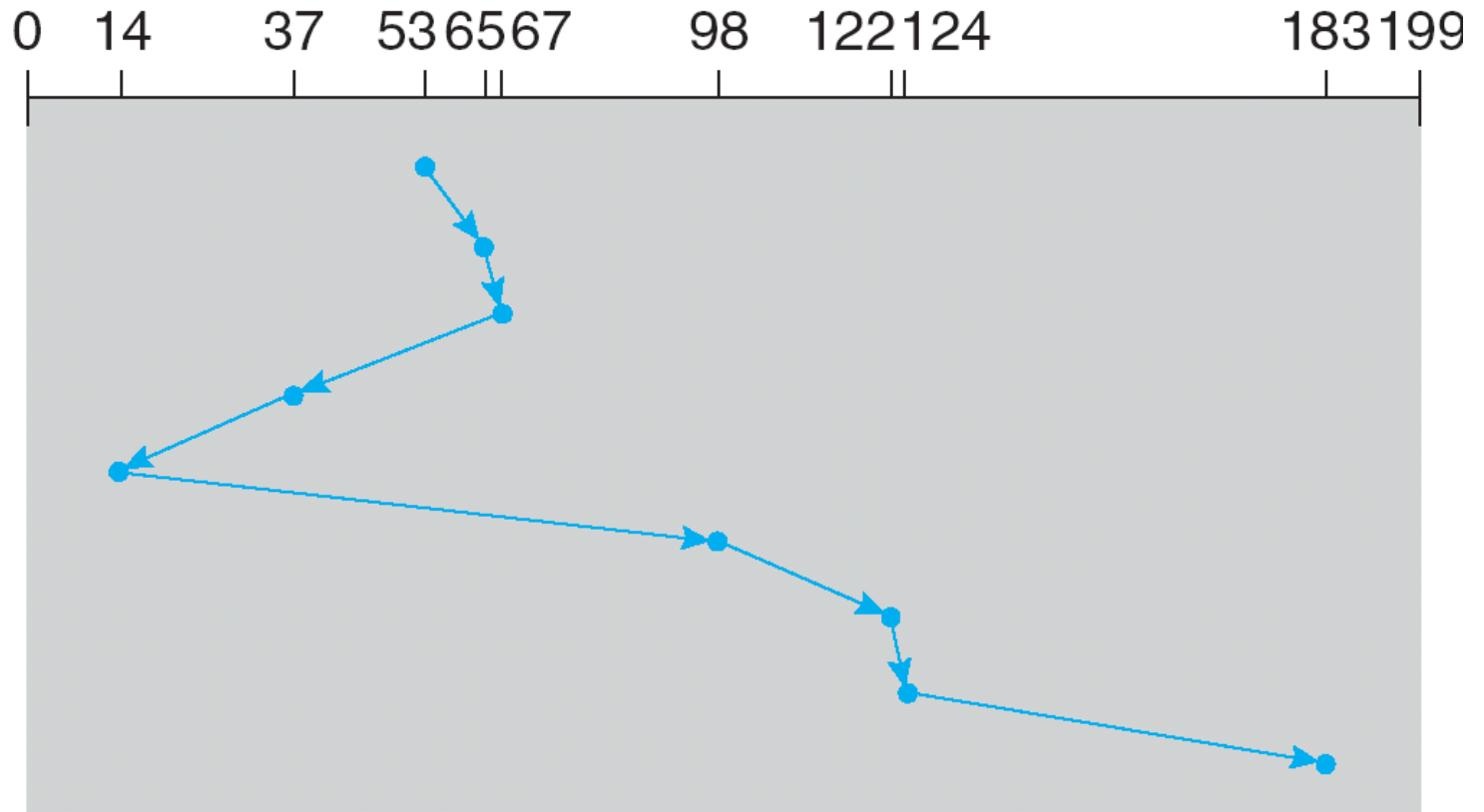


SSTF (1)

- Z fronty požadavků vybírá ten požadavek, který vyžaduje minimální dobu vystavení od současné pozice hlavičky
- SSTF (shortest seek time first) algoritmus je variantou algoritmu SJF (shortest job first); může způsobit stárnutí požadavků.
- Náš příklad vyžaduje přesun o 236 válců

SSTF (2)

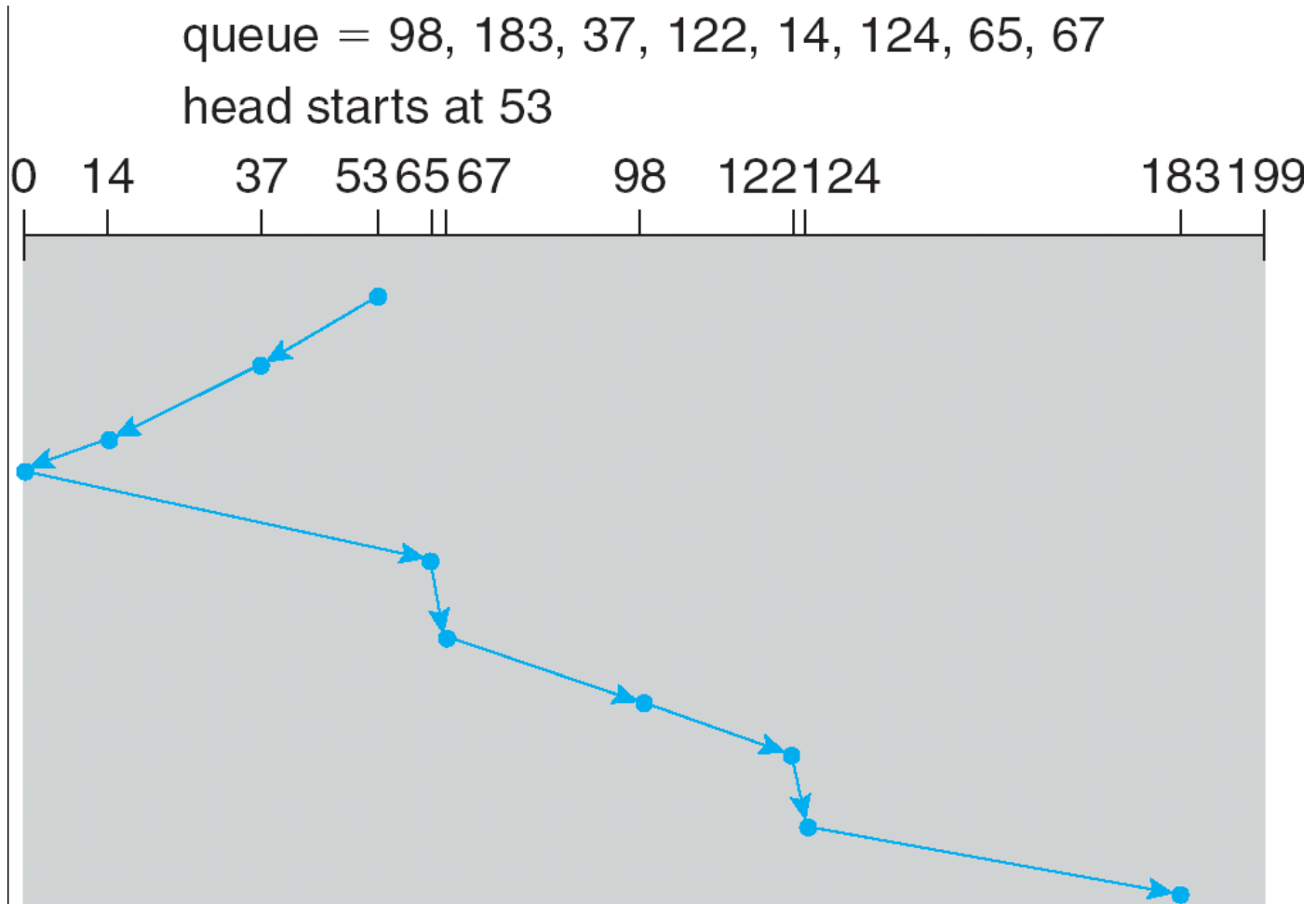
queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



SCAN (1)

- Hlavička disku začíná na jedné straně disku a přesune se při splňování požadavků ke druhé straně disku. Pak se vrací zpět a opět plní požadavky
- Někdy nazývané algoritmus typu výtah
- Náš příklad vyžaduje přesun o 208 válců

SCAN (2)



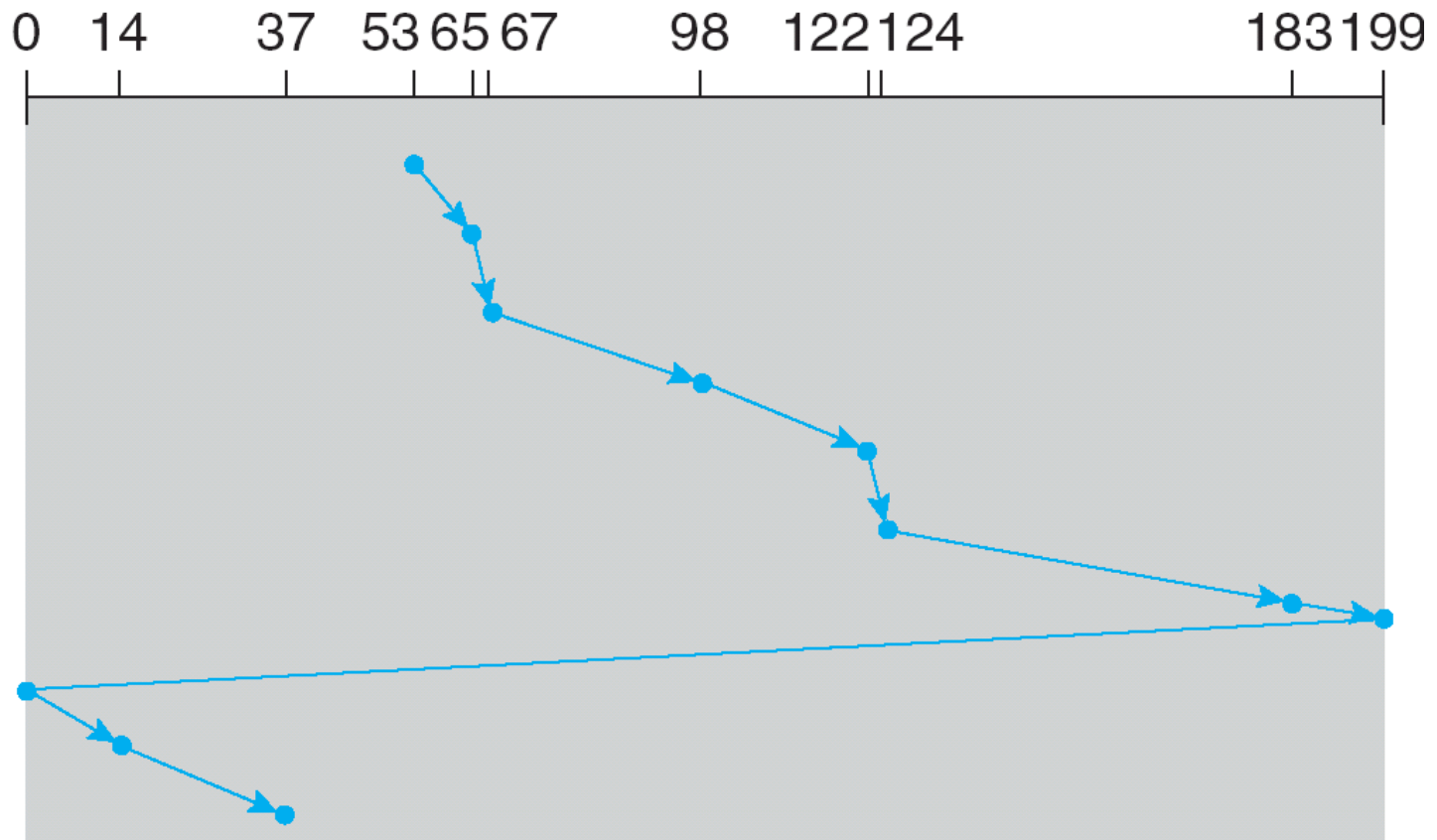
C-SCAN (1)

- Poskytuje jednotnější čekací dobu než SCAN
- Hlavička se posouvá z jednoho konce disku na druhý a zpracovává požadavky. Potom se vrací zpět bez vyřizování požadavků a opět začíná vyřizovat požadavky z prvního konce
- Válce považuje za kruhový seznam, který za posledním válcem pokračuje opět prvním válcem

C-SCAN (2)

queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53

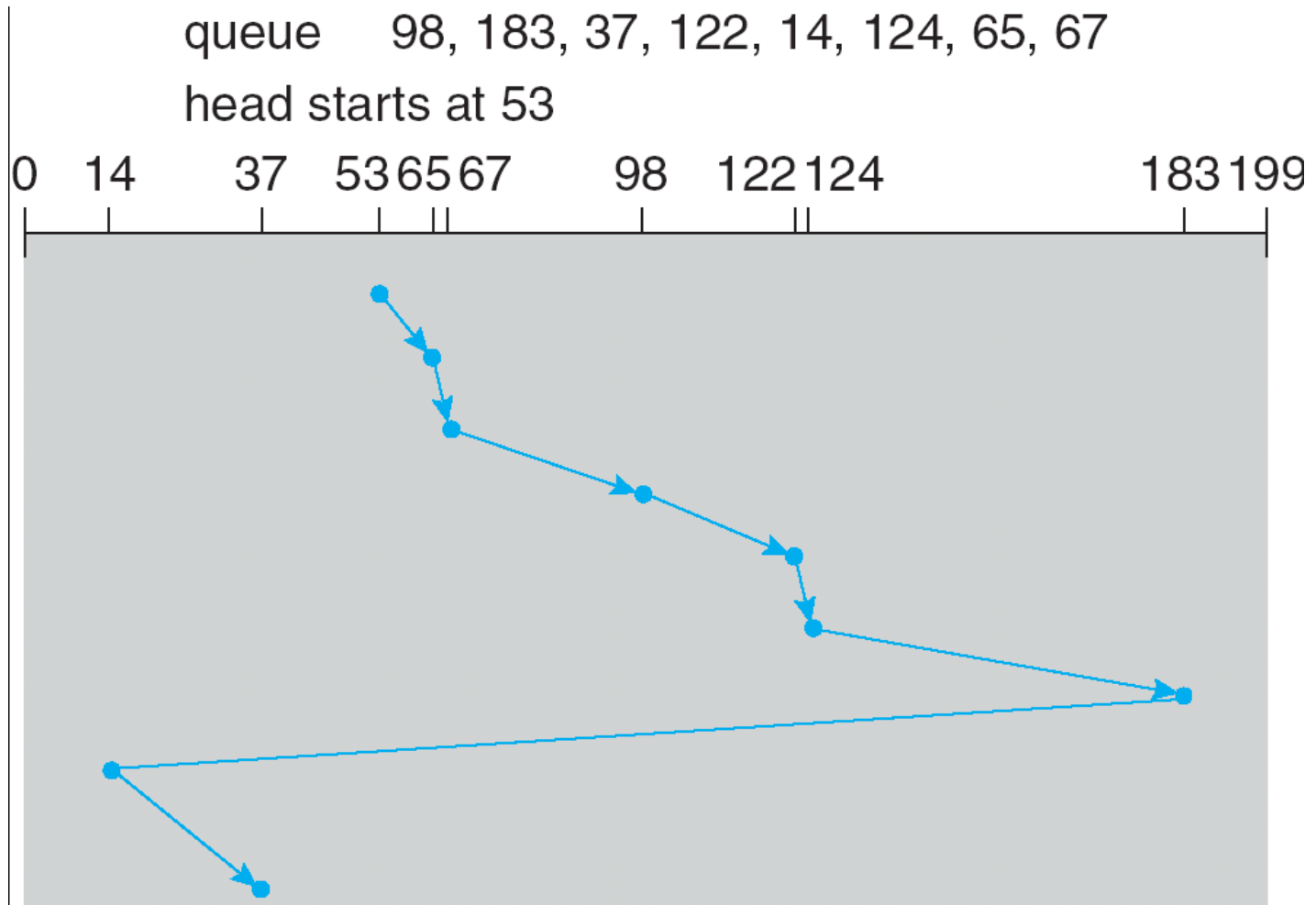




C-LOOK (1)

- Obdoba C-SCAN, ale hlavička jen potud do kraje, pokud existují požadavky.
- Pak se vrací zpět

C-LOOK (2)



Výběr algoritmu

- SSTF je přirozený, má přirozené chápání
- SCAN a C-SCAN jsou vhodnější pro těžkou zátěž disku
- Výkon závisí na počtu a typech požadavků
- Požadavky na disk mohou být ovlivněny metodami organizace souborů v souborovém systému
- Plánovací algoritmus by měl být napsán jako samostatný modul, aby plánovací algoritmus bylo možné zaměňovat
- Častá implicitní volba bývá SSTF nebo LOOK

Moderní HW

- U moderních disků nemusí být známé mapování logických bloků na fyzické adresy
- Disku předáme skupinu požadavků a disk si pořadí optimalizuje sám
- OS přesto může mít zájem na vlastním řazení požadavků
 - Priorita I/O operací z důvodu výpadků stránek
 - Pořadí operací zápisu dat a metadat souborového systému

Technologie RAID (1)

- RAID: Redundant Arrays of Independent (Inexpensive) Disks
 - Organizace disků řízená tak, že poskytuje dojem jednoho (logického) disku
 - S velkou kapacitou a rychlostí díky tomu, že mnoho disků pracuje paralelně
 - S velkou spolehlivostí, data se uchovávají redundantně, lze je obnovit i po poruše některého z disků

Technologie RAID (2)

- Pravděpodobnost, že některý disk z množiny N disků selže je mnohem vyšší, než pravděpodobnost, že selže jediný disk
 - $N = 100$ disků, každý má $MTTF = 100\,000$ hodin (cca 11 let), celý systém bude mít $MTTF = 1000$ hodin (cca 41 dní)
 - Techniky na bázi redundance chránící před ztrátou dat jsou pro systémy s velkým počtem komponent (disků) kritické
- Původní záměr
 - Levná alternativa nahrazující velké drahé disky
 - „I“ je interpretováno jako „independent“

RAID - zvýšení spolehlivosti

- Redundance
 - Nadbytečnost, doplňková informace použitelná pro obnovu informace po poruše (disku)
- Zrcadlení (stínování), Mirroring (shadowing)
 - Každý disk je duplikován, 1 logický disk je tvořen 2 fyzickými disky
 - Každý zápis se provede na obou discích, čte se z jednoho disku
 - Jestliže se jeden disk porouchá, data jsou k dispozici na druhém disku
 - Ke ztrátě dat dojde při výpadku obou disků, když zrcadlový disk selže dříve, než se systém opraví
 - Průměrná doba do ztráty dat závisí na průměrné době do poruchy a průměrné době opravy
 - Např. MTTF = 100 000 hodin, průměrná doba opravy 10 hodin, dává u zrcadlené dvojice disků průměrnou dobu ztráty dat $100\,000^2 / (2 * 10) = 500 * 10^6$ hodin (čili 57 000 let), když budeme ignorovat požáry apod.

RAID - zvýšení výkonu

- Dva hlavní cíle paralelismu v diskových systémech
 - Zvýšení propustnosti vyvážením zátěže malými přístupy
 - Paralelizace velkých přístupů s cílem zkrácení doby odpovědi
- Zvýšení přenosové rychlosti paralelním zápisem do více disků (dělení, striping)
 - Bit-level striping
 - Blok-level striping

RAID - Bit-level striping

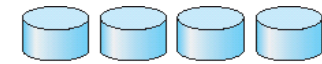
- Dělení bitů každého bytu mezi samostatné disky
- V poli 8 disků se zapisuje bit i každého bytu na disk i
- Čtení dat probíhá 8x rychleji než z jednoho disku
- Vystavení je delší než v případě jednoho disku
- Dnes se bit-level striping de facto už nepoužívá

Block level striping

- Systém s n disky, blok souboru i se zapisuje na disk $(i \bmod n) + 1$
- Požadavky na různé bloky se mohou realizovat paralelně, jestliže bloky leží na různých discích
- Požadavek na dlouhou posloupnost bloků může použít všechny disky paralelně

Úrovně RAID, přehled

- RAID Level 0: Žádná redundance, jen souběžnost
- RAID Level 1: Spolehlivost dosažená zrcadlením disků
- RAID Level 2: Hamming code error correction
- RAID Level 3: 1 kontrolní disk na skupinu, dělení bitů
- RAID Level 4: Nezávislé operace read/write, dělení bloků
- RAID Level 5: Data/parity přes všechny disky (více souběžný přístup)
- RAID Level 6: Odolnost při více než jedné poruše disku



(a) RAID 0: non-redundant striping.



(b) RAID 1: mirrored disks.



(c) RAID 2: memory-style error-correcting codes.



(d) RAID 3: bit-interleaved parity.



(e) RAID 4: block-interleaved parity.

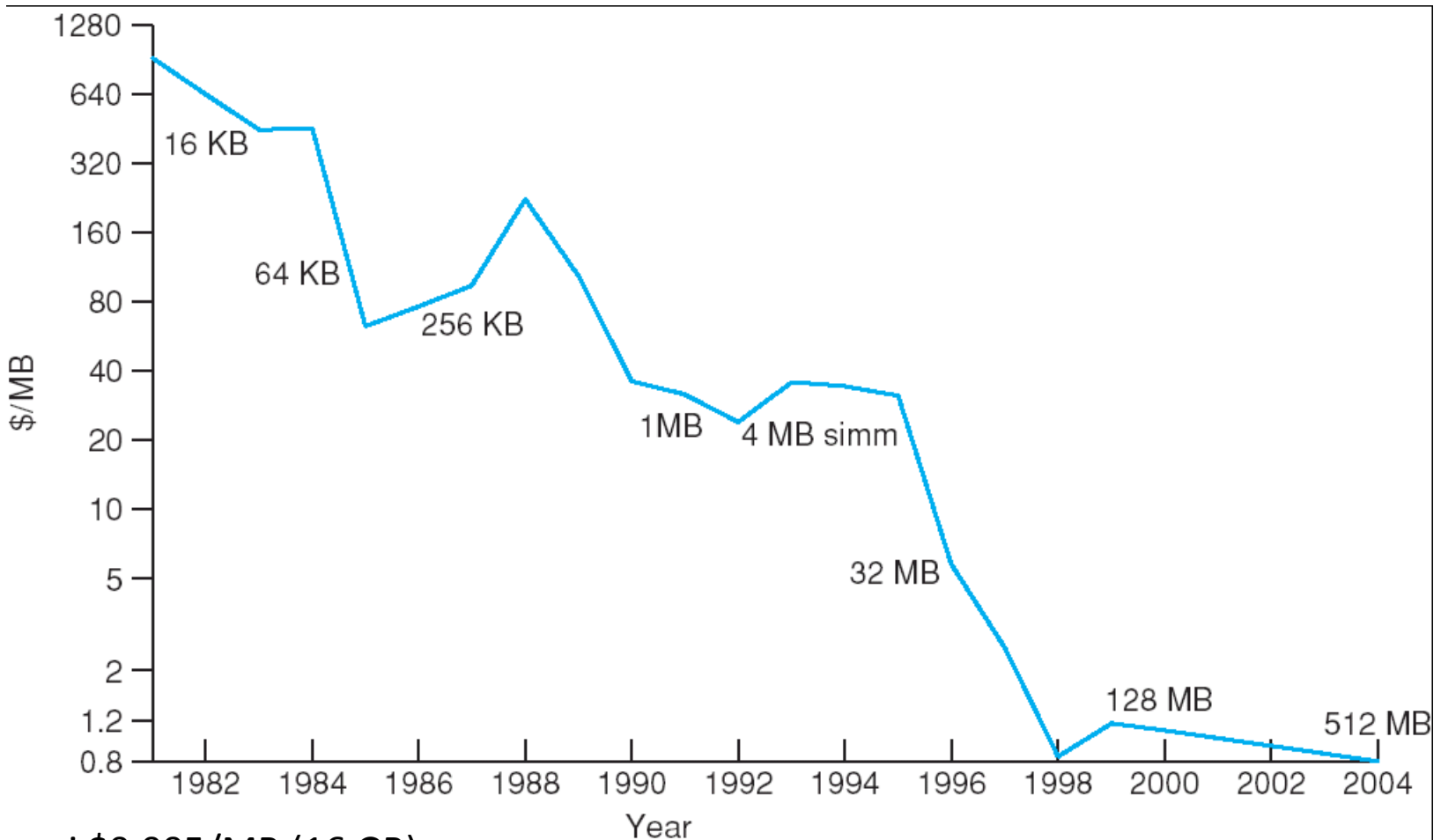


(f) RAID 5: block-interleaved distributed parity.



(g) RAID 6: P + Q redundancy.

Cena MB RAM (1981 – 2004)



Dnes: asi \$0.005/MB (16 GB)

Cena MB pevných disků (1956 – 2012)

