



Zadání bakalářské práce

Název:	Záloha dat na optické disky - implemtace technologie RAID
Student:	Petr Jiříček
Vedoucí:	Ing. Viktor Černý
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

Vytvořte backend pro aplikaci na spolehlivou zálohu dat za pomoci optických datových nosičů a redundance dat. Analyzujte způsoby jak lze využít redundanci dat na zvýšení spolehlivosti datové zálohy.

- Navrhněte vlastní způsob redundance dat, nebo adaptujte existující technologii RAID, které se primárně používá pro pevné disky.
- Dále proveďte analýzu spolehlivosti takto uložených dat (vycházejte z dostupných statistických údajů). Na základě analýzy vyberte nejvhodnější způsob ukládání dat, který nabídne vysokou úroveň spolehlivosti a zároveň je dostatečně uživatelsky přívětivý.
- Vytvořte knihovnu, která implementuje vybrané techniky a utilitu unixového stylu pro příkazový řádek, která umožní uživateli pracovat s vytvořenou knihovnou.
- Pro implementaci zvolte vhodnou technologii.
- Výsledná utilita/knihovna bude řádně zdokumentována, aby ji bylo možné rozšiřovat např. o grafické nadstavby a jiné typy zálohy nebo rekonstrukce dat.
- Výsledné řešení řádně otestujte.

Bakalářská práce

ZÁLOHA DAT NA OPTICKÉ DISKY - IMPLEMENTACE TECHNOLOGIE RAID

Petr Jiříček

Fakulta informačních technologií
Katedra softwarového inženýrství
Vedoucí: Ing. Viktor Černý
20. června 2022

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2022 Petr Jiříček. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení, je nezbytný souhlas autora.

Odkaz na tuto práci: Jiříček Petr. *Záloha dat na optické disky - implementace technologie RAID*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2022.

Obsah

Poděkování	vii
Prohlášení	viii
Abstrakt	ix
Seznam zkratk a pojmů	x
Úvod	1
1 Teoretická část	3
1.1 Možnosti zálohování	3
1.1.1 Pevný disk HDD	3
1.1.2 Pevný disk SSD	3
1.1.3 Magnetická páska	4
1.1.4 Optická média	4
1.1.5 Požadavky na zálohování ve firemním prostředí	4
1.1.6 Požadavky na zálohování v domácím prostředí	5
1.1.7 Porovnání výhodnosti jednotlivých technologií	5
1.2 Optická média	6
1.2.1 CD	6
1.2.2 DVD	6
1.2.3 Blu-ray	6
1.2.4 Srovnání CD, DVD a Blu-ray	8
1.3 Specifika zálohování na optické disky	8
1.3.1 Udržení dlouhé životnosti disků	8
1.4 Rešerše podobných produktů	9
1.4.1 IBM - zálohování na optické disky	9
1.4.2 Nástroj dd	9
1.4.3 Sada nástrojů cdrtools	9
1.4.4 Nástroj udffsck	10
1.4.5 Analýza disků a metadat pod OS Linux	10
1.4.6 Rozšířené zálohování pro OS Windows	11
1.5 Pole RAID a nástroj mdadm	11
1.5.1 Analýza technologie RAID	11
1.5.2 Analýza využití postupu pro účely zálohování dat na optická média	15
1.5.3 Binární kopie optického média	15
1.5.4 Tvorba pole RAID 4 ve Windows WSL / Linux	15
1.6 Souborové systémy optických disků	17
1.6.1 ISO 9660	17
1.6.2 Joliet	17
1.6.3 UDF	18
1.6.4 Srovnání souborových systémů	18
1.6.5 Formát souborů ISO	18

1.7	Testování čitelnosti disků	18
2	Praktická část	19
2.1	Model procesů	19
2.2	Funkční a nefunkční požadavky	19
2.3	Model případů užití	21
2.3.1	Případy užití	21
2.4	Doménový model	21
2.5	Programovací jazyk a implementace	25
2.5.1	Algoritmus tvorby paritního souboru	25
2.5.2	Algoritmus obnovení chybějícího disku	26
2.5.3	Algoritmus pro opravení disku s vadnými sektory	26
2.5.4	Sestavení projektu dsb	27
2.5.5	Dokumentace generovaná nástrojem doxygen	28
3	Testování	29
3.1	Nápověda nástroje dsb	29
3.2	Vytvoření zálohy	30
3.3	Zobrazení informací v paritním souboru	30
3.4	Otestování konzistence disků	30
3.5	Obnovení ztraceného disku	31
3.6	Obnovení poškozeného disku	32
4	Budoucí rozvoj	33
5	Závěr	35
A	Ceny médií v obchodech	39
	Obsah přiloženého média	41

Seznam obrázků

1.1	Graf závislosti ceny na době uložení dat do 100 let	7
1.2	Graf závislosti ceny na době uložení dat do 1000 let	7
1.3	Rozvržení RAID 0	12
1.4	Rozvržení RAID 1	12
1.5	Rozvržení RAID 3	13
1.6	Rozvržení RAID 4	13
1.7	Rozvržení RAID 5	14
1.8	Rozvržení RAID 6	14
2.1	Diagram modelu procesu	20
2.2	Diagram modelu funkčních požadavků	22
2.3	Diagram modelu případů užití	22
2.4	Diagram doménového modelu	23
A.1	Ceny uložení 1 TB za rok pro různé <i>optické disky</i>	40

Seznam tabulek

1.1	Srovnání klíčových vlastností <i>optických disků</i>	8
1.2	Srovnání souborových systémů	18
2.1	Pokrytí funkčních požadavků případy užití	21

Seznam výpisů kódu

1.1	Vytvoření obrazu disku ve formátu iso	10
1.2	Vypálení obrazu disku	10
1.3	Vymazání <i>optického disku</i>	10
1.4	Binární kopie optického média	15
1.5	Vytvoření 1 prázdného disku	15
1.6	Vytvoření dalších 2 disků nakopírováním 1. disku	15
1.7	Připojení všech disků k operačnímu systému	16

1.8	Vytvoření diskového pole RAID 4 nad připojenými disky	16
1.9	Zobrazení stručné informace o nově vytvořeném poli	16
1.10	Zobrazení detailní informace o poli	16
1.11	Zobrazení informace o blokových zařízeních	16
1.12	Naformátování disku na systém souborů NTFS	16
1.13	Vytvoření adresáře pro bod připojení	16
1.14	Připojení svazku	16
1.15	Pokusné zapsání dat na disk	17
1.16	Odpojení svazku	17
1.17	Zastavení diskového pole	17
1.18	Zobrazení nápovědy pro použité příkazy mdadm	17
2.1	Hlavička skupiny disků	21
2.2	Hlavička každého disku ve skupině	24
2.3	Příkaz pro vytvoření <i>paritního souboru</i> ve formátu <i>DSB</i>	25
2.4	Příkaz obnovení chybějícího disku <code>disk2.iso</code>	26
2.5	Příkaz pro opravení disku <code>disk2.iso</code>	27
2.6	Obsah souboru <code>invsect.txt</code>	27
2.7	Nápověda nástroje <code>make</code> pro <code>dsb</code>	27
3.1	Nápověda nástroje <code>dsb</code>	29
3.2	Tvorba <i>paritního souboru</i> <code>parity.dsb</code>	30
3.3	Zobrazení informace souboru <code>parity.dsb</code>	30
3.4	Testování konzistence <i>záložních disků</i>	31
3.5	Testování konzistence disků našlo chybu	31
3.6	Obnovení ztraceného disku	31
3.7	Chyba při obnovování ztraceného disku	31
3.8	Obnovení částečně poškozeného disku	32
3.9	Chyba při obnovování porušeného disku	32

Rád bych poděkoval Ing. Viktoru Černému za odborné vedení této práce, srovnání mých myšlenek tím správným směrem, a dovedení do zdárného konce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 20. června 2022

.....

Abstrakt

Tato práce se zabývá spolehlivou zálohou dat na dlouhé časové období 50 let i více s použitím optických disků. Tato technologie nabízí zajímavý poměr mezi datovou stabilitou a cenou, takže by mohla být atraktivní pro domácí uživatele. Pro zvýšení spolehlivosti zálohy byl vytvořen nástroj, který umí připravit soubor s pomocnými daty pro obnovu ztraceného nebo poškozeného disku.

Klíčová slova nástroj příkazové řádky, záloha dat, optický disk, RAID, C/C++

Abstract

This work deals with reliable data backup over a long period of time of 50 years or more using optical disks. This technology offers an interesting ratio between data stability and price, so it could be attractive for home users. To increase the reliability of the backup, a tool has been created that can prepare a file with auxiliary data for recovery of lost or damaged disk.

Keywords command line tool, backup reliability, optical disc, RAID, C/C++

Seznam zkratek a pojmů

CD	Kompaktní disk (Compact disk)
CDFS	Souborový systém kompaktních disků (Compact Disc File System)
DVD	Digitální video disk (Digital Video Disc)
ECMA	Evropská asociace výrobců počítačů (European Computer Manufacturers Association)
EFM	Kódování osm do čtrnácti (Eight-to-Fourteen Modulation)
GUID / UUID	Univerzální unikátní identifikátor (Global / Universally Unique Identifier)
HDD	Pevný disk (Hard Disk Drive)
ISO/IEC	Mezinárodní organizace pro normalizaci a Mezinárodní elektrotechnická komise (International Organization for Standardization and the International Electrotechnical Commission)
OOP	Objektově orientované programování (Object-oriented programming)
OSTA	Organizace sdružující výrobce a další firmy pracující v oblasti optických paměťových médií (Optical Storage Technology Association).
POSIX	Přenosné rozhraní operačního systému (Portable Operating System Interface)
RAID	Vícenásobné pole nezávislých disků (Redundant Array of Independent Disks)
SSD	Polovodičový disk (Solid State Drive)
UDF	Univerzální formát disku (Universal Disk Format)

<i>Optický disk</i>	Optický disk je ploché, obvykle kruhové médium, na které se zaznamenávají binární data jako prohlubně nebo výstupky.
<i>Paritní soubor</i>	Soubor ve formátu <i>DSB</i> , se kterým mohou pracovat pomocí nástroje <i>dsb</i> .
<i>Záložní soubor</i>	Soubor, který obsahuje vlastní zálohované data. V této práci používám formát ISO, což je bitová kopie dat z optického disku. Ale obecně lze použít libovolný formát dat.

Úvod

Jistě jste někdy řešili problém jednoduchého a spolehlivého zálohování dat, ptali jste se, na která média zálohovat. Možná již máte nějaké starší zálohy hotové a chtěli byste jen zvýšit spolehlivost takových záloh. Možná máte strach zálohovat do cloudu kvůli bezpečnosti, bojíte se zneužití, nebo nechcete utrácet mnoho peněz za pronájem kapacity v cloudu.

Z těchto důvodů vytvořím nástroj pro zálohování. Soustředím se zde na zálohování dat na optická média, která jsou podle některých zdrojů nejlevnější a nejdostupnější variantou zálohování pro dlouhodobé uložení. Prozkoumám existující řešení včetně základního nástroje pro práci s RAID mdadm. Popíši a zdůvodním, proč jsem zvolil algoritmus pro vytvoření zálohy, jaký jsem zvolil. Také informuji o spolehlivosti obnovy dat pro různé algoritmy zálohování.

Nástroj bude jednoduchý na použití, nebude mít žádné závislosti. Nebude třeba instalace ani žádné složité nastavování. Stačí ho jen stáhnout a začít používat. Veškeré nastavení lze zrealizovat pomocí argumentů na příkazové řádce, případně editací textového souboru libovolným editorem. Veškerá dokumentace je přístupná společně s nástrojem ve formátu kompatibilním s ostatními nástroji příkazové řádky.

V nástroji implementuji nízkourovňové operace s obrazy optických disků. V budoucnosti se plánuje použití tohoto nástroje v komplexnějším řešení celého zálohování, které ale není součástí této práce.

Cíl

V této práci jsem si vytyčil za cíl usnadnit zálohování dat (souborů nebo obrazů disku) na optická média tak, aby byla zaručena vysoká spolehlivost obnovy i po velmi dlouhé době. Důraz kladu na jednoduchost řešení a uživatelskou přívětivost. Proto v této práci vytvořím nástroj použitelný z příkazového řádku, který je plně kompatibilní s ostatními nástroji příkazového řádku a všechny vstupní argumenty je možné zadat buď jako argument nástroje nebo lze vytvořit pomocí libovolného textového editoru soubor s potřebnými údaji a ten předat jako vstupní argument nástroji.

V teoretické části práce provedu rešerši existujících řešení a porovnáím je s požadavky mojí práce. Prozkoumám různé způsoby zvýšení spolehlivosti uložených dat - speciálně se budu soustředit na technologii RAID. Zjistím aktuální ceny médií a vyberu nejvhodnější a nejlevnější způsob uložení s ohledem na délku zálohování dat. Dále vypočítám pravděpodobnost obnovení dat na základě odhadů poruchovosti médií a naleznu neoptimálnější způsob uložení dat v závislosti na ceně médií a pravděpodobnosti obnovení dat. Posoudím také možnost využití algoritmů pro RAID a jeho vhodnost pro účely nástroje.

V praktické části zdůvodním, který algoritmus ukládání dat jsem zvolil a také jeho podrobný popis. Také popíši všechny argumenty nástroje včetně podrobných vysvětlení jejich účelu. V příloze jsou dostupné kompletní zdrojové kódy.

Na závěr zhodnotím vytvořený nástroj z hlediska uživatelské přívětivosti a optimálnosti uložení vstupních dat. Také uvedu další možnosti rozvoje nástroje.

Teoretická část

V teoretické části nejprve prozkoumám nejpoužívanější možnosti zálohování a porovnáám je se zálohováním na optická média. Dále popíši pravidla, které je vhodné dodržovat pro dosažení maximální životnosti optických disků.

1.1 Možnosti zálohování

V této kapitole nejprve seznámím s možnostmi zálohování a dále srovnám tyto metody mezi sebou a ukáži, proč je *optický disk* jednou z nevhodnějších možností pro domácí uživatele na dlouhodobé a spolehlivé zálohování dat. Nakonec ještě popíši způsob, který jsem využil pro další navýšení spolehlivosti zálohy vytvořením speciálního disku, který pomůže s případnou obnovou ztraceného nebo jinak poškozeného disku.

1.1.1 Pevný disk HDD

Výhoda je, hlavně pro krátkodobé ukládání dat, nízká cena cca. 500 Kč/TB [Příloha A]. Pokud vezmu v úvahu také délku uložení, komplikovanost a čas strávený při přepírovávání dat, cena již takto nízká není. Hlavní nevýhodu spatřuji v krátké životnosti a náchylnosti disku k mechanickému poškození [14]. I když nejsou samotné plotny poškozené, je velmi obtížné z nich přečíst data. Existují specializované firmy, které se zabývají obnovou dat z těchto disků, ale i tak je výsledek nejistý.

1.1.2 Pevný disk SSD

Hlavní výhody SSD oproti HDD jsou [18]:

- násobně větší rychlost čtení a zápisu dat,
- fragmentace má zanedbatelný vliv na rychlost přístupu k datům,
- protože neobsahuje pohyblivé mechanické části, tak větší odolnost proti otřesům a vibracím a tím i menší náchylnost k mechanickému poškození, nižší hluk,
- magnetické pole má jen malý dopad na paměť,
- nižší spotřeba energie.

Mezi nevýhody patří:

- Cena disku, i když poslední dobou se SSD zlevňují, tak stále ještě nedosahují kapacitně ani cenově na HDD.
- Nižší počet zápisů do stejného místa oproti HDD.

Zde bych rád upozornil na to, že ceny této technologie klesají a kapacita roste. Je tedy možné, že časem již nebude levnější technologie HDD, ale bude výhodnější použít SSD.

1.1.3 Magnetická páska

Zálohování na magnetické pásky v současnosti využívají hlavně firmy, které potřebují ukládat velké objemy dat, typicky databáze nebo multimediální soubory. Toto je také největší výhodou - objem, který lze na pásku uložit dosahuje až stovek TB. Jako další výhodu spatřuji cenu těchto pásek, která je příznivá. Například uložení 1 TB na 30 let vyjde na méně než 100 Kč [7]. Naproti tomu jako velkou nevýhodu shledávám cenu mechaniky, která umí na magnetickou pásku zapsat zálohovaná data. Ta dosahuje 55 000 Kč [6] což je o mnoho více než cena vypalovačky v případě optických médií, kterou mohu koupit od 700 Kč [22]. Proto tuto technologii nedoporučuji pro domácí využití.

1.1.4 Optická média

Největší výhodou spatřuji v ceně a dostupnosti médií, a rovněž zařízení pro vypalování jednotlivých disků. V některých scénářích použití může být výhodou také fakt, že *optické disky* jsou zapisovatelné jen jednou a poté je možné je pouze číst. Jedná se o ty případy, kdy je připojený počítač napadený ransomwarem, který se pokouší zašifrovat všechna připojená datová média.

Mezi hlavní nevýhody tohoto způsobu zálohování patří nízká kapacita. U Blu-ray disků je kapacita 25-100 GB, u DVD je to pouze 4,7 / 8,5 GB.

U DVD disků je nevýhodou i jejich slibovaná výdrž, která je pouze 10 let [14]. Blu-ray média jsou na tom lépe. U nich výdrž dosahuje až 50 let [14]. Problém kratší životnosti disků výrobci médií řeší zavedením technologie AZO, u které je slibovaná výdrž 100 let [12]. Další možností je využití technologie M-DISC, u které výrobce slibuje výdrž 1000 let.

V případě *optických disků* CD je vzhledem k jejich velikosti 700 MB cena za uložení 1 TB dat již příliš vysoká viz Obrázek A.1. Také manipulace s velkým počtem disků není pohodlná. Proto tyto disky nedoporučuji pro zálohování – je to již příliš stará technologie, která se dále nerozvíjí.

1.1.5 Požadavky na zálohování ve firemním prostředí

Shrunuji seznam nejdůležitějších firemních požadavků:

- Množství dat je velké
- Finanční prostředky vyčleněné pro zálohování jsou větší než v případě domácího použití.
- Firma požaduje po havárii rychlejší obnovu dat.
- Existuje zaměstnanec, který se přímo stará o zálohování.

Srovnávání jednotlivých možností zálohování se věnuje řada závěrečných prací studentů různých vysokých škol v České republice. Mnoho z nich se zabývá návrhem způsobu zálohování přímo pro konkrétní organizaci a za konkrétních podmínek [17]. Proto zde nebudu procházet jednotlivé možnosti více dopodrobna.

1.1.6 Požadavky na zálohování v domácím prostředí

Moje řešení se soustředí na domácí uživatele, kteří požadují vysokou spolehlivost obnovení zálohy. Typické požadavky pro zálohování takového uživatele jsou:

- Nižší objem dat než ve firemním prostředí.
- Nízká cena médií a zařízení, které se používá pro zápis na média.
- Jednoduchá správa disků. Jednoduchá tvorba *paritního souboru* pomocí nástroje příkazového řádku.

1.1.7 Porovnání výhodnosti jednotlivých technologií

Pro porovnání výhodnosti ukládání dat na dlouhou dobu používám cenu za uložení 1 TB v Kč na dobu, na kterou plánuji data uchovávat. Dále se zamýšlím nad tím, jak takové dlouhodobé uložení bude probíhat. Nejprve si uložím všechna data na médium a musím tedy zaplatit za toto médium. Poté, co uplyne výdrž média, tak data zkopíruji z jednoho média na nové a opět musím zaplatit za další médium. Takto to opakuji až do doby, na kterou plánuji zálohování dat.

Nejprve si zjistím cenu za uložení 1 TB v Kč. Protože se tato hodnota běžně u záznamových médií neuvádí, tak ji vypočítám pomocí vzorce:

$$cena_ulozeni_TB = \frac{cena_balicku_s_medii}{pocet_medii_v_balicku * velikost_jednoho_media_TB}$$

- *cena_ulozeni_TB* - výsledná cena za uložení 1 TB na dané médium
- *cena_balicku_s_medii* - cena v Kč celého balíčku s médii (např. 1 špindl se 100 kusy DVD stojí 509 Kč)
- *pocet_medii_v_balicku* - počet kusů médií v jednom balíčku (např. 100 ks DVD na jednom špindlu)
- *velikost_jednoho_media_TB* - velikost jednoho média v TB (např. velikost jednoho DVD je 4,7 GB = 0,0047 TB)

Dále potřebuji vypočítat cenu v Kč za uložení 1 TB na L let:

$$cena_ulozeni_TB_na_L_let = \left[\frac{L}{vydrz_let} \right] * cena_ulozeni_TB$$

- *vydrz_let* - výdrž jednoho média v letech
- *cena_ulozeni_TB_za_rok* - výsledná cena za uložení 1 TB za rok v Kč

Seznam všech potenciálně vhodných *optických disků* včetně celkové ceny a ceny za uložení 1 TB/rok v Kč jsem připravil v tabulce A.1. Nyní mohu zvolit vhodného zástupce z každé skupiny médií, které uvažuji použít pro zálohování dat. To bude ten, který má nejnižší cenu za zálohování 1 TB. Pro názornější vyjádření nárůstu ceny v závislosti na čase, na který data ukládám, takto spočítané hodnoty zobrazím do grafu na obrázku 1.1. Delší časové období znázorňuji v dalším grafu viz Obrázek 1.2. Na něm je dobře vidět, jak se zvýhodňují média s životností až 1000 let v případě M-DISC. Nicméně cena této technologie je lepší až od 200 let, což je mnohem více, než na jako dlouhou dobu cílím já. Mohu tedy říct, že pro ukládání záloh jsou za současných podmínek nejvýhodnější obyčejná média Blu-ray.

Z chování jednotlivých skupin médií vidím, že pro uložení na délku života člověka (50 let) je nejvýhodnější použití jednoduchých Blu-ray disků s cenou 663 Kč za TB. Oproti tomu technologie

M-DISC, použitá na Blu-ray disku tolik výhodná není. Cena, která je srovnatelná s obyčejným Blu-ray diskem nebo DVD s vrstvou AZO, je až od více než 100 let a to je mnohem více než délka lidského života.

Některé skupiny medií již nedoporučuji vůbec používat, protože jejich cena je příliš velká ve srovnání s ostatními médii. Jedná se hlavně o CD, které jsou opravdu příliš malé a cena příliš vysoká. U obyčejných DVD je cena výhodnější, ale stále nedosahuje hodnot těch nejvýhodnějších typů medií. Nedoporučuji ani pevné disky hlavně kvůli jejich krátké životnosti 5 let. Mají sice velkou kapacitu, ale bylo by s nimi dost práce při překopírovávání dat z disku na disk po 5 letech.

V této práci se budu zabývat nejvíce zálohováním na optická média typu Blu-ray nebo dvd s vrstvou AZO, která se vyznačují nejlepší cenou pro naše požadavky výdrže na 50 let.

1.2 Optická média

V této práci se zabývám hlavně optickými médii, proto o nich zmíním některé základní informace. Také se zmíním o pracích, které přináší další podrobnosti.

1.2.1 CD

V roce 1984 firmy Sony a Philips představily nový typ formátu *optických disků* – Compact Disc Read-Only Memory (CD-ROM), který vycházel z formátu Audio CD a umožňoval ukládání počítačových dat o kapacitě nejčastěji 650–700 MB [11].

Data jsou uložena ve spirále, která začíná ve středu disku a tvoří ji posloupnost prohlubní a výstupků [16]. Prohlubeň má hloubku 120 nm, šířku 500 nm a délku 870-3180 nm. Data se čtou pomocí laseru s vlnovou délkou 780 nm v infračerveném spektru. Informace není uložena přímo pomocí výstupků nebo prohlubní, ale na přechodu mezi nimi a na délce souvislých oblastí. Logická jednička je uložena jako přechod mezi výstupkem a prohlubní nebo prohlubní a výstupkem. V opačném případě je na médium kódovaná logická nula. Z fyzikálního hlediska není možné mít dvě jedničky (dva přechody) hned vedle sebe. Proto se používá kódování EFM, které k zakódování 8 bitů využije 14 bitů, kde je také vyřešen problém s po sobě jdoucími jedničkami [4].

1.2.2 DVD

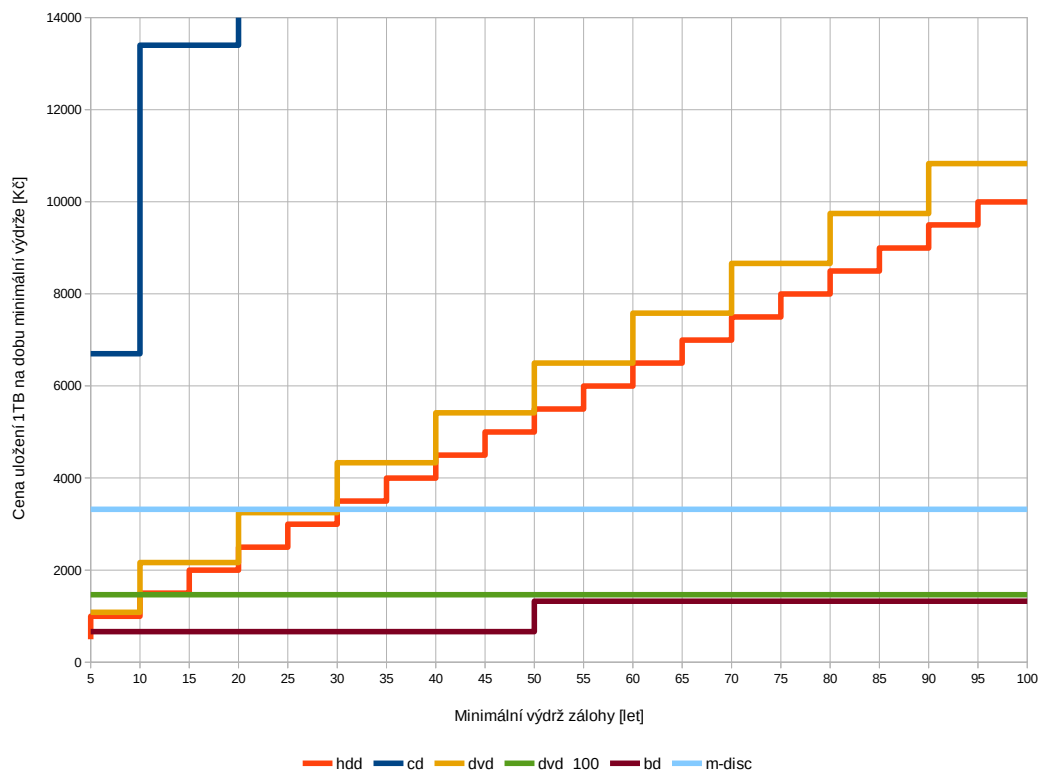
Protože kapacita CD-ROM byla nedostatečná pro ukládání filmů a videí, bylo v roce 1995 poprvé představeno médium DVD. Toto médium je navrženo jako zpětně kompatibilní, takže lze v DVD mechanice přečíst jak DVD média, tak i předchozí média CD.

DVD čerpá z návrhů, které byly vytvořeny pro CD. DVD má průměr 12 cm a tloušťku 1,2 mm stejnou jako disk CD-ROM. Aby se na DVD mohlo zapsat více dat, než na CD, je velikost prohlubní a výstupků menší. Protože je hustota dat vyšší, tak je nutné se chovat opatrněji při zacházení s diskem, abych se mechanicky nepoškodil. Na disk DVD lze uložit 4,7-17 GB dat. Z cenového přehledu viz Tabulka A.1 vidím, že nejvýhodnější poměr kapacita / cena mají média o kapacitě 4.7 GB.

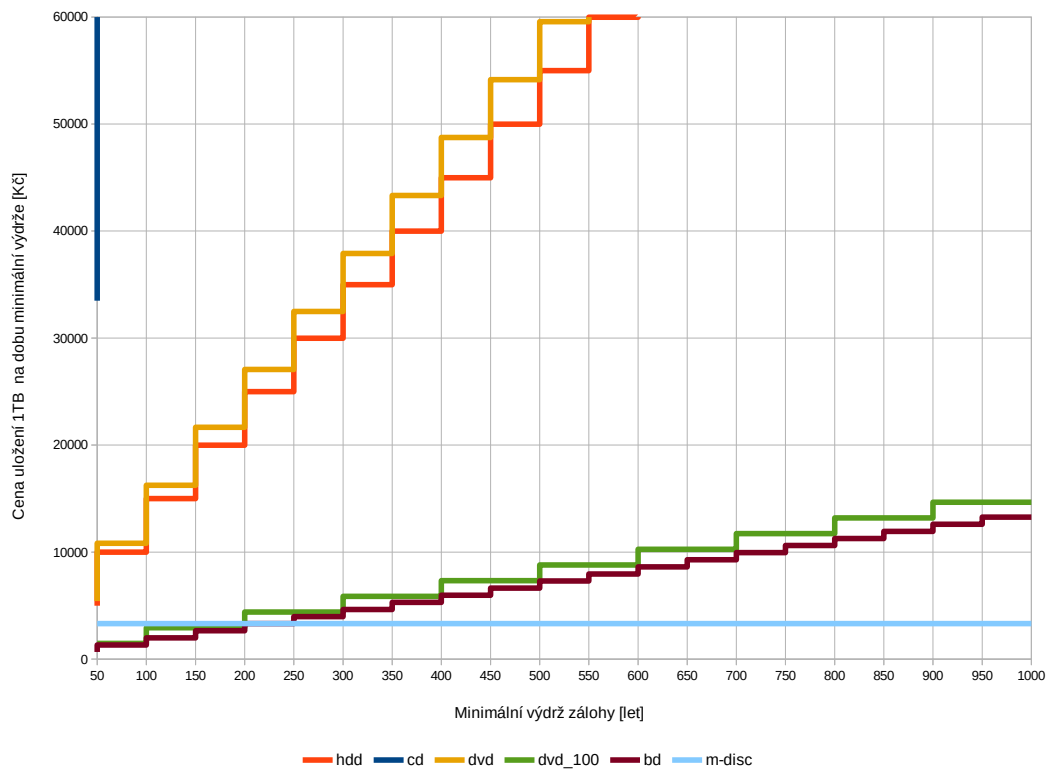
1.2.3 Blu-ray

Protože rozlišení videí, distribuovaných pomocí DVD disků rostlo, objevil se také požadavek na ještě větší kapacitu *optických disků*. Nejpoužívanějším formátem se nakonec ukázal formát Blu-ray. Název je z anglického blue ray (modrý paprsek), protože pro práci s těmito médii se používá světelný paprsek modré barvy. Toto médium je také zpětně kompatibilní a v Blu-ray mechanice lze číst i DVD a CD média. Mezi formáty medií patří:

- BD-ROM – disk je určen pouze pro čtení,



Obrázek 1.1 Graf závislosti ceny na době uložení dat do 100 let



Obrázek 1.2 Graf závislosti ceny na době uložení dat do 1000 let

	CD	DVD	Blu-ray
Úložná kapacita 1 vrstva	700 MB	4,7 GB	25 GB
Úložná kapacita 2 vrstvy	nemá	8,5 GB	50 GB
Vlnová délka laseru	780 nm	650 nm	405 nm
Numerická apertura	–	0,60	0,85
Průměr disku	120 mm	120 mm	120 mm
Tloušťka disku	1,2 mm	1,2 mm	1,2 mm
Ochranná vrstva	–	0,6 mm	0,1 mm
Rozteč stop	1,6 μm	0,74 μm	0,32 μm
Rychlost přenosu dat	1,23 MB/s	11,08 MB/s (1x)	36,0 MB/s (1x)
Rychlost přenosu dat (audio/video)	–	10,08 MB/s (<1x)	54,0 MB/s (1,5x)
Rozlišení videa (maximální)	–	720 x 480; 720 x 576	1920 x 1080
Rychlost přenosu videa (maximální)	–	9,8 MB/s	40,0 MB/s

■ **Tabulka 1.1** Srovnání klíčových vlastností *optických disků*

- BD-R – na disk je možné jednou zapsat a mnohokrát číst,
- BD-RW – na disk je možné zapsat vícekrát.

Stejně jako DVD a CD má i Blu-ray průměr 12 cm a tloušťku 1,2 mm. Na tato média je možné zapsat 25-100 GB dat. Pohledem do cenového přehledu A vidím, že nejvýhodnější cenu má médium o kapacitě 25 GB.

Také jsme si v předchozím textu ukázali, že pro dlouhodobé zálohování jsou média Blu-ray nejvýhodnější (viz kapitola 1.1.7).

1.2.4 Srovnání CD, DVD a Blu-ray

Všechny klíčové parametry jsem pro přehlednost zobrazil v tabulce 1.1.

1.3 Specifika zálohování na optické disky

Pokud pro zálohování dat používám *optické disky*, musím si uvědomit, že mám k dispozici pouze jeden zápis. Proto si nejprve připravím data pro konkrétní disk a poté je naráz vypálím.

1.3.1 Udržení dlouhé životnosti disků

Pro udržení co nejdelší životnosti *optických disků* dodržuji následující doporučení [5]:

- Vždy držím *optický disk* za vnější okraje, nebo středový otvor, abych se vyhnul otiskům prstů.
- Disk vkládám do obalů na suchém a chladném místě.
- Udržuji disk v čistotě.
- Nevystavuji disk nadměrné teplotě a vlhkosti nebo mrazu. Teploty by se měly pohybovat v intervalu 5–23 °C, relativní vlhkost by měla být přibližně 40–50%.
- Uchovávám disk mimo přímé sluneční nebo ultrafialové světlo.
- Nikdy se nesnažím disk ohýbat.
- Nepoužívám lepící etikety, které by mohly poškodit obsah disku.

- K popsání disku používám nerozpuštěnou fixu na štítku, pokud je na disku přítomen, nebo na zadní straně disku.

1.4 Rešerše podobných produktů

Při rešerši podobných produktů nenacházím žádný volně dostupný nástroj, který řeší funkční požadavky tak, jak jsou požadované v zadání. Ovšem nalézám a blíže zkoumám nástroj od firmy IBM, který řeší zálohování na optická média pro firmy. Prozkoumám i další nástroje, které lze použít při práci s optickými médii při jejich vytváření nebo při načítání souborů z poškozených médií. Také zde analyzuji práci, která umí vytvářet zálohy jednotlivých disků.

1.4.1 IBM - zálohování na optické disky

Toto řešení, je implementované v operačním systému *IBM i* je určeno pro podnikové řešení zálohování [8]. Mezi jeho hlavní výhody patří:

- Obsahuje speciální příkaz CPYOPT, který umí kopírovat soubory na optická média a příkaz DUPOPT, který umí klonovat disky.
- Umí také jednoduše zálohovat soubory, které se nevejdou na jedno optické médium tak, že je rozloží na více částí a ty uloží na více *optických disků*.
- Automaticky se stará o databázi všech uložených souborů.

Ale má současně i několik nevýhod:

- Řešení funguje výhradně pod operačním systémem *IBM i*, který není tolik rozšířený mezi mojí cílovou skupinou. Zde se soustředím na OS Windows a OS Linux.
- Neumí jednoduše vytvářet *paritní soubory*, které zvyšují šanci na správné a úplné přečtení dat.

Tento produkt jsem při rešerši neměl k dispozici, takže jsem nemohl otestovat všechny funkcionality, které jsou popsány v dokumentaci. Ale rozhodně, při dalším kroku — budování komplexního zálohovacího systému, tento nástroj stojí za další zkoumání.

1.4.2 Nástroj dd

Tento nástroj příkazové řádky je součástí standardní knihovny. Je specifikován v IEEE Std 1003.1-2008, který je součástí Single UNIX Specification [3]. Hlavní využití v mém případě je pro kopírování obrazů *záložních disků* do souborů iso (viz Výpis kódu 1.4). Ty mohou použít buď ke generování *paritního souboru* nebo k obnovení chybějícího či porušeného disku.

1.4.3 Sada nástrojů cdrtools

Sada nástrojů cdrtools slouží k práci s *optickými disky*. Pro moje potřeby jsou nejdůležitější nástroje mkisofs, který používám k vytváření souborů iso, a nástroje cdrecord, který využívám pro vlastní vypalování těchto obrazů na *optické disky*.

1.4.3.1 Nástroj mkisofs

Nástroj příkazové řádky mkisofs slouží k vytvoření obrazu disku ve formátu ISO 9660, Joliet nebo UDF, který je připraven k vypálení. V příkladu 1.1 vytvářím obraz disku z adresáře `tobackup`.

■ Výpis kódu 1.1 Vytvoření obrazu disku ve formátu iso

```
$ mkisofs -r -udf -input-charset utf-8 -o disk1.iso ./tobackup/
Total translation table size: 0
Total rockridge attributes bytes: 661
Total directory bytes: 0
Path table size(bytes): 10
Max brk space used 0
465 extents written (0 MB)
```

1.4.3.2 Nástroj cdrecord

Nástroj příkazové řádky cdrecord mohou využít při ukládání dat na *optické disky* CD nebo DVD (viz Výpis kódu 1.2).

■ Výpis kódu 1.2 Vypálení obrazu disku

```
$ cdrecord -v speed=20 dev=/dev/sr1 ../disk1.iso
...
Last chance to quit, starting real write in 0 seconds. Operation starts.
Waiting for reader process to fill input buffer ... input buffer ready.
Performing OPC...
Starting new track at sector: 24329
Track 01: 58 of 58 MB written (fifo 100%) [buf 99%] 4.2x.
Track 01: Total bytes read/written: 61104128/61104128 (29836 sectors).
Writing time: 102.923s
...
```

V případě prepisovatelného *optického disku* ho mohou také vymazat a tím ho připravit na zápis jiných dat (viz Výpis kódu 1.3).

■ Výpis kódu 1.3 Vymazání optického disku

```
$ cdrecord dev=/dev/sr1 blank=fast
...
Starting to write CD/DVD at speed 4.0 in real BLANK mode for single session.
Last chance to quit, starting real write in 0 seconds. Operation starts.
```

1.4.4 Nástroj udffsck

Tento nástroj příkazové řádky, který byl vytvořen v rámci diplomové práce NÁSTROJE PRO DIAGNOSTIKU INTEGRITY SOUBOROVÉHO SYSTÉMU V OS LINUX, slouží pro detekci a opravu konzistence dat souborového systému UDF až do verze 2.01 v GNU/Linux. V mém případě může velmi dobře sloužit pro obnovu špatně čitelných *optických disků* [23].

1.4.5 Analýza disků a metadat pod OS Linux

Bakalářská práce FORENZNÍ ANALÝZA DISKŮ A METADAT POD OS LINUX [10] se zabývá tvorbou nástroje pro obnovení souboru pro různé souborové systémy. Mimo jiné i pro ISO 9660 a UDF. Aplikace obnovuje soubory pomocí vyhledávání jejich začátků a konců. Dokáže

uspět i v případech, kdy se nedají k obnově souborů využít informace ze žurnálu. Tato práce také srovnává aplikace s již existujícími nástroji zabývající se problematikou obnovování dat z poškozených disků.

1.4.6 Rozšířené zálohování pro OS Windows

V bakalářské práci ROZŠÍŘENÉ ZÁLOHOVÁNÍ PRO OS WINDOWS [20] autor popisuje nej-používanější volně dostupná řešení pro zálohování a na základě získaných poznatků vytvoří program, který je schopný inteligentně zálohovat a obnovovat data včetně jejich verzování. Autor se spoléhá na to, že zálohované soubory budou dostupné bez chyb a vůbec se nezabývá dlouho-dobým skladováním záloh. Program je napsán v jazyce C# v prostředí .NET pro operační systém Windows. Tato práce se sice zabývá jiným problémem, než mám já v zadání, ale v případě dalšího rozvoje komplexního zálohování je určitě dobrým zdrojem informací.

1.5 Pole RAID a nástroj mdadm

Ze zadání plyne, že se mám soustředit na prozkoumání technologie RAID a využít její základní algoritmus k vytvoření vlastního postupu tvorby *paritního souboru*. Proto nejprve zkoumám, co je to technologie RAID, jak lze vytvořit pole RAID a poté získané informace využiji pro tvorbu mého algoritmu.

1.5.1 Analýza technologie RAID

RAID je technologie, která umožňuje z několika fyzických disků vytvořit disk logický, který má následující výhody [19] [15]:

- rychlejší čtení a zápis dat,
- a také bude větší, než každý jednotlivý disk,
- zvýšení kapacity úložiště oproti samostatnému disku [2].

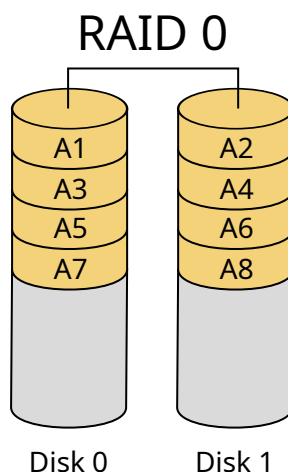
Existují 3 hlavní metody, které jsou v praxi v rámci technologie RAID využívány:

- prokládání (stripping) – rovnoměrné prokládání dat na 2 nebo více disků, kdy první blok je uložen na 1. disku, druhý na 2. disku ... N-tý blok na N-tém disku, $N + 1$ blok opět na prvním disku atd.,
- zrcadlení (mirroring) – prosté zkopírování dat na 2 nebo více disků,
- parita (parity) – využívá speciální – paritní bloky, kde jsou uloženy speciální samoopravné kódy.

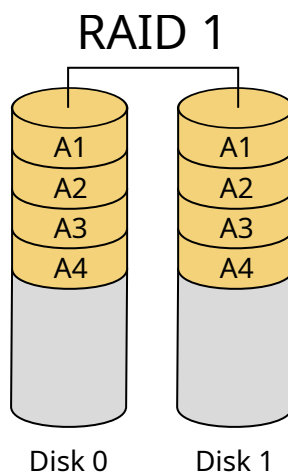
Ve všech případech uvažuji velikosti disků jako stejné. Jinak, v případě např. zrcadlení, nejsem schopen na menší disk zrcadlit všechna data z disku většího.

1.5.1.1 RAID 0

RAID 0 využívá pouze metodu prokládání (viz Obrázek 1.3) a tím navyšuje kapacitu logického pole na součet kapacity použitých disků [15]. Vlastní data jsou uložena v blocích $A1, \dots, A8$. V mém případě tato technologie není příliš vhodná, protože mám k dispozici již hotové samotné disky s daty a není proto nutné, abych je vytvářel znova.



■ **Obrázek 1.3** Rozvržení RAID 0 Zdroj: [1]



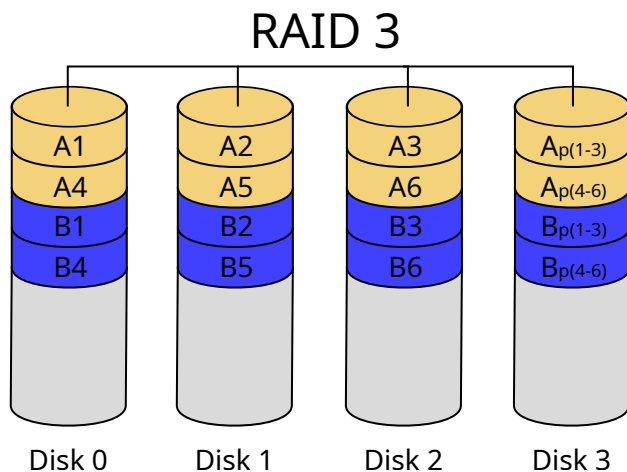
■ **Obrázek 1.4** Rozvržení RAID 1 Zdroj: [1]

1.5.1.2 RAID 1

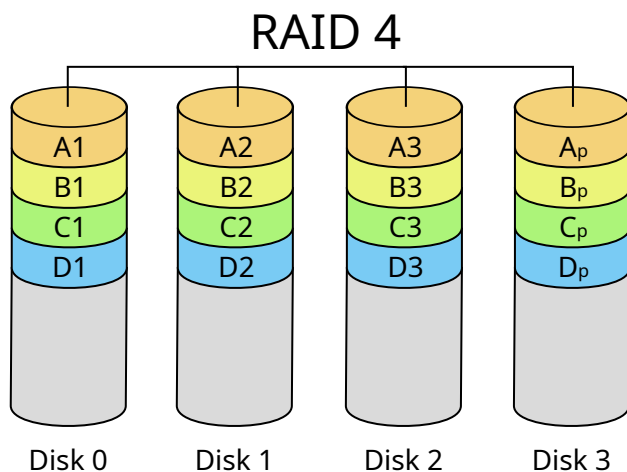
RAID 1 využívá metodu zrcadlení, kdy jsou na 2 disky uloženy stejná data. Jejich velikost je v tomto případě stejná jako velikost jednoho disku. Na obrázku 1.4 data tvoří bloky A_1, \dots, A_4 . Nevýhodou je, že je zde pouze $\frac{1}{2}$ využitelné kapacity, což je ve srovnání s ostatními algoritmy nejméně.

1.5.1.3 RAID 3, 4, 5

Základní vlastností této skupiny technologií je využití $N + 1$ disků o stejné velikosti v bytů. Prostor N disků se využívá pro uložení vlastních dat a prostor 1 disku na uložení takzvaných *paritních dat*. Z toho mohou dopočítat, že pomocí tohoto způsobu mohou uložit maximálně $v \cdot N$ bytů. Jednotlivé technologie RAID se poté liší způsobem, jakým jsou data mezi disky rozdělena [15].



■ Obrázek 1.5 Rozvržení RAID 3 Zdroj: [1]

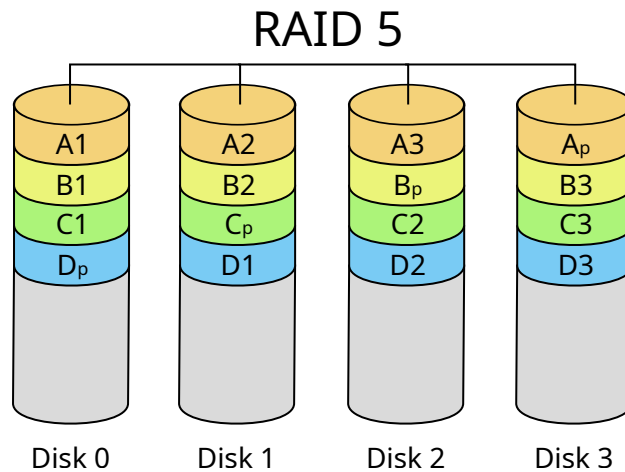


■ Obrázek 1.6 Rozvržení RAID 4 Zdroj: [1]

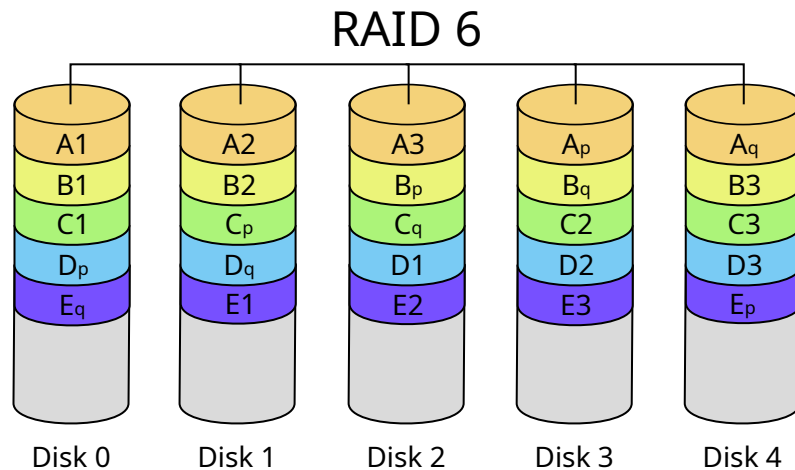
RAID 3 využívá principu prokládání a parity. Pro ten je třeba N datových disků a jeden *paritní disk* pro ukládání *paritních dat*. Na obrázku 1.5 tvoří vlastní data byty $A_1, \dots, A_6, B_1, \dots, B_6$. Byty $A_{p(1-3)}, A_{p(4-6)}, B_{p(1-3)}, B_{p(4-6)}$ jsou příslušné paritní byty. Nevýhodou tohoto řešení je velké vytížení *paritního disku*, protože při každém zápisu dat je třeba na něj zapsat změny.

RAID 4 používá, podobně jako RAID 3, jeden *paritní disk*. Ovšem při ukládání dat se prokládají celé bloky místo jednotlivých bytů. Na obrázku 1.6 tvoří data bloky $A_1, A_2, A_3, B_1, B_2, B_3, C_1, C_2, C_3, D_1, D_2, D_3$ a příslušné paritní bloky jsou A_p, B_p, C_p, D_p . V tomto případě ovšem bloky netvoří byty, ale větší skupiny bytů. Podobně jako RAID 3 trpí problémem častého přepisování *paritního disku*.

RAID 5 je nejvíce populární způsob uložení dat z této kategorie hlavně proto, že řeší problém příliš častého přístupu k *paritním datům* tak, že tuto část rovnoměrně rozdělí mezi všechny disky. Na obrázku 1.7 tvoří data bloky $A_1, A_2, A_3, B_1, B_2, B_3, C_1, C_2, C_3, D_1, D_2, D_3$ a



■ **Obrázek 1.7** Rozvržení RAID 5 Zdroj: [1]



■ **Obrázek 1.8** Rozvržení RAID 6 Zdroj: [1]

paritní bloky jsou A_p, B_p, C_p, D_p . Nevýhodou je zde pomalejší zápis o čas, který je třeba strávit výpočtem *paritních dat*.

1.5.1.4 RAID 6

RAID 6 je podobný RAID 5. Používá dva paritní bloky střídavě na všech discích, kde každý z nich vypočítá jiným způsobem. Je odolný proti výpadku dvou disků. Rychlost čtení je srovnatelná s RAID 5, ale zápis je pomalejší, protože algoritmus musí vypočítat dvě sady paritních informací. Na obrázku 1.8 data tvoří bloky $A_1, A_2, A_3, B_1, B_2, B_3, C_1, C_2, C_3, D_1, D_2, D_3, E_1, E_2, E_3$ a příslušné paritní bloky jsou $A_p, B_p, C_p, D_p, E_p, A_q, B_q, C_q, D_q, E_q$. RAID 6 má tedy smysl používat s 5 a více disky. Při menším počtu disků a zachování kapacity lze nahradit technologií RAID 1, kdy algoritmus nemusí počítat paritu, ale stačí provést jednoduché zrcadlení.

1.5.1.5 Výběr vhodného algoritmu RAID pro moje účely

Jako výslednou technologii pro nástroj `dsb` volím modifikovanou verzi RAID 4, protože tento typ nejvíce vyhovuje zadání. Místo prokládání dat mezi jednotlivými disky použiji již vytvořené disky celé jako data a zbývá mi dopočítat pouze *paritní soubor*. V mém případě není nevýhodou, že nemám parity rozdělené mezi jednotlivé disky z následujících důvodů:

- *Paritní soubor* nepotřebuji, s výjimkou situace, když obnovuji ztracený disk nebo opravuji disk s porušenými sektory.
- Samostatné disky mají svůj vlastní souborový systém a lze je tedy používat nezávisle na sobě.

Stačí mi tedy vytvořit nástroj, který zvýší šanci pro obnovení *záložních disků* vytvořením *souboru* s *paritními* daty.

1.5.2 Analýza využití postupu pro účely zálohování dat na optická média

- Pole RAID je určeno primárně pro disky, které se používají v reálném čase. Optimalizuje se také rychlost přístupu k datům.
- Naproti tomu mně stačí zapsat data jen jednou a pak je číst jen výjimečně. Bud' při kontrole dostupnosti dat nebo jejich vlastní obnově.

1.5.3 Binární kopie optického média

K vytvoření binární kopie optického média využiji standardní nástroj příkazové řádky `dd`. Ten zkopíruje veškerá data z *optického disku* a nakopíruje je do souboru ve formátu `iso`.

■ Výpis kódu 1.4 Binární kopie optického média

```
$ dd if=/dev/sr1 of=disk1.iso bs=1M
154+1 records in
154+1 records out
161558528 bytes (162 MB, 154 MiB) copied, 2.17909 s, 74.1 MB/s
```

Tento příkaz zkopíruje obraz *optického disku* ve formátu `iso`. Tyto soubory pak využívám pro tvorbu *paritního souboru*.

1.5.4 Tvorba pole RAID 4 ve Windows WSL / Linux

V dalším textu podrobně popíši postup tvorby diskového pole RAID 4. Postup je tvořen posloupností Výpisů kódu 1.5 ... 1.17. Poslední Výpis kódu 1.18 ukazuje, jak zobrazit nápovědu pro jednotlivé kroky.

■ Výpis kódu 1.5 Vytvoření 1 prázdného disku

```
$ dd if=/dev/zero of=image1.img iflag=fullblock bs=1M count=100 && sync
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 0.0603768 s, 1.7 GB/s
```

■ Výpis kódu 1.6 Vytvoření dalších 2 disků nakopírováním 1. disku

```
$ cp image1.img image2.img
$ cp image1.img image3.img
```

■ Výpis kódu 1.7 Připojení všech disků k operačnímu systému

```
$ sudo losetup loop1 image1.img
$ sudo losetup loop2 image2.img
$ sudo losetup loop3 image3.img
```

■ Výpis kódu 1.8 Vytvoření diskového pole RAID 4 nad připojenými disky

```
$ sudo mdadm --create /dev/md0 --level=4 --raid-devices=3 /dev/loop1
/dev/loop2 /dev/loop3
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

■ Výpis kódu 1.9 Zobrazení stručné informace o nově vytvořeném poli

```
$ sudo mdadm --query /dev/md0
/dev/md0: 196.00MiB raid4 3 devices, 0 spares. Use mdadm --detail for more detail.
```

■ Výpis kódu 1.10 Zobrazení detailní informace o poli

```
$ sudo mdadm --detail /dev/md0
      Version : 1.2
  Creation Time : Sat Jan 1 00:00:00 2022
    Raid Level : raid4
   Array Size : 200704 (196.00 MiB 205.52 MB)
  Used Dev Size : 100352 (98.00 MiB 102.76 MB)
  ...
```

■ Výpis kódu 1.11 Zobrazení informace o blokových zařízeních

```
$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
loop1 7:1 0 100M 0 loop
+-md0 9:0 0 196M 0 raid4
loop2 7:2 0 100M 0 loop
+-md0 9:0 0 196M 0 raid4
loop3 7:3 0 100M 0 loop
+-md0 9:0 0 196M 0 raid4
...
```

■ Výpis kódu 1.12 Naformátování disku na systém souborů NTFS

```
$ sudo mkfs -t ntfs /dev/md0
Cluster size has been automatically set to 4096 bytes.
Initializing device with zeroes: 100% - Done.
Creating NTFS volume structures.
mkntfs completed successfully. Have a nice day.
```

■ Výpis kódu 1.13 Vytvoření adresáře pro bod připojení

```
$ sudo mkdir -p /mnt/md0
```

■ Výpis kódu 1.14 Připojení svazku

```
$ sudo mount -t auto /dev/md0 /mnt/md0
```

■ Výpis kódu 1.15 Pokusné zapsání dat na disk

```
$ echo "Hello World!" > /mnt/md0/helloworld
$ cat /mnt/md0/helloworld
Hello World!
```

■ Výpis kódu 1.16 Odpojení svazku

```
$ sudo umount /mnt/md0
```

■ Výpis kódu 1.17 Zastavení diskového pole

```
$ sudo mdadm --stop --scan
```

■ Výpis kódu 1.18 Zobrazení nápovědy pro použité příkazy mdadm

```
mdadm --create --help
mdadm --config --help
mdadm --help
```

1.6 Souborové systémy optických disků

V mé práci se také ujistím, že obrazy *optických disků* mohou považovat za bloky o délce, která je násobkem 512 bytů. Ale protože souborových systémů pro *optické disky* je mnoho, já se zde budu soustředit pouze na ty nejrozšířenější [11].

1.6.1 ISO 9660

Standard ISO 9660 vznikl již v roce 1988 a bývá označován také jako CDFS. Tento formát se hodně rozšířil hlavně proto, že je univerzální a tato média lze číst na mnoha operačních systémech včetně Windows, Linux, Mac. Díky tomuto rozšíření je tedy stále ještě možné narazit na CD s tímto souborovým systémem. Velikost disku je obvykle násobkem 2 KiB [9] [13].

Standard ISO 9660 má 3 úrovně [11]:

- ISO 9660 Level 1: Jméno souboru může mít maximálně 8 znaků + 3 znaky pro příponu a může obsahovat pouze znaky A-Z, číslice 0-9 a podtržítka .. Maximální hloubka adresářů je 8. Soubor nesmí být fragmentovaný.
- ISO 9660 Level 2: Maximální délka souboru je prodloužena na 31 znaků včetně oddělovací tečky mezi jménem souboru a jeho příponou. Název souboru může obsahovat speciální znaky.
- ISO 9660 Level 3: Navíc k předchozím vlastnostem umožňuje, že data mohou být fragmentovaná.

1.6.2 Joliet

Formát disku Joliet je rozšířením normy ISO 9660, který vyvinula firma Microsoft pro operační systém Windows od verze Windows 95. Současně je ale možné jej číst ve většině ostatních operačních systémů.

Hlavní rozdíl oproti ISO 9660 je v

- Maximální délka názvu souboru je až 64 znaků.

■ **Tabulka 1.2** Srovnání souborových systémů

	ISO 9660 L1	ISO9660 L2	ISO9660 L3	Joliet	UDF
Použitelné znaky	A-Z, 0-9, -	speciální znaky	spec. znaky	Unicode	Unicode
Délka názvu souboru	8 + 3	30	30	128	255
Délka názvu adresáře	8 + 3	31	31	128	255
Délka cesty	8 adresářů	8 adresářů	8 adresářů	neomezená	neomezená
Podpora fragmentace	ne	ne	ano	ano	ano
Podpora multisession	ne	ne	ne	ano	ano

- Podporuje kódování unicode v názvech souborů.
- Není omezena délka cesty k souboru.
- Podporuje postupné přidávání dat na disk (technologie multisession).

1.6.3 UDF

Souborový systém UDF je vytvořen jako nástupce formátu ISO 9660 mezinárodní obchodní asociací OSTA v roce 1996, který je založený na standardu ISO/IEC 13346 a ECMA-167. Mohu ho použít pro většinu typů optických médií CD, DVD, Blu-ray. Podporuje jej také většina současných operačních systémů včetně Windows, Linux, Mac. Další podrobnosti o tomto nejrozšířenějším souborovém systému pro optická média jsou [21] [23].

1.6.4 Srovnání souborových systémů

Shromáždil jsem všechny parametry souborových systémů, které jsou popsány výše do tabulky 1.2 [11].

1.6.5 Formát souborů ISO

ISO je asi nejznámější formát obrazů disku pro optická média. Název je původně odvozený z názvu souborového systému ISO 9660 popsaného výše. Nyní může být také v souboru s příponou iso uložený souborový systém UDF.

1.7 Testování čitelnosti disků

Pro dosažení spolehlivé dostupnosti zálohovaných dat je nutné *optické disky* pravidelně testovat a nahrazovat je po vypršení deklarované výdrže za nové.

Praktická část

V praktické části vytvořím nástroj, který umožní vytváření *paritního souboru* pro *optické disky* a tím zvýší šanci na úspěšné obnovení záložních dat.

2.1 Model procesů

Ze zadání práce a z následné komunikace s vedoucím práce jsem vytvořil model procesu využití výsledného nástroje viz Obrázek 2.1. Dověděl jsem se také, že bude součástí většího celku, který se bude zabývat zálohováním pro domácí uživatele jako celku. včetně možnosti vyhledávání souborů a zálohování z aplikací třetích stran.

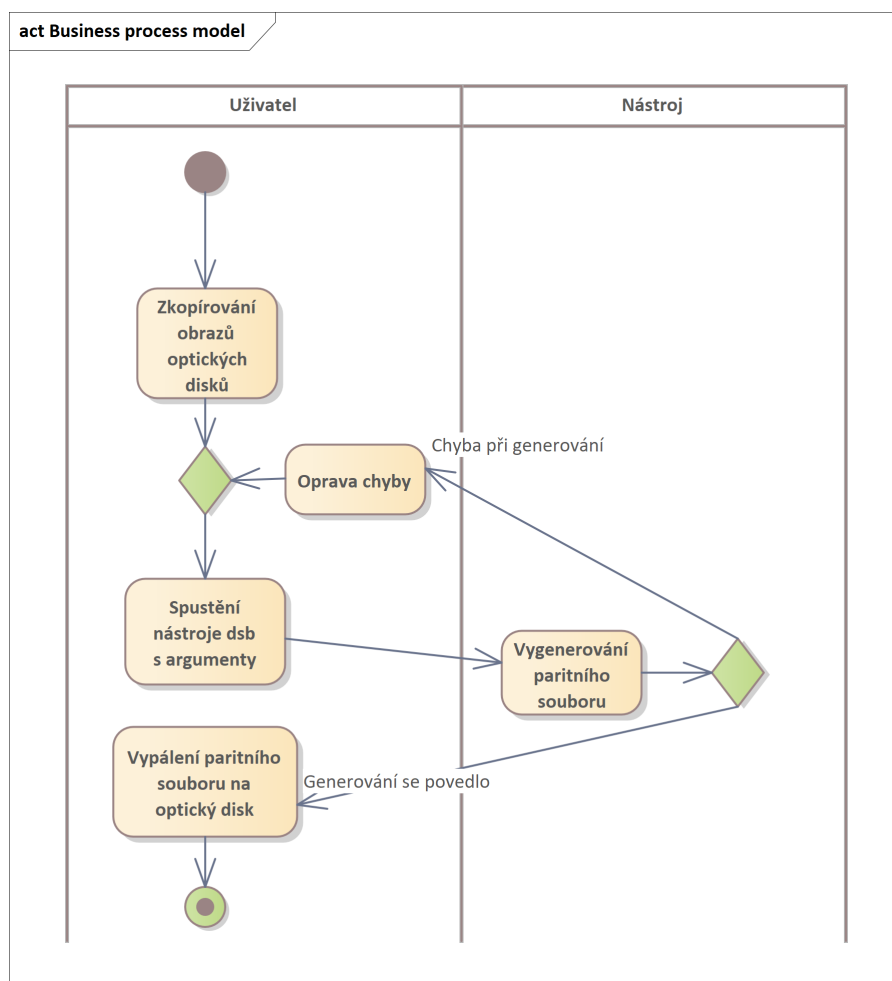
Nicméně v této práci se zabývám pouze nástrojem, který umí zvýšit šanci na obnovení dat ze záložních disků vytvořením *paritního souboru*.

2.2 Funkční a nefunkční požadavky

- F1: Uživatel chce zvýšit spolehlivost zálohy vytvořením 1 nového disku, když má již vytvořené disky (2 a více) s vlastním souborovým systémem.
- F2: Uživatel chce vygenerovat chybějící disk a má k dispozici všechny ostatní disky a *paritní soubor*. Nástroj obnoví chybějící disk a pomocí kontrolního součtu ověří, že je vytvořen správně.
- F3: Uživatel chce vygenerovat nepoškozený disk a má k dispozici všechny disky a *paritní soubor*, kdy některé z nich mohou být poškozené. Má také seznam všech poškozených bloků na těchto discích. Nástroj ověří, že poškozené části se nepřekrývají a vygeneruje disk bez poškozených bloků a ověří kontrolní součet opraveného disku.
- F4: Uživatel chce ověřit, že existující záloha je správně čitelná.

Nefunkční požadavky

- N1: Při rekonstrukci chybějícího disku nástroj sám zkontroluje, že vstupní disky jsou ty, které byly použity při tvorbě *paritního souboru*.
- N2: Spuštění nástroje musí být jednoduché, všechny požadované hodnoty jde zadat jako vstupní argumenty nástroje nebo jsou uloženy v textovém souboru a cesta k nim je poté předána jako vstupní argument.



■ Obrázek 2.1 Diagram modelu procesu

■ **Tabulka 2.1** Pokrytí funkčních požadavků případy užití

	F1	F2	F3	F4
UC1	X			
UC2		X		
UC3			X	
UC4	X			X

- N3: Všechny funkční požadavky budou dostupné také jako knihovna pro případné využití jinými komponentami.

2.3 Model případů užití

2.3.1 Případy užití

- UC1: Vytvoření záložního souboru – zálohovač vytvoří z optických disků *paritní soubor* a ten vypálí na optické médium.
- UC2: Obnovení ztraceného disku ze skupiny disků – restaurátor z optických disků a *paritního souboru* vytvoří kopii ztraceného optického disku a ujistí se, že je funkční.
- UC3: Obnovení částečně porušeného disku – restaurátor z částečně porušených optických disků vytvoří kopii optického disku bez chyb.
- UC4: Ověření konzistence disků – restaurátor prověří, že všechny optické disky a *paritní soubor* jsou bez chyby.

Seznam případů užití je také v diagramu viz Obrázek 2.2. Pokrytí funkčních požadavků případy užití jsem zobrazil do tabulky 2.1. Mapování aktérů na jednotlivé případy užití je v diagramu viz Obrázek 2.3.

2.4 Doménový model

V diagramu viz Obrázek 2.4 zobrazuji doménový model, který pokrývá všechny požadavky na výsledný nástroj. DiscGroup je objekt, který popisuje celou skupinu disků, pro kterou jsem vygeneroval *paritní soubor*. Ten se skládá z jednotlivých disků s daty (Disc - Raw) a také jedním *paritním souborem* (Disc - Parity).

Na základě doménového modelu mohu vytvořit příslušné hlavičky *paritního souboru*, které budou udržovat informace o skupině disků (viz Výpis kódu 2.1) a i o jednotlivých discích 2.2 (Výpis kódu 2.2).

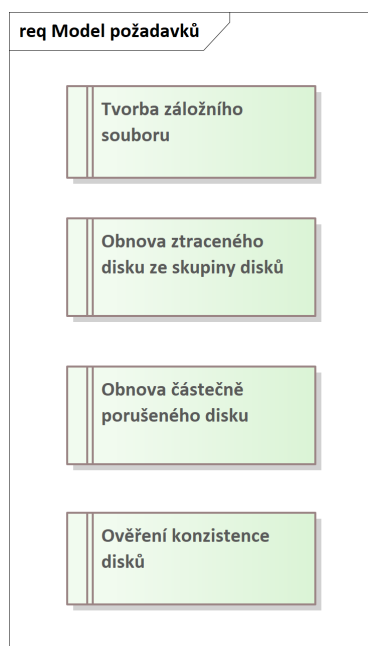
■ **Výpis kódu 2.1** Hlavička skupiny disků

```

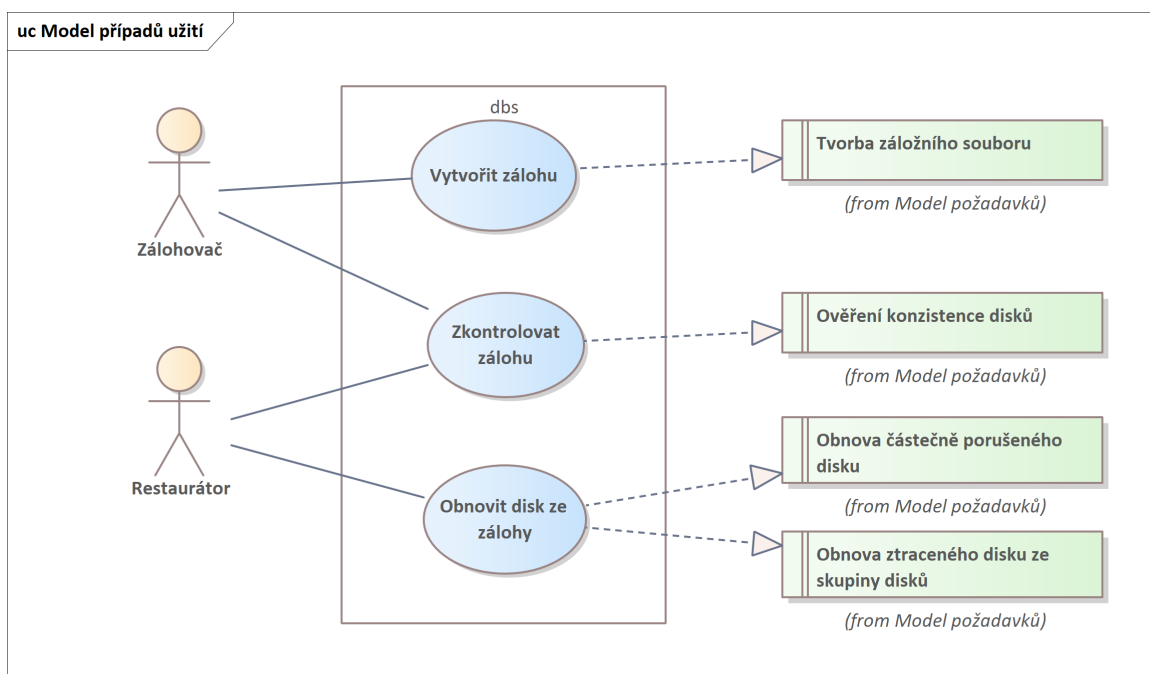
/// Maximum number of disks in one group
#define DSB_COUNTOF_DISKS 10

typedef struct _dsb_header_t {
    ULONG64 magic;                ///< Magic number DSB_HEADER_MAGIC
    UCHAR sizeo;                  ///< Size of header in sectors
    UCHAR countofDisks;          ///< Count of all disks in the pool (= valid
        elements in disks)
    UCHAR reserved1[2];          ///< Reserved values equals to zero

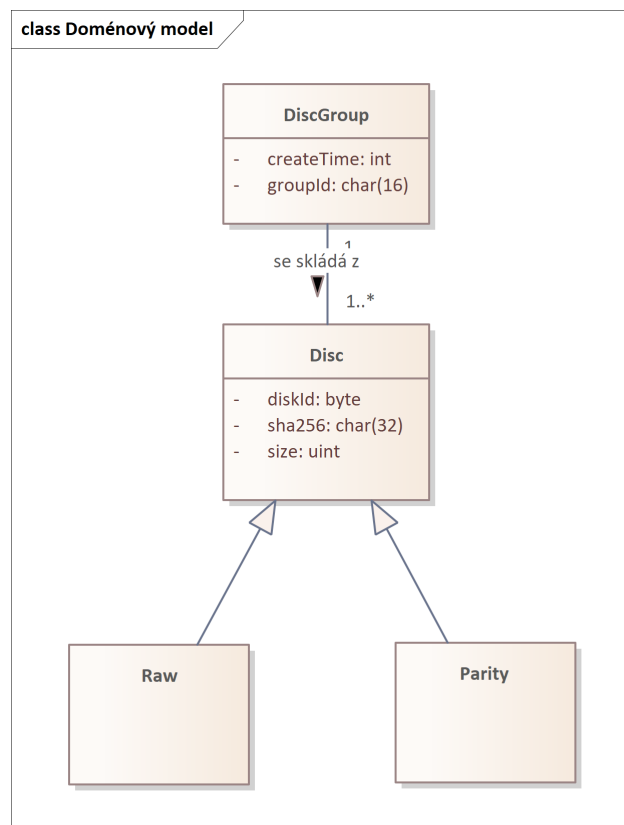
```



■ Obrázek 2.2 Diagram modelu funkčních požadavků



■ Obrázek 2.3 Diagram modelu případů užití



■ **Obrázek 2.4** Diagram doménového modelu

```

ULONG crc32Header;          ///< Crc of this header when this field is zero
GUID groupId;              ///< Identifier for given group
ULONG64 createTime;       ///< time, when the header was created
UCHAR reserved2[72];      ///< Reserved values equals to zero

dsb_disk_info_t disks[DSB_COUNTOF_DISKS]; ///< [0] = my disk, rest are all other
                                     disks in group
} dsb_header_t, *pdsb_header_t;

```

■ Výpis kódu 2.2 Hlavička každého disku ve skupině

```

/// Size of sha256 hash in bytes
#define DSB_SHA256_SIZE      32

typedef struct dsb_disk_info_t {
    dsb_typeof_disk_t typeofDisk;    ///< Type of disk
    UCHAR diskId;                   ///< Unique identifier of this disk
    UCHAR reserved[2];              ///< Reserved value equals zero
    ULONG size;                     ///< Size of data in sectors
    UCHAR sha256[DSB_SHA256_SIZE];  ///< Hash of the disk
} dsb_disk_info_t, *pdsb_disk_info_t;

```

Nyní popíši jednotlivé položky hlavičky *paritního souboru dsb_header_t*:

<i>magic</i>	Hodnota "DSB 1.0", která jde dobře číst a podle ní se pozná začátek vlastní hlavičky. Také obsahuje číslo verze hlavičky – v mém případě je to verze 1.0.
<i>sizeo</i>	Velikost hlavičky ve stránkách. 1 stránka má 512 bytů.
<i>countofDisks</i>	Počet disků v celé skupině včetně <i>paritního souboru</i> .
<i>reserved1</i>	Rezervované hodnoty. Při vytváření se nastaví na 0, při čtení se tato hodnota vůbec nepoužívá.
<i>crc32Header</i>	Crc32 z celé hlavičky. Pro účely výpočtu se hodnota <i>crc32Header</i> nastaví na 0, vypočítá se <i>crc32</i> a na závěr se vypočtená hodnota zapíše na tuto pozici.
<i>groupId</i>	Unikátní identifikátor skupiny uložený jako GUID.
<i>createTime</i>	Čas vytvoření <i>paritního souboru</i> . Protože využívám standardní POSIX funkci <i>time</i> , je tento čas počet sekund od 1.1.1970 0:00:00.
<i>reserved2</i>	Další blok rezervovaných hodnot.
<i>disks</i>	Vlastní pole disků o velikosti 10 položek. Odsud plyne omezení, že ve skupině disků může být maximálně 9 disků + 1 <i>paritní soubor</i> .

Hlavička jednotlivého disku *dsb_disk_info_t* obsahuje tyto položky:

<i>typeofDisk</i>	Typ disku. Nyní to mohou být hodnoty <i>typeof_disk_RAW</i> nebo <i>typeof_disk_PARITY</i> . Hodnota <i>typeof_disk_0</i> je rezervovaná pro chybový stav a nebude nikdy využita v platné hlavičce disku.
<i>diskId</i>	Unikátní identifikátor disku v rámci skupiny.
<i>reserved</i>	Další rezervované byty.
<i>size</i>	Velikost disku ve stránkách. Připomínám, že stránka má 512 bytů.
<i>sha256</i>	Hash SHA256 celého disku. Tato hodnota se počítá v okamžiku vytváření <i>paritního souboru</i> a využívá se pro ověření, že disk obsahuje nezměněná data.

Při řešení problému obnovení dat při porušení nějakého libovolného bloku dat nesmím zapomenout ani na vlastní hlavičku. Již v popisu položek uvádím *crc32Header*, pomocí které umím otestovat, jestli je samotná hlavička uložena bez chyby nebo nikoliv. Pro případ jejího porušení jsem navrhl uložení dvou stejných kopií – jedna je na začátku *paritního souboru* a druhá na jejím konci.

2.5 Programovací jazyk a implementace

Pro implementaci jsem vybral programovací jazyk C++14. V maximální možné míře jsem využil standardní funkce POSIX, protože jsou tyto funkce použitelné na platformách, které potřebují tzn. Windows a Linux. Dále je zde jednoduché a rychlé načítání potřebných dat z obrazu disků. Produkt jsem rozdělil podle modulárního přístupu do tříd, kde každá má jasně definovanou zodpovědnost a naplnil jsem tedy základní požadavky pro dobře navržený systém.

2.5.1 Algoritmus tvorby paritního souboru

Před zahájením algoritmu si naalokují potřebné vstupní a výstupní vyrovnávací paměti. Otevřou si všechny vstupní soubory s obrazy disků a také výstupní soubor, který bude obsahovat výsledná data ve formátu *DSB*.

Do výstupního souboru si připravím dostatek místa pro uložení úvodní hlavičky posunutím ukazatele v souboru na pozici za hlavičkou – v mém případě, na pozici 512 B. Poté načtu blok dat z prvního datového disku do vstupní vyrovnávací paměti. Aktualizuji výpočet hashe pro první disk a zkopíruji do výstupní vyrovnávací paměti. Podobným způsobem zpracuji data z druhého vstupního disku. Nejprve je načtu, pak aktualizuji výpočet hashe pro druhý disk a na závěr provedu xor celého bloku s výstupní vyrovnávací pamětí. Tento algoritmus zopakují pro všechny zbylé vstupní obrazy disků.

Nyní doplním poslední chybějící hodnoty do hlavičky. Vygeneruji náhodný unikátní identifikátor skupiny disků a jako indexy jednotlivých disků v rámci skupiny použiji jejich pořadí, jak byly zadány do nástroje *dsb*. Doplním hashe jednotlivých disků a jejich velikosti, počet těchto disků a datum vytvoření zálohy. Vypočtu hodnotu *crc32* celé hlavičky a zaznamenám ji na příslušnou pozici.

Po vytvoření kompletní hlavičky ji zapíši na konec disku za *paritní soubor*, a stejnou kopii také na začátek do prostoru, který jsem si předem připravil.

Při načítání dat může nastat speciální situace, kdy velikost obrazu jednoho disku je menší než velikost ostatních disků. V tom případě načtu pouze ta data, která jsou k dispozici a výpočet hashe a xor provedu pouze s nimi. Pokud na chvíli připustím, že doplním načtená data nulami, tak ale mohu lehce ověřit $A \wedge 0 = A$. Neboli s nulami nemusím vůbec počítat a mohu je jednoduše vynechat z výpočtu. Stačí si tedy zapamatovat, že pro takový disk již není třeba načítat další data a mohu ho při generování vynechat.

► **Příklad 2.1.** Vytvoření *paritního souboru* ve formátu *DSB*

■ **Výpis kódu 2.3** Příkaz pro vytvoření *paritního souboru* ve formátu *DSB*

```
./dsb build -i disk1.iso -i disk2.iso -o parity.dsb
```

Na vstupu má algoritmus zadané 2 disky. První soubor s obrazem disku *D1* má 105 MB a druhý *D2* má 206 MB. Velikost vyrovnávací paměti je 10 MB. Před vlastním zahájením algoritmus naalokuje vstupní a výstupní vyrovnávací paměti o velikosti 10 MB. Výstupní paměť celou vynuluje.

Algoritmus nejprve načte 10 MB dat ze souboru *D1* prvního disku do vstupní vyrovnávací paměti, aktualizuje výpočet hashe pro tento disk a na závěr připojí do výstupní vyrovnávací paměti tak, že se na každý bit provede operace xor. Poté načte prvních 10 MB ze souboru *D2* a opět aktualizuje výpočet hashe pro druhý disk a připojí do výstupní paměti pomocí operace xor podobně jako v předchozím případě. V tomto okamžiku použije výstupní paměť k aktualizaci hashe *paritního souboru* a paměť zapíše do výstupního souboru.

Toto zopakuje 10 krát a při jedenáctém načtení dat ze souboru *D1* přečte pouze 5 MB. Nyní operuje pouze s těmito 5 MB. Pouze ty použije k aktualizaci hashe disku *D1* a ty nakopíruje do výstupního souboru. Dále se již souborem *D1* nebude zabývat. Celý další načtený blok o velikosti 10 MB ze souboru *D2* připojí do výstupní paměti pomocí operace xor a po aktualizaci hashe

paritního souboru zapíše do výstupního souboru. 9x zopakuje generování výstupního souboru pouze ze souboru *D2*. Poslední blok o velikosti 6 MB ze souboru *D2* zpracuje podobně jako to udělal pro disk *D1*. Tím vytvoří datovou část výsledného souboru formátu *DSB*, která bude mít velikost stejnou jako největší soubor – v tomto případě 206 MB.

Nakonec algoritmus připraví a zapíše hlavičku na začátek i konec a tím je výstupní soubor formátu *DSB* vytvořen.

2.5.2 Algoritmus obnovení chybějícího disku

Obnovení disku zahájím alokací potřebné vstupní a výstupní vyrovnávací paměti. Jako první vstupní argument otevřu *paritní soubor* ve formátu *DSB*, který také obsahuje informace o ostatních discích. Dále otevřu všechny soubory s obrazy disků zadané ve vstupních argumentech. Pokud jsem pro vytvoření *paritního souboru* použil N disků, tak nyní potřebuji $N - 1$ disků. Nakonec otevřu výstupní soubor, který bude obsahovat obnovený chybějící obraz disku.

Z *paritního souboru* se pokusím načíst první kopii hlavičky, která je uvedena na začátku souboru. Ověřím, jestli je validní a neporušená. V případě, že není, tak využiji existenci druhé kopie na konci souboru a opět ověřím její neporušenost. V případě, že ani druhá hlavička není v uložena pořádku, tak nahlásím chybu ve čtení hlaviček, protože informace v nich jsou klíčové pro obnovování jakýchkoliv dat.

Jako první krok provedu ověření vstupních argumentů. Prověřím, že velikosti a hashe všech souborů souhlasí s hodnotami, které jsou uloženy v hlavičce, že na vstupu není zadán žádný soubor vícekrát. Pokud nastane libovolná z těchto chyb, informuji uživatele a ukončím algoritmus s chybou.

Jádro algoritmu je velmi podobné tomu, co jsem implementoval v případě generování *paritního souboru*. Nejprve načtu první blok z *paritního souboru* do vstupní vyrovnávací paměti a ten připojím do výstupní vyrovnávací paměti pomocí operace xor na každý bit vstupu. Potom provedu stejný postup se všemi zbývajícimi soubory obrazu všech disků. Aktualizuji hash z dat ve výstupní paměti a zapíši je do výstupního souboru. Tyto operace provedu se všemi bloky až dosáhnu původní velikosti *obrazu disku*. Zde připomínám, že pokud nastane situace, kdy velikost některého souboru s obrazem disku je menší než velikost výsledného disku, tak zpracuji data pouze do velikosti tohoto disku a dále tento obraz nepoužívám ze stejného důvodu, jako jsem to dělal v případě vytváření zálohy.

■ **Výpis kódu 2.4** Příkaz obnovení chybějícího disku `disk2.iso`

```
$ ./dsb restore -i parity.dsb -i disk1.iso -o disk2~.iso
```

2.5.3 Algoritmus pro opravení disku s vadnými sektory

Při tvorbě algoritmu pro opravení disku jsem také vycházel z případu pro generování *paritního souboru*. Na vstupu mám k dispozici seznam porušených bloků pro všechny obrazy disků a také *paritní soubor* ve formátu *DSB*. Podobně, jako v předchozí kapitole, z hlavičky *paritního souboru* načtu buď první nebo její druhou kopii. Z ní ověřím, že mám k dispozici všechny potřebné soubory. Dále ověřím, že se žádné 2 intervaly poškozených bloků nepřekrývají a mám tedy zaručeno, že půjdou obnovit všechna data. Procházím opravovaný soubor od začátku a kopíruji nepoškozená data do generovaného souboru. V případě, že narazím na poškozená data, využiji ostatních souborů a pomocí operace xor vytvořím správný blok, který nakonec také nakopíruji do generovaného souboru. Během těchto operací průběžně také počítám jeho hash.

Na závěr po dokončení generování ověřím, že mnou napočítaná hash odpovídá hodnotě, která je uložena v hlavičce *paritního souboru*. V případě výskytu jakékoliv chyby informuji uživatele výpisem do chybového výstupu.

► **Příklad 2.2.** Opravení disku `disk2.iso`

■ Výpis kódu 2.5 Příkaz pro opravení disku disk2.iso

```
$ ./dsb restore -i parity.dsb -i disk1.iso -i disk2.iso -s invsect.txt -d 1 -o disk1~.iso
```

-i parity.dsb	<i>paritní soubor</i>
-i disk1.dsb -i disk2.iso	soubory s obrazy disků
-s invsect.txt	soubor s popisem vadných sektorů
-d 1	chci obnovit disk 1
-o disk1 .iso	jméno výstupního souboru pro obraz disku 1

■ Výpis kódu 2.6 Obsah souboru invsect.txt

```
$ cat invsect.txt
disk 1
32768 512
65024 1536
disk 2
2048 512
```

- `disk 1` - Identifikuje disk, na kterém se vyskytují nějaké chyby. V našem případě je to disk 1.
- `32768 512` - První číslo je offset na disku, na kterém se vyskytuje chyba v bytech. Druhé číslo je délka chybného bloku v bytech. V našem případě je chybný sektor na offsetu 32768 B a délku má 512 B.
- Všechny položky, až do dalšího řádku `disk 2`, jsou položky, které popisují vadné sektory disku 1.
- Položka `disk 2` obsahuje informace o poškozených sektorech disku 2.

2.5.4 Sestavení projektu dsb

Pro sestavení projektu `dsb` v OS Linux používám standardní nástroj `make`.

■ Výpis kódu 2.7 Náповěda nástroje make pro dsb

```
$ make help
Usage: make <target>

target:
  all
  check
  clean
  compile
  doc
  help
  run
```

Dokumentace jednotlivých příkazů:

- `dsb all` – sestavení spustitelného souboru a vygenerování dokumentace,
- `dsb check` – otestování běhu pomocí nástroje `valgrind`,
- `dsb clean` – smazání všech vygenerovaných souborů,
- `dsb compile` – sestavení spustitelného souboru,

- `dsb compile_a` – sestavení statické knihovny,
- `dsb compile_so` – sestavení dynamické knihovny,
- `dsb doc` – vygenerování dokumentace nástrojem doxygen,
- `dsb help` – zobrazení vlastní nápovědy nástroje make,
- `dsb run` – sestavení a spuštění nástroje `dsb`, např. `make run ARGS="verify -i parity.dsb"`.

V OS Windows k vývoji nástroje `dsb` používám Visual Studio 2019. Pro vytvoření spustitelného souboru otevřu `dsb.sln` a poté mohu sestavit projekt pro platformy x86 nebo x64 pro konfigurace Debug, kterou použiji při ladění projektu, nebo Release, která optimalizuje výsledný kód a je tak vhodná pro závěrečné nasazení.

2.5.5 Dokumentace generovaná nástrojem doxygen

Do nástroje příkazové řádky `dsb` vkládám také komentáře ke všem veřejným funkcím a položkám tříd. Potom pomocí nástroje doxygen generuji dokumentaci a tu vkládám společně ze zdrojovým kódem na médium a je tak součástí této práce.

V této kapitole postupně projdu všechny hlavní funkční požadavky a prověřím, že je implementace nástroje `dsb` vyřeší správně. U všech operací, které běží dlouho se zobrazuje průběh operace v procentech. Jedná se o operace vypočtení hashe disku, vytvoření *paritního souboru* a obnovení porušeného nebo chybějícího disku.

3.1 Náповěda nástroje `dsb`

Nástroj `dsb` zobrazuje nápovědu stejným způsobem jako ostatní podobné nástroje příkazové řádky viz Výpis kódu 3.1.

■ Výpis kódu 3.1 Nápověda nástroje `dsb`

```
$ dsb -h
Usage: dsb ( printInfo | build | restore | verify ) [OPTIONS]
-h, --help            this help
-v, --verbose         verbose output
-i, --input           input file
-o, --output          output file
-s, --invalidBlocks <path> file with invalid blocks
-d, --outputDiskId <diskId> diskId to restore

InvalidBlocks file format:
[disk <diskId>          first line defines diskId
 [<offset> <length>]*]+ next is block of tuples offset and length of block
                        which are invalid

# Examples

- Generate parity
dsb build -i disk1.dd -i disk2.dd -o parity.dsb

- Verify images
dsb verify -i parity.dsb -i disk1.dd -i disk2.dd

- Restore missing image
dsb restore -i parity.dsb -i disk1.dd -o restored_disk2.dd

- Restore image from other images with invalid blocks
```

```
dsb restore -i parity.dsb -i disk1.dd -i disk2.dd -s invalid_blocks.txt -d 2 -o
  restored_disk2.dd
```

```
File invalid_blocks.txt looks like:
disk 1
0 512
```

3.2 Vytvoření zálohy

Vytvořím *paritní soubor* `parity.dsb` ve formátu *DSB* z existujících disků `disk1.iso` a `disk2.iso`:

■ Výpis kódu 3.2 Tvorba *paritního souboru* `parity.dsb`

```
$ dsb build -i disk1.iso -i disk2.iso -o parity.dsb
BuildImage: Build starting. (100%)
GenerateGuid: Generated guid= {07bc5277-e7f9-8e41-b3c9-9d6d485d9a8a}
BuildImage: Build succeeded.
```

3.3 Zobrazení informací v paritním souboru

Zobrazím informace uložené v *paritním souboru* `parity.dsb`, který jsem vytvořil v předchozím kroku:

■ Výpis kódu 3.3 Zobrazení informace souboru `parity.dsb`

```
$ dsb printInfo -i parity.dsb
parity.dsb:
  groupId=      {07bc5277-e7f9-8e41-b3c9-9d6d485d9a8a}
  sizeo=        1 blocks | 512 B
  crc32Header=  0xb191072a
  countofDisks= 3
  createTime=   Mon May 23 00:32:39 2022
  disk 0:
    typeofDisk= 2 | PARITY
    diskId=      0
    size=        1266184 blocks | 648286208 B
    sha256=      68c4c749a06f849fdab8f205f86021b40a354d292eca68a999d8315f8796803d
  disk 1:
    typeofDisk= 1 | RAW
    diskId=      1
    size=        315544 blocks | 161558528 B
    sha256=      1a715ae4196edc17af4b9cd39e1be7df6dcd83885782c4a3cf0d8ce381129fa6
  disk 2:
    typeofDisk= 1 | RAW
    diskId=      2
    size=        1266184 blocks | 648286208 B
    sha256=      efe3eab2b3bb6ee959a13f997978369f5576b95d0736fd5e4a3eb9946663a358
```

3.4 Otestování konzistence disků

Pro otestování konzistence *záložních disků* mohu využít informací, které jsou uloženy v *paritním souboru*. Jedná se hlavně o velikost a hash disku. K tomu nástroj `dsb` podporuje příkaz `verify`:

■ Výpis kódu 3.4 Testování konzistence záložních disků

```
$ dsb verify -i parity.dsb -i disk1.iso -i disk2.iso
VerifyImage: Verify starting.
VerifyImage: Counting hash of disk 0. (100%)
VerifyImage: Verification of disk 0 succeeded. diskId= 0
VerifyImage: Counting hash of disk 1. (100%)
VerifyImage: The image has been found in the table. diskId= 1 name= disk1.iso
VerifyImage: Counting hash of disk 2. (100%)
VerifyImage: The image has been found in the table. diskId= 2 name= disk2.iso
VerifyImage: Verify succeeded.
```

Zde vidím, že všechny disky byly zkontrolovány a obsahují očekávaná data. V případě, když některý soubor neobsahuje správná data, můj nástroj o tom náležitě informuje:

■ Výpis kódu 3.5 Testování konzistence disků našlo chybu

```
$ dsb verify -i parity.dsb -i disk1.iso -i disk2_mod.iso
VerifyImage: Verify starting.
VerifyImage: Counting hash of disk 0. (100%)
VerifyImage: Verification of disk 0 succeeded. diskId= 0
VerifyImage: Counting hash of disk 1. (100%)
VerifyImage: The image has been found in the table. diskId= 1 name= disk1.iso
VerifyImage: Counting hash of disk 2. (100%)
VerifyImage: ERROR: The image has not been found in the table or it is not a valid
image. path= disk2_mod.iso
VerifyImage: ERROR: There are some invalid images. count= 1
```

3.5 Obnovení ztraceného disku

Obnovím ztracený disk `disk2~.iso` z existujícího disku `disk1.iso` a *paritního souboru* `parity.dsb`:

■ Výpis kódu 3.6 Obnovení ztraceného disku

```
$ dsb restore -i parity.dsb -i disk1.iso -o disk2~.dsb
Restore: Restore starting.
RestoreMissingImage: Counting hash of disk parity.dsb. (100%)
RestoreMissingImage: Counting hash of disk disk1.iso. (100%)
RestoreMissingImage: Restoring disk 2. (100%)
Restore: Restore succeeded.
```

Ve výpisu vidím, že se obnovení disku povedlo bez chyby. V případě, když *paritní soubor* nebo některý obraz disku neobsahuje data, která byla použita při vytváření *paritního souboru*, nástroj nás na tuto skutečnost upozorní:

■ Výpis kódu 3.7 Chyba při obnovování ztraceného disku

```
$ dsb restore -i parity.dsb -i disk1.iso -o disk2~.iso
Restore: Restore starting.
RestoreMissingImage: Counting hash of disk parity.dsb. (100%)
RestoreMissingImage: Counting hash of disk disk1.iso. (100%)
RestoreMissingImage: ERROR: Image is not in group or hash does not match. path= disk1.
iso
```

3.6 Obnovení poškozeného disku

Obnovím disk `disk1.iso` s vadnými sektory z existujících disků `disk1.iso` a *paritního souboru* `parity.dsb` a uložím ho pod jménem `disk2~.iso`:

■ **Výpis kódu 3.8** Obnovení částečně poškozeného disku

```
$ cat invalid_sectors.txt
disk 1
32768 512
65024 1536
disk 2
2048 512

$ dsb restore -i parity.dsb -i disk1.iso -i disk2.iso -s invalid_dvd_sectors.txt -d 1 -
  o disk1~.iso
Restore: Restore starting.
RestoreImageWithInvalidBlocks: Writing output data. (100%)
Restore: Restore succeeded.
```

Zde vidím, že operace proběhla bez chyby. V případě, že by nastala chyba, dozvíme se to z výstupního logu nástroje:

■ **Výpis kódu 3.9** Chyba při obnovování porušeného disku

```
$ cat invalid_sectors_overlap.txt
disk 1
32768 512
65024 1536
disk 2
65536 512

$ dsb restore -v -i parity.dsb -i disk1.iso -i disk2.iso -s invalid_sectors_overlap.txt
  -d 1 -o disk1~.iso
ParseInvalidBlocksFile: ERROR: Overlapping intervals (65024,L1536), (65536,L512) of
  invalid blocks found.
```

V tomto výpisu jsem zadal překrývající se intervaly, které popisují stejný blok na dvou nebo více obrazech disku. Tuto chybu nástroj `dsb` neumí opravit. Vždy je třeba mít maximálně 1 chybu ve všech vstupních souborech.

Cílem práce bylo popsat postup, jak vytvářet spolehlivé zálohy libovolných dat na *optické disky*.

Při rešerši jsem nenašel žádné podobné řešení, které by se soustředilo na domácí uživatele, nízkou pořizovací cenu nejen médií, ale i zařízení pro zápis na tato média, na vysokou spolehlivost záloh, jednoduchost použití nástrojů, případně multiplatformnost, kdy lze všechny postupy provádět jak na OS Windows, tak na OS Linux.

Přesvědčil jsem se, že využití moderních *optických disků* pro domácího uživatele je ekonomické s ohledem na životnost jednotlivých médií. Speciálně pokud se bavíme o Blu-ray / DVD discích využívajících technologii M-DISC dosahuje výdrže mnoha stovek let, což je mnohem více, než délka lidského života.

Dále jsem zkoumal způsob ukládání dat pomocí technologie RAID, kde jsem se inspiroval při návrhu algoritmu, který jsem posléze využil v další části při tvorbě parity pro skupinu disků.

V praktické části této práci jsem se soustředil na zvýšení spolehlivosti zálohy na *optické disky*. Využil jsem k tomu vytvoření dalšího souboru, který spojí obrazy disků a vytvoří *paritní soubor* jako xor jednotlivých obrazů disků. Při navrhování tohoto algoritmu jsem využil znalostí, které jsem získal z předchozí analýzy technologie RAID a jeho způsobu ukládání dat. Pro potřeby *optických disků* jsem vyšel z technologie RAID 4 a tento algoritmus jsem modifikoval způsobem, který více vyhovuje faktu, že zápis na disky lze provést jen jednou.

Testování vytváření záloh různých souborů jsem provedl na obrazech disků CD o velikosti do 700 MiB a DVD o velikosti 4.1 GiB ve formátu iso, abych také prověřil možnost práce s diskem, který je větší než 4 GiB. Nejprve jsem vyzkoušel první implementovaný algoritmus pro obnovu ztraceného disku a poté i druhý, který umí obnovit data z disků které mají některé bloky dat porušené tak, že je nelze přechít. Zde vybírám všechny kombinace porušených bloků dat.

Zadání práce jsem splnil. Vytvořil jsem postup, pomocí kterého si může kdokoliv vytvořit vlastní zálohu dat, a nástroj, který spolehlivost zálohy ještě zvýší.

Literatura

- [1] CBURNETT. Tento vektorový obrázek byl vytvořen programem Inkscape., CC BY-SA 3.0, [online]. 31.12.2006 [cit. 2022-05-19]. Dostupné z: https://commons.wikimedia.org/wiki/File:RAID_0.svg
- [2] ČERNÝ, Radek. Zálohování dat a datová úložiště [online]. Brno, 2020 [cit. 2020-05-15]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/127661>. Diplomová práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Jiří Kříž.
- [3] Dd (Unix) – Wikipedie [online]. [cit. 2022-06-01]. Dostupné z: [https://cs.wikipedia.org/wiki/Dd_\(Unix\)](https://cs.wikipedia.org/wiki/Dd_(Unix))
- [4] EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION. Standard ECMA-130: data interchange on read-only 120 mm optical data disks (cd-rom). [online]. 1988 [cit. 2022-06-12]. Dostupné z: <https://books.google.cz/books?id=fPaXQwAACAAJ>
- [5] How long will DVD discs last? [online]. 14. dubna 2021 [cit. 2022-05-13]. Dostupné z: <https://www.sony.com/electronics/support/articles/00009195>
- [6] HP StorageWorks LTO-6 Ultrium 6250 SAS External - BOHEMIA COMPUTERS [online]. [cit. 2022-05-13]. Dostupné z: <https://www.bohemiapc.cz/hp-storageworks-lto-6-ultrium-6250-sas-external/>
- [7] HPE LTO-8 30TB RW Data Cartridge - Kazeta — Alza.cz [online]. [cit. 2022-05-13]. Dostupné z: <https://www.alza.cz/hpe-lto-8-30tb-rw-data-cartridge-d6870425.htm>
- [8] IBM CORPORATION. Optical volume backup [online]. 2022-05-03 [cit. 2022-06-13]. Dostupné z: <https://www.ibm.com/docs/en/i/7.5?topic=storage-optical-volume-backup>
- [9] ISO 9660 - OSDev Wiki [online]. [cit. 2022-06-01]. Dostupné z: https://wiki.osdev.org/ISO_9660
- [10] KOCNOVÁ, Jitka. Forenzní analýza disků a metadat pod OS Linux. Brno, 2014. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. 2014-06-18. Vedoucí práce Očenášek Pavel. Dostupné z: <https://www.fit.vut.cz/study/thesis/16564/>
- [11] LANGER, Langer. *Optická paměťová média, principy, využití, trendy* [online]. Brno, 2011 [cit. 2022-06-07]. Dostupné z: <https://is.muni.cz/th/b9tj4/>. Bakalářská práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce Jan STAUDEK.

- [12] More than 100 years projected lifetime for DVD+R 2.4X / 4X [online]. [cit. 2022-05-13]. Dostupné z: http://www.verbatim-europe.com/includes/binary_details.php?id=454
- [13] Optical disc image - Wikipedia [online]. [cit. 2022-06-01]. Dostupné z: https://en.wikipedia.org/wiki/Optical_disc_image
- [14] PARHAMI, Behrooz. Data Longevity and Compatibility [online]. 2018 [cit. 2022-05-13]. Dostupné z: https://web.ece.ucsb.edu/~parhami/pubs_folder/parh19f-ebdt-data-longevity-compatib.pdf
- [15] RAID – Wikipedie [online]. [cit. 2022-05-27]. Dostupné z: <https://cs.wikipedia.org/wiki/RAID>
- [16] REICHL, Jaroslav a Martin VŠETIČKA. Princip záznamu a reprodukce zvuku [online]. [cit. 2022-06-12]. Dostupné z: <http://fyzika.jreichl.com/main.article/view/409-princip-zaznamu-a-reprodukce-zvuku>
- [17] Seznam závěrečných prací na téma Zálohování dat a datová úložiště [online]. [cit. 2022-04-22]. Dostupné z: <https://theses.cz/vyhledavani/?search=Z%C3%A1lohov%C3%A1n%C3%AD+dat+a+datov%C3%A1+C3%BA1o%C5%BEi%C5%A1t%C4%9B>
- [18] Solid-state drive – Wikipedie [online]. [cit. 2022-05-27]. Dostupné z: https://cs.wikipedia.org/wiki/Solid-state_drive
- [19] STORAGE NETWORKING INDUSTRY ASSOCIATION (SNIA). Common RAID Disk Data Format Specification [online]. Version 2.0 Revision 19. 2009-03-27 [cit. 2022-06-19]. Dostupné z: https://www.snia.org/sites/default/files/SNIA_DDF_Technical_Position_v2.0.pdf
- [20] ŠILHÁN, Jiří. Rozšířené zálohování pro OS Windows. Brno, 2013. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. 2013-06-10. Vedoucí práce Očenášek Pavel. Dostupné z: <https://www.fit.vut.cz/study/thesis/15258/>
- [21] Universal Disk Format – Wikipedie [online]. [cit. 2022-05-29]. Dostupné z: https://cs.wikipedia.org/wiki/Universal_Disk_Format
- [22] VERBATIM CD/DVD Slimline, černá - Externí vypalovačka — Alza.cz [online]. [cit. 2022-05-31]. Dostupné z: <https://www.alza.cz/verbatim-mobile-dvd-rewriter-usb-2-0-black-light-version-d5513938.htm>
- [23] VLADYKA, Vojtěch. Nástroje pro diagnostiku integrity souborového systému v OS Linux [online]. Brno, 2017 [cit. 2022-05-28]. Dostupné z: <http://hdl.handle.net/11012/65230>. Diplomová práce. Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav automatizace a měřicí techniky. Vedoucí práce Petr Petyovský.

Ceny médií v obchodech

Ceny uložení 1 TB za rok pro různé druhy *optických disků* (cd, dvd, dvd_100, bd, m-disc) jsou v tabulce A.1. Pro srovnání je také uvedena cena uložení dat na pevný disk (hdd).

Popis jednotlivých sloupců:

- **Minimální výdrž** – je slovní popis, jak je uveden přímo v příslušném obchodu.
- **Výdrž numericky** – převedení předchozí hodnoty na číslo. Pokud není hodnota výdrže známá přímo z původního zdroje, využil jsem jiné zdroje, které popisují tuto hodnotu [14].
- **Cena balení** – cena celého balení.
- **Disků v balení** – počet jednotlivých disků v balení.
- **Velikost média** – velikost jednoho disku.
- **Cena za uložení TB/rok Kč** – cena za uložení 1 TB na rok v Kč.
- **Cena za uložení TB/50 let Kč** – cena za uložení 1 TB na 50 let v Kč. Zde aplikuji algoritmus, který blíže popisuji v kapitole 1.1.7.

Šedě jsou zvýrazněny ty řádky, které jsou pro mě nejvýhodnější, protože nabízejí nejnižší cenu za uložení 1 TB.

Prodejce obchod	Typ média	Jméno	Cena platná dne	Minimální výdř	Výdř numerický	Cena balení	Disků v balení	Velikost média	Cena za uložení	Cena za uložení	Cena za uložení na 50 let
https://www.alza.cz			datum	popis	let	kč	#	GB	TB	TB/rok Kč	TB/50 let Kč
	cd	MediaRange CD-R blank 100ks shrink	13.03.2022			10	469	100	0,7	6700	670
	dvd	MEDIARANGE DVD-R 4,7GB 16x slim case 5ks	13.03.2022			10	69	5	4,7	2936	294
	dvd	MEDIARANGE DVD-R 4,7GB 16x spindle 25ks	13.03.2022			10	179	25	4,7	1523	152
	dvd	MEDIARANGE DVD-R 4,7GB 16x spindle 50ks	13.03.2022			10	359	50	4,7	1528	153
	dvd	MEDIARANGE DVD-R 4,7GB 16x blank 60ks 100ks	13.03.2022			10	509	100	4,7	1088	108
	dvd	MEDIARANGE DVD-R 4,7GB 16x spindle 10ks	13.03.2022			10	79	10	4,7	1681	168
	dvd_100	MEDIARANGE DVD-R 4,7GB 8x Dual Layer spindle 10ks	13.03.2022			10	179	10	8,5	2106	211
	dvd_100	VERBATIM DVD-R AZO 4,7GB .16x, colour, slim case 5 ks	13.03.2022	více než 100 let		100	90	5	4,7	3830	3830
	dvd_100	VERBATIM DVD-R AZO 4,7GB .16x, wrap 10 ks	13.03.2022	více než 100 let		100	90	10	4,7	2106	211
	dvd_100	VERBATIM DVD-R AZO 4,7GB .16x, spindle 25 ks	13.03.2022	více než 100 let		100	209	25	4,7	1779	1779
	dvd_100	VERBATIM DVD-R DL AZO 8,5GB .8x, spindle 10 ks	13.03.2022	více než 100 let		100	339	10	8,5	3988	3988
	bd	VERBATIM DVD+R AZO 4,7GB .16x, printable, spindle 50 ks	13.03.2022	více než 100 let		100	389	50	4,7	1655	1655
	bd	VERBATIM BD-R SL DataLife 25GB, 6x, jewel case 5 ks	13.03.2022			50	169	5	25	1352	1352
	bd	MEDIARANGE BD-R BLU-RAY 25GB 4x spindle 10ks	13.03.2022			50	189	10	25	756	756
	bd	MEDIARANGE BD-R BLU-RAY 25GB 4x spindle 25ks	13.03.2022			50	459	25	25	734	734
	bd	VERBATIM BD-R DL 50GB, 6x, jewel case 5 ks	13.03.2022			50	499	6	50	1663	1663
	bd	VERBATIM BD-R SL DataLife 25GB, 6x, printable, spindle 50 ks	13.03.2022			50	829	50	25	663	663
	bd	VERBATIM BD-R DL 50GB, 6x, spindle 10 ks	13.03.2022			50	939	10	50	1878	1878
	bd	VERBATIM BD-R XL 100GB, 4x, printable, jewel case 5 ks	13.03.2022			50	1299	5	100	2598	2598
	bd	VERBATIM M-DISC BD-R SL 25GB, 4x 60min case 3 ks	13.03.2022	1000 let		1000	249	3	25	3320	3320
	m-disc	VERBATIM M-DISC BD-R SL 25GB, 4x 60min case 5 ks	13.03.2022	1000 let		1000	509	5	25	4072	4072
	m-disc	VERBATIM M-DISC BD-R XL 100GB, 4x, printable, jewel case 5 ks	13.03.2022	1000 let		1000	1999	5	100	3998	3998
	hdd	Seagate Barracuda 1TB	16.03.2022			5	999	1	1000	999	9990
	hdd	Seagate Barracuda 2TB	16.03.2022			5	1182	1	2000	591	118
	hdd	Seagate Barracuda 3TB	16.03.2022			5	1999	1	3000	666	133
	hdd	WD Blue 4TB	16.03.2022			5	2199	1	4000	550	110
	hdd	Seagate Barracuda Laptop 5TB	16.03.2022			5	2790	1	5000	558	112
	hdd	WD Blue 6TB	16.03.2022			5	2999	1	6000	500	100
	hdd	Seagate Barracuda 8TB	16.03.2022			5	4690	1	8000	586	117
	hdd	WD Elements Desktop 10TB	16.03.2022			5	6129	1	10000	613	123
	hdd	WD BLACK D10 Game drive 12TB, černý	16.03.2022			5	7499	1	12000	625	125
	hdd	Seagate Exos X16 14TB	16.03.2022			5	8790	1	14000	628	126
	hdd	Seagate Exos X16 16TB	16.03.2022			5	8990	1	16000	562	112
	hdd	Western Digital 18TB Ultrastar DC HC550 SATA	16.03.2022			5	9899	1	18000	550	110
	hdd	Seagate Exos X20 20TB SATA	16.03.2022			5	13290	1	20000	665	133
	hdd	Verbatim BD-R, 6x HTL, 25GB, printable, 10 ks, spindle	13.03.2022			50	167	10	25	668	668
	m-disc	Verbatim BD-R, M-Disc, 4x, 25GB, printable, 5 ks, jewel	13.03.2022	1000 let		1000	509	5	25	4072	4072
	m-disc	Verbatim BD-R, M-Disc, 4x, 100GB, jewel 5 ks	13.03.2022	1000 let		1000	949	10	25	3796	3796
	m-disc	Verbatim BD-R 25GB 4x, M-Disc, printable, jewel, 5ks (43823)	16.03.2022			1000	1999	5	100	3998	3998
	m-disc	Verbatim BD-R 100GB 4x, M-Disc, printable, jewel, 5ks (43834)	16.03.2022			1000	461	5	25	3688	3688
	m-disc	Verbatim BD-R 25GB 4x, M-Disc, printable, spindle, 10ks (43825)	16.03.2022			1000	1841	5	100	3682	3682
	m-disc	Verbatim BD-R 25GB 4x, M-Disc, printable, spindle, 10ks (43825)	16.03.2022			1000	947	10	25	3788	3788

■ **Obrázek A.1** Ceny uložení 1 TB za rok pro různé *optické disky*

Obsah přiloženého média

dsb.....	Zdrojové kódy implementace
└─ doc.....	Adresář s vygenerovanou dokumentací pomocí nástroje doxygen
─ exe.....	Adresář se spustitelnou verzí nástroje
─ earchitect	Diagramy ve formátu Enterprise Architect
─ thesis	Zdrojová forma práce ve formátu L ^A T _E X
─ jiricpe1-BACHELOR.pdf.....	Text práce ve formátu PDF