10., Správa I/O

ZOS 2024,V2, L. PEŠIČKA

OS

- Modul pro správu procesů
- Modul pro správu paměti
- **Modul pro správu I/O**
- Modul pro správu souborů
- Síťování

Důležité pojmy

- Instrukce **IN, OUT** pro práci s periferním zařízením
  - Privilegované
  - Používají adresy I/O portu – jiný adresní prostor, než do RAM
  - Adresa 70 v IN instrukci je jiná než adresa 70 do RAM
  - Signál (drát v počítači) M/IO určuje, zda je adresa na adresních vodičích do paměti nebo I/O portu

Vývoj rozhraní mezi CPU a zařízeními

1. CPU řídí přímo periferii
2. CPU – řadič – periferie
   (řadič a aktivní čekání CPU na dokončení operace)
3. řadič umí vyvolat přerušení
4. řadič umí DMA
5. I/O modul
6. I/O modul s vlastní pamětí

1. CPU řídí přímo periferii

- CPU přímo vydává potřebné signály
- CPU dekóduje signály poskytované zařízením
- Nejjednodušší HW
- Nejméně efektivní využití CPU

- Jen v jednoduchých mikroprocesorem řízených zařízeních (dálkové ovládání televize)

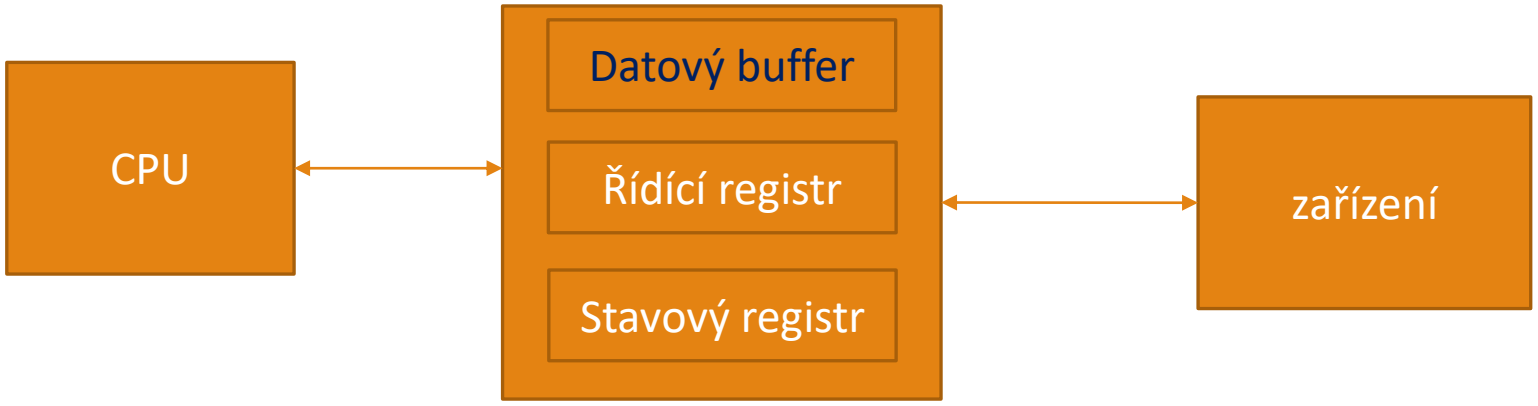
2. CPU – řadič - periférie

Řadič (device controller)

- Převádí příkazy CPU na elektrické impulzy pro zařízení
- Poskytuje CPU info o stavu zařízení
- Komunikace s CPU pomocí registrů řadiče na známých I/O adresách
- HW buffer pro alespoň 1 záznam (blok, znak, řádka)
- Rozhraní řadič-periférie může být standardizováno (SCSI, IDE, …)

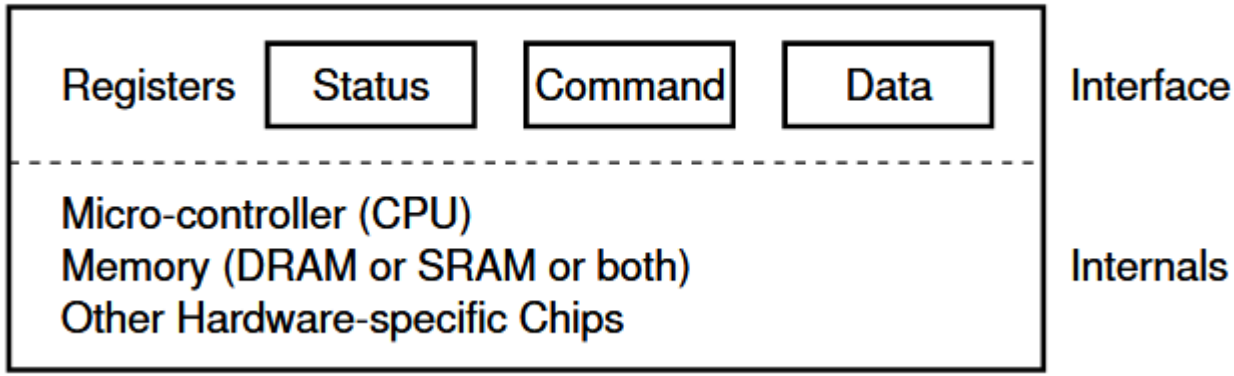
Povel – příkaz, který dává CPU řadiči
Stav – informace o stavu, např. přenos OK
Data – buffer na předávaná data

CPU, řadič, zařízení



Povel – příkaz, který dává CPU řadiči (přes řídící registr)
Stav – informace o stavu, např. přenos OK (stavový registr)
Data – buffer na předávaná data

Řadič


Registers
Status
Command
Data
Interface
Micro-controller (CPU)
Memory (DRAM or SRAM or both)
Other Hardware-specific Chips
Internals

2. řadič – příklad operace zápisu

- CPU zapíše data do bufferu,
  Informuje řadič o požadované operaci (chci zapsat data)
- Po dokončení výstupu zařízení nastaví příznak, který může CPU otestovat (stav nastaví na OK)
- if přenos == OK, může vložit další data

- CPU musí dělat všechno (programové I/O)
- Významnou část času stráví CPU
  čekáním na dokončení I/O operace

3. Řadič umí vyvolat přerušení

- CPU nemusí testovat příznak dokončení
- Při dokončení I/O vyvolá řadič **přerušení**
- CPU začne obsluhovat přerušení
  - Obslužná procedura přerušení určí co dále
- Postačuje pro pomalá zařízení, např. sériové I/O

4. Řadič může přistupovat k paměti pomocí DMA (!!!!)

- DMA přenosy mezi pamětí a I/O zařízením
- CPU inicializuje přenos, ale sám ho nevykonává
- Řadič DMA – speciální obvod, zajišťuje blokové přenosy mezi I/O zařízením a pamětí

1. CPU zadá požadavek řadiči DMA (odkud: adresu I/O zařízení, kam: adresu v RAM, kolik: počet bytů, lze i naopak)
2. DMA obvod provede přesun dat bez zásahu CPU
3. CPU se zatím může věnovat dalším věcem (ale je omezen ve využití sběrnice)
4. Po ukončení přenosu DMA obvod vyvolá přerušení

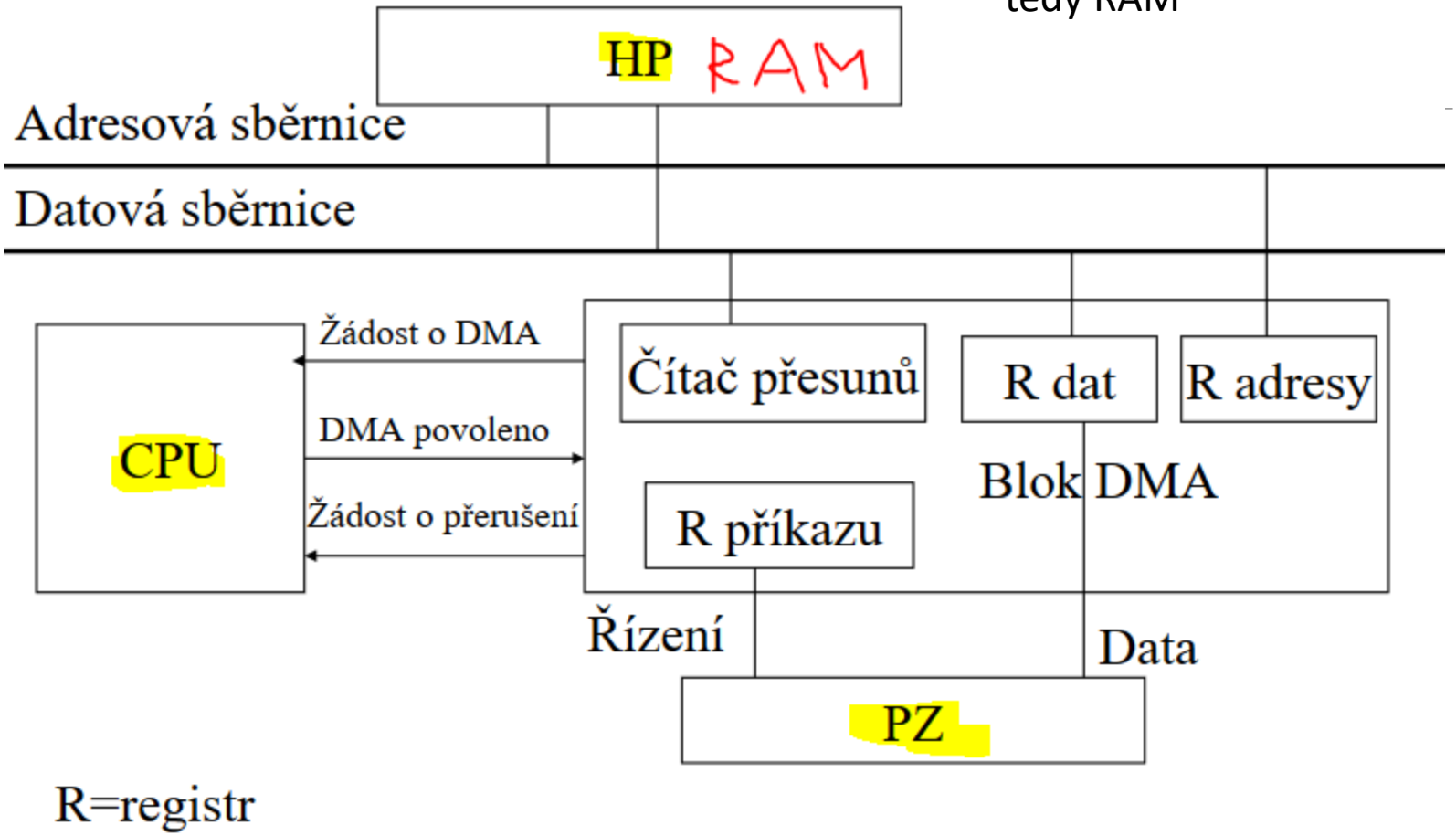
4. poznámky

- Bus mastering – zařízení převezme kontrolu nad sběrnicí a přenos provede samo (PCI sběrnice)

- Vhodné pro rychlá zařízení – řadič disků, síťová karta, zvuková karta, grafická karta atd.

DMA

HP = hlavní paměť,
tedy RAM



R=registr

Zdroj: http://slideplayer.cz/slide/3685221/

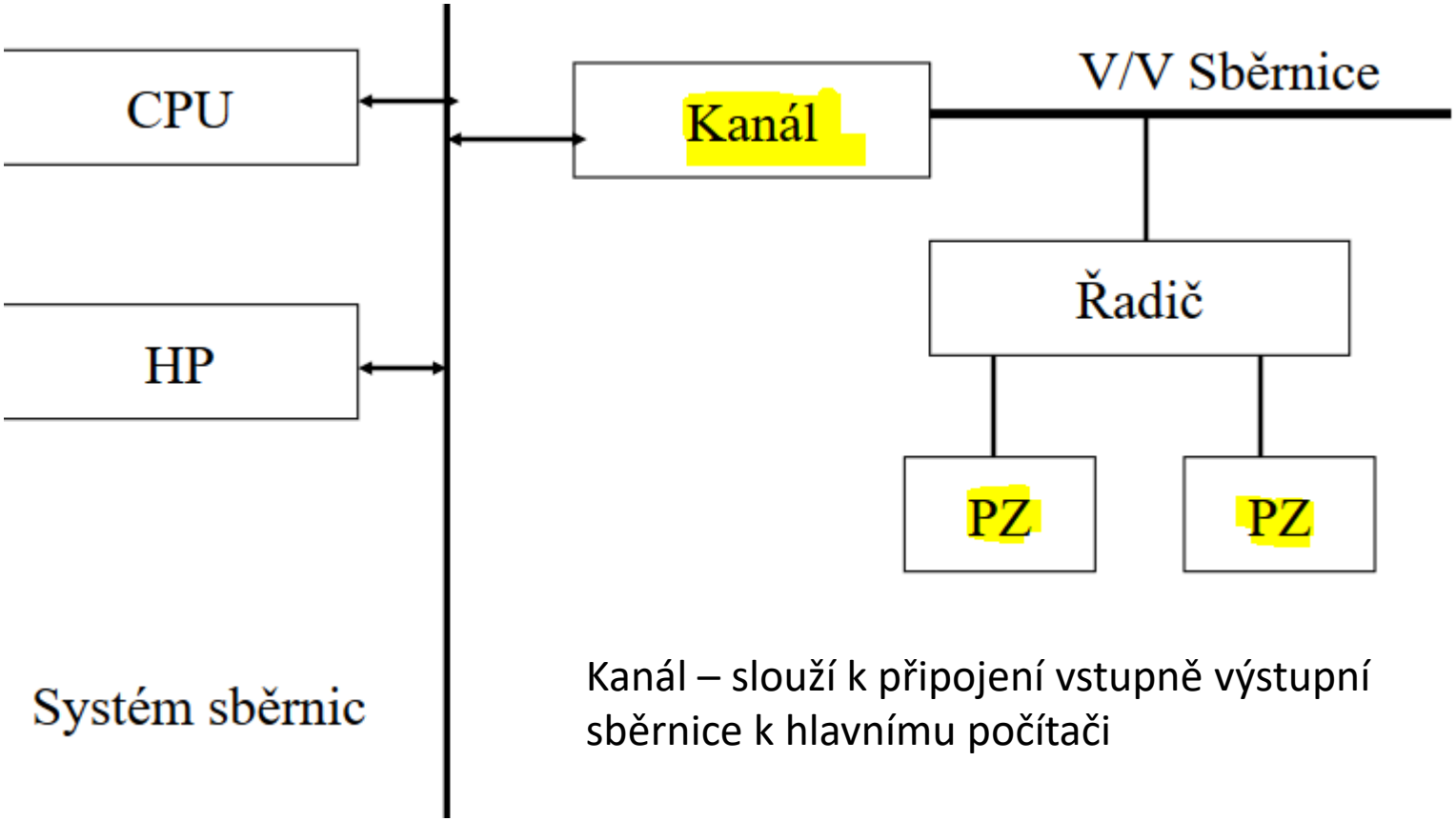
5. I/O modul umí interpretovat speciální I/O programy

- I/O procesor
- Interpretuje programy v hlavní paměti (RAM)
- CPU spustí I/O procesor
  - I/O procesor provádí své instrukce samostatně
  - V podstatě samostatný počítač, program má v RAM

6. I/O modul s vlastní pamětí

- I/O modul provádí programy
- Má vlastní paměť(!)
  - Je vlastně samostatným počítačem
- Složité a časově náročné operace
  grafika, šifrování, …

Kanálová architektura (5,6)



Kanál – slouží k připojení vstupně výstupní sběrnice k hlavnímu počítači

Zdroj: http://slideplayer.cz/slide/3685221/

Komunikace CPU s řadičem

Tři varianty:

- Odlišné adresní prostory (I/O prostor)
  - CPU zapisuje do registrů řadiče pomocí speciálních I/O instrukcí
  - Vstup: IN R, port
  - Výstup: OUT R, port
- 1 adresní prostor (RAM)
- Hybridní schéma (I/O, RAM)

Důležitá poznámka

Mezi privilegované instrukce patří:

- řízení CPU
- zákaz přerušení
- práce se speciálními registry
- práce se vstupními a výstupními zařízeními
- nastavení a mapování paměti

Viz https://cs.wikipedia.org/wiki/Privilegovan%C3%BD_re%C5%BEim

Ad – 1 adresní prostor (RAM)

- Používá **vyhrazené** adresy
- Nazývá se *paměťově mapované I/O*
- HW musí pro dané adresy umět vypnout cachování

- Danou oblast můžeme namapovat do virtuálního adresního prostoru nějakého procesu
(zpřístupnění I/O zařízení)

Ad – hybridní schéma

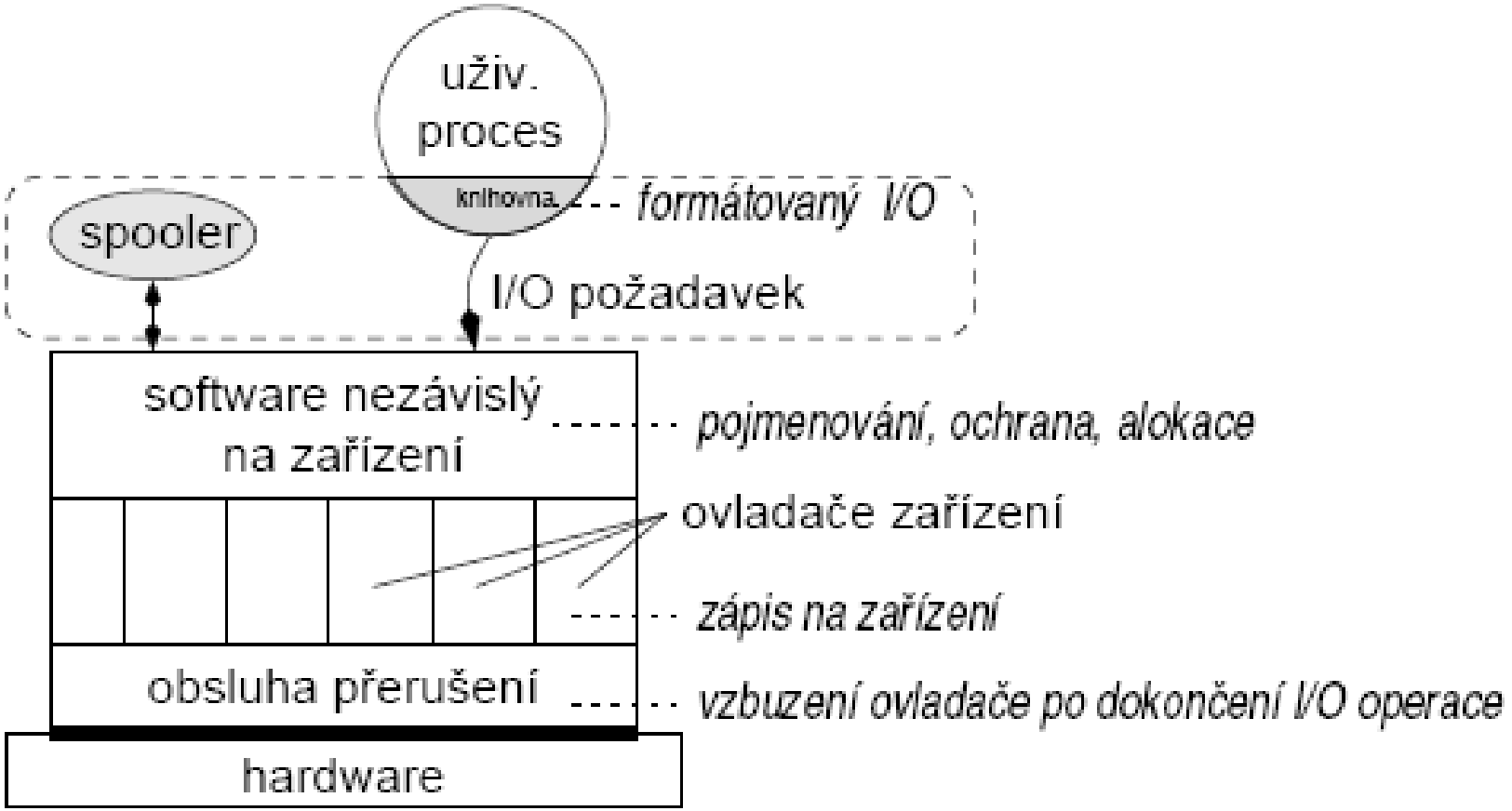
- Řídící registry
  - Přístup pomocí I/O instrukcí (instrukce IN, OUT)
- HW buffer
  - Mapován do paměti (RAM)
- Např. na PC
  (buffery mapovány do oblasti 640K až 1MB)

Principy I/O software (!!!!)

typicky strukturován do 4 úrovní (od nejnižší):

1. obsluha přerušení (nejnižší úroveň v OS)
2. ovladač zařízení
3. SW vrstva OS nezávislá na zařízení
4. uživatelský I/O SW

Toto je potřeba znát !!


uživ.
proces
knihovna
formátovaný I/O
spooler
I/O požadavek
software nezávislý
na zařízení
pojmenování, ochrana, alokace
ovladače zařízení
zápis na zařízení
obsluha přerušení
vzbuzení ovladače po dokončení I/O operace
hardware

1. Obsluha přerušení

- řadič vyvolá přerušení ve chvíli dokončení I/O požadavku
- snaha, aby se přerušením nemusely zabývat vyšší vrstvy

- ovladač zadá I/O požadavek, usne - P(sem)
- po příchodu přerušení ho obsluha přerušení vzbudí - V(sem)

- časově kritická obsluha přerušení – co nejkratší

2. Ovladače zařízení

- obsahují veškerý kód závislý na konkrétním I/O zařízení (např. ovladač zvukovky od daného výrobce)
- ovladač zná jediný hw podrobnosti
  - způsob komunikace s řadičem zařízení
  - zná detaily – např. ví o sektorech a stopách na disku, pohybech diskového raménka, start & stop motoru
- může ovládat všechna zařízení daného druhu nebo třídu příbuzných zařízení
  - např. ovladač SCSI disků – všechny SCSI disky

Funkce ovladače zařízení

1. ovladači předán příkaz od vyšší vrstvy
   - např. zapiš data do bloku n
2. nový požadavek zařazen do fronty
   - může ještě obsluhovat předchozí
3. ovladač zadá příkazy řadiči (požadavek přijde na řadu)
   - např. nastavení hlavy, přečtení sektoru
4. zablokuje se do vykonání požadavku
   - neblokuje při rychlých operacích – např. zápis do registru
5. vzbuzení obsluhou přerušení (dokončení operace) – zkontroluje, zda nenastala chyba

Funkce ovladače zařízení – pokrač.

6. pokud OK, předá výsledek (status + data) vyšší vrstvě
   - status – datová struktura pro hlášení chyb
7. Zpracuje další požadavky ve frontě
   - jeden vybere a spustí

Poznámky

- ovladače často vytvářejí výrobci HW
  - dobře definované rozhraní mezi OS a ovladači
- ovladače podobných zařízení – stejná rozhraní
  - např. sítové karty, zvukové karty, ...

Problémy s ovladači

- Chyba ovladače – pád systému
  - Běh v privilegovaném režimu (jádře)
  - Chyba v ovladači může způsobit pád systému

- Ovladač pro určitý HW i určitý OS
  - Můžete mít starší kameru s ovladačem pro Windows XP, ale třeba nebude použitelná ve Windows 8.1

3. SW vrstva OS nezávislá na zařízení

- I/O funkce společné pro všechna zařízení daného druhu
  - např. společné fce pro všechna bloková zařízení
- definuje rozhraní s ovladači
- poskytuje jednotné rozhraní uživatelskému SW

Tato vrstva často dělá studentům potíže. Přestože má jasně definované funkce – viz další slide

Poskytované funkce (!!)

- pojmenování zařízení
  - LPT1, COM1 (paralelní a sériový port), /dev/lp0
- ochrana zařízení (přístupová práva)
- alokace a uvolnění vyhrazených zařízení
  - v 1 chvíli použitelná pouze jedním procesem
  - např. tiskárna, plotter, magnetická páska
- vyrovnávací paměti
  - bloková zařízení – bloky pevné délky
  - pomalá zařízení – čtení / zápis s využitím bufferu

Poskytované funkce – pokračování

- hlášení chyb
- jednotná velikost bloku pro bloková zařízení

- v moderních OS se zařízení jeví jako objekty v souborovém systému (v mnoha OS je tato vrstva součástí logického souborového systému)

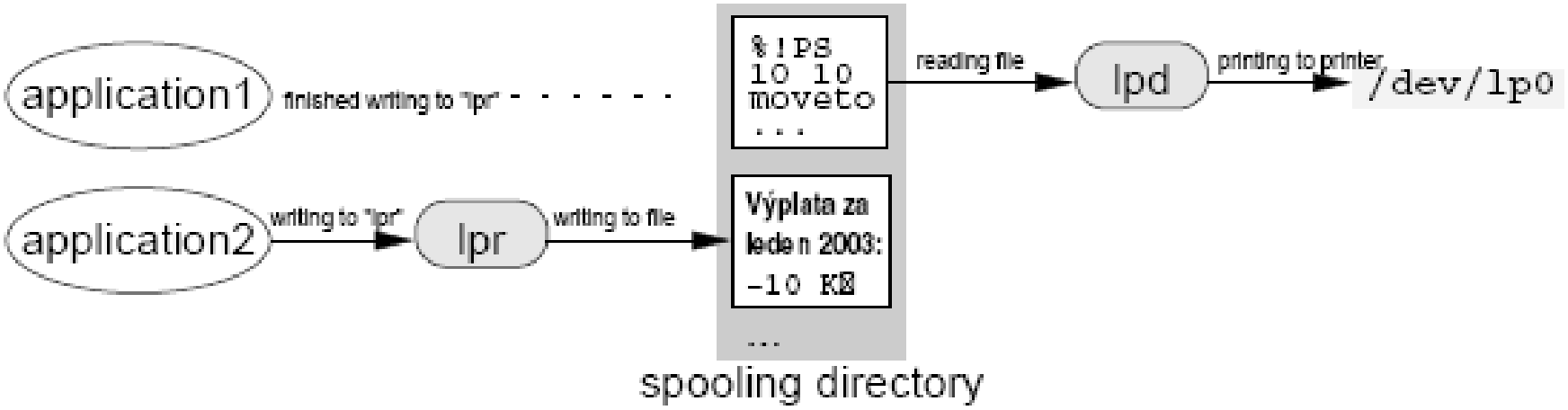
4. I/O sw v uživatelském režimu

- programátor používá v programech I/O funkce nebo příkazy jazyka
  - např. printf v C
  - knihovny sestavené s programem
  - formátování - printf("%.2d:%.2d\n", hodin, minut)
  - často vlastní vyrovnávací paměť na jeden blok
- spooling
  - implementován pomocí procesů běžících v uživ. režimu
  - způsob obsluhy vyhrazených I/O zařízení (multiprogram.)
  - např. proces by alokoval zařízení a pak hodinu nic nedělal

Příklad spoolingu – tisk Unix

- přístup k fyzické tiskárně – pouze 1 speciální proces
  - daemon **lpd**
- proces vygeneruje celý soubor, lpd ho vytiskne
  - proces chce tisknout, spustí **lpr** a naváže s ním komunikaci
  - proces předává tisknutá data programu lpr
  - lpr zapíše data do souboru v určeném adresáři
    - spooling directory – přístup jen lpr a lpd
  - dokončení zápisu – lpr oznámí lpd, že soubor je připraven k vytisknutí, lpd soubor vytiskne a zruší

tisk Unix



lpd – démon (služba) čte ze spoolovacího adresáře a přistupuje k tiskárně

lpr – data, která chce aplikace vytisknout se zapisují do spoolovacího adresáře

Poznámka - spooling lze použít např. i pro přenos elektronické pošty

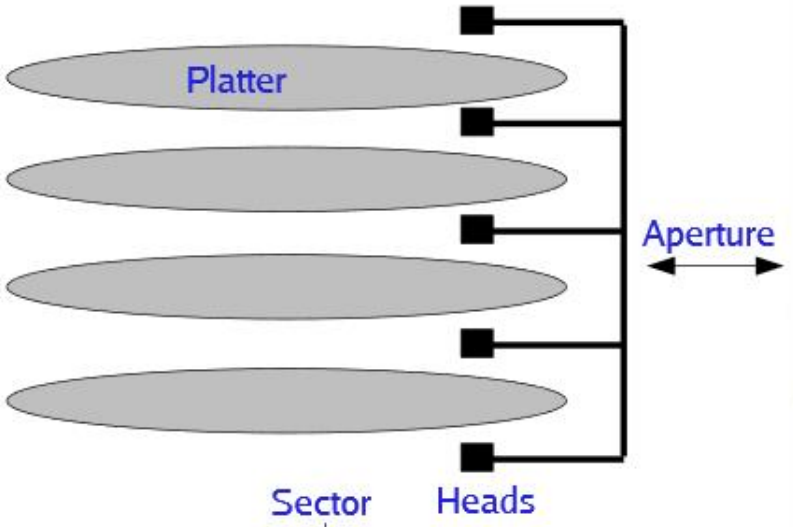
Disk

- Rotační disky
  - doba vystavení (posun hlaviček) + rotační zpoždění
  - Stopa, sektor
- SSD disky
  - dražší, menší kapacita
- Mix
  - SSD disk (systém) v kombinaci s rotačním diskem (velká data)

A Disk Primer

Disks consist of one or more *platters* divided into *tracks*

- Each platter may have one or two *heads* that perform read/write operations
- Each track consists of multiple *sectors*
- The set of sectors across all platters is a *cylinder*



plotna
stopa (track)
hlava
sektor – 512B nebo 4KB
cylindr

SSD disk

- Vyšší cena za stejnou kapacitu
- Nemá pohyblivé části
- Rychlá přístupová doba (0.1ms)
- Musí vymazat celý blok, než do něj zapíše
- Omezený počet cyklů mazání/zápisu

Proč RAID?

- pevný disk
  - elektronická část + mechanická
  - náchylnost k poruchám
  - cena dat >> cena hw
- odstávka při výměně zařízení
  - náhrada hw, přenos dat ze zálohy - prostoje
  - SLA 24/7 (Service Level Agreement)
- větší disková kapacita než 1 disk
- RAID
  - Redundant Array of Independent (Inexpensive) disks

RAID

- data jsou distribuována na více disků
- datová operace je realizována paralelně (např. na disk 1 a 2)
- kromě distribuování dat na více disků zvýšení spolehlivosti (mimo RAID 0) – redundance informace (zdvojení disků nebo parita)
  - -> vliv na rychlost a/nebo spolehlivost úložiště
- sada fyzických disků, OS je následně vidí jako jeden disk
- Redundant Array of Independent Disks
- Redundant Array of Inexpensive Disks

RAID – používané úrovně

- RAID 0 .. jen zvýšení rychlosti, nikoliv spolehlivosti
- RAID 1, 5 , 6 .. nejpoužívanější
- RAID 5 .. distribuovaná parita (může vypadnout 1 disk)
- RAID 6 .. zdvojená parita (můžou vypadnout 2 disky)
- RAID 10 .. kombinace 1 a 0
- RAID 0+1 .. Kombinace 0 a 1
- RAID 50 .. kombinace RAIDů 0 a 5

- pojmy:
  - SW (RAID dělá OS) nebo HW RAID (speciální HW karta)
  - hot plug
  - hot spare
  - Degradovaný režim – jeden (či více dle typu RAIDu) z disků v poli je porouchaný, ale RAID stále funguje

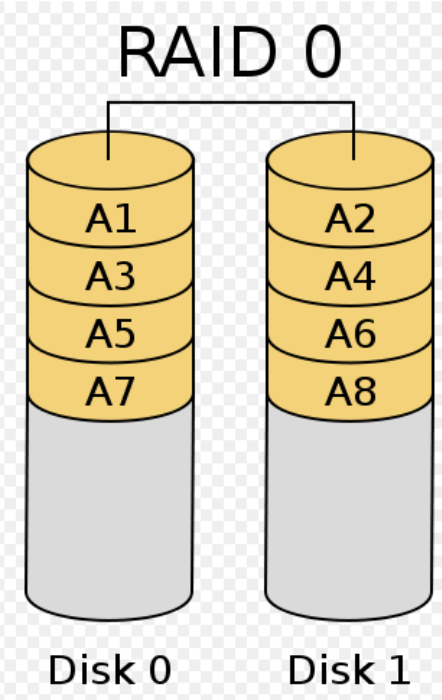
RAID 0

Dva režimy RAID 0:
zřetězení a prokládání

- není redundantní, neposkytuje ochranu dat
- ztráta 1 disku – ztráta celého pole nebo části (dle režimu)
- důvod použití – může být výkon při režimu prokládání (např. střih videa)
- Dva režimy – zřetězení nebo prokládání (stripping)

RAID 0 (!!)

- Zřetězení
  - Data postupně ukládána na několik disků
  - Zaplní se první disk, pak druhý, atd.
  - Snadné zvětšení kapacity, při poruše disku ztratíme jen část dat
- Prokládání
  - Data ukládána na disky cyklicky po blocích (stripy)
  - Při poruše jednoho z disků – přijdeme o data
  - Větší rychlost čtení / zápisu
    - Jeden blok z jednoho disku, druhý blok z druhého disku

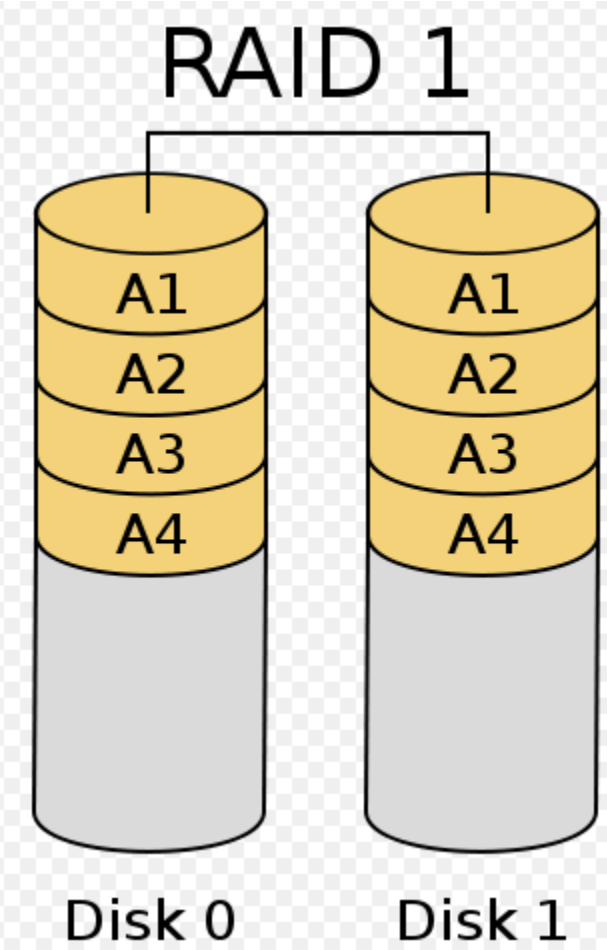


Na obrázku je režim prokládání, zdroj: wikipedia (i u dalších obrázků)

RAID 1 (!!)

- mirroring .. zrcadlení
- na 2 disky stejných kapacit totožné informace
- výpadek 1 disku – nevadí
- jednoduchá implementace – často čistě sw
- nevýhoda – využijeme jen polovinu kapacity
- zápis – pomalejší (stejná data na 2 disky)
  ovlivněn diskem, na němž bude trvat déle
- čtení – rychlejší
  (řadič - lze střídat požadavky mezi disky)

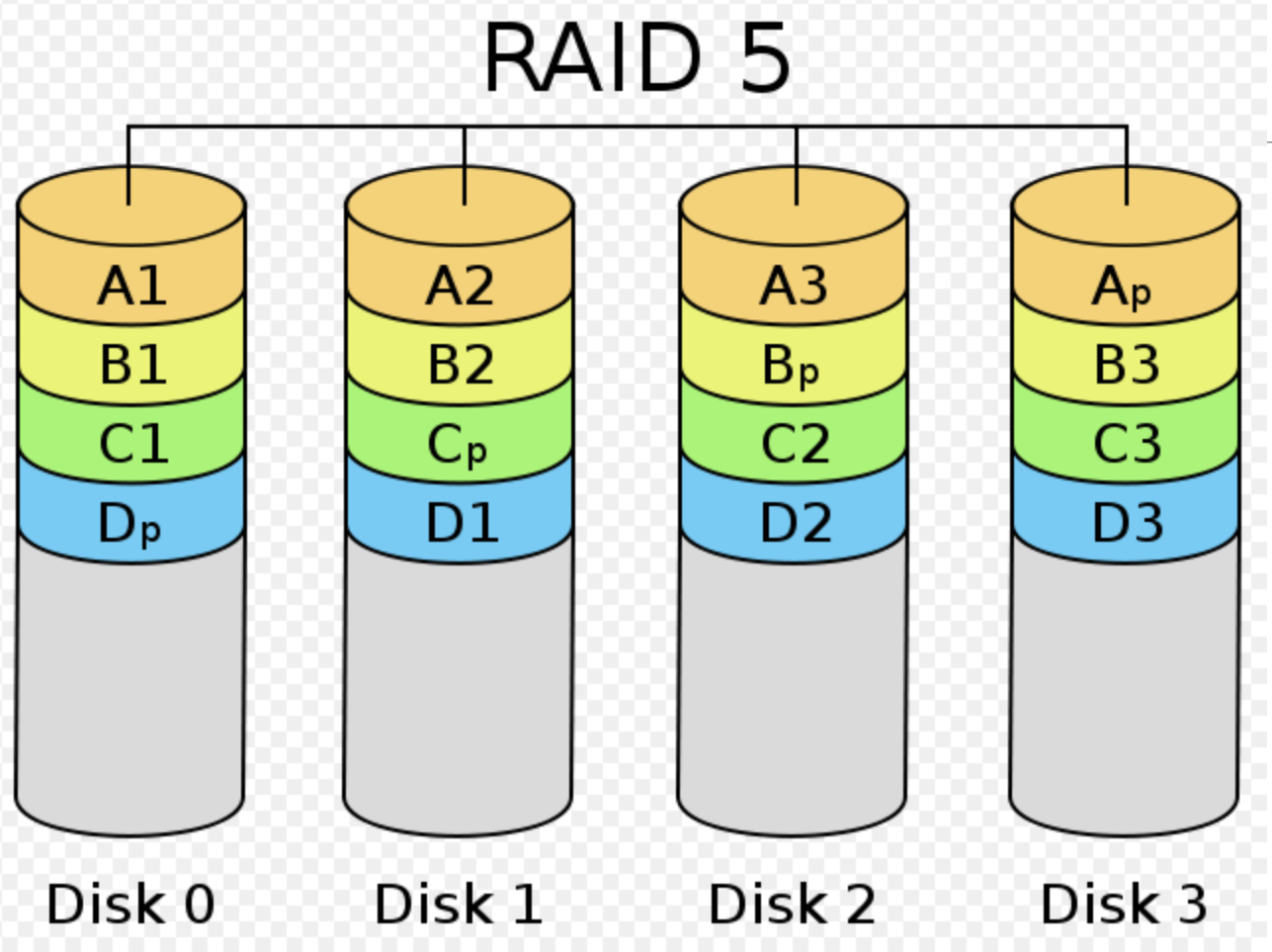
RAID 1



RAID 5 (!!)

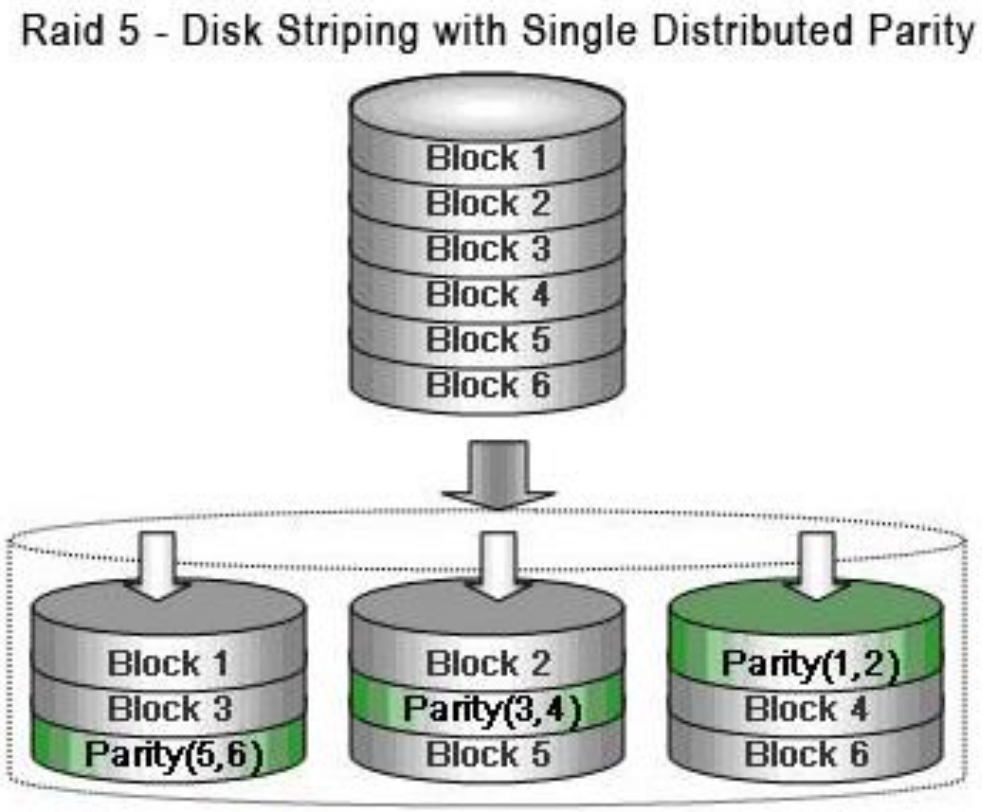
- redundantní pole s distribuovanou paritou
- minimálně 3 disky
- režie: odpovídá kapacitě 1 disku z pole n disků
  - 5 disků 100GB : 400GB pro data
- výpadek 1 disku nevadí
- čtení – výkon ok
- zápis – pomalejší

  1 zápis – čtení starých dat, čtení staré parity, výpočet nové parity, zápis nových dat, zápis nové parity



Např. RAID 5 z 4 disků 1TB, výsledná kapacita: 3 TB
Může vypadnout 1 z disků a o data nepřijdeme

RAID 5



Zdroj:
http://www.partitionwizard.com/resize-partition/resize-raid5.html

RAID 5 – jak funguje?

- Pro zjednodušení si představme, že máme tři disky, blok bude mít 3 bity a používáme sudou paritu
- Červeně bude označen dopočtený paritní bit.

Disk 1	Disk 2	Disk 3	Co ukládáme například:
1	0	1	1 0
0	1	1	0 1
0	0	0	0 0
1	1	0	1 1

Vidíme, že se nám parita střídá mezi jednotlivými disky 3, 2, 1

RAID 5 jak funguje?

- Přijdeme o libovolný disk, např. disk 2
- Až to administrátor zjistí, nahradí ho novým a je třeba na něm dopočítat data

Disk 1	Disk 2	Disk 3	Co ukládáme například:
1	?	1	1 0
0	?	1	0 1
0	?	0	0 0
1	?	0	1 1

Dopočet dat

Disk 1	Disk 2	Disk 3	Co ukládáme například:
1	?	1	1 0
0	?	1	0 1
0	?	0	0 0
1	?	0	1 1

Disk 1	Disk 2	Disk 3	Co ukládáme například:
1	0	1	1 0
0	1	1	0 1
0	0	0	0 0
1	1	0	1 1

V každém řádku dopočteme, aby mezi sloupci disk1, 2, 3 byl sudý počet jedniček

Delší blok dat

Disk 1	Disk 2	Disk 3	Co ukládáme například:
10	00	10	10 00
01	10	11	01 11
00	00	00	00 00
11	10	01	11 10

RAID 5 - poznámky

- Proč je parita distribuována a není jen na jednom disku?
  - Disk s paritou by byl více vytížený
  - Zápis na libovolný disk -> zápis na paritní disk

- Degradovaný režim
  - Máme disky v RAID 5
  - Jeden z disků nám vypadne
    - Uživatel to nepozná
    - Pole běží v degradovaném režimu, maskuje poruchu
    - Je třeba, aby se dozvěděl administrátor – jinak vypadne další disk a přijdeme o data

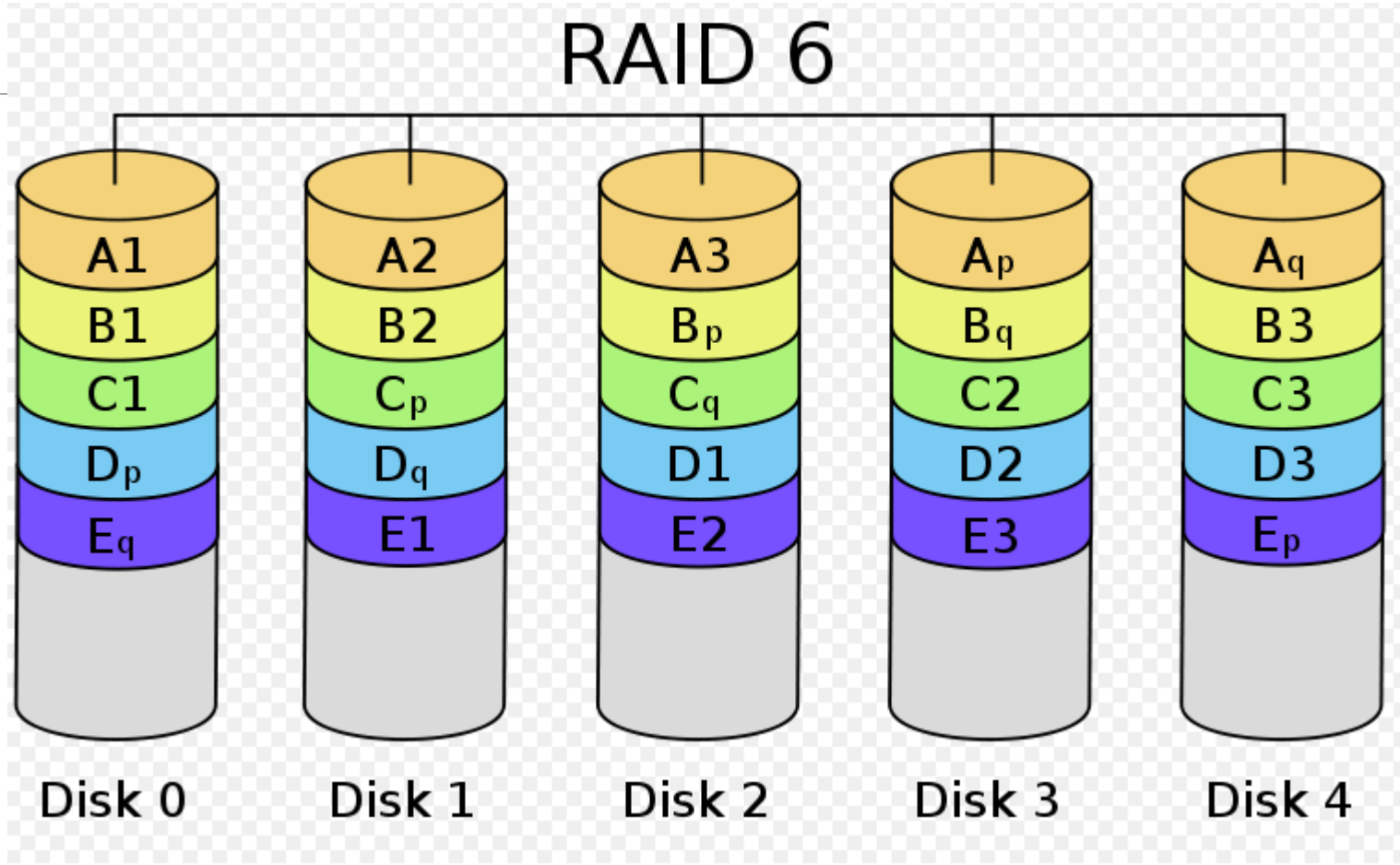
RAID 5

- Nejpoužívanější v serverech spolu s RAID 6

- detekce poruchy v diskovém poli
- hot spare disk
- použití hot plug disků (lze vyjmout z přední stěny za běhu)

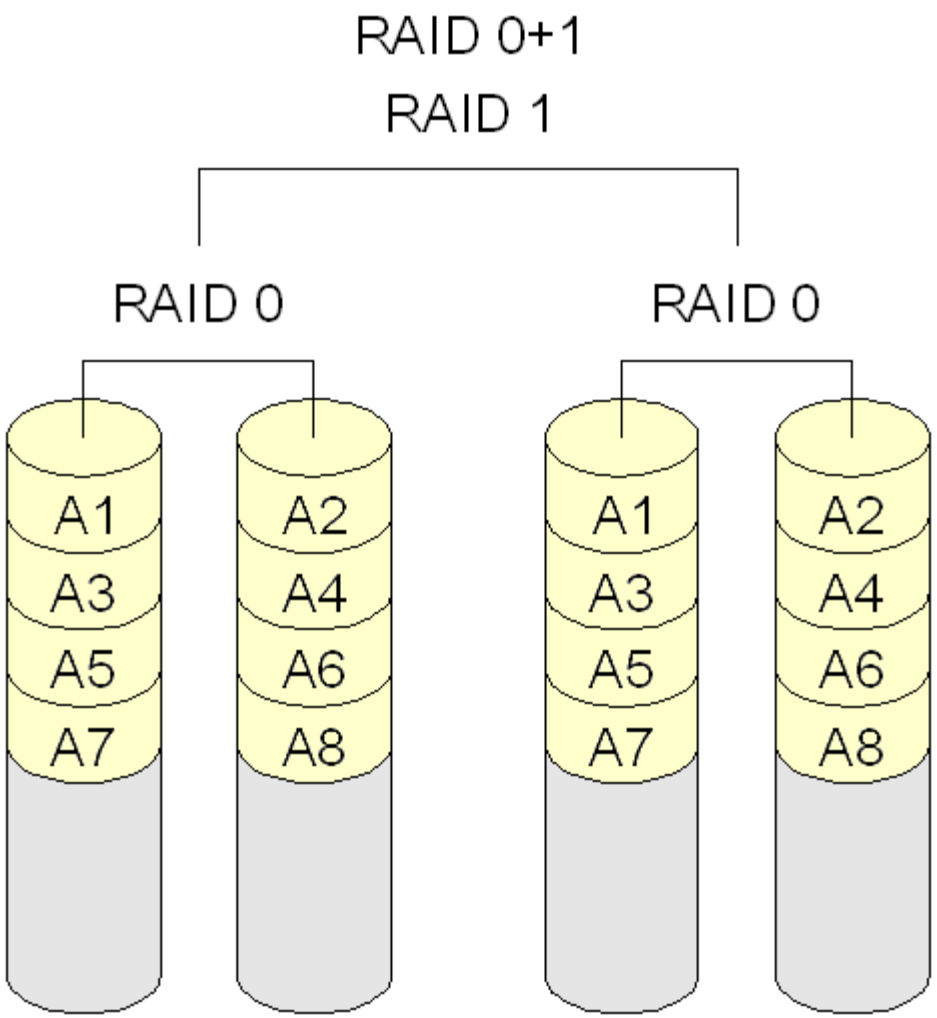
RAID 6 (!!)

- RAID 5 + navíc další paritní disk
- odolné proti výpadku dvou disků
- rychlost čtení srovnatelná s RAID 5
- zápis pomalejší

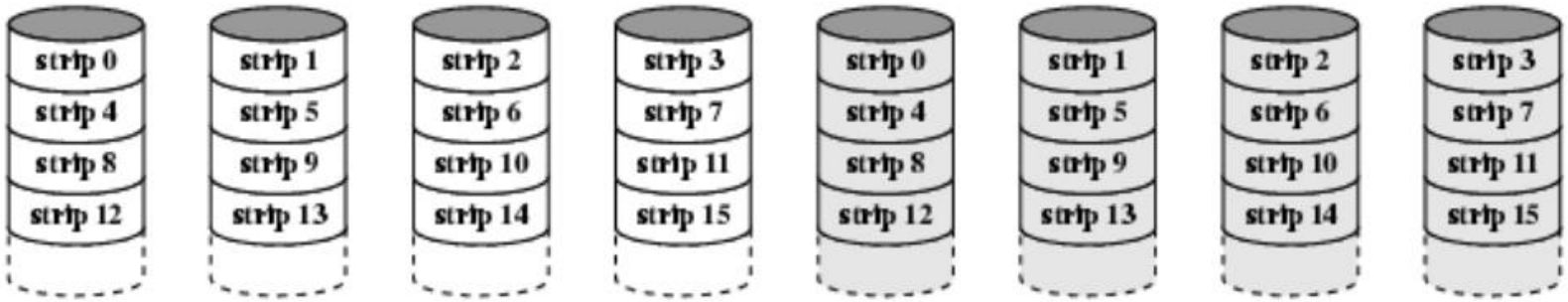

RAID 6
A1
B1
C1
Dp
Eq
A2
B2
Cp
Dq
E1
A3
Bp
Cq
D1
E2
Ap
Bq
C2
D2
E3
Aq
B3
C3
D3
Ep
Disk 0
Disk 1
Disk 2
Disk 3
Disk 4

RAID 10

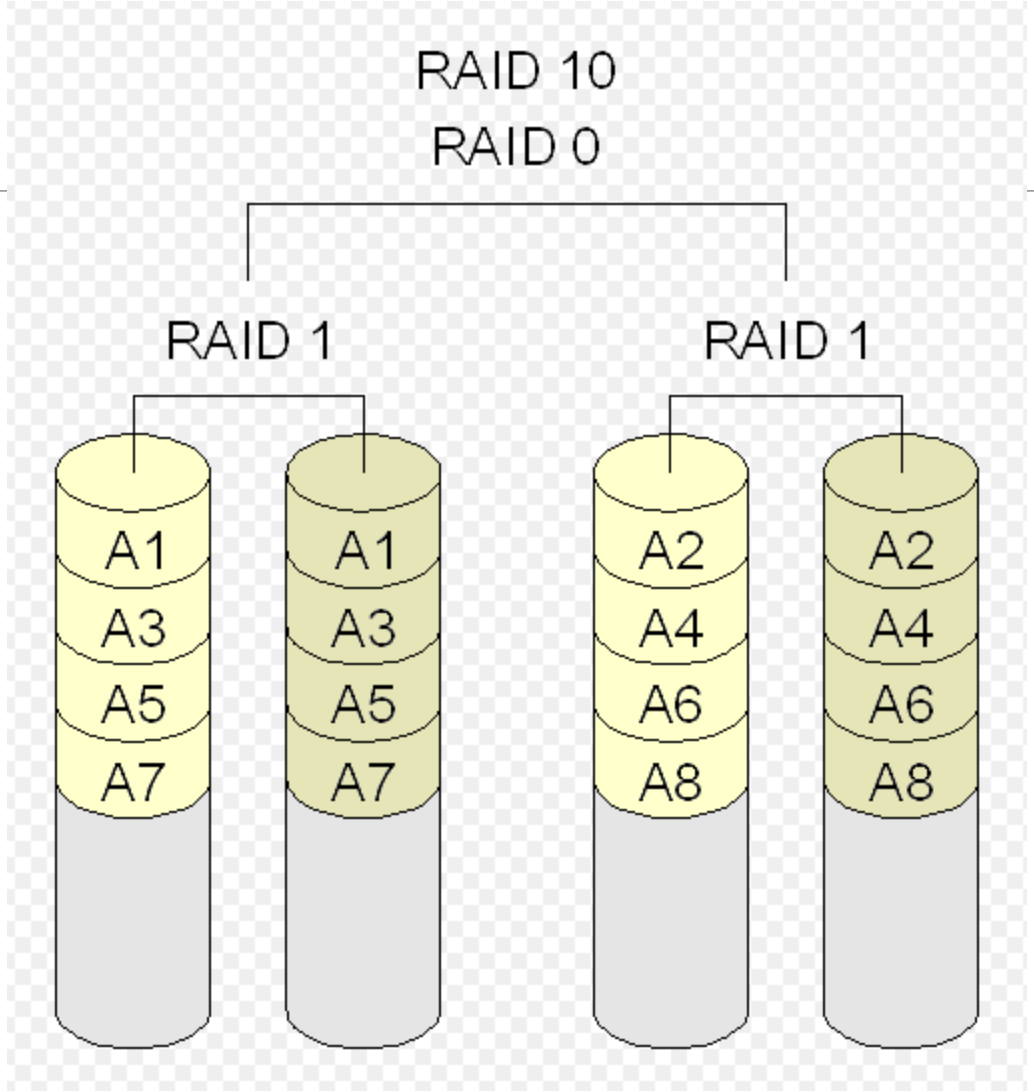
- kombinace RAID 1 (zrcadlo) a RAID 0 (stripe)
  - První číslo „spodní technologie“ další číslo „horní technologie“
  - Často se značí i RAID 1+0
  - RAID 0+1 znamená RAID 0 a nad ním RAID 1
- min. počet disků 4
- režie 100% diskové kapacity navíc
- vyšší výkon v bezpečných typech polích
- podstatně rychlejší než RAID 5, při zápisu
- odolnost proti ztrátě až 50% disků x RAID 5


RAID 0+1
RAID 1
RAID 0
RAID 0
A1
A3
A5
A7
A2
A4
A6
A8
A1
A3
A5
A7
A2
A4
A6
A8

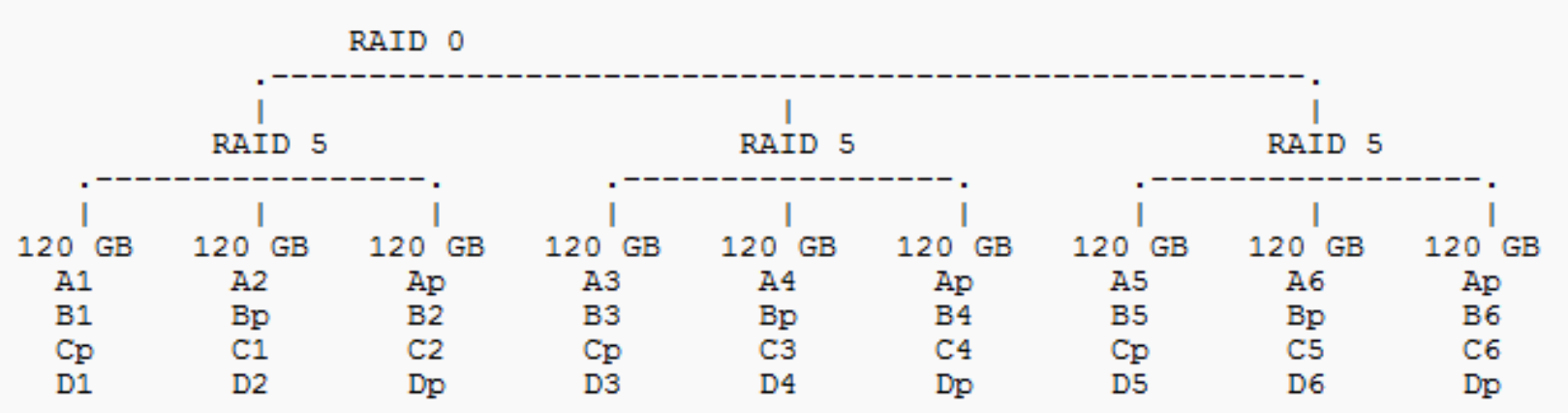
Příklad: Zrcadlení s využitím 8 disků



Zdroj: http://www.fit.vutbr.cz/study/courses/ITP/public/itp07/raid01.pdf


RAID 10
RAID 0
RAID 1
RAID 1
A1
A3
A5
A7
A1
A3
A5
A7
A2
A4
A6
A8
A2
A4
A6
A8

RAID 50

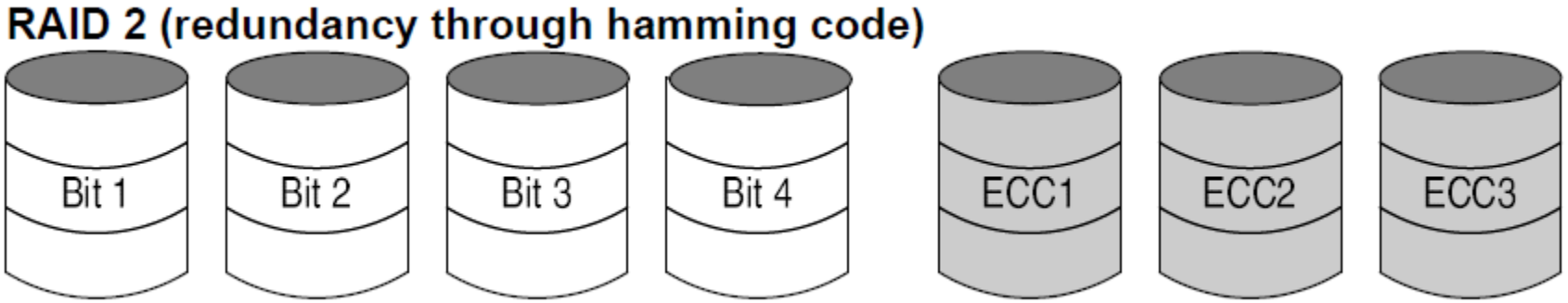


Zdroj obrázků a doporučená literatura:
http://cs.wikipedia.org/wiki/Raid

RAID 2

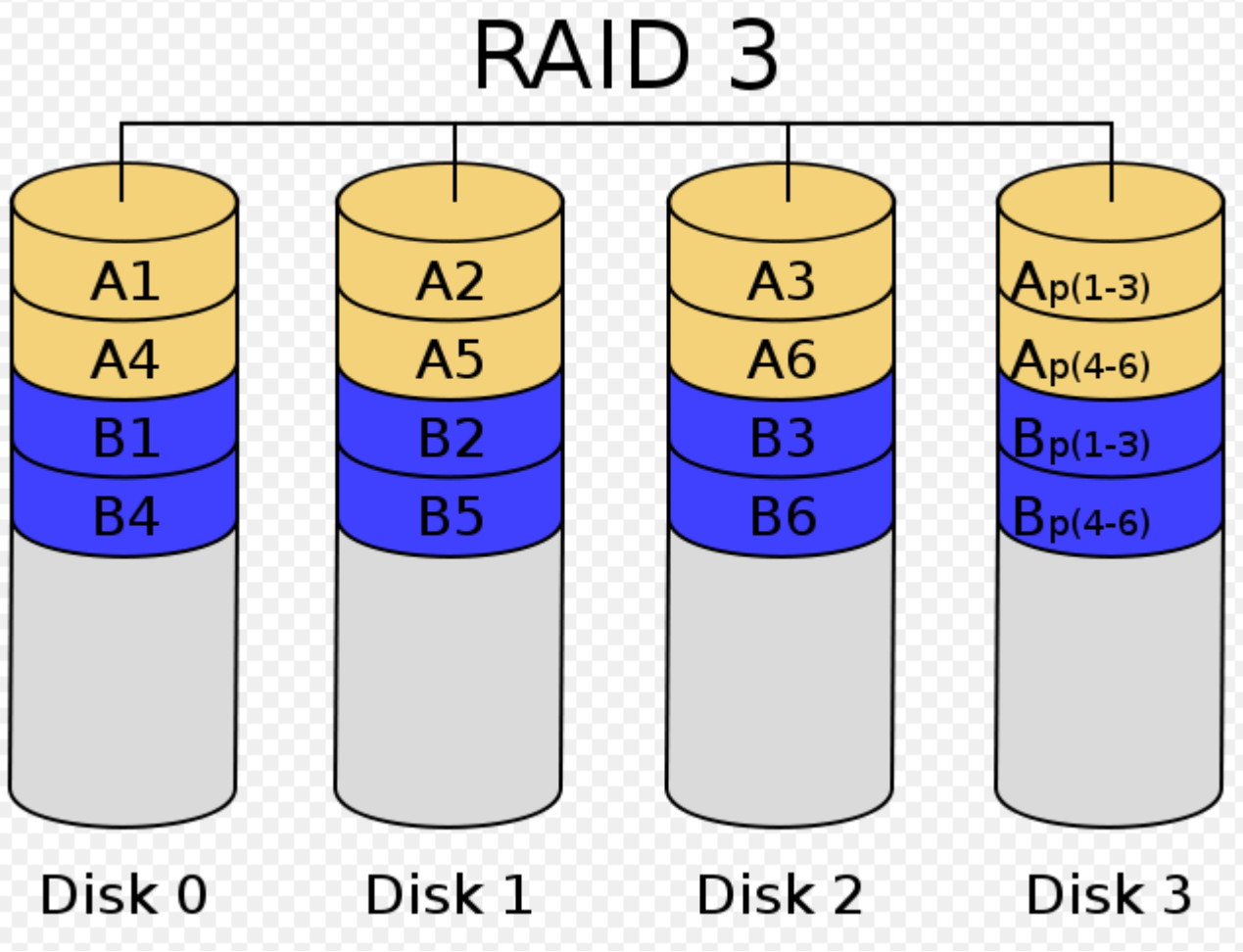
- Data po bitech stripována mezi jednotlivé disky
- Rotačně synchronizované disky
  (na všech discích jsou hlavy ve stejné pozici z hlediska otáčení disků a jejich vystavení)
- Zabezpečení Hammingovým kódem
- Např. 7 disků
  - 4 bity datové
  - 3 bity Hammingův kód

RAID 2


RAID 2 (redundancy through hamming code)
Bit 1
Bit 2
Bit 3
Bit 4
ECC1
ECC2
ECC3

RAID 3

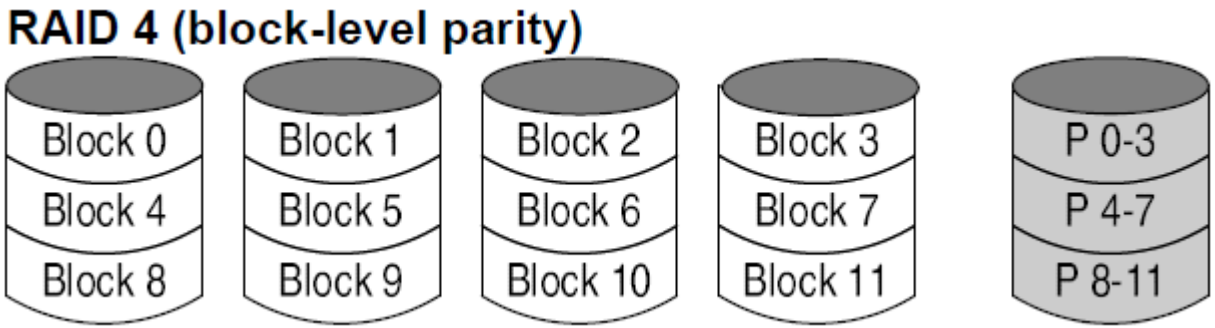
- N+1 disků, bitové prokládání
- Rotačně synchronizované disky
- Na N data, poslední disk XOR parita
- Jen 1 disk navíc
- Paritní disk vytížen při zápisu na libovolný disk – vyšší opotřebení

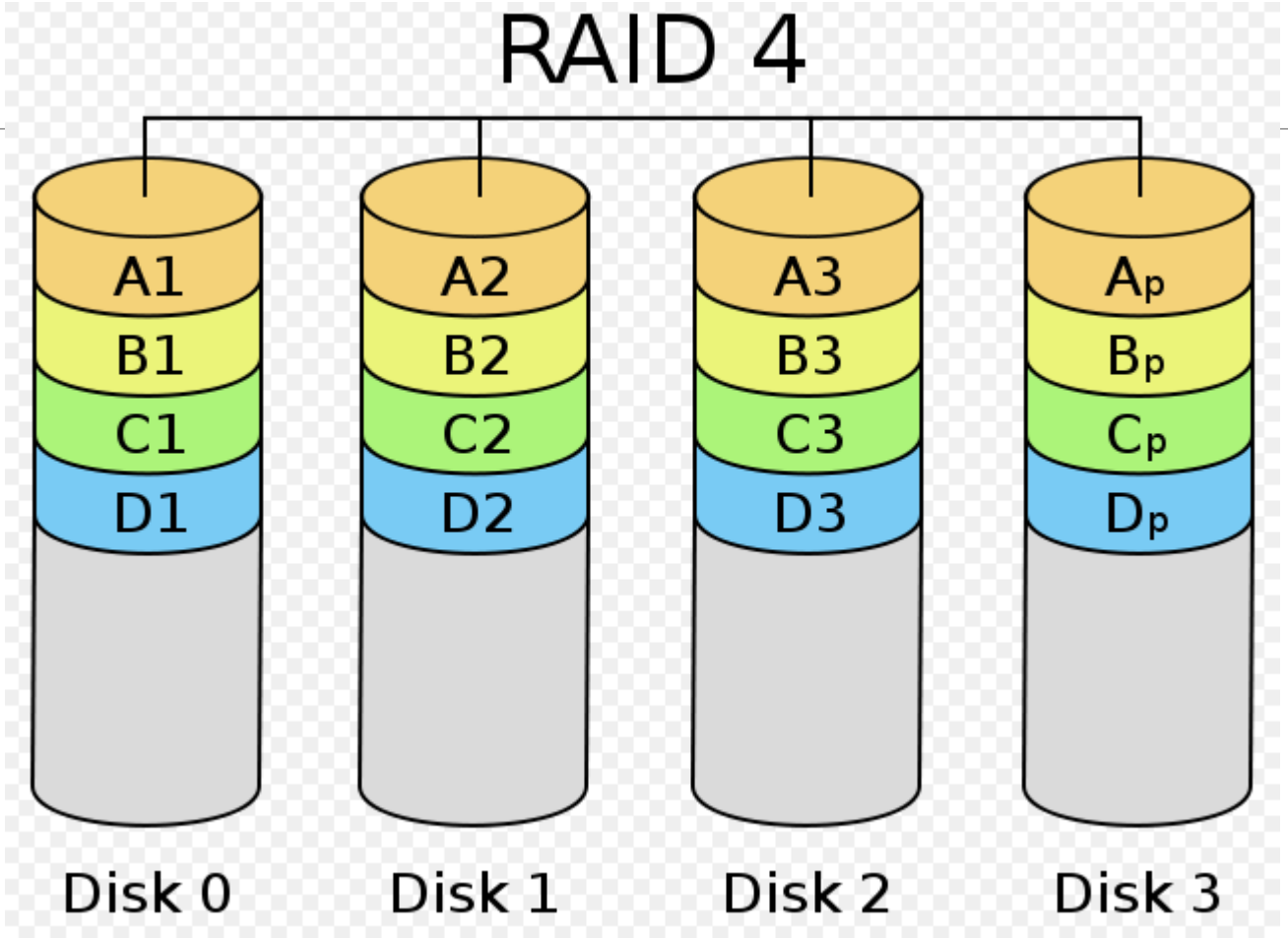

RAID 3
A1
A4
B1
B4
A2
A5
B2
B5
A3
A6
B3
B6
Ap(1-3)
Ap(4-6)
Bp(1-3)
Bp(4-6)
Disk 0
Disk 1
Disk 2
Disk 3

RAID 4

Disky stripovány po blocích, ne po bitech
Každý disk je nezávislý

Parita je opět po blocích




RAID 4
A1
B1
C1
D1
A2
B2
C2
D2
A3
B3
C3
D3
Ap
Bp
Cp
Dp
Disk 0
Disk 1
Disk 2
Disk 3

Problém rekonstrukce pole

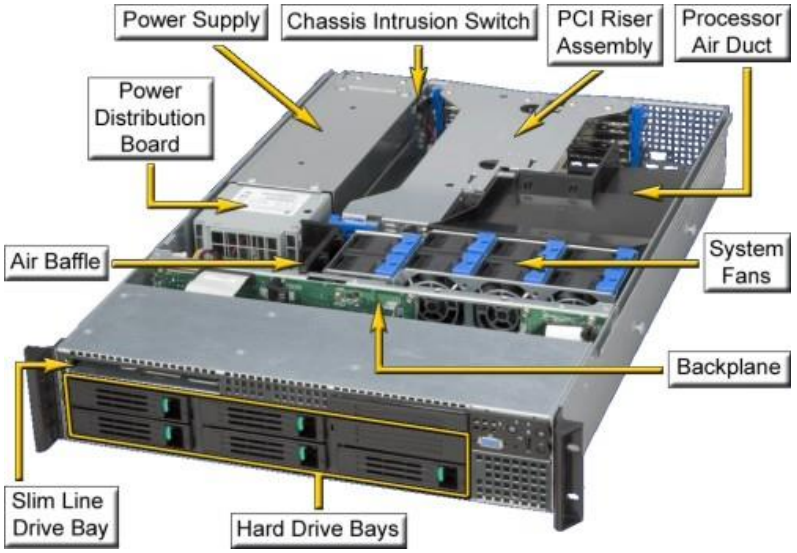
- rekonstrukce pole při výpadku – trvá dlouho
  - po dobu rekonstrukce není pole chráněno proti výpadku dalšího disku
  - náročná činnost – může se objevit další chyba, řadič disk odpojí a … přijdeme o data…

HOT **SPARE** DISK

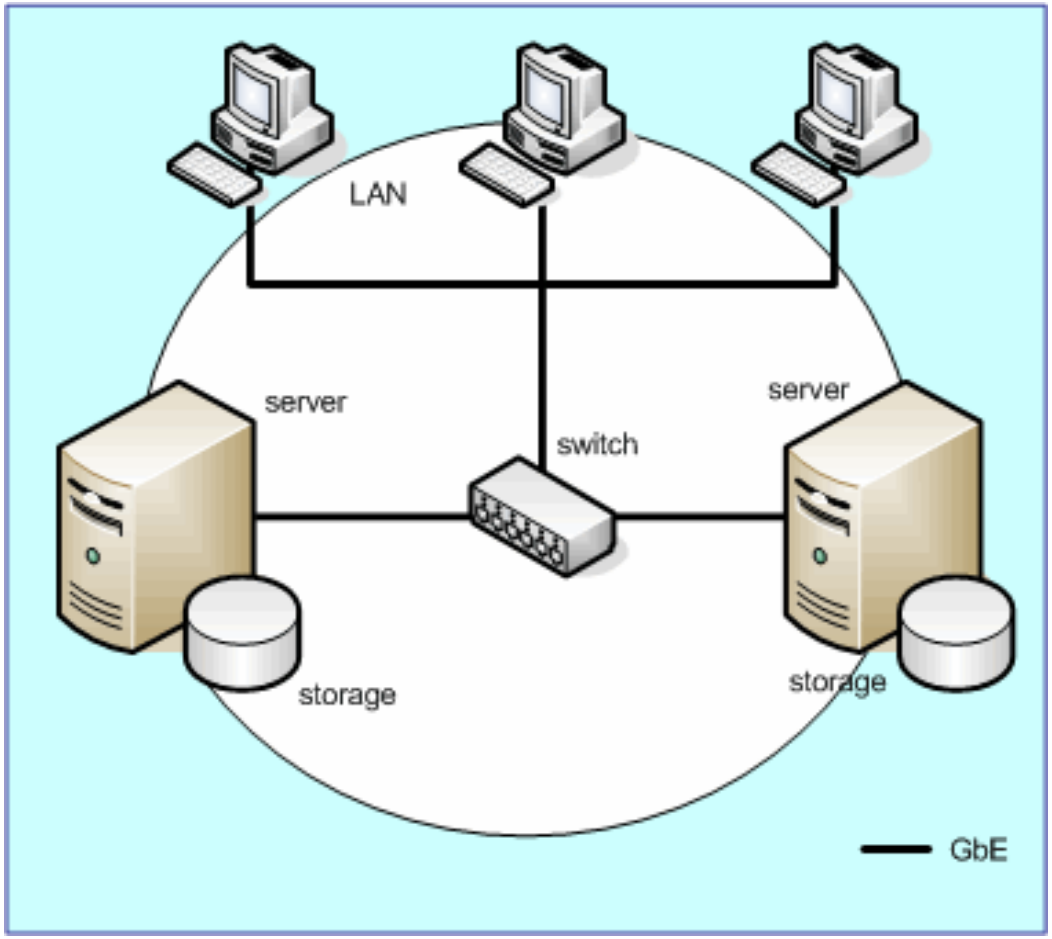
- záložní disk okamžitě připravený k nahrazení vadného disku
- při výpadku disku v poli automaticky aktivován hot spare disk a dopočítána data
- minimalizace rizika (časové okno)
  - Pole je degradované a je třeba vyměnit disk
  - Administrátor nemusí být poblíž
- hot spare disk lze sdílet mezi více RAIDy (někdy)

HOT **PLUG**

- Snadná výměna disku za běhu systému
- Není třeba vypnout server pro výměnu disku
- „šuplík z přední strany serveru“



DAS (Directy Attached Storage)



Každý sever má své úložiště.

Zatímco jedno může být poloprázdné a nevyužité,

druhému serveru může docházet místo.

SAN

Storage Area Network

- oddělení storage a serverů

- Propojení
  - Různé technologie
  - Fibre Channel – optický kabel
  - Rychlý Ethernet

- např. clustery, společná datová oblast
- high availability solution

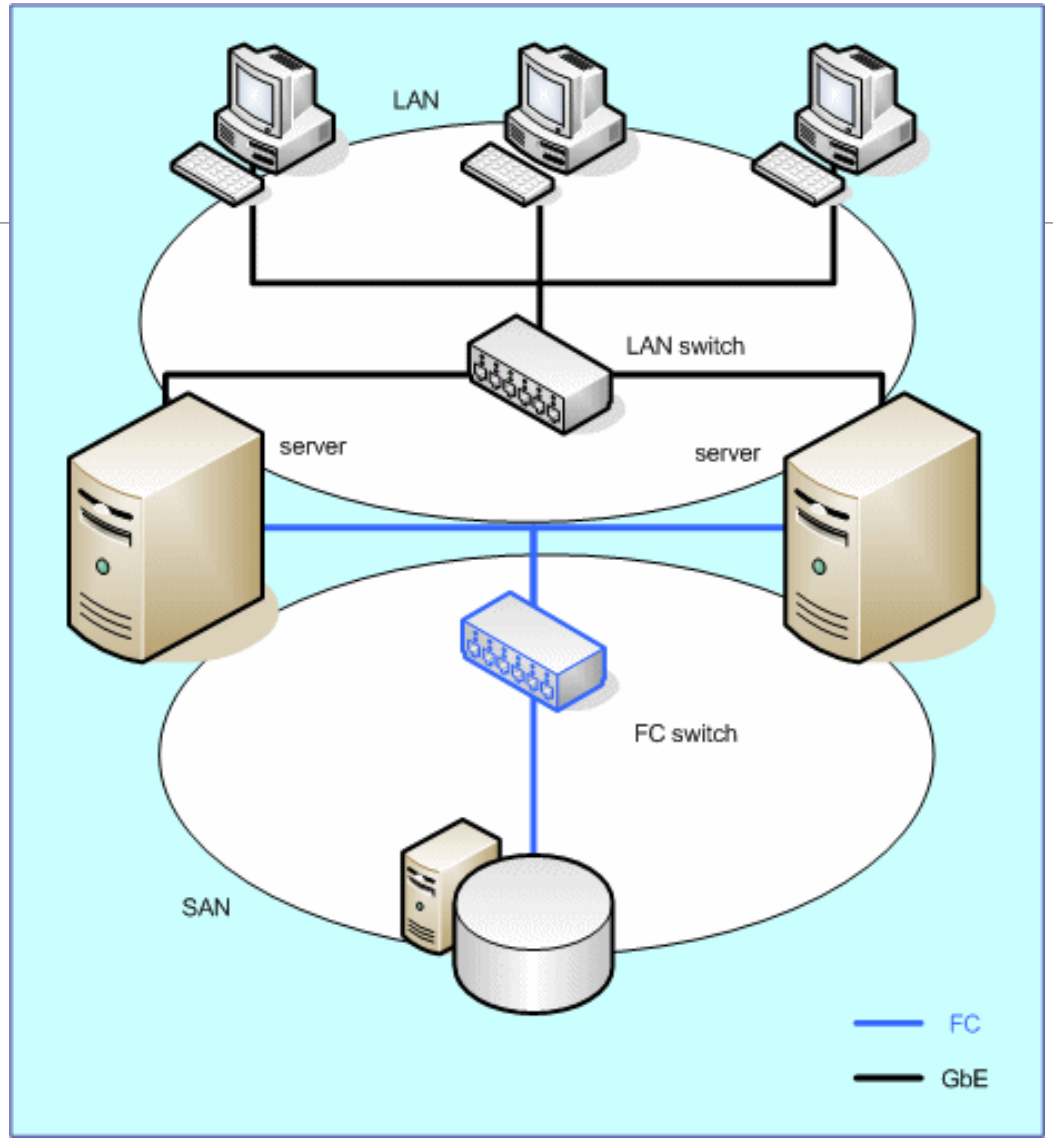
SAN

Storaga Area Network

Servery využívají společné úložiště.

Můžou mít např. dvě sítové karty – jednu pro komunikaci s úložištěm, druhou pro komunikaci s klienty.

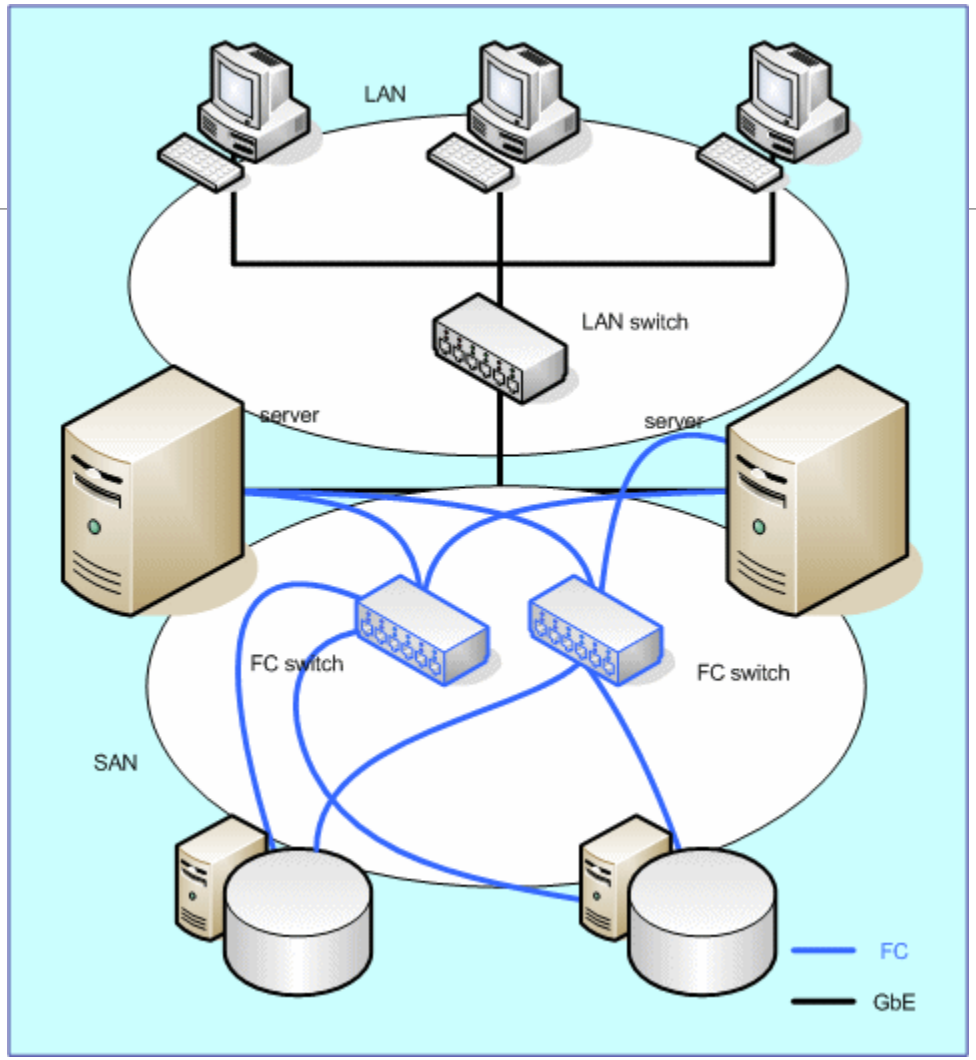
Zde na obrázku se pro komunikaci s úložištěm používá FiberChannel.



SAN

Zdvojení kritických prvků.

Může odejít jeden FC Switch, může odejít kabel k němu vedoucí, může odejít jedno z úložišť.



Použité zdroje

http://www.vahal.cz/cz/podpora/technicke-okenko/ukladani-dat-iscsi.html

(obrázky, v současné době stránky odkazu přestrukturované)