Operační systémy 2

I/O: zařízení

Petr Krajča

Katedra informatiky
Univerzita Palackého v Olomouci

25. říjen, 11. listopad 2010

I/O: zařízení

- zásadní složka Von Neumannova architektury
- různé pohledy na I/O zařízení: inženýrský (dráty, motory) vs. programátorský (rozhraní)
- různé rychlosti od 10 B/s (klávesnice) po 1 GB/s (PCI Express)
- různé druhy přístupu

Bloková zařízení

- data jsou přenášena v blocích stejné velikosti (typicky 512 B až 32 kB)
- možné nezávisle adresovat/zapisovat/číst data po jednotlivých blocích
- HDD, SSD, CD, DVD, páska?, ...

Znaková zařízení

- proud znaků/bytů (nelze se posouvat)
- klávesnice, myš, tiskárna, terminál

Ostatní

- nespadají ani do jedné z kategorií
- hodiny (přerušení), grafické rozhraní (mapovaná paměť)

Přístup k zařízením (1/2)

Port-Mapped I/O

- registry jednotlivých zařízení mají samostatný adresní prostor (oddělený od paměti)
- přístupné přes operace `in`, `out` – zápis/čtení hodnoty z portu
- nevýhody: omezené na speciální operace (jen zápis/čtení), omezené řízení přístupu

Memory-Mapped I/O

- registry jednotlivých zařízení jsou namapovány v paměťi
- data se čtou přímo na sběrnici; zapisuje je na sběrnici zařízení, ne do paměti
- výhoda: k zařízení se přistupuje jako k paměti (možné používat všechny instrukce); řízení přístupu—lze použít to, co se používá pro paměť
- problém: cache, oddělená sběrnice pro paměť
- rozdělení paměti na oblasti: 640 kB, 3 GB

Přístup k zařízením: přenos bez účasti CPU (2/2)

- přesun dat v předchozích případech vyžaduje účast CPU $\Longrightarrow$ neefektivní
- čtení z disku
  - řadič disku dostane požadavek: čtení
  - disk načte do interního bufferu data
  - řadič disku vyvolá přerušení
  - CPU čte postupně data z bufferu a ukládá do paměti
- Direct Memory Access (DMA)
  - obr. tan. 277
  - řadič DMA (DMAC) dostane požadavek: čtení + cílovou adresu
  - předá požadavek řadiči disku
  - zapisuje data do paměti
  - dokončení je oznámeno řadiči DMA
  - DMAC vyvolá přerušení
- různé varianty (např. přenos přes řadič DMA)
- spolupráce DMA s MMU
- PCI zařízení nepotřebují DMA (PCI Bus Master)

Přístup k I/O z aplikace (1/2)

- OS by měl zajistit přístup k zařízením uniformním způsobem bez ohledu na zařízení (IDE, SCSI, CD)
- např. zápisu bloku, přečtení znaku
- API, zápis do speciálních souborů (Linux, e.g., `/dev/hda`)
- Linux: major (ovladač) a minor (zařízení) číslo zařízení
- obr. Tan. p.299
- synchronní/asynchronní (blokující/neblokující) přístup k zařízení
- výlučný/sdílený přístup

Přístup k I/O z aplikace (2/2)

Aktivní čekání

- data se kopírují z bufferu do registru
- podle stavového registru se čeká až budou přenesena
- jednoduchá implementace, ale neefektivní

I/O s přerušeními

- není nutné čekat na přenos dat
- v průběhu přenosu může procesor provádět další činnost
- data předána jádru, aplikace zablokována
- přenos dat řídí obsluha přerušení

I/O přes DMA

- analogické přerušením, ale přenos dat řídí řadič DMA $\Longrightarrow$ míň přerušení

Bufferování

- optimalizace přenosu dat – zpoždění zápisu/čtení

Ovladače zařízení

- zajišťují přístup k zařízení $\Longrightarrow$ zápis, čtení, inicializace, správa napájení, logování
- typicky součást jádra OS (může být i v uživatelském prostoru $\Longrightarrow$ oddělení ovladačů)
- zakompilování do jádra vs. dynamické načítání
- měl by být dodáván výrobcem HW $\Longrightarrow$ definovaný model ovladačů (např. bloková zařízení)
- $\Longrightarrow$ spolupráce s ostatními částmi OS, sdílení funkcionality
- zjednodušení vývoje ovladačů; jednotný přístup aplikací
- Hardware Abstraction Layer (HAL)
- Windows, v Linuxu (*per se* není)
- SoRo str. 541

Bloková zařízení: HDD

- disk – plotny, stopy ($\Longrightarrow$ cylindry), sektory (typicky 512 B)
- původně se adresovaly sektory ve formě CHS (praktická omezení, mj. velikost disku)
- nahrazeno LBA (logical block addressing)
- low-level formát $\Longrightarrow$ hlavička + data + ECC
- připojené typicky přes (P)ATA, SATA
- rychlost přístupu ovlivňuje
  - nastavení hlavičky na příslušný cylindr (seek time; nejzásadnější)
  - rotace (nastavení sektoru) pod hlavičku
  - přenosová rychlost
- nezávislá cache (hromadí požadavky $\Longrightarrow$ eliminuje přesuny)

HDD: optimalizace

- víc požadavků se bude řešit najednou
- místo sektorů se pracuje s clustery sektorů (velikost podle velikosti disku)
- cache disku
- cache OS $\Longrightarrow$ společně s VM; cachuje se na úrovni FS
- odpovídající algoritmy – LRU, LFU, . . . , jejich kombinace
- zjednodušení OS $\Longrightarrow$ otevření souboru $\Longrightarrow$ namapování do cache; demand paging
- write-through cache: data se po zapsaní zapisují přímo na disk
- write-back cache: data se zapisují až po čase (možnost optimalizací zápisu)
- vynucení uložení cache (`flush`)
- sekvenční čtení
  - read-ahead – data se načítají dopředu
  - free-behind – proaktivně uvolňuje stránky, při načítání nových
- „spolupráce" – OS & HW (spoon-feeding); databáze

HDD: algoritmy přístupu

Varianty

- FCFS (First Come First Serve)
- SSFT (Shortest Seek Time First) – vybrán ten, kam bude nejrychlejší přesun (má tendenci zůstat uprostřed)
- SCAN (výtahový algoritmus) – raménko je systematicky přesouváno od jednoho okraje k druhému a postupně zapisuje data (data uprostřed zapisována rychleji)
- C-SCAN – jako SCAN, ale data se zapisují jen v jednom směru
- LOOL a C-LOOK – jako SCAN a C-SCAN, ale nepřesunují hlavičky až na konec, když tam není požadavek

Poznámky

- SSFT vhodný pro sekvenční práci; SCAN pro zatížené systémy
- zmíněné algoritmy počítají s ,,virtuální reprezentací" disku

Bloková zařízení: RAID

- SLED: Single Large Expensive Disk
- RAID: Redundant Array of Inexpensive/Independent disks
- Mean Time to Failure: $MTTF_{pole} = \frac{MTTF_{disk}}{N}$
- hardware vs. software RAID
- RAID-0 (stripping): zvýšení propustnosti, problém selhání pořád existuje
- RAID-1 (mirroring): zvýšení propustnosti (kopie), řeší problém selhání
- RAID-2: dělí data na po bitech; Hammingův kód; disk pro paritu (napoužívá se)
- RAID-3: dělí data po bytech; XOR; disk pro paritu (zátěž); zvládne výpadek jednoho disku
- RAID-4: jako RAID-3 používá bloky (zátěž)
- RAID-5: jako RAID-4; paritní bloky jsou, ale distribuovány
- RAID-6: jako RAID-5; Reed-Solomon kód; dva paritní bloky; výpadek až dvou disků
- kombinace: RAID-0+1, RAID-1+0

Bloková zařízení: SSD, CD, DVD

Solid-state Drives

- flash paměti; popř. rozhraní jako HDD
- bez rotujících částí $\Longrightarrow$ rychlý přístup (výrazně víc IOPS)
- problematický zápis
  - omezení na počet přepsání jednoho místa
  - paměť musí být nejdřív vymazána
  - často lze zapisovat po stránkách, ale mazat je nutné po blocích $\Longrightarrow$ rychlější zápis než přepis
- wear levelling
  - *žádný* – data se přepisují na místě
  - *dynamický* – změněné bloky označeny jako neplatné a data zapsány jinam (USB)
  - *statický* – jako dynamický, ale přesouvá i nezměněné data (SSD)
  - softwarová vs. hardwarová implementace (JFFS2, LogFS)
- garbage collection + TRIM

Compact Disc (CD)

- data umístěna ve spirále $\Longrightarrow$ pomalé vyhledávání; rychlé sekvenční čtení
- vysoká redundance dat
- symbol - k zakodávání 8 b se používá 14 b
- 42 symbolů tvoří rámec o velikosti 588 b (192 b data, zbytek ECC)
- jeden sektor obsahující 2048 B dat je tvořen 98 rámci (zahrnuje 16 B hlavičku a 288 B pro ECC)
- efektivita 28%!

DVD

- analogicky jako CD

Bloková zařízení: SAN, NAS, NBD

SAN (Storage Area Network)

- umožnuje připojit disky přes síť, aby se jevily jako lokální
- SCSI přes Fiber Channel
- ATA over Ethernet
- iSCSI
- virtulizace a konsolidace úložných zařízení
- možnost řešit výpadky HW

NAS (Network Attached Storage)

- poskytuje zařízení na úrovni souborového systému

NBD (Network Block Device)

- zpřístupňuje blokové zařízení přes síťové spojení

Znaková zařízení

Terminál

- většina počítačů: klávesnice + monitor $\Longrightarrow$ terminál
- osobní počítače
- síťové terminály
- samostatné terminály (RS-232) $\Longrightarrow$ převod znaků na sériovou linku a zpět
- vstup z klávesnice a výstup řeší odlišné ovladače
- možnost předávat znaky přímo aplikaci (RAW mode) nebo počkat (backspace; cooked mode) $\Longrightarrow$ ovladač musí mít buffer (echoing)
- speciální znaky pro speciální chování (Ctl+D $\Longrightarrow$ EOF; Ctl+H $\Longrightarrow$ backspace; Ctl+\ $\Longrightarrow$ SIGQUIT)
- aplikace často vyžadují sofistikovaný přístup k výpisu textu (editory)
- $\Longrightarrow$ escape sekvence (rozdíly mezi terminály; ANSI)
- speciální znak ESC (0x1B)
- např. ESC[*n*A $\Longrightarrow$ posun kurzoru o *n* řádků nahoru
- další operace: vkládání/mazaní řádků, posun doleva/doprava, změna barvy

Hodiny (časovače)

- krystal generující pravidelné pulzy (např. 1000 MHz)
- programovatelný časovač
  - nastavením registru na určitou hodnotu se inicializuje
  - při každém pulzu snížena hodnota o jedna
  - při nula vygeneruje přerušení a zastaví se
- různé funkce
- může jich být víc (příp. možnost emulovat jedním)
  - evidence reálného času
  - plánování procesů (proces nesmí využít víc času než mu bylo přiděleno)
  - uložení cache
  - systémové volání `alarm`

Shrnutí I/O

- blokující (synchronní) – aplikace vydá požadavek a je uspána do doby než je vyřešen
- neblokující – nedochází k uspání; požadavek je vyřešen okamžitě pokud to jde; např. `read` vrátí dostupná data
- asynchronní – požadavek je předán; v momentě, kdy je vyřešen, ja o tom aplikace infomována
- buffer – paměť určená k přenosu dat mezi zařízeními (přip. zařízením a aplikací)
  - vyrovnání se s odlišnými přenosovými rychlostmi
  - vyrovnání se různými velikostmi přenášených dat
  - sémentika kopírování (copy semantics) – jádro vs. aplikace
- cache – rychlá paměť, která umožňuje zryhlit přístup k jinak pomalejšímu zařízení
- cache a buffer – odlišné funkce, i když stejná paměť může být použita pro oba účely