



Operační systémy 2

# I/O: zařízení

Petr Krajča



Katedra informatiky  
Univerzita Palackého v Olomouci

- 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 128 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, optické disky, páska?, ...

## **Znaková zařízení**

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

## **Ostatní**

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



## Port-Mapped I/O

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

## Memory-Mapped I/O

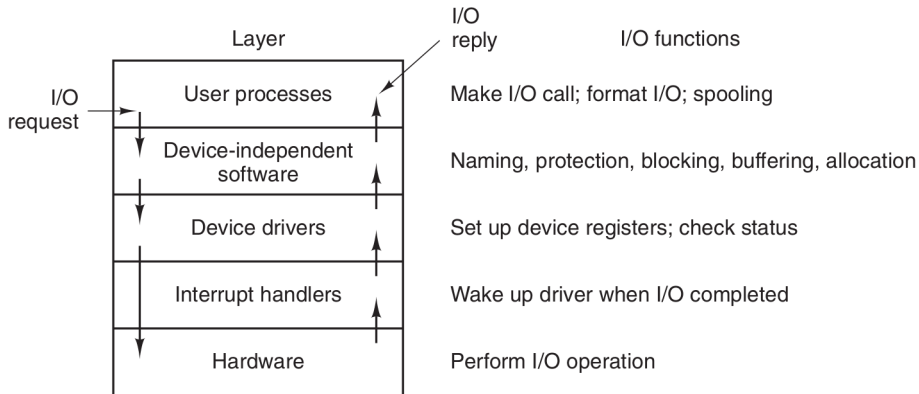
- registry jednotlivých zařízení jsou namapovány do paměťového prostoru
- data se čtou/zapisují přímo na sběrnici; přesměrováno na sběrnici zařízení, ne do paměti
- výhoda: k zařízení se přistupuje jako k paměti (možné používat všechny instrukce); řízení přístupu—lze použít to, co se používá pro paměť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



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



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



Zdroj: A. Tanenbaum. Modern Operating systems.

## Aktivní čekání

- data se kopírují z bufferu do registru zařízení (popř. opačně)
- podle stavového registru se čeká, až budou přenesena
- jednoduchá implementace (neefektivní)

## I/O s přerušeními

- není nutné čekat na dokončení přenosu dat
- v průběhu přenosu může procesor provádět další činnost
- počas přenosu dat proces zablokován
- přenos dat řídí obsluha přerušení

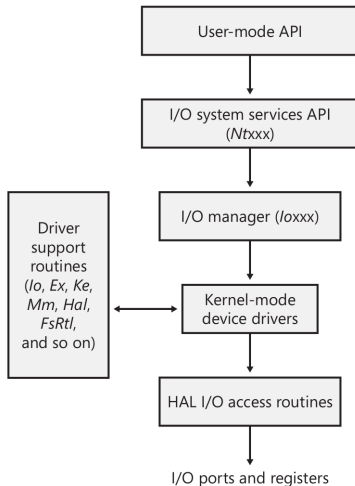
## I/O přes DMA

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

## Bufferování

- optimalizace přenosu dat – zpoždění zápisu/čtení

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



Zdroj: M. Russinovich, D. Solomon. Windows Internals, 5ed.

- disk – plotny, stopy ( $\implies$  cylindry), sektory (typicky 512 B, 4096 B)
- původně se adresovaly sektory ve formě CHS (praktická omezení, mj. velikost disku)
- nahrazeno LBA (logical block addressing)
- low-level formát  $\implies$  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  $\implies$  eliminuje přesuny)

- víc požadavků se bude řešit najednou
- místo sektorů se pracuje s clustery sektorů (velikost podle velikosti disku)
- cache disku
- cache OS  $\implies$  společně s VM; cachuje se na úrovni FS
- odpovídající algoritmy – LRU, LFU, . . . , jejich kombinace
- zjednodušení OS  $\implies$  otevření souboru  $\implies$  namapování do cache; demand paging
- write-through cache: data se po zapsání 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

- 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 (nepoužívá 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

## Solid-state Drives

- flash paměti; fyzické rozhraní jako HDD nebo M.2 (PCI Express)
- bez rotujících částí  $\implies$  rychlý přístup (výrazně víc IOPS)
- problematický zápis
  - omezení na počet přepsání jednoho místa
  - paměť musí být nejdříve vymazána
  - často lze zapisovat po stránkách, ale mazat je nutné po blocích  $\implies$  rychlejší zápis než přepis
- wear levelling
  - *žádný* – data se přepisují na místě
  - *dynamický* – změněné bloky označeny jako neplatné a data zapsány jinde (USB)
  - *statický* – jako dynamický, ale přesouvá i nezměněné data (SSD)
  - softwarová vs. hardwarová implementace (JFFS2, LogFS)
- garbage collection + TRIM

## CD

- data umístěna ve spirále  $\implies$  pomalé vyhledávání; rychlé sekvenční čtení
- vysoká redundance dat
- symbol - k zakodování 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%!
- laser 780 nm

## DVD

- analogicky jako CD
- laser 650 nm

## BD (Blue-ray disc)

- analogicky jako CD & DVD
- laser 405 nm



## **SAN (Storage Area Network)**

- umožňuje připojit disky přes síť, aby se jevíly jako lokální
- SCSI přes Fiber Channel
- ATA over Ethernet
- iSCSI
- virtualizace 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

## Terminál

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

- 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 přerušeni a zastaví se
- může jich být víc (příp. možnost emulovat jedním)
- různé funkce
  - evidence času
  - plánování procesů (proces nesmí využít víc času než mu bylo přiděleno)
  - uloženi cache
  - systémové volání `alarm`

- blokující (synchronní) – proces vydá požadavek a je uspán 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, je o tom proces informován
- buffer – paměť určená k přenosu dat mezi zařízeními (příp. zařízením a procesem)
  - vyrovnání se s odlišnými přenosovými rychlostmi
  - vyrovnání se s různými velikostmi přenášených dat
  - sémantika kopírování (copy semantics) – jádro vs. procesem
- cache – rychlá paměť, která umožňuje zrychlit přístup k jinak pomalejšímu zařízení
- cache a buffer – odlišné funkce, i když stejná paměť může být použita pro oba účely