VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

POROVNÁNÍ SCHOPNOSTÍ NÁSTROJŮ PRO DATA CARVING Z PEVNÝCH DISKŮ

COMPARISON OF FILE CARVING TOOLS FOR HARD DRIVE IMAGES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **LUKÁŠ DUKA**
AUTHOR

VEDOUCÍ PRÁCE **Ing. JAN PLUSKAL, Ph.D.**
SUPERVISOR

BRNO 2025

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

164549

Ústav: Ústav informačních systémů (UIFS)
Student: **Duka Lukáš**
Program: Informační technologie
Název: **Porovnání schopností nástrojů pro data carving z pevných disků**
Kategorie: Data mining
Akademický rok: 2024/25

Zadání:

1. Proveďte průzkum existujících formátů diskových oddílů, souborových systémů, které se používají v desktopových počítačích, pracovních stanicích, serverech a vestavěných zařízeních dle konzultace s vedoucím. Neopomeňte multimediální data, která jsou typická pro obnovu nástroji z bodu 3.
2. Nastudujte problematiku obnovy smazaných/poškozených souborů, souborových systémů a formátů diskových oddílů na identifikovaných položkách z bodu 1. Proveďte rešerši odborných publikací a identifikujte možnosti smazání i poškození položek z bodu 1.
3. Nalezněte volně dostupné i komerční nástroje schopné řešit problematiku z bodu 2. Zaměřte se na porovnání vlastností těchto nástrojů s využitím běžně používaných metrik, které jsou k dispozici a použité autory jednotlivých řešení.
4. Zjistěte, zdali existují volně dostupné datové sady pro porovnání nástrojů z bodu 3. Pokud ano, diskutujte jejich vlastnosti vzhledem k získaným informacím z bodu 3. Pokud identifikujete, po konzultaci s vedoucím, že nějaká vlastnost chybí či sady neexistují, navrhněte a vytvořte vlastní. Datová sada musí zahrnovat všechny identifikované položky z bodu 1, izolovaně i navzájem kombinovaně.
5. Proveďte měření nástrojů identifikovaných v bodu 3 a dbejte připomínek vedoucího.
6. Zhodnoťte úspěšnost měření z bodu 5 a diskutujte dosažené výsledky.

Literatura:

- BURGHARDT, Aaron a Adam J. FELDMAN, 2008. Using the HFS+ journal for deleted file recovery. *Digital Investigation* [online]. **5**, The Proceedings of the Eighth Annual DFRWS Conference, S76–S82. ISSN 1742-2876.
- CASEY, Eoghan, Alex NELSON a Jessica HYDE, 2019. Standardization of file recovery classification and authentication. *Digital Investigation* [online]. 31, 100873. ISSN 1742-2876.
- LEE, Seokjun a Taeshik SHON, 2014. Improved deleted file recovery technique for Ext2/3 filesystem. *The Journal of Supercomputing* [online]. 70(1), 20–30. ISSN 1573-0484.
- NA, Gi-Hyun, Kyu-Sun SHIM, Ki-Woong MOON, Seong G. KONG, Eun-Soo KIM a Joong LEE, 2014. Frame-Based Recovery of Corrupted Video Files Using Video Codec Specifications. *IEEE Transactions on Image Processing* [online]. 23(2), 517–526. ISSN 1941-0042.

Při obhajobě semestrální části projektu je požadováno:
Body 1, 2, 3 a rozpracovaný bod 4.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Pluskal Jan, Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1.11.2024
Termín pro odevzdání: 14.5.2025
Datum schválení: 18.10.2024

Abstrakt

Tato práce se zabývá komplexní analýzou a hodnocením různých nástrojů pro obnovu dat z pevných disků. Práce poskytuje přehled vybraných souborových systémů využívající se v praxi. Zabývá se také principy multimediálních dat a technikami jejich obnovy. Hlavní část je věnována komparativní analýze a hodnocení vybraných nástrojů pro obnovu dat, zahrnující jak volně dostupné, tak komerční programy. Pro účely této analýzy byla navržena a pomocí vlastních skriptů automatizovaně vygenerována komplexní datová sada, která slouží pro měření účinnosti jednotlivých programů. Výsledkem práce je vyhodnocení a doporučení vhodných programů v různých kontextech obnovy dat.

Abstract

This thesis focuses on a comprehensive analysis and comparison of different tools for data carving from hard disks. It provides an overview of selected file systems commonly used in practice. It also discusses the principles of multimedia data and techniques of data recovery. The main part of the thesis is devoted to a comparative analysis and evaluation of selected carving tools, including open-source and commercial programs. For the purposes of this analysis, a complex dataset was designed and automatically generated using custom scripts to assess the performance of individual tools. The result of the work is an evaluation and recommendation of appropriate programs in different data recovery contexts.

Klíčová slova

obnova dat, rekonstrukce dat, smazané soubory, vyřezávání dat, digitální forenzní analýza, souborový systém

Keywords

data recovery, data reconstruction, deleted files, data carving, digital forensic analysis, file system

Citace

DUKA, Lukáš. *Porovnání schopností nástrojů pro data carving z pevných disků.* Brno, 2025. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jan Pluskal, Ph.D.

Porovnání schopností nástrojů pro data carving z pevných disků

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jana Pluskala, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal. Pro gramatickou kontrolu a stylistickou úpravu vlastního textu jsem použil nástroje generativní AI.

.....
Lukáš Duka
12. května 2025

Poděkování

Rád bych poděkoval Ing. Janu Pluskalovi, Ph.D. za jeho odborné vedení bakalářské práce a poskytnuté rady během tvorby práce. Rovněž bych chtěl poděkovat své rodině za pomoc a podporu během studia.

Obsah

Seznam obrázků

Kapitola 1

Úvod

V současné době je pod pojmem „data“ obecně chápána informace v digitální podobě, uložená na nejrůznějších typech paměťových médií. Mezi ně lze zařadit mobilní telefony, osobní počítače, pevné disky či paměťové karty. Zatímco dříve šlo především o jednodušší typy dat, jejich rozsah se v posledních dekádách výrazně rozšířil. Zahrnuje širokou škálu informací od záznamů každodenních událostí zachycených digitálními kamerami, přes rozsáhlá kancelářská až po komplexní vědecká data generovaná výzkumnými projekty. Data se tak stala nepostradatelnou součástí lidského fungování. Podílejí se také na efektivním fungování rozmanitých průmyslových odvětví, včetně dopravy a obchodu. Každá transakce, logistický pohyb nebo rozhodovací proces zanechává digitální stopu, která slouží k monitorování, řízení i optimalizaci systémů. Z toho plyne, že současná lidská existence je ve značné míře závislá na využívání dat a jakákoli ztráta či poškození může představovat závažný problém.

Tato situace je úzce spojená s neustálým nárůstem objemu dat ukládaných na paměťová média. Z těchto důvodů nabývá obnova dat stále většího významu, a to nejen v kontextu forenzní analýzy, ale i při běžném užívání výpočetní techniky. Smazané, poškozené nebo nedostupné soubory mohou mít závažné důsledky, a proto existuje řada metod a nástrojů, které umožňují jejich rekonstrukci. Na trhu je dostupných mnoho nástrojů, jejichž cílem je efektivní obnova dat. Každý z nich se ovšem liší mírou spolehlivosti, podporovanými formáty, výkonností či metodikou obnovy. Právě tyto rozdíly představují hlavní motivaci pro vznik této práce. Jejím cílem je poskytnout porovnání vybraných nástrojů a zhodnotit jejich efektivitu v procesu obnovy dat.

Kapitola 2 se zaměřuje na logické rozdělení dat v rámci úložiště. Zahrnuje popis hlavních formátů diskových oddílů, které slouží k organizaci a správě dat. Přestože existuje velké množství souborových systémů, tato práce se omezuje pouze na vybrané. Kapitola 3 se věnuje multimediálním datům a jejich metodám obnovy. Kapitola 4 prezentuje výběr volně dostupných a komerčních nástrojů, jejichž primárním zaměřením je obnova dat. Cílem je teoretické srovnání vlastností na základě dostupných informací. Pro účely měření vybraných programů byla vytvořena komplexní datová sada zahrnující různé kombinace souborových systémů a multimediálních dat. Samotný návrh je popsán v kapitole 5, na kterou navazuje kapitola 6 s jeho implementací. Závěrečná kapitola 7 je věnována průběhu měření jednotlivých programů s využitím vytvořené datové sady. Součástí této kapitoly je také vyhodnocení dosažených výsledků.

Kapitola 2

Logická struktura datových úložišť

Ukládání informací v digitální podobě je pro naprostou většinu populace běžnou součástí každodenního života. Počítače, chytré telefony a další elektronická zařízení umožňují uchovávat rozsáhlé objemy dat, a to v nejrůznějších formátech. Aby bylo možné s těmito daty pracovat, je důležité, aby jejich ukládání probíhalo podle předem definovaných pravidel. Bez zavedení vhodné logické struktury by veškerá data existovala pouze jako neuspořádaná posloupnost bitů, což by znemožnilo jejich systematickou správu. Z tohoto důvodu je nutné využívat mechanismy, které umožní jednoznačně určit, kde a jak je každý jednotlivý bit na paměťovém médiu uložen.

V období před nástupem digitální revoluce byla situace zcela odlišná. Ukládání dat bylo omezeno na fyzická média, jako je papír, které představovaly primární nosiče informací. Archivace dat probíhala především manuálně, často závisela na fyzickém uspořádání dokumentů. První principy organizační struktury se začaly tvořit právě v této době. S příchodem výpočetní techniky se otevřely nové možnosti pro ukládání a správu dat. Jedním z prvních automatizovaných typů úložišť se stal děrný štítek. Nicméně, vzhledem ke svým omezením, byl postupem času nahrazen magnetickou páskou, která nabízela vyšší rychlost zápisu a čtení dat. Dalším krokem bylo zdokonalení magnetické pásky do podoby magnetických disků. Ty se v současnosti běžně využívají v různých typech elektronických zařízení ve formě pevných disků [16]. Následující kapitola se proto věnuje přehledu hlavních prostředků pro správu dat v rámci datových úložišť[1].

2.1 Správa logických disků

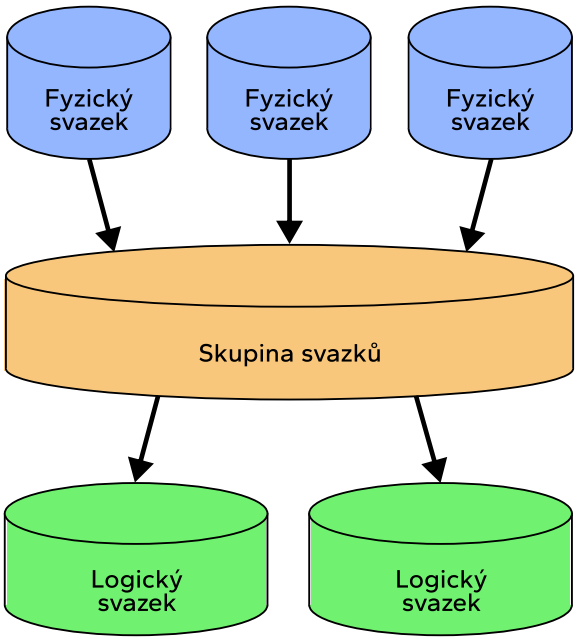
Správa logických disků, také známá pod zkratkou LVM (*Logical Volume Management*) představuje flexibilní vrstvu pro správu úložných zařízení. Ta funguje jako mezivrstva mezi fyzickými disky a souborovými systémy vytvořenými nad ní. LVM umožňuje dynamickou a efektivní správu diskového prostoru ve srovnání se statickým dělením disků. Informace uvedené v této sekci byly čerpány z publikace [27].

Tato architektura je založena na třech základních komponentách:

- **Fyzické svazky** – jsou základní stavební bloky systému LVM, obvykle celé disky nebo diskové oddíly, které byly inicializovány pro použití v rámci LVM.

[1] Po konzultaci s vedoucím byly pro bod 1 zadání vybrány následující souborové systémy: *FAT*, NTFS, ext3/4, btrfs, HFS+, APFS* a tabulky oddílů: *MBR, GPT*.

- **Skupiny svazků** – reprezentují abstrakční vrstvu, která spojuje jeden nebo více fyzických svazků do skupiny. Ta poskytuje souvislý prostor pro alokaci logických svazků.
- **Logické svazky** – se vytvářejí z prostoru definovaného skupinou svazků. Tyto svazky se pak operačnímu systému jeví jako standardní bloková zařízení, na kterých lze vytvořit souborový systém (např. ext4).



Obrázek 2.1: Schéma propojení jednotlivých komponent LVM. Převzato z [27].

Hlavní výhody LVM spočívají ve vysoké flexibilitě při správě úložného prostoru. Logické svazky lze snadno vytvářet, mazat a dynamicky měnit jejich velikost. V určitých případech toho lze dosáhnout i bez nutnosti odpojení souborového systému. LVM umožňuje, aby byl logický svazek rozložen přes více fyzických disků. Tím dochází k oddělení logické struktury svazku od fyzického rozmístění dat. Data jednoho svazku tak mohou být rozptýlena mezi různá zařízení, což zvyšuje efektivitu využití úložné kapacity.

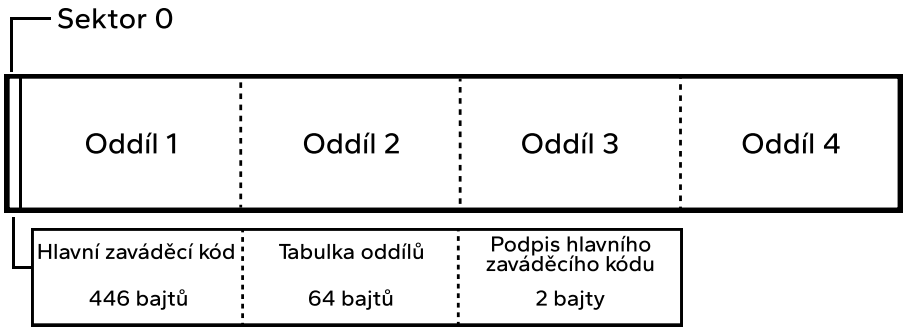
LVM také umožňuje vytváření snímků, které zachycují stav logického svazku v určitém časovém okamžiku. Tato funkcionalita je důležitá pro účely zálohování a obnovy systémů. Na rozdíl od pokročilých souborových systémů, jako ZFS nebo Btrfs, jsou však na snímky v LVM vázána určitá omezení. Při vytvoření snímku je alokováno pevné množství úložného prostoru z objemu skupiny svazků, který je následně využíván k uchovávání kopií původních bloků při jejich modifikaci (tzv. mechanismus *copy-on-write*). Díky podpoře migrace dat lze obsah logických svazků přesouvat mezi různými fyzickými zařízeními bez nutnosti jakéhokoliv přerušení provozu systému.

2.2 Formáty diskových oddílů

Správa diskových oddílů je důležitým prvkem organizační struktury pevných disků. Hlavním cílem rozdělení disku do několika oblastí je zajištění nezávislé správy dané oblasti. Ta se označuje jako oddíl. Jednou z primárních výhod správy diskových oddílů je možnost logického oddělení dat. Takovéto rozdělení umožňuje využití různých souborových systémů na jednotlivých oddílech. To je také vhodné pro instalaci více operačních systémů na jednom fyzickém disku. Každý operační systém totiž vyžaduje vlastní oddíl s odpovídajícím souborovým systémem. Jednotlivé formáty tabulek oddílů definují způsoby, jakými jsou data na disku strukturována a organizována. Tyto formáty mají vliv nejen na rychlost přístupu k datům, ale také na spolehlivost úložiště a jeho schopnost obnovy v případě selhání. Tato sekce se zaměřuje na přehled jednotlivých formátů diskových oddílů.

2.2.1 MBR

Master Boot Record, zkáceně MBR, je starší formát tabulky oddílů, který se stal standardem pro organizaci oddílů na pevných discích. Jednou z jeho hlavních limitací je podpora diskových médií o maximální kapacitě 2 TB. Kvůli omezenému množství alokovaného prostoru v rámci načítacího sektoru je možné vytvořit nejvýše 4 primární oddíly. Struktura MBR je umístěna v prvních 512 bajtech disku [8]. Jak ukazuje obrázek 2.2, zahrnuje hlavní spouštěcí kód, tabulku diskových oddílů a identifikační kód. Hlavní spouštěcí kód o velikosti 446 bajtů zajišťuje předání řízení operačnímu systému, jehož umístění je zaznamenáno v tabulce diskových oddílů. Tabulka diskových oddílů o celkové velikosti 64 bajtů uchovává informace o jednotlivých diskových oddílech a jejich fyzickém umístění na disku. Pro každý z oddílů je maximální velikost 16 bajtů. Celý sektor MBR je zakončen dvoubajtovým identifikačním kódem, který slouží k jeho validaci jako platného spouštěcího záznamu.



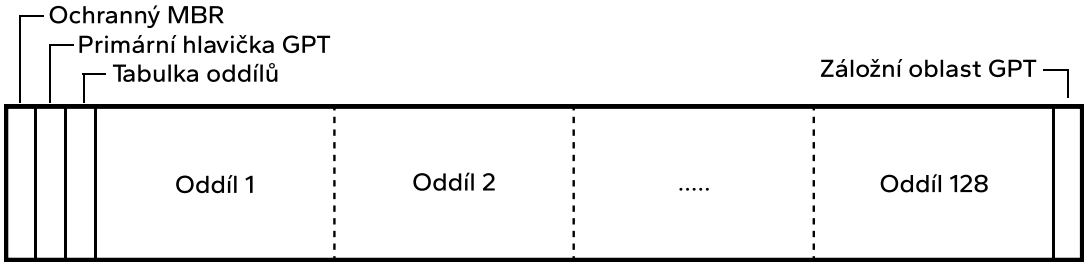
Obrázek 2.2: Schéma rozložení diskového úložiště s použitím MBR. Převzato z [29].

Nevýhodou MBR je jeho zranitelnost v případě fyzického poškození disku nebo narušení uložených dat. Všechny informace, včetně spouštěcího kódu, tabulky oddílů a identifikačního kódu, jsou uloženy na jediné souvislé oblasti na disku. Při poškození MBR jsou data obtížně obnovitelná. Pokud je nutné vytvořit více než 4 primární oddíly, je možné využít tzv. rozšířený oddíl (*Extended Partition*). V rámci tohoto rozšířeného oddílu lze definovat více logických oddílů. V praxi tento mechanismus funguje tak, že rozšířený oddíl má vlastní spouštěcí sektor, označovaný jako *Extended Boot Record* (EBR) [29]. Ten obsahuje informace o logickém rozložení oddílů uvnitř rozšířeného oddílu. Každý logický oddíl má vlastní EBR, který odkazuje na následující logický oddíl, nebo signalizuje konec rozšířeného oddílu.

2.2.2 GPT

GUID Partition Table (GPT) je moderní standard pro uspořádání tabulek diskových oddílů na fyzických úložištích. Následující popis tohoto formátu a jeho struktury v této sekci vychází z knihy [8]. Byl vyvinut jako součást standardu UEFI (*Unified Extensible Firmware Interface*), který nahrazuje zastaralý systém BIOS. Ve srovnání se schématem MBR přináší GPT řadu výhod a řeší některá jeho zásadní omezení. Mezi tyto výhody patří například podpora až 128 primárních oddílů na jednom fyzickém disku a použití 64bitové logické blokové adresace. Každý diskový oddíl v rámci GPT je jednoznačně identifikován pomocí svého globálně unikátního identifikátoru (GUID). Tato vlastnost eliminuje omezení MBR, kde identifikace byla omezena na číselné hodnoty v rozmezí 1 až 4. Rovněž je umožněna práce s úložnými kapacitami úložišť přesahujícími 2 TB, což výrazně překonává limity předchozího standardu MBR.

První oblastí této struktury je ochranný MBR. Tato oblast obsahuje tabulku diskových oddílů s jedním záznamem, jenž je označen kódem `0xEE`. Tato informace slouží k zajištění kompatibility se staršími systémy, které by disk se strukturou GPT nerozpoznaly a mohly by se pokusit o jeho nechtěné přeformátování. V sektoru s logickou adresou 1 začíná hlavička GPT. Ta uchovává metainformace o disku, jako je umístění, velikost tabulky oddílů nebo unikátní identifikátor disku. Kromě toho obsahuje hlavička také kontrolní součet samotné hlavičky a tabulky oddílů, aby bylo možné detekovat případné chyby. Třetí část obsahuje tabulku diskových oddílů, která uchovává záznamy o jednotlivých oddílech na disku. Každý takový záznam nese informace o počáteční a koncové logické adrese oddílu, jeho typu a výše zmíněném unikátním identifikátoru. Následují samotné datové oddíly GPT, které mohou sloužit k různým účelům. Jednou z významných vlastností tohoto formátu je redundance. Na konci fyzického disku se nachází záložní kopie hlavičky GPT a tabulky oddílů, což zvyšuje odolnost proti chybám. V případě detekce poškození primární kopie může systém použít záložní kopii k obnovení informací o diskových oddílech. Celou strukturu tabulky oddílů GPT znázorňuje obrázek 2.3.

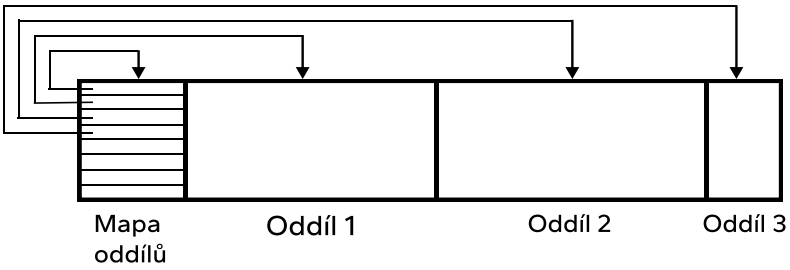


Obrázek 2.3: Schéma rozložení diskového úložiště s použitím GPT. Převzato z [29].

2.2.3 APM

Apple Partition Map (APM) je formát tabulky diskových oddílů, který byl historicky využíván na systémech společnosti Apple. Nacházel se zejména na starších počítačích Macintosh, které byly založeny na architektuře PowerPC. Po zavedení novějších zařízení využívajících procesory Intel došlo k postupnému nahrazení formátu tabulky diskových oddílů APM modernějším formátem GPT. Tento přechod byl především v důsledku změny architektury a výhod, které nový formát přinášel. Poslední verzí operačního systému, která umožňovala plnou instalaci na diskových jednotkách využívajících APM, byl Mac OS X 10.5 Leopard.

Od verze Mac OS X 10.6 Snow Leopard je oficiálně podporována instalace pouze na discích s formátem tabulky diskových oddílů GPT. Některé starší počítače, především ty s architekturou PowerPC, však stále používají diskovou tabulku APM [3]. Každá položka v rámci této mapy oddílů obsahuje metainformace o daném diskovém oddílu. To zahrnuje počáteční sektor oddílu, alokovanou velikost, typ souborového systému a název svazku. První záznam v diskové tabulce APM je obvykle záznam o samotné tabulce, včetně maximální povolené velikosti, kterou tato disková tabulka může dosáhnout. Apple využívá vyhrazené oddíly pro ukládání ovladačů hardwaru. Z tohoto důvodu primární systémový disk u počítačů Apple často obsahuje několik oddílů vyhrazených pro ovladače a další systémově specifický obsah, který není přímo spojen s uživatelskými souborovými systémy [8]. Strukturu APM znázorňuje obrázek 2.4. Jedná se o uspořádání s třemi oddíly vyhrazenými pro souborové systémy a jedním oddílem alokovaným pro diskovou tabulku APM.



Obrázek 2.4: Schéma rozložení diskového úložiště s použitím APM. Převzato z [8].

2.3 Souborový systém

S cílem usnadnit přístup a správu dat jsou informace na pevném disku ukládány ve formě souborů. Souborový systém představuje metody a datové struktury, které operační systém využívá k uspořádání souborů na disku. Z tohoto pohledu představuje souborový systém mechanismus pro organizaci dat. Bez něj by veškeré uložené informace byly pouze hromadou dat bez možnosti zjištění, kde daná informace začíná a kde končí. Hlavním účelem souborového systému je zaznamenávat informace o využitém a nevyužitém prostoru na úložišti, udržovat adresáře a soubory či zaznamenávat fyzická umístění souborů [4]. Tímto způsobem poskytuje strukturu pro ukládání, vyhledávání a organizaci dat na pevném disku.

Každý operační systém využívá svůj souborový systém, což znamená, že platformy jako Windows, macOS či Linux využívají odlišné přístupy k organizaci a správě dat. Ideální by byl jeden univerzální souborový systém, který by splňoval všechny potřebné požadavky napříč různými operačními systémy. Implementace takového jednotného formátu by teoreticky mohla zjednodušit nebo úplně eliminovat problémy jednotlivých souborových systémů. Zatím bohužel neexistuje žádný takový standard, který by vyhovoval potřebám ve všech operačních systémech. Různorodost existujících souborových systémů je dána jejich specifickými požadavky, které odrážejí historický vývoj a cílové použití. Každý z nich nabízí specifické výhody a nevýhody v oblasti úrovně zabezpečení dat, rychlosti přístupu k datům nebo podpory velikosti souborů. Právě těmto rozdílům mezi různými souborovými systémy se věnuje tato sekce.

2.3.1 FAT

FAT, neboli *File Allocation Table*, představuje jeden z nejstarších souborových systémů. Objevil se na počátcích osobních počítačů a stal se jedním z prvních standardních souborových systémů pro operační systém MS-DOS. Jeho název je odvozen od základní datové struktury – alokační tabulky (dále jen FAT), která tvoří jeho architekturu. Princip alokace prostoru v systému FAT je založen na přidělení prvního volného klastru [8]. To znamená, že nemá žádné složitější mechanismy pro optimalizaci umístění dat. Tato metoda alokace je primární příčinou fragmentace dat, kdy jsou části jednoho souboru uloženy v nesouvislých oblastech disku. Důsledkem fragmentace je snížení výkonu systému při čtení a zápisu dat. Jednou z hlavních vlastností FAT je jeho jednoduchost, která plyne z omezeného počtu typů datových struktur. Podle zdroje [5] byly postupem času vyvinuty 3 verze:

- **FAT12** – byl původně navržen pro disky s malou kapacitou a podporoval maximální velikost disku do 32 MB.
- **FAT16** – je rozšířená verze, která zvýšila limit maximální velikost disku až na 2 GB.
- **FAT32** – je nejnovější a nejrozšířenější varianta z rodiny FAT. Dokáže adresovat disky až do velikosti 2 TB. Nicméně, má omezení na maximální velikost jednotlivého souboru, která činí 4 GB.

Souborový systém je poměrně citlivý na poškození, zejména alokační tabulka. K jejímu narušení může dojít například při náhlém odpojení úložného média během probíhající operace zápisu. Proto si FAT uchovává kopii této tabulky, která se nachází hned za hlavní verzí. Absence mechanismu pro zaznamenávání a obnovu transakcí znamená, že FAT nemá vestavěné nástroje pro automatickou obnovu konzistence dat po takovýchto událostech. Poškození tabulky FAT může vést ke ztrátě informací o umístění souborů a tím nemožnosti k nim přistupovat. Smazání dat v systému FAT probíhá na logické úrovni. V alokační tabulce je první bajt názvu souboru nahrazen hexadecimální hodnotou `0xE5` [38], což slouží jako indikátor smazání. Klastry, které soubor zabíral, jsou následně označeny jako volné. Fyzicky však data na disku zůstávají, dokud nejsou přepsána.

Pro svou kompatibilitu napříč různými operačními systémy a nízké nároky na systémové prostředky z něj dělá ideální volbu pro externí úložiště. Proto je v současnosti FAT stále hojně využíván v různých přenosných a výměnných úložných zařízeních, jako jsou USB flash disky, paměťové karty (SD, microSD) a další externí média.

2.3.2 ExFAT

ExFAT (*extended FAT*) byl vyvinut společností Microsoft a následně uveden v roce 2006. Cílem bylo poskytnout efektivní řešení pro správu velkých souborů a rozsáhlých úložných zařízení. ExFAT představuje nástupce souborového systému FAT32 a řeší některé jeho omezení. Na rozdíl od tradičních hierarchických adresářových struktur implementovaných v jiných souborových systémech, exFAT využívá centralizovanou adresářovou tabulku, která obsahuje informace o souborech a adresářích. Pro alokaci datového prostoru využívá alokační bitovou mapu, která spravuje volné klastry efektivněji než předchozí tabulka FAT [18]. Podobně jako u FAT32, při smazání souboru v exFAT dochází k aktualizaci adresářové tabulky a alokační bitové mapy, které indikují, že je daný prostor volný. Samotná data na disku obvykle zůstávají do přepsání.

Návrh exFAT byl ovlivněn potřebou optimalizace pro flash paměťová média, jako jsou přenosné USB disky nebo paměťové karty. To zahrnuje minimalizaci nadbytečných a opakujících se zápisů, čímž se snižuje opotřebení paměťových buněk a prodlužuje jejich životnost. Původně byl exFAT vytvořen pro použití s operačním systémem Windows, avšak postupem času byla jeho podpora rozšířena i na další platformy, včetně systému macOS a vybraných distribucí operačního systému Linux [18]. Díky svým vlastnostem se často uplatňuje v zařízeních, jako jsou digitální fotoaparáty, videokamery, mobilní telefony a další přenosná elektronika. I když exFAT řeší omezení FAT32, stále nemá některé pokročilé funkce, jako je například žurnálování souborového systému. Absence této vlastnosti může znamenat nižší odolnost proti chybám a potenciální riziko ztráty dat v případě neočekávaných událostí, kterými jsou například výpadky napájení během operací zápisu.

2.3.3 NTFS

NTFS, zkratka pro *New Technology File System*, je souborový systém vyvinutý společností Microsoft. Stal se hlavním souborovým systémem pro operační systémy Windows od verze Windows 2000 a novější, proto je možné ho objevit na pracovních strojích i serverech. Nejčastěji se však vyskytuje u osobních počítačů. NTFS představuje vylepšený a rozšířený souborový systém oproti svému předchůdci FAT. Základní strukturou je tabulka *Master File Table* (dále jen MFT), která uchovává informace o všech souborech a adresářích. Každá položka je reprezentována samostatným záznamem v této tabulce. Ta obsahuje metadata jako název, velikost, přístupová práva, časové značky nebo umístění datových klastrů. Samotná tabulka je také uložena jako soubor a obsahuje záznam sama o sobě. První položka tabulky popisuje samotnou MFT, čímž je zajištěna její identifikace a správa. Jednou z hlavních vlastností je implementace žurnálování [24]. Tato technika spočívá v zaznamenávání informací o chystaných změnách souborového systému (přidání, odstranění nebo úpravy souborů a metadat) před jejich samotným provedením na disku. K tomu se využívá speciální soubor `$LogFile`. V případě neočekávaných událostí umožňuje žurnál obnovit konzistentní stav souborového systému a minimalizovat riziko poškození nebo ztráty dat.

Kromě vyšší spolehlivosti nabízí NTFS také kompresi souborů pro úsporu místa na disku, šifrování dat pro zvýšenou bezpečnost dat a seznamy řízení přístupu (ACL) [24]. Výhodou je ukládání metadat o souborech a složkách, což usnadňuje rychlé vyhledávání a organizaci dat. Obsahuje mechanismy pro kontrolu integrity dat, což umožňuje detekovat a opravovat chyby v souborech nebo na disku. Při smazání souboru je jeho záznam odstraněn z tabulky MFT. Samotná data ovšem zůstávají na disku, dokud nejsou přepsána jinými daty.

2.3.4 Ext2, Ext3, Ext4

Souborový systém Ext2, zavedený v roce 1993, byl jedním z prvních souborových systémů pro operační systém Linux. Mezi jeho významné charakteristiky patřila podpora blokových adres pro zvýšení účinnosti alokace paměťového prostoru. Pro každý soubor a adresář existuje inode, což je datová struktura obsahující jejich metadata. Souborový systém je organizován hierarchicky, přičemž adresáře obsahují mapování názvů souborů na jejich odpovídající čísla inode. Alokace diskového prostoru probíhá na úrovni bloků. Stav volných či obsazených bloků je sledován pomocí bitmapy bloků, zatímco stav obsazenosti inodů je sledován pomocí bitmapy inodů. Ext2 využívá přímý přístup k ukládání dat bez zavedení žurnálu, což se projevilo ve vyšší rychlosti zápisových operací [8]. Nicméně, absence žurnálu má za následek nižší úroveň odolnosti proti chybám. V případě, kdy dojde ke smazání souboru v Ext2, dojde k odstranění záznamu o souboru z adresářové struktury a k násled-

nému uvolnění odpovídajícího inodu [21]. Stejně jako u předchozích souborových systémů zůstávají data souboru na disku, dokud nejsou přepsána novými.

Jako nástupce Ext2 byl představen Ext3. Ten kladl důraz na zvýšenou spolehlivost a odolnost proti chybám. Byl integrován od verze linuxového jádra 2.4.15. Ext3 podporuje maximální velikost jednotlivého souboru od 16 GB do 2 TB a celkovou maximální velikost souborového systému od 2 TB do 32 TB [6]. Největším přínosem bylo zavedení žurnálování. Ten nabízí 3 režimy pro vyvážení výkonu a integrity dat [28]:

- **Plné žurnálování** – zapisuje jak metadata souborového systému, tak samotná data souborů. Před zápisem změny do souborového systému se kompletně zapíše do žurnálu. Jedná se o nejvyšší úroveň integrity dat. Nevýhodou je nižší výkon, protože se data zapisují dvakrát.
- **Řazené žurnálování** – je výchozí volbou a nejčastěji používaným režimem. Do žurnálu se zapisují pouze metadata. Data souborů, která souvisejí s danou metadatovou transakcí, jsou zapsána na disk předtím, než jsou zapsána samotná metadata do žurnálu. Tím je zajištěno, že po obnově z žurnálu bude souborový systém konzistentní a nedojde ke ztrátě dat, i když by mohla být verze souboru starší.
- **Zpětný zápis** – zapisuje do žurnálu také pouze metadata. Pořadí, v jakém jsou data souborů zapisována na disk, však není nijak garantováno. Data se mohou na disk zapsat až po zapsání souvisejících metadat do žurnálu. Tento režim poskytuje nejvyšší výkon, protože se minimalizuje zátěž spojená s žurnálováním dat.

Smazání souborů v Ext3 probíhá podobně jako v Ext2. Podle článku [21] dochází k odstranění záznamu z adresáře a následnému uvolnění inode a ukazatelů datových bloků. To je řešeno v tabulce inode inicializací na hodnoty 0. Samotné žurnálování primárně chrání integritu metadat souborového systému, ale přímo neusnadňuje obnovu smazaných dat.

Souborový systém Ext4 se zaměřil na další optimalizaci výkonu a zavedení moderních funkcionalit. Mezi jeho významné vlastnosti patří například rozšířená podpora pro manipulaci s většími soubory a rychlejší operace zápisu. Ext4 zavedl extenty pro lepší správu souvislých bloků a využívá alokaci s odloženým zápisem. Ta odkládá alokaci bloků do skutečného zápisu dat, což umožňuje efektivnější alokaci souvislých bloků a snižuje fragmentaci souborů [14]. Stejně jako u předchozích verzí Ext, i v Ext4 probíhá smazání souboru primárně na úrovni metadat, kdy jsou uvolněny odkazy na datové bloky souboru.

2.3.5 Btrfs

B-tree File System (Btrfs) je primárně navržený pro operační systémy založené na linuxovém jádře. Pro svou datovou strukturu využívá B-strom. Jedná se o stromovou strukturu, která se dokáže sama vyvažovat. Uchovává indexové klíče ve vnitřních uzlech stromu. Samotná data spojená s klíči jsou uložena v listech. Veškerá metadata souborového systému, včetně informací o souborech, adresářích a alokaci prostoru, jsou organizována v rámci této stromové struktury. Modifikace probíhají prostřednictvím metody copy-on-write [20]. Při jakékoliv operaci zápisu je nejprve vytvořena kopie modifikovaných dat a veškeré změny jsou prováděny převážně v této kopii. To představuje alternativní přístup k udržení souborového systému snadno obnovitelného po havárii, stejně jako u žurnálování.

Jednou z výrazných vlastností Btrfs jsou snímky. Btrfs podporuje softwarové RAID[2] úrovně, které umožňují vytváření a správu diskových polí s redundancí dat na softwarové úrovni. Pro optimalizaci využití úložného prostoru a snížení redundance dat implementuje Btrfs kompresi dat. Btrfs umožňuje přidávání a odebírání disků. Tím se usnadňuje dynamická správa diskové kapacity a rozšiřování úložiště za běhu systému bez nutnosti přerušení provozu. Umožňuje ukládání rozsáhlých metadat, což zahrnuje různé atributy souborů [32]. Během smazání dochází k odstranění záznamu z metadat a uvolnění vazby na dané datové bloky. Díky principu copy-on-write zůstávají původní data na disku, dokud nejsou přepsána nebo odstraněna během údržby systému. Pokud ovšem existuje aktivní snímek souborového systému, jsou soubory uchovány i po jejich smazání, protože snímek stále odkazuje na původní datové bloky a tím brání jejich uvolnění pro zápis nových dat. Btrfs se běžně využívá v linuxových distribucích zaměřených na správu serverů, datových úložišť a virtualizaci.

2.3.6 ZFS

Zettabyte File System (ZFS) je pokročilý souborový systém vyvinutý společností Sun Microsystems. Podle článku [32] integruje tento souborový systém správu souborového systému, diskových oddílů i svazků do jedné architektury. Jedním ze základních principů ZFS je technika copy-on-write. Integrita dat je v ZFS prioritou. Každý diskový blok je chráněn kontrolním součtem, který je při každém čtení ověřován. Pokud je detekována chyba a je dostupné redundantní úložiště, systém automaticky provede obnovu poškozených dat. Pro správu fyzických zařízení využívá ZFS koncept úložištních fondů tvořených virtuálními zařízeními. Tato zařízení mohou být jednotlivé disky, zrcadlené skupiny nebo konfigurace typu RAID-Z . Tento přístup umožňuje vysokou flexibilitu při správě úložných kapacit bez nutnosti oddílování disku nebo použití LVM.

ZFS se vyznačuje funkcemi, jako jsou transparentní komprese, deduplikace dat a dynamická alokace úložného prostoru. Další vlastností je možnost okamžitého vytvoření snímku na úrovni jednotlivých souborových systémů. Ten představuje konzistentní a neměnný obraz dat v konkrétním čase. ZFS také nabízí jejich klonování, čímž lze vytvořit zapisovatelnou kopii existujícího stavu souborového systému. Klony sdílejí data se svým zdrojovým snímkem až do okamžiku, kdy dojde ke změně dat.

Mezi nevýhody ZFS patří vyšší paměťové nároky a větší složitost konfigurace ve srovnání s jinými souborovými systémy. Kvůli svým možnostem správy je široce nasazován v prostředí serverů a síťových úložišť.

2.3.7 HFS a HFS+

Hierarchical File System (HFS) byl vyvinut společností Apple. K představení došlo v roce 1985. Jednalo se o souborový systém vyvinutý pro tehdejší operační systém Macintosh Operating System (Mac OS). Zavedl hierarchickou strukturu adresářů, která umožňovala organizovat soubory do stromové struktury. Ta byla implementována pomocí B-stromů. Každý soubor a adresář byl v rámci této hierarchie identifikován jednoznačným identifikátorem a umístěním. HFS měl několik omezení, mezi které patřila omezená délka názvů souborů, omezená velikost souborů a chybějící podpora pro žurnálování [34].

V reakci na výše uvedená omezení byl společně s operačním systémem Mac OS 8.1 představen *Hierarchical File System Plus* (HFS+). Ten se stal standardem v pozdějších verzích

[2] RAID (*Redundant Array of Independent Disks*) – technologie pro ukládání dat, která spojuje více pevných disků do jednoho celku a tím dochází k zajištění redundance dat.

systému Mac OS 9 a Mac OS X, kde nahradil původní HFS. Tento souborový systém, rovněž známý jako Mac OS Extended nebo HFS Extended, přinesl zlepšení v oblastech správy dat a překonal omezení svého předchůdce. Zavedl katalogový soubor, který využívá B-strom pro indexování názvů souborů a jejich přidružených metadat, včetně informací o umístění dat na disku. Přidáním souboru rozšířených atributů umožňuje uchovávat doplňující metadata, například informace o ikonách, barevných značkách a dalších atributech. HFS+ také obsahuje rozšířenou podporu pro kódování názvů souborů v Unicode. Zavedl také žurnálování a podporu pro 32bitovou alokační mapovací tabulku, která umožňuje práci s většími svazky a soubory [12]. Podle článku [7] dochází při smazání souboru na HFS+ k odstranění jeho záznamu z katalogového B-stromu a k označení využívaných bloků jako volných v alokační tabulce. Při aktualizaci B-stromu dojde k okamžitému přepsání metadat smazaného souboru.

2.3.8 APFS

Apple File System (APFS) je moderní souborový systém vyvinutý společností Apple pro své operační systémy na jejich vlastních zařízeních. Poprvé byl představen v roce 2016, kdy postupně nahradil předchozí souborový systém HFS+. Architektura APFS je založena na kontejnerech, které umožňují vytváření více logických svazků v rámci jednoho fyzického oddílu. Jednotlivé svazky se mohou chovat jako samostatné oddíly, ale sdílí jeden společný prostor. Svazky nemají v rámci kontejneru pevně přidělenou kapacitu, ale využívají princip dynamické alokace. Využívají pouze tolik úložného prostoru, kolik skutečně potřebují. Veškeré volné místo je sdíleno zbytkem celého kontejneru. Do tohoto souborového systému byla přidána podpora pro tvorbu snímků a možnost vytvářet klony souborů i složek. To umožňuje sdílet data mezi různými umístěními v rámci souborového systému bez nutnosti duplikace. APFS byl navržen s ohledem na moderní technologie úložišť, zejména SSD (*Solid State Drive*) a flash paměti. Z tohoto důvodu využívá efektivněji prostor na disku, minimalizuje počet opakovaných zápisů a zvyšuje celkový výkon na těchto typech médií. Pro zajištění integrity metadat využívá souborový systém kontrolní součty. Na rozdíl od svého předchůdce, APFS nepoužívá žurnálovací systém. Místo toho využívá techniku copy-on-write. Tento přístup zajišťuje, že veškeré operace zápisu jsou atomické, což Apple označuje termínem *Atomic Safe-Save*. Tím je zajištěno, že se operace dokončí nebo k ní vůbec nedojde [19]. Případná obnova dat je založena na tom, že při každé změně souborového systému jsou generovány kontrolní body, které zaznamenávají jeho předešlé stavy. Pomocí nich lze rekonstruovat předchozí verzi systému. Při smazání souboru dochází k aktualizaci metadat, kdy se odstraní záznam o souboru a uvolní se alokované bloky. K uvolnění dochází pouze u nesdílených bloků s klony či snímky.

Kapitola 3

Obnova multimediálních dat

Obnova dat je proces, při kterém jsou poškozená, ztracená či smazaná data navrácena do původní podoby. Bez ohledu na příčinu ztráty existují ve většině případů metody, jak tato data získat zpět. Výjimku představují především případy fyzického poškození úložiště, kdy nelze využít softwarové postupy. Ani smazání souboru není vždy nevratné. Operační systémy často přesouvají odstraněné soubory do dočasného úložiště, jako je koš. Z něho je možné data obnovit před jejich trvalým odstraněním. I po vysypání koše však data obvykle zůstávají na úložišti a nejsou okamžitě přepsána. V takových případech lze využít specializované nástroje, které umožňují analyzovat úložiště a obnovit soubory, které nejsou viditelné běžnými prostředky. Proto tato kapitola pojednává o multimediálních datech, jejich možném poškození a následných technikách používaných k obnovení.

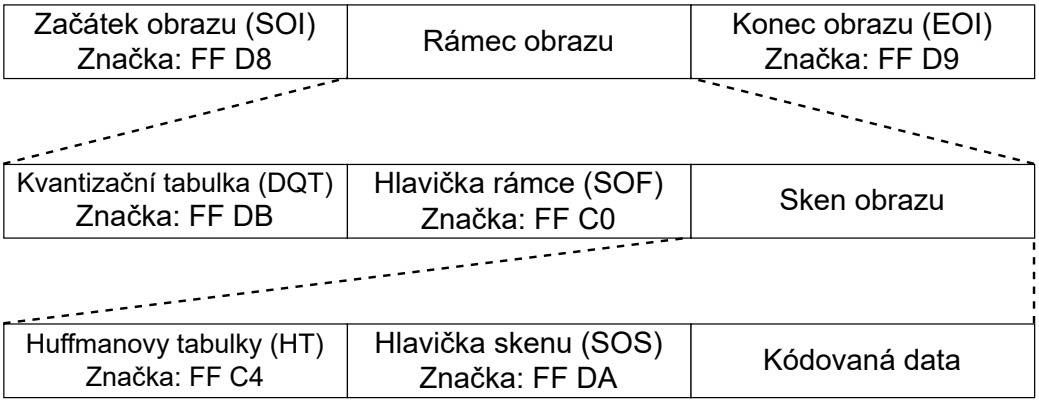
3.1 Multimediální data

Multimediální data představují kategorii digitálních informací, která obsahuje širokou škálu formátů pro reprezentaci textu, obrázků, zvuku a videa [22]. Ačkoli jsou tato data na nejnižší úrovni uložena identicky, každý z typů se vyznačuje svou specifickou vnitřní strukturou a způsobem reprezentace. Ta ovlivňuje jejich identifikaci, zobrazení a možnosti rekonstrukce v případě poškození. Následující podsekce se proto věnují popisu jednotlivých kategorií multimediálních dat. Z těchto základních principů následně čerpá i sekce 6.3, která se věnuje metodám porovnávání obsahu dvou souborů s těmito datovými typy.

3.1.1 Obrazová data a videozáznamy

Obrazová data tvoří základní a velmi rozšířenou složku multimediálního obsahu. Digitální obraz je složen z jednotlivých bodů, které se nazývají pixely. Tyto pixely jsou organizovány do pravidelné mřížky a každý z nich nese informaci o své barvě a jasu. Obraz je primárně definován dvěma parametry. Prvním je rozlišení, které udává počet pixelů na šířku a výšku. Druhým je barevná hloubka, která určuje počet bitů použitých k reprezentaci barvy každého pixelu. Pro barevné obrazy se nejčastěji využívá barevná hloubka s hodnotou 24 bitů. Těchto 24 bitů zahrnuje 8 bitů pro každou ze základních složek barevného modelu RGB, který je tvořen z červené, zelené a modré barvy. U těchto typů souborů se běžně používají jak bezeztrátové, tak ztrátové kompresní techniky. Ztrátová komprese umožňuje výrazně zmenšit velikost souboru, ale nevratně ztratit části obrazových informací. To může vést ke zhoršení kvality obrazu, zvláště při vysokém stupni komprese [22]. Příkladem ztrátové komprese je formát `JPEG`. Podle článku [33] je tvořen posloupností značek a segmentů.

Značky reprezentují dvoubajtové kódy, které označují začátek specifické části souboru. Mezi ně patří například začátek nebo konec obrazových dat či definice kompresních tabulek. Po každé značce následuje segment, který obsahuje konkrétní data nebo metadata, jak je ukázáno na obrázku 3.1. Pokud jsou značky zachovány, obnova může být úspěšná, i když část obrazových dat chybí. Naopak, pokud chybí segment s komprimovaným obsahem nebo dojde k jeho přepsání jinými daty, obnova již často není možná. Proto se po pokusu o jeho zobrazení objeví viditelné defekty v podobě šedých, černých nebo barevně zkreslených bloků.



Obrázek 3.1: Schéma struktury formátu JPEG. Převzato z [15].

U video souborů dochází ke kombinaci vizuální a zvukové složky. Na rozdíl od statických obrazových formátů je video tvořeno sekvencí jednotlivých snímků, které jsou zobrazovány dostatečně rychle za sebou. Právě tato návaznost snímků vytváří pro lidské oko pocit plynulého pohybu. Plynulost videa je určena snímkovou frekvencí, tedy počtem snímků zobrazených za sekundu. Může nabývat různých hodnot, ale běžně se využívá 25, 30 nebo 60 snímků za sekundu. Vzhledem k obrovskému objemu dat, který video soubory obsahují, je nezbytná jejich komprese. Tento proces je řízen pomocí kodeků, které implementují kompresní algoritmy [22]. Velké množství dat ve video souborech zároveň zvyšuje pravděpodobnost jejich fragmentace na disku. I když metadata obsažená v záhlaví souboru zůstávají neporušená, nemusí být jejich přítomnost dostatečná pro úspěšnou obnovu, pokud došlo k přepsání některého z fragmentů. Jak zmiňuje článek [26], překážkou pro obnovu jsou přepsané segmenty video dat, které jsou typicky považovány za neobnovitelné, jelikož původní informace byly nahrazeny jinými daty. Každopádně článek představuje techniku zaměřenou na obnovu dat na úrovni video snímků, která dokáže identifikovat, shromažďovat a spojovat izolované snímky pomocí specifikace video kodeku. Při jejich extrakci a následném dekódování je umožněna jejich rekonstrukce i z částečně přepsaných souborů. Samotný kodek totiž vkládá identifikátory do každého snímku videa, které jsou využity při extrakci a následném ověřování snímků pomocí dekodéru. Propojení obnovených snímků je následně možné s využitím metadat souboru.

3.1.2 Zvuková data

Zvuk je fyzikální jev, který vzniká jako mechanické vlnění šířící se prostředím, nejčastěji vzduchem. Toto vlnění představuje spojitý analogový signál, který je pro zpracování v digitálních systémech nutné převést do digitální podoby. Tento proces se provádí ve dvou hlavních krocích – vzorkování a kvantizace. Vzorkování převádí časově spojitý analogový signál na diskrétní posloupnost hodnot. Amplituda zvukové vlny je zaznamenávána v pravidelných

časových intervalech. Frekvence, s jakou jsou tyto hodnoty zaznamenány, se označuje jako vzorkovací frekvence. Jedná se o parametr, který určuje počet vzorků za sekundu. V následující fázi je každému získanému vzorku přiřazena číselná hodnota z předem definované množiny úrovní. Rozsah hodnot je omezen bitovou hloubkou. Ta určuje počet bitů, které jsou použity pro reprezentaci každého vzorku. Při vyšší bitové hloubce je umožněno jemnější rozlišení úrovní hlasitosti [22]. Výsledkem celého procesu je digitální zvukový signál, který je tvořen jako sekvence číselných hodnot uspořádaných v čase. Každý vzorek představuje hodnotu amplitudy v daném časovém okamžiku s přesností určenou bitovou hloubkou. Tyto parametry jsou uloženy v metadatech souboru a při jejich poškození dochází k problémům s otevřením či přehráním. Při poškození některých vzorků zvuku jsou v tomto důsledku slyšitelné defekty jako praskání, zkreslení nebo dokonce výpadky.

3.1.3 Komprimované archivy

Komprese dat je proces, jehož cílem je redukce objemu dat při zachování jejich informačního obsahu. U archivů se komprese používá především pro účely přenosu více souborů v jedné komprimované struktuře. Příkladem jsou formáty `zip` nebo `rar`. Jelikož jsou komprimovaná data transformovanou a strukturálně upravenou podobou původních dat, je jejich obnova složitější než u nekomprimovaných dat. Poškození komprimovaného souboru, zejména v oblasti hlavičky obsahující metadata, může znemožnit dekompresi celého archivu nebo jeho částí. Nachází se zde informace o počtu souborů, použitém algoritmu, délkách bloků nebo pozicích jednotlivých segmentů. V případě, že došlo k přepsání segmentu dat, je tato část považována za neobnovitelnou. U některých kompresních algoritmů, jako například DEFLATE používaný u formátu `zip`, jsou data rozdělena do bloků s různými stupni závislosti. Pokud algoritmus použil nastavení zajišťující nezávislost bloků, je možné obnovit dané části i bez přístupu k úvodním metadatům [30]. Pokud jsou bloky vzájemně závislé, poškození první části znemožňuje obnovu dalších segmentů, i kdyby byly nepoškozené.

3.2 Obnova dat s využitím metadata

Souborové systémy, jako jsou FAT32, NTFS nebo ext4, využívají metadata k udržování informací o souborech, včetně jejich umístění na úložišti, časů modifikace, vlastnictví či přístupových oprávnění. Když je soubor smazán, souborové systémy samotný obsah souboru okamžitě neodstraní. Dochází pouze k označení daných bloků jako dostupné pro opětovné použití při dalších alokacích jinými soubory [23].

Obnova probíhá prostřednictvím nalezení záznamů, které obsahují informace o původním umístění dat smazaných souborů. Například u souborového systému NTFS, jak již bylo zmíněno v sekci 2.3.3, jsou metadata uložena v tabulce MFT, která obsahuje záznamy o každém souboru i adresáři. Samotná obnova se svým principem podobá čtení souboru ze známého umístění. Problém nastává ve chvíli, kdy jsou původní datové bloky již alokovány novým souborům a jejich obsah je přepsán. V takovém případě je obtížné určit, zda daný blok skutečně náleží původnímu smazanému souboru, nebo již obsahuje jiná data. I když jsou části původních metadat dostupné, nemusí být zaručena správnost a úplnost obnoveného obsahu [8]. Tato metoda je proto účinná pouze tehdy, pokud metadata zůstala nedotčena a datové bloky nebyly přepsány novými daty.

3.3 Technika vyřezávání dat

Tato sekce čerpá z článku [31] a popisuje *data carving*, neboli techniku vyřezávání dat. Ta se využívá především v oblasti digitální forenzní analýzy. Jedná se o nízkoúrovňové skenování médií, při němž jsou vyhledávány specifické sekvence bajtů a identifikovány jejich vzory, tzv. signatury. Vzory odpovídají známým typům souborů, které se nacházejí v jejich hlavičkách a v některých případech i v zápatích. Tabulka 3.1 uvádí příklady charakteristických vzorů pro vybrané typy souborů. V některých případech mohou různé formáty souborů sdílet stejné identifikační vzory.

Typ souboru	**Hex. vzor v záhlaví**	**Hex. vzor v zápatí**
jpeg	FFD8	FFD9
png	89504E470D0A1A0A	49454E44
gif	47494638	003B
pdf	25504446	2525454F46
html	3C48544D4C3E	3C2F68746D6C3E
avi	52494646	Není pevně definovaný
mp4	66747970	Závisí na struktuře

Tabulka 3.1: Hexadecimální vzory souborů obsaženy v záhlaví a zápatí souborů [31].

Vyřezávání dat se uplatňuje v případech, kdy jsou standardní metadata souborového systému poškozena nebo chybí, což omezuje obnovu pomocí jiných metod. Pomocí analýzy nealokovaného prostoru či prostoru označeného jako nepoužitelný operačním systémem lze rozpoznat soubory bez ohledu na jejich původní název nebo umístění. Počáteční bajty souborů se často nacházejí na hranicích klastrů nebo sektorů, což umožňuje optimalizovat vyhledávání analýzou pouze jejich počátků. Tato technika se potýká s problémy spojenými s fragmentací souborů. Dochází k porušení předpokladu souvislého uložení dat a tím se znemožňuje jednoduché získání celého souboru pouze na základě detekce hlavičky a zápatí. V těchto případech se obvykle obnoví pouze první fragment. U vložených nebo fragmentovaných souborů je proto nutné provádět sekvenční analýzu bajt po bajtu v rámci celého prostoru. Tuto techniku lze uplatnit i při obnově dat z dokumentů, do kterých byly vloženy další soubory. Příkladem je vložený obrázek v souboru `pdf`. V souborovém systému nemusí být identifikovány jako samostatné položky, ale pomocí vyřezávání by je bylo možné z obsahu extrahovat. Jedním ze způsobů ukončení vyhledávání je využití vzorů v zápatí, které indikují konec souboru.

Tato metoda je však náchylná ke vzniku tzv. falešně pozitivních nálezů. K těm dochází tehdy, když určitá sekvence bajtů odpovídá známému vzoru souboru a je identifikována v datech. Ve skutečnosti nemusí tvořit samostatný soubor ani jeho relevantní část. Může se jednat o náhodné shody vzorů v rámci jiných datových struktur nebo o fragmenty, které nepředstavují kompletní či funkční soubor. Falešná pozitiva tak zbytečně zvyšují počet potenciálně nalezených souborů.

Kapitola 4

Nástroje pro obnovu dat

V digitálním prostředí je běžné odstraňovat soubory z různých datových úložišť. Co když nastane případ, kdy je potřeba daná data obnovit zpět? V takových případech představují specializované nástroje pro obnovu dat možné řešení pro znovuzískání ztracených informací v jejich původním stavu. Efektivita těchto nástrojů se však značně liší. Některé nástroje vykazují omezenou kompatibilitu s různými typy paměťových médií, zatímco jiné jsou zaměřeny na úzkou množinu formátů souborů. Z tohoto důvodu je dobré zvolit nástroje, které podporují daný typ zařízení, souborový systém a formáty souborů, které je potřeba obnovit.

Využívání těchto nástrojů je především důležité v oblasti kriminalistiky a forenzního vyšetřování. Pachatelé se často pokoušejí maskovat svou činnost odstraňováním digitálních stop, což vyžaduje od vyšetřovatelů schopnost obnovovat data z různých digitálních zařízení. Z těchto důvodů potřebují nástroje, které jsou schopny obnovovat data napříč širokým spektrem. Důležité je nejen identifikovat smazaná nebo upravená data, ale také je obnovit do původní podoby. Následně totiž mohou být použity jako důkazní materiály.

Tato kapitola se zaměřuje na přehled několika vybraných nástrojů pro obnovu dat, které jsou běžně využívány v praxi. Součástí jsou metriky pro jejich porovnání, které definuje sekce 4.1. V rámci sekce 4.2 jsou představeny volně dostupné programy a sekce 4.3 popisuje komerční programy. Obě sekce obsahují porovnání jejich technických parametrů a dostupných funkcionalit pro obnovu.

4.1 Metriky pro porovnání nástrojů

Stanovené metriky pro porovnání nástrojů na obnovu dat vycházejí z článků [25, 1], které se zabývají porovnáním těchto programů. S ohledem na měření v kapitole 7 jsou vybraná kritéria následně využita. Použití standardizovaných metrik zároveň umožní porovnání dosažených výsledků s výstupy jiných porovnání, které se věnují obdobné problematice.

Úspěšnost obnovy

Tato metrika hodnotí, s jakou přesností dokáže nástroj obnovit ztracená data ve srovnání s jejich původní podobou. Vyjádřená procentuální shoda poskytuje představu o míře zachování autentičnosti obnovených dat. Čím vyšší je procento shody, tím věrněji nástroj zrekonstruoval původní strukturu a obsah souborů během procesu obnovy. Od kvalitního nástroje pro obnovu dat se očekává vysoká úspěšnost při obnově různých typů souborů a paměťových médií.

Kompatibilita

Důležitým aspektem při hodnocení nástrojů pro obnovu dat je jejich kompatibilita s různými souborovými a operačními systémy. Nicméně, důraz na kompatibilitu se neomezuje pouze na počítačová data. Nástroj by měl být schopen obnovovat data i z odlišných typů paměťových zařízení. Nezbytná je rovněž široká podpora různých formátů souborů, včetně multimediálních dat.

Rychlost obnovy

Rychlost obnovy představuje jednu z hlavních metrik hodnocení, která zohledňuje časovou náročnost procesu obnovy. Měření průměrné doby potřebné k obnovení dat poskytuje indikátor celkové efektivity nástroje z hlediska rychlosti. Rychlý nástroj by měl minimalizovat čas potřebný k obnovení dat, což je zejména důležité při zpracování rozsáhlých objemů dat. Některé nástroje mohou být efektivnější při obnově souborů menší velikosti, zatímco jiné mohou dosahovat lepších výsledků v rychlosti obnovy rozsáhlých datových souborů.

Uživatelská přívětivost

Uživatelské rozhraní tvoří nedílnou součást nástrojů. Nástroj by měl nabízet intuitivní a snadno pochopitelné uživatelské rozhraní i pro uživatele bez hlubších technických znalostí. V opačném případě, pokud ovládání aplikace vykazuje nedostatky nebo je narušena její přehlednost, může uživatele od dalšího využívání odradit. Výjimku mohou tvořit konzolové aplikace, kde je absence grafického uživatelského prostředí.

4.2 Volně dostupné nástroje

Volně dostupný nástroj, často označovaný termínem *freeware*, představuje typ softwaru, který je uživatelům poskytován bezplatně. Uživatelé nejsou povinni hradit žádné poplatky za jejich stažení, instalaci či užívání [2]. Tato kategorie zahrnuje různé typy programových řešení. Je ovšem důležité podotknout, že se vždy nemusí jednat pouze o programy s otevřeným zdrojovým kódem (angl. *open-source*), u něhož je možná uživatelská modifikace v souladu s licencí. Freeware může rovněž představovat omezenou verzi komerčního produktu, která bezplatně nabízí pouze vybrané funkcionality.

Mezi hlavní výhody patří bezplatné použití a snadná dostupnost. Nicméně, volně dostupné nástroje mají i určité limitace. Omezená funkčnost, nižší výkonnost či efektivita v porovnání s komerčními alternativami mohou ve specifických scénářích představovat značná omezení. Dle zdroje [2] je u programů s otevřeným zdrojovým kódem technická podpora závislá na aktivitě komunitních přispěvatelů, kteří se podílejí na odstraňování chyb a implementaci vylepšení. V tomto ohledu se nemusí jednat o specializované vývojářské týmy s primárním zaměřením na danou problematiku. Tyto aspekty se promítají také do kvality samotné dokumentace, která nemusí dosahovat požadované úrovně. S tímto úzce souvisí i možná bezpečnostní rizika.

V následujících podsekcích jsou představeny a popsány specifikace tří konkrétních nástrojů zaměřených na problematiku obnovy dat, které spadají do kategorie volně dostupných nástrojů. Jedná se o programy Recuva, PhotoRec a Scalpel. Informace pro popis programu Recuva jsou čerpány z [37], pro PhotoRec z [10] a pro Scalpel z [17].

4.2.1 Recuva

Recuva, produkt společnosti Piriform, představuje široce využívaný nástroj pro obnovu dat ceněný zejména pro svou intuitivní ovladatelnost. Jedná se o nástroj, který je určen pro operační systém Windows. Aktuálně nejnovější dostupná verze je Recuva `v1.53.2096`. Kromě bezplatné verze je k dispozici také profesionální verze za cenu $18 ročně, která rozšiřuje funkčnost o podporu virtuálních pevných disků, automatické aktualizace a prioritní zákaznickou podporu. Tento uživatelsky přívětivý program je schopen obnovit smazaná data z různých paměťových médií, včetně pevných disků, paměťových karet a USB zařízení. Mezi podporované souborové systémy patří pouze FAT, exFAT, NTFS, Ext2, Ext3, Ext4 či EFS. Recuva nabízí možnost hloubkové analýzy pro důkladnější obnovu datových struktur. Mezi typy souborů, které dokáže obnovit, lze zařadit obrazové formáty, multimediální soubory, dokumenty, e-mailové zprávy z poštovních klientů a komprimované archivy.

4.2.2 PhotoRec

PhotoRec je open-source projekt vyvinutý společností CGSecurity. Představuje nástroj schopný obnovit soubory z většiny souborových systémů, které byly popsány v sekci 2.3, s výjimkou btrfs, ZFS a APFS. Stabilní verze `7.1` tohoto softwaru je dostupná ke stažení pro širokou škálu operačních systémů. Je zahrnuta podpora pro Windows Vista a novější, Mac OS X od verze `10.6` a Linux s kernelem `2.6.18` a vyšším. PhotoRec disponuje schopností rozpoznat a obnovit rozmanitou škálu formátů souborů. Primárně se jedná o multimediální obsah, archivy a kancelářské dokumenty. Výčet všech podporovaných formátů zahrnuje **více než 480 různých přípon** souborů[1]. Dokáže identifikovat typy souborů na základě specifických podpisů. Pro účely obnovy využívá techniku vyřezávání dat a porovnání datových bloků. V situacích, kdy je souborový systém poškozen, je PhotoRec schopen autonomně dopočítat velikost datového bloku na základě analýzy prvních deseti souborů uložených na daném médiu.

4.2.3 Scalpel

Scalpel je také open-source nástroj, který je určený pro obnovu dat. Jedná se o nástupce programu Foremost a je distribuován pod licencí *GNU General Public License*. Ke stažení jsou dostupné dvě verze: Scalpel `1.60`, který je nejrozšířenějším veřejným vydáním, a novější vydání Scalpel `2.02`. Obě vydání jsou kompatibilní s operačními systémy Windows, Linux i macOS. Tento nástroj si zakládá na technice vyřezávání dat. Z tohoto důvodu dokáže pracovat téměř s kterýmkoli souborovým systémem. Základním prvkem funkčnosti nástroje je konfigurační soubor, v němž jsou definovány rozpoznávací vzory označující začátek a konec jednotlivých typů souborů. Na jejich základě Scalpel dokáže identifikovat a extrahovat datové bloky, které odpovídají známým formátům, a to i v případech, kdy souborový systém není dostupný nebo je poškozen. Mezi hlavní výhody nástroje patří modularita a možnost přizpůsobení konfigurace konkrétním požadavkům analýzy. Naopak mezi nevýhody lze zařadit nutnost přesného nastavení vzorů pro jednotlivé formáty souborů a omezenou schopnost rekonstruovat fragmentované soubory, jelikož metoda vyřezávání předpokládá, že data jsou uložena v souvislém bloku mezi počáteční a koncovou sekvencí.

[1] Výčet všech podporovaných formátů souborů, které program PhotoRec dokáže zpracovat je dostupný z: https://www.cgsecurity.org/wiki/File_Formats_Recovered_By_PhotoRec.

4.2.4 Shrnutí

V rámci sekce 4.2 byly představeny vybrané volně dostupné nástroje. Jejich vlastnosti jsou shrnuty v tabulce 4.1, která poskytuje přehled informací o podporovaných souborových systémech a operačních systémech daných jednotlivými programy. Všechny z nich podporují několik různých typů souborů. Kvůli velkému množství nejsou zahrnuty do tabulky.

Shrnutí volně dostupných nástrojů.			
	Recuva	**PhotoRec**	**Scalpel**
Podpora OS	Win	Win, Mac, Linux	Win, Mac, Linux
Podporované souborové systémy			
NTFS	✓	✓	✓
FAT12	✓	✓	✗
FAT16	✓	✓	✓
FAT32	✓	✓	✓
ExFAT	✓	✓	✓
Ext2	✓	✓	✓
Ext3	✓	✓	✓
Ext4	✓	✓	✓
Btrfs	✗	✗	✓
HFS	✗	✗	✓
HFS+	✗	✓	✓
APFS	✗	✗	✗

✓ Souborový systém je nástrojem podporován.
✗ Souborový systém není nástrojem podporován.

Tabulka 4.1: Souhrn podporovaných souborových systémů pro volně dostupné nástroje.

4.3 Komerční nástroje

Tento typ programu bývá typicky charakterizován uzavřeným zdrojovým kódem, což uživateli zamezuje nahlédnout na interní implementační detaily. Jakékoli modifikace jsou přitom striktně regulovány licenčními podmínkami [2]. Pro využívání komerčního softwaru je od uživatele požadováno zakoupení licence, ať už formou jednorázové platby, nebo prostřednictvím pravidelného měsíčního či ročního předplatného. Ta bývá často vázána na konkrétní počet koncových zařízení nebo uživatelských účtů. Od těchto produktů se standardně očekává vyšší úroveň kvality a stability, přičemž by mohly nabízet pokročilejší funkcionality v porovnání s volně dostupnými alternativami.

Mezi zásadní výhody lze zařadit profesionální zákaznickou podporu poskytovanou vývojářskými týmy. Pravidelné aktualizace a bezpečnostní záplaty zajišťují aktuálnost programu. Jednou z hlavních nevýhod tohoto typu softwaru mohou být vyšší pořizovací náklady licence. Další nevýhodou jsou často omezené možnosti uživatelského přizpůsobení. V některých případech mohou komerční aplikace obsahovat dodatečné funkcionality či dokonce celé aplikace, které jsou pro uživatele irelevantní. Tato sekce se zaměřuje na představení komerčních nástrojů R-Studio, Disk Drill a EaseUS Data Recovery Wizard. Uvedené informace v podsekcích čerpají z oficiálních webových stránek programů, pro R-Studio z [36], pro EaseUS Data Recovery Wizard z [13] a pro Disk Drill z [11].

4.3.1 R-Studio

R-Studio, vyvinutý společností R-Tool Technology Inc., patří mezi známější nástroje pro profesionální obnovu dat. Tento forenzní nástroj je nabízen s trvalou licencí od částky $80 a umožňuje multiplatformní využití. Je však nutné zmínit, že licence je specifická pro daný operační systém. Uživatelé jej mohou využít na různých operačních systémech, konkrétně na systémech Windows (od verze 2000 a novějších), Windows Server (od verze 2003 a novějších), macOS (od verze 10.14) a různých distribucích Linux. U nich je podmínka umožnit instalaci balíčků ve formátech `.rpm` nebo `.deb`.

Program podporuje širokou škálu souborových systémů, včetně těch, které byly popsány v sekci 2.3 s výjimkou Btrfs. Dále zahrnuje podporu pro UFS1/UFS2, XFS a ReFS/ReFS2. Významnou funkcí je obnova dat z poškozených polí RAID na standardních úrovních 0, 1, 4, 5 a 6 s automatickou detekcí parametrů pole. K obnově se využívá hledání souborů podle jejich specifických datových vzorů, ale dokáže využít i informace ze souborového systému. Kromě toho nástroj nabízí další pokročilé funkce, jako je obnova dat ze síťových úložišť, vytváření bitových kopií pevných disků, hexadecimální editor pro modifikaci atributů souborů a zobrazení informací S.M.A.R.T.[2].

4.3.2 EaseUS Data Recovery Wizard

EaseUS Data Recovery Wizard se řadí mezi komplexní forenzní nástroje. Je určen pro dva operační systémy. Verze `17.0` pro platformu Windows je kompatibilní se systémy Windows 7 a novějším, nebo na Windows Server 2003 a novějším. Podpora souborových systémů zahrnuje FAT, ext2/ext3, HFS+, ReFS, exFAT a NTFS. Ve verzi `14.2.0` pro macOS je podpora souborových systémů podobná. Přidává podporu pro HFS a APFS, ale neumožňuje práci s rodinou Ext a FAT12. Obě z verzí mají rozsáhlou podporu různých typů souborů zahrnující multimediální data, dokumenty a mnoho dalších. Cena trvalé licence se pohybuje kolem $150. K dispozici je rovněž bezplatná verze s omezením na maximální velikost obnovených dat do 2 GB. Tato bezplatná varianta však nepodporuje obnovu poškozených video souborů, fotografií a dokumentů. Mezi hlavní funkce nástroje patří obnova smazaných i poškozených souborů a diskových oddílů. Uživatelské rozhraní aplikace je navrženo s ohledem na jednoduchost a intuitivnost ovládání. Autor softwaru neposkytuje bližší informace o metodách použitých v rámci nabízených funkcí pro obnovu dat.

4.3.3 Disk Drill

Disk Drill je komplexní nástroj pro obnovu dat, určený pro platformy Windows a macOS. Aktuálně nejnovější verze `5.3.826` je dostupná pouze pro Windows 10 a 11. Pro kompatibilitu se staršími verzemi systému Windows je nutné využít předchozí verze programu. Umožňuje práci se souborovými systémy FAT, exFAT, NTFS, HFS i HFS+, APFS, ReFS a všemi verzemi Ext. Nástroj je schopen obnovit **více než 370 různých formátů** souborů. Pro obnovu primárně využívá funkci rychlého skenování pro nalezení nedávno smazaných souborů. Ta pracuje s informacemi z metadat, které jsou dostupné na souborovém systému. Podobně jako výše uvedené programy i Disk Drill analyzuje sektory paměťového média a na základě detekovaných datových vzorů identifikuje formát souboru. Pro tento účel využívá metodu hloubkové analýzy. Kromě primární funkce obnovy smazaných dat nabízí Disk Drill i doplňkové funkce jako prevence datových ztrát, kontrolu integrity diskových

[2] S.M.A.R.T. (*Self-Monitoring Analysis and Reporting Technology*) – monitorovací systém v pevných discích, jehož úkolem je detekovat a hlásit spolehlivost disku s cílem předpovědět blížící se selhání.

jednotek nebo zálohování v podobě bitové kopie disku. V programu je také integrováno monitorování S.M.A.R.T., podobně jako u předchozího programu R-Studio. Doživotní licenci k plné verzi je možné zakoupit za cenu od $80.

4.3.4 Shrnutí

V rámci sekce 4.3 byly představeny vybrané komerční nástroje. Jejich vlastnosti jsou shrnuty v tabulce 4.2, která poskytuje přehled informací o podporovaných souborových systémech na daných operačních systémech jednotlivých programů. Mimo to jsou zde uvedeny také licenční poplatky. Jelikož všechny z nástrojů podporují několik různých typů souborů, nejsou z důvodu velkého množství zahrnuty do tabulky se shrnutím.

Shrnutí komerčních nástrojů					
	R-Studio	**EaseUS Data Recovery Wizard**		**Disk Drill**	
Podpora OS	Win, Mac, Linux	Win	Mac	Win	Mac
Cena	od $80	od $70		od $80	
Podporované souborové systémy					
NTFS	✓	✓	✓	✓	✓
FAT12	✓	✓	✗	✗	✓
FAT16	✓	✓	✓	✓	✓
FAT32	✓	✓	✓	✓	✓
ExFAT	✓	✓	✓	✓	✓
Ext2	✓	✓	✗	✓	✗
Ext3	✓	✓	✗	✓	✓
Ext4	✓	✗	✗	✓	✓
Btrfs	✗	✗	✗	✗	✗
HFS	✓	✗	✓	✓	✓
HFS+	✓	✓	✓	✓	✓
APFS	✓	✗	✓	✗	✓

✓ Souborový systém je nástrojem podporován.
✗ Souborový systém není nástrojem podporován.

Tabulka 4.2: Souhrn podporovaných souborových systémů pro komerční nástroje, které budou využity při měření.

Kapitola 5

Návrh automatizované datové sady

Datové sady představují ideální a důležitý zdroj pro vývoj a výzkum nových technologií, neboť umožňují zkoumání a testování nových algoritmů, metod či nástrojů. Hlavním cílem této kapitoly je představení návrhu automatizovaných skriptů, které zajistí reprodukovatelné generování komplexní datové sady. Tato datová sada umožní simulovat různé scénáře ztráty dat pro testování účinnosti technik obnovy, přičemž lze postupně vytvářet varianty pro různé souborové systémy. Sekce 5.1 se zaměřuje na analýzu stávajících řešení a vyhodnocení jejich vlastností. Sekce 5.2 specifikuje vlastnosti nově navržené datové sady a sekce 5.3 navazuje s návrhem pro automatizovaný skript zajišťující její generaci. Poslední sekce 5.4 se věnuje návrhu pro doplňující skripty, které řeší připojování virtuálních obrazů disků a automatizované vyhodnocení úspěšnosti výsledků obnovených dat.

5.1 Analýza existujících datových sad

Proces výběru datové sady nepředstavuje náhodný výběr souborů. Je potřebné zvolit data, která co nejvěrněji odrážejí různé situace vyskytující se v digitálních systémech. Hlavním cílem je tedy simulovat praktické scénáře, které mohou nastat v prostředí digitální forenzní analýzy. Z tohoto důvodu jsem se v průzkumu existujících datových sad zaměřil na testovací sady publikované v rámci projektu Testování nástrojů počítačové kriminalistiky (CFTT), který spravuje Národní institut pro standardy a technologie (NIST). Tento projekt se zaměřuje na vývoj spolehlivých testovacích postupů pro hodnocení forenzních nástrojů. Snahou je poskytnout odborníkům z praxe, výzkumníkům i dalším uživatelům podložené informace o přesnosti a spolehlivosti těchto nástrojů. Proto se jedná o významný zdroj referenčních dat v oblasti digitální forenzní analýzy.

V rámci provedeného průzkumu se mi podařilo nalézt několik menších datových sad zaměřených především na případy fragmentace dat. Tyto scénáře byly typicky reprezentovány obrazy disků bez souborového systému nebo obsahujícími souborové systémy FAT32 a NTFS. Jejich zaměření bylo cíleno na testování techniky vyřezávání dat. Z rozsáhlejších sad jsem vybral tři zástupce pro podrobnější analýzu, z nichž se každý zaměřuje na odlišné řešení daných scénářů. Tato sekce se věnuje detailnímu popisu každé z vybraných datových sad, které slouží k testování aplikací na obnovu dat. Všechny analyzované sady jsou veřejně dostupné prostřednictvím portálu Počítačových forenzních referenčních datových sad (CFReDS[1]). Cílem analýzy je identifikovat přínosy a omezení již existujících řešení.

[1]Ostatní datové sady lze nalézt na portálu CFReDS, který je dostupný z: https://cfreds.nist.gov.

5.1.1 Datová sada s multimediálním obsahem

Tato datová sada[2] se vyznačuje širokým pokrytím různých typů souborů a testovacích scénářů. Celkově zahrnuje 6 různých stupňů fragmentace, které začínají od souborů uložených v souvislém bloku až po komplexnější formy fragmentace.

1. **Nefragmentované soubory** – data jsou uložena v původním neporušeném stavu.
2. **Sekvenční fragmentace** – jednotlivé části jsou rozděleny, ale fyzicky uložené v postupném pořadí.
3. **Nesekvenční fragmentace** – části dat jsou rozloženy nesouvisle, často na různých místech v úložišti.
4. **Chybějící fragmenty** – některé části jsou nedostupné nebo ztracené, což vede k neúplnosti výsledného obsahu.
5. **Vnořené fragmenty** – struktura obsahuje další vložené prvky, které mohou být rovněž částečné nebo nekompletní.
6. **Spletené soubory** – datové segmenty několika různých souborů jsou proloženy a vzájemně promíchány.

V datové sadě se vyskytuje celkem 27 různých typů souborů, které jsou kategorizovány podle svého formátu, jak ilustruje tabulka 5.1.

Kategorie souboru	**Formáty**
Grafika	JPG, PNG, BMP, GIF, TIF, PCX
Dokumenty	DOC, XLS, PPT, PDF
Archivy	7Z, BZ2, GZ, TAR, WIM, RAR, ZIP
Zvuk	MP3, WAV, AU, WMA
Video	MP4, AVI, MOV, FLV, MPG, WMV

Tabulka 5.1: Typy souborů zahrnuté v datové sadě s multimediálním obsahem.

K dispozici je celkem 30 obrazů disku, které jsou komprimovány do formátu `bz2`. Každý obraz disku představuje kombinaci jedné z výše uvedených kategorií souborů a jednoho z definovaných typů fragmentace. Žádný z obrazů neobsahuje více než 10 souborů stejného formátu. První soubor je vždy uložen od pevně stanovené pozice sektoru s hodnotou 10 000. Soubory se nahrávají přímo na surový diskový prostor, který neobsahuje žádný souborový systém. Jednotlivé obrazy disků jsou systematicky pojmenovány podle zvoleného stupně fragmentace a kategorie obsažených dat. Ke každému vytvořenému disku ve formátu `dd` jsou rovněž poskytnuty původní, tj. nefragmentované soubory definované v tabulce 5.1. To umožňuje přímé porovnání výsledků obnovy dat s původními daty. Součástí je také podrobný popis uložených dat na každém obrazu. Ten zahrnuje položky, jako je název souboru, jeho velikost a rozsah alokovaných sektorů.

[2]Datová sada s multimediálním obsahem: https://cfreds-archive.nist.gov/FileCarving/.

5.1.2 Generovaná datová sada

Podobně jako výše popsaná datová sada ze sekce 5.1.1, i tato se zaměřuje na fragmentaci souborů. Obsahuje identické varianty fragmentací, které jsou dostupné[3] v rámci jednoho rozsáhlého archivu obsahujícího všechny obrazy disků ve formátu `dd`. Archivy jsou rozděleny na 2 části. První část uchovává disky s grafickými soubory, druhá pak disky s video soubory. Mezi grafickými soubory se nachází formáty `jpg`, `png`, `bmp`, `tiff` a `gif`. U video souborů jsou k dispozici formáty `mov`, `avi`, `ogv`, `3gp`, `wmv` a `mp4`. Všechny položky se nahrávají na disky bez souborových systémů, proto se datová sada zaměřuje na techniky vyřezávání dat.

Výhodou této datové sady je poskytnutí originálních, nefragmentovaných souborů, což umožňuje zpětnou kontrolu obnovených dat. Kromě hotových obrazů disků jsou přiloženy také zdrojové skripty, které umožňují generovat specifické části datové sady dle vlastních potřeb. K tomuto účelu se využívá skriptovacího jazyka C Shell spolu s programovacím jazykem Python. Nicméně, dokumentace k využití zdrojových skriptů je na nízké úrovni a neposkytuje bližší informace k jejich použití. Kvůli značné velikosti jsou již vygenerované obrazy komprimovány a dostupné v archivech `bz2` a `zip`.

5.1.3 Datová sada se souborovými systémy

Datová sada[4] se zaměřuje na obnovu dat v široké škále testovacích scénářů. Je vytvořena především pro nástroje, které obnovují data na základě metadat. Na rozdíl od technik obnovy založených na vyřezávání dat, zde se předpokládá využití metadat v souborovém systému. Obrazy pevných disků v této sadě obsahují několik běžně používaných souborových systémů, a to v jednom či více diskových oddílech rozdělených pomocí tabulky oddílů MBR. Přesné rozdělení souborových systémů podle pojmenování obrazů je následující:

- **FAT** – obsahuje tři oddíly formátované souborovými systémy FAT12, FAT16 a FAT32.
- **XFAT** – obsahuje jeden oddíl formátovaný souborovým systémem exFAT.
- **NTFS** – obsahuje jeden oddíl formátovaný souborovým systémem NTFS.
- **EXT** – obsahuje tři oddíly se souborovými systémy ext2, ext3 a ext4.
- **OSX** – obsahuje čtyři oddíly, na které jsou nahrány varianty souborového systému HFS+ s žurnálováním a bez něj.

Autor v dokumentaci datové sady upozorňuje na specifickou vlastnost oddílu se souborovým systémem FAT12, který je pod linuxovým nástrojem `fdisk` identifikován jako souborový systém FAT12, avšak ve skutečnosti obsahuje 16bitovou tabulku FAT. Tato skutečnost ve většině případů způsobuje jeho chování jako souborového systému FAT16.

Obsahuje celkem 27 různých testovacích scénářů. Mezi ně se řadí různé typy fragmentace souborů a adresářů, obnova jednoho a několika rozsáhlých souborů, které byly přepsány jinými soubory, a testování zobrazení velkého počtu nepoškozených souborů. V datové sadě se využívá pouze formát `txt`. Každý obraz je navržen tak, aby obsahoval právě jeden soubor. To umožňuje pozorovat chování testovaných nástrojů v konkrétních situacích. V popisu datové sady autor uvádí obecný postup pro tvorbu těchto obrazů, společně s informacemi o potřebných nástrojích.

[3] Generovaná datová sada: https://cfreds-archive.nist.gov/filecarvingtestreports.html.
[4] Datová sada se souborovými systémy: https://cfreds-archive.nist.gov/dfr-test-images.html.

5.1.4 Shrnutí

Z provedeného průzkumu existujících datových sad vyplývá, že popisované sady jsou především zaměřené na fragmentaci souborů. Je tomu tak i u datové sady ze sekce 5.1.3, která se navíc zaměřuje i na další pokrytí scénářů. V datových sadách se hojně využívá ukládání dat přímo na surový diskový prostor, bez jakéhokoliv souborového systému. To umožňuje detailní přehled o obnově dat bez použití metadat souborového systému, ale zároveň to omezuje možnosti testování obnovy dat v kontextu reálných souborových systémů a jejich specifických struktur.

Velkou výhodou zmíněné datové sady ze sekce 5.1.3 je, že pokrývá nejběžnější souborové systémy, ale zastupuje pouze textový dokument formátu `txt`. V ideálním případě by mohla zahrnovat širší pokrytí formátů souborů. Namísto dalšího rozšiřování scénářů spojených s fragmentací by bylo vhodné zvážit návrh datové sady, která by se soustředila na odlišnou problematiku spojenou s obnovou dat na jednotlivých souborových systémech.

5.2 Popis vlastní datové sady

Na základě analýzy vybraných datových sad v sekci 5.1 a jejich zjištěných nedostatků jsem se rozhodl navrhnout vlastní datovou sadu. Cílem je komplexní datová sada, která by umožňovala automatizované, plně reprodukovatelné a flexibilní generování testovacích dat. Tento přístup eliminuje nutnost stahování rozsáhlých, již hotových datových sad. Možnost dynamického generování zajistí vytvoření specifického scénáře a přizpůsobení velikosti výstupních dat aktuálním požadavkům. Návrh zahrnuje specifikaci podporovaných tabulek oddílů, souborových systémů, formátů souborů a scénářů zahrnutých do datové sady. Datová sada zahrnuje vybrané souborové systémy ze sekce 2.3, a to konkrétně:

- FAT16 a FAT32,
- ExFAT,
- NTFS,
- Ext2, Ext3 a Ext4,
- Btrfs,
- HFS+,
- APFS.

Výběr těchto souborových systémů jsem provedl na základě analýzy poskytované podpory u vybraných nástrojů ze sekcí 4.2 a 4.3. Jednotlivé souborové systémy mohou být v rámci disku také kombinovány s tabulkami oddílů MBR a GPT, které byly rozebrány v sekci 2.2. Souborové systémy HFS+ a APFS budou vytvářeny na operačním systému macOS a pouze v kombinaci s formátem GPT. Ostatní zmíněné souborové systémy používají pro své vytvoření prostředí Linux. Tabulku oddílů APM jsem v návrhu datové sady vynechal, z důvodu její omezené podpory v novějších operačních systémech. Obdobně není do návrhu zahrnut souborový systém HFS, jehož podpora je také omezena. V linuxových distribucích navíc umožňuje pouze čtení a jeho vytváření, nikoliv plnohodnotnou správu. Souborový systém FAT12 je vynechán s ohledem na jeho limitovanou maximální velikost,

která by i u relativně malých disků vedla k automatické konverzi na formát FAT16. Tento fakt potvrzuje poznatek autora datové sady v sekci 5.1.3.

Scénáře pro datovou sadu byly zvoleny na základě reálných situací ztráty dat. Proto jsem vybral dva různé scénáře, které se využívají také při testování účinnosti nástrojů pro obnovu dat v článku [35]. Jedná se o následující scénáře:

- **Smazané soubory**: Tento scénář simuluje běžné odstranění souboru na úrovni operačního systému, kdy nedochází k okamžitému fyzickému přepsání obsahu. Jedná se o běžný případ v reálném prostředí, který je závislý na době odchycení od smazání.
- **Přepsané souborové systémy**: Tento scénář reprezentuje situaci, kdy je původní souborový systém nahrazen novým například prostřednictvím tzv. rychlého formátování. Tím dochází k přepsání původních metadat či nahrazení celé struktury novým souborovým systémem.

Pro obě varianty se budou využívat stejné sady souborů. Vybral jsem nejběžněji používané formáty souborů v rámci každé kategorie. Zdrojem souborů je multimediální sada zmíněná v sekci 5.1. Celkem je použito 21 různých formátů multimediálních dat, které jsou kategorizovány v tabulce 5.2.

Kategorie souboru	**Vybrané formáty**
Zvuk	MP3, WAV, WMA
Video	MP4, AVI, MOV, MPG, WMV
Grafika	PNG, JPG, TIF, GIF, BMP
Dokument	PDF, XLSX, DOCX, PPTX
Archiv	7z, TAR, RAR, ZIP

Tabulka 5.2: Zvolené soubory obsaženy ve vlastní datové sadě.

Pro určení celkového počtu možných kombinací souborů, které mohou být vygenerovány z množiny 21 souborů, jsem použil kombinační vzorec.

$$\sum_{k=1}^{21} \binom{21}{k} = 2^{21} - 1 = 2\,097\,151 \tag{5.1}$$

Výsledná hodnota z rovnice 5.1, tedy 2 097 151, udává teoretický maximální počet všech unikátních možných kombinací typů souborů, které se vytvoří pro jeden testovaný případ v rámci daného souborového systému. Celková komplexnost datové sady, uvažuje-li se jeden konkrétní scénář (např. smazání dat), je dána kombinací deseti souborových systémů a tří variant rozložení disku. To znamená, že by existovalo 62 914 530 různých konfigurací obrazů disku z hlediska kombinace souborů a rozložení disku. Tato obrovská potenciální variabilita jasně ukazuje, že manuální vytvoření takto rozsáhlé části datové sady je nemožné. Při zavedení selektivního omezení, například filtrování pouze největších kombinací souborů, je možné velikost výsledné datové sady výrazně snížit. Pokud by pro každý souborový systém byla generována právě jedna taková maximální kombinace, odpovídal by výsledný počet diskových obrazů přímo počtu testovaných souborových systémů.

5.3 Návrh automatizovaných skriptů pro tvorbu datové sady

S ohledem na uvedenou variabilitu datové sady v předchozí sekci 5.2, je potřeba vytvořit prostředky pro její automatizované generování. Skript je řízen sadou vstupních parametrů, které umožňují konfiguraci vlastností generované datové sady. Mezi parametry jsem zahrnul volbu souborových systémů, tabulek diskových oddílů a rozsah jednotlivých kombinací. Rozsah pro generování obsahu datové sady je omezen na tři režimy:

1. **Všechny kombinace souborů bez opakování** – ze všech souborů jsou vytvořeny jejich unikátní kombinace.
2. **Největší kombinace** – obraz obsahuje vždy všechny dostupné soubory najednou.
3. **Izolované kombinace** – každý obraz obsahuje pouze jeden soubor daného typu.

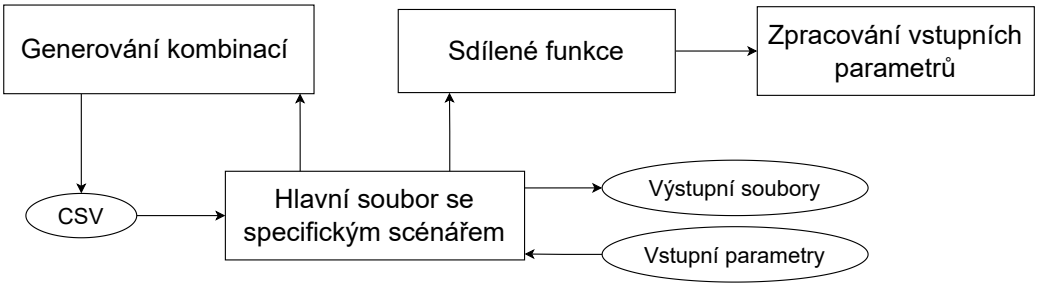
Všechny kombinace jsou následně uloženy do souboru ve formátu `csv`, který slouží jako vstupní data pro další zpracování. Při spuštění celého skriptu dochází nejprve ke zpracování vstupních argumentů. Pokud již existuje soubor `csv` obsahující kombinace souborů, provádí se jejich postupné načítání. V opačném případě jsou kombinace nejprve vygenerovány a uloženy do tohoto souboru. Další postup zpracování zahrnuje následující kroky:

1. **Vytvoření diskového obrazu**: V prvním kroku dochází k vytvoření prázdného souboru formátu `dd`, který slouží jako obraz disku o zvolené velikosti. Tvorba probíhá pomocí příkazu `dd`.
2. **Inicializace tabulky oddílů**: Následuje krok inicializace tabulky oddílů na vytvořeném diskovém obrazu. Tento krok se provádí pouze v případech, kdy má výsledná datová sada obsahovat diskové oddíly. K tomu se využívají nástroje `gdisk` a `fdisk` na Linux, `diskutil` na macOS.
3. **Formátování souborových systémů**: Po případné inicializaci tabulky oddílů dochází k formátování vytvořeného diskového oddílu nebo přímo celého diskového obrazu na specifikovaný souborový systém. Na Linux se využívá nástroj `mkfs` a na macOS nástroj `diskutil`.
4. **Připojení diskového obrazu**: Obraz disku je připojen k operačnímu systému do dočasné složky. Připojení probíhá na Linux pomocí příkazů `losetup` a `mount`, na macOS pomocí příkazu `hdiutil`.
5. **Kopírování souborů**: Po úspěšném připojení diskového obrazu následuje operace kopírování předem vybraných testovacích souborů do jeho struktury. Výběr kopírovaných souborů je řízen aktuálně zpracovávanou kombinací datové sady. K samotnému kopírování souborů se používá příkaz `cp`.
6. **Simulace daného scénáře**: V tomto kroku dochází k aplikaci specifického scénáře ztráty dat na připojený diskový obraz. V závislosti na zvoleném scénáři jsou prováděny odpovídající operace.
7. **Odpojení diskového obrazu**: Po dokončení dochází ke korektnímu odpojení diskového obrazu od operačního systému. Na Linux se používají příkazy `losetup` a `umount`, na macOS zas nástroj `hdiutil`.

8. **Záznam informací**: Posledním krokem v procesu generování datové sady je zaznamenání informací popisujících vytvořený obraz. Tyto informace, které zahrnují specifikaci použitého souborového systému, nahrané kombinace souborů a další, jsou ukládány do strukturovaného datového souboru ve formátu `csv`.

9. **Komprese výstupního obrazu**: Po uložení obrazu může dojít k volitelnému kroku, kterým je komprese. K tomu se využívá kompresní algoritmus `Zstandard` (formát zstd), který dokáže výslednou velikost obrazu snížit o 60-90 %. Míra komprese se ovšem odvíjí od původní velikosti a jeho zaplnění.

Celkový postup, jakým skript postupuje bez detailního popisu zpracování disku, zjednodušeně zachycuje vývojový diagram na obrázku A.1. Celý proces jsem navrhl s ohledem na modularitu, což usnadní případnou budoucí rozšiřitelnost. Jednotlivé kroky, jako je formátování souborových systémů nebo vykonání daných scénářů ztráty dat, budou z těchto důvodů samostatnými funkcemi a moduly. Obecné rozdělení do samostatných skriptů a jejich propojení zobrazuje obrázek 5.1.



Obrázek 5.1: Propojení jednotlivých skriptů pro generování datové sady.

5.4 Návrh doprovodných skriptů

Pro testování obnovy dat bylo potřeba vytvořit sadu doprovodných skriptů, jejichž hlavním cílem je automatizovat rutinní operace, které by jinak bylo nutné provádět manuálně. To by vedlo ke zvýšené chybovosti a časové náročnosti. Skripty se zaměřují na usnadnění práce s virtuálními obrazy disků, včetně jejich zpracování na různých operačních systémech, a na vyhodnocení úspěšnosti obnovy dat. Tato sekce se věnuje návrhu funkcionality daných skriptů. První část (sekce 5.4.1) se zabývá návrhem pro připojování a odpojování diskových obrazů pro systémy Windows a Linux. Druhá část (sekce 5.4.2) představuje návrh automatizovaného vyhodnocování úspěšnosti obnovy dat z disků datové sady.

5.4.1 Skripty pro připojování virtuálních obrazů disků

Pro účely práce s rozsáhlou datovou sadou a zajištění opakovatelného zpřístupnění obsahu virtuálních obrazů disků v různých operačních systémech jsem navrhl systém poloautomatického připojování těchto disků. Navržené řešení se skládá ze dvou částí. Každá z nich je určena pro jiný operační systém – Windows a Linux. V obou případech je cílem automatizovat proces výběru a připojení obrazových souborů ve formátu `dd` se zajištěním použití správných parametrů.

Skript nabízí interaktivní výběr obrazových souborů ze zvolené složky. Výběr se může provést jako jednotlivý, hromadný nebo postupný. Na platformě Windows předpokládám využití prostředku schopného emulovat fyzické zařízení. To umožňuje kompatibilitu s forenzními nástroji, které vyžadují nízkoúrovňový přístup k disku. Obě varianty z výše uvedených skriptů představují důležitý podpůrný prvek pro část s měřením, které se věnuje sekce 7.1. Standardizací procesu připojování obrazů a vynucení určitých parametrů, které by se jinak musely složitě nastavovat, výrazně zrychluje celý proces.

5.4.2 Automatizované vyhodnocení výsledků obnovy

Pro usnadnění a urychlení vyhodnocení úspěšnosti obnovy dat jsem navrhl automatizovaný validační skript. Ten má za úkol analyzovat obnovená data a porovnávat je s původními soubory, které byly nahrány do datové sady. Porovnávání probíhá na základě obsahu souborů. Skript využívá `csv` soubor s informacemi o vytvořeném disku. Z něho čerpá především název disku a nahrané kombinace souborů. Tyto informace slouží k dohledání odpovídajících původních souborů a jejich následnému porovnání s obnovenými daty. Předpokládá se, že složka s obnovenými daty je pojmenována stejně jako zpracovávaný disk, což umožňuje zpětné dohledání potřebných informací.

Pro vyhodnocení jednotlivých kategorií jsem musel využít vhodné metody. Navržený přístup je založen na dvoustupňové kontrole. V první fázi se kontroluje, zda jsou oba soubory identické. Pokud se ovšem liší, přistupuje se k podrobnějšímu porovnání obsahu, které je specifické pro daný typ souboru. Výsledkem vyhodnocení analýzy mohou být 4 různé případy, které specifikuje tabulka 5.3. Pro kategorizaci těchto výsledků a metodiku jejich vyhodnocení jsem se inspiroval článkem [9], který se zabývá standardizací klasifikace a autentizace obnovených dat. Výsledky nástroj zapíše do souboru ve formátu `csv`, který shrnuje procentuální shodu obnovených dat s původními pro každý načtený obraz disku.

Hodnota	Význam
I	Soubory jsou bitově identické.
M	Soubor chybí v obnovených datech.
C	Soubor je poškozený nebo nelze zpracovat.
0 % – 100 %	Procentuální shoda obsahu mezi soubory. 0 % znamená, že soubory se liší, 99 % znamená minimální rozdíl

Tabulka 5.3: Význam výstupních hodnot ve výsledcích validace obnovených dat.

Kapitola 6

Implementace

Tato kapitola se zaměřuje na implementaci nástrojů, které byly navrženy v rámci předchozí kapitoly 5. Cílem je poskytnout přehled o jejich technickém řešení a souvisejících vývojových rozhodnutích. První sekce 6.1 popisuje sadu skriptů pro automatizované generování datové sady a vysvětluje jednotlivé dílčí části, které se k tomu využívají. Sekce 6.2 rozebírá skript pro práci s virtuálními diskovými obrazy na systémech Windows a Linux. Poslední sekce 6.3 se zabývá nástrojem pro automatizované vyhodnocení úspěšnosti obnovených dat. Popisuje princip zpracování a použité metody pro porovnávání u jednotlivých kategorií.

6.1 Skripty pro automatizované generování datové sady

Pro zajištění plně automatizované tvorby datové sady jsem implementoval sadu skriptů, které využívají skriptovací jazyk Bash. Volba tohoto jazyka vycházela ze snahy minimalizovat závislost na externích nástrojích, které vždy nemusí být součástí operačního systému. Tvorba jednotlivých skriptů vycházela z návrhu v sekci 5.3. Celý generovací proces je proto založen na modulární architektuře. Ta se skládá z hlavních skriptů, které jsou umístěny v adresáři `scripts/` a zpracovávají scénáře ztráty dat. Dále také z pomocných skriptů, které poskytují sdílenou funkcionalitu pro hlavní skripty. Všechny pomocné skripty jsou umístěny v podadresáři `common/`. Vytvořené komponenty jsou následující:

- **Sdílená knihovna funkcí** (*common_function.sh*)

  Tento soubor slouží jako centrální místo pro často používané operace. Obsahuje sadu funkcí, které umožňují snadno a opakovaně provádět úkony, jako je tvorba diskového obrazu, tabulek diskových oddílů či formátování jednotlivých souborových systémů.

- **Zpracování argumentů** (*process_args.sh*)

  Script je zodpovědný za zpracování a validaci argumentů předaných hlavním skriptům jako vstupní parametry. Na základě argumentů nastavuje globální proměnné např. pro zpracovávané souborové systémy či cesty k adresářům. Navíc zajišťuje detekci operačního systému, podle kterého se v ostatních modulech volí vhodné nástroje a příkazy. Je přímo napojena na knihovnu sdílených funkcí, proto je možné ji využívat i bez explicitního připojení k hlavnímu skriptu.

- **Generátor kombinací souborů** (*combinations.sh*)

  Hlavním úkolem tohoto pomocného skriptu je příprava konfigurace pro jednotlivé diskové obrazy. Původní implementaci jsem zpracoval v jazyce Bash. Avšak kvůli

omezeným možnostem efektivního zpracování dat a absenci pokročilých nástrojů pro generování kombinací jsem výpočetní krok nahradil implementací v jazyce Python. Z důvodu kompatibility byl tento kód vložen přímo do skriptu Bash. Pro vygenerování kombinací ze zdrojových souborů se využívá knihovna `itertools`, která zajišťuje jejich vytvoření bez opakování. Pro každou kombinaci se navíc vypočítá celková velikost a tyto informace se zapíší do výstupního souboru `combinations.csv`.

Skripty jsou tvořeny tak, aby byly přenositelné mezi systémy Linux a macOS. Vzhledem k tomu, že výchozí verze Bash na macOS (obvykle verze 3.2) nepodporuje některé pokročilé funkce, je potřeba aktualizovat alespoň na verzi 4.0 či novější. Jedná se například o podporu pro asociativní pole. To lze jednoduše provést prostřednictvím nástroje `Homebrew`. Všechny z uvedených skriptů byly vyvíjeny a testovány s verzí 5.2.

Následující část této sekce se věnuje hlavním skriptům, které zpracovávají specifické scénáře. V rámci implementace jsem vytvořil čtyři hlavní skripty, které pokrývají dva různé scénáře představené v sekci 5.2. Podle nich jsou rozděleny soubory implementující jejich funkcionalitu. Pojmenování jednotlivých skriptů vychází ze vzoru `pt-fs_nazev_scenare`, kde každá z částí identifikuje specifický aspekt. Použité zkratky a jejich význam jsou shrnuty v rámci tabulky 6.1.

Zkratka	Význam
`pt`	Skript využívá disk s tabulkou oddílů a vytváří 1 oddíl.
`fs`	Dochází k formátování oddílu nebo celého disku souborovým systémem.
`del_files`	Simuluje scénář, při kterém jsou soubory na disku smazány.
`replace_fs`	Reprezentuje scénář, kdy je existující souborový systém přeformátován.

Tabulka 6.1: Konvence pojmenování skriptů pro generování datové sady.

Každý z těchto skriptů pracuje se souborem `combinations.csv`, který definuje kombinace zdrojových souborů. Skript iterativně prochází tento soubor a pro zpracovávanou kombinaci generuje diskové obrazy pro všechny zvolené souborové systémy. Postup generování v jednotlivých skriptech odpovídá navrženému postupu v sekci 5.3, včetně využití zvolených příkazů. Velikost při tvorbě diskového obrazu je dynamicky určena na základě celkové velikosti zpracovávané kombinace souborů. Pro zajištění dostatečného prostoru pro souborový systém a data je k celkové velikosti souborů připočtena rezerva 30 %. Následně je vybrána nejbližší vyšší hodnota z předdefinovaných velikostí: 32 MB, 50 MB, 80 MB a 150 MB. Úkony, které se vykonávají na vytvořeném obrazu disku, se liší v závislosti na vybraném skriptu s jeho scénářem. Tyto rozdíly jsou rozebrány níže.

Skript fs_del_files.sh

Tento skript simuluje scénář běžné ztráty dat způsobený smazáním souborů na úrovni souborového systému. Při tvorbě obrazu se formátuje souborovým systémem celý prostor. Na něj se postupně kopíruje daná kombinace souborů pomocí `cp`. K synchronizaci změn se volá příkaz `sync` a následně se odstraní soubory, což je řešeno příkazem `rm`. Informace o každém obrazu jsou zapsány do souboru `fs_del_files_info.csv`, který je uložen k vytvořené datové sadě do adresáře `datasets/del_files/`. Ten obsahuje název obrazu, použitý souborový systém, seznam zpracované kombinace a také kontrolní součty (MD5 a SHA512) pro případné ověření integrity. Výsledkem je sada diskových obrazů obsahujících souborový systém se stopami po smazaných souborech.

Skript fs__replace__fs.sh

Skript modeluje scénář, kdy jsou data ztracena v důsledku rychlého přeformátování souborovým systémem. Stejně jako předchozí skript využívá celý diskový prostor pro formátování souborového systému a následné kopírování kombinací. Dalším krokem je opětovné naformátování celého obrazu jiným souborovým systémem. Tím dojde k přepsání původních struktur souborového systému na novou prázdnou strukturu. V posledním kroku dochází také k zápisu informací do souboru `fs_replace_fs_info.csv`, kde se navíc zapisuje nově vytvořený souborový systém. To je uloženo do adresáře `datasets/replace_fs/`. Tento postup se iterativně opakuje pro každou konkrétní kombinaci souborů a výchozí souborový systém. K ukončení dojde až po vytvoření všech obrazů, ve kterých je původní souborový systém nahrazen každým z ostatních definovaných souborových systémů. Tím vznikají různé kombinace přepisů, které pokrývají všechny možné varianty nahrazení jednoho souborového systému jiným souborovým systémem.

Skripty pt-fs__del__files.sh a pt-fs__replace__fs.sh

Oba skripty rozšiřují výše popsané scénáře o práci s diskovou strukturou obsahující tabulky oddílů. Po vytvoření diskového obrazu se nejprve vytvoří vybraná tabulka oddílů, která definuje jeden oddíl zabírající většinu dostupného prostoru. Teprve tento oddíl je následně naformátován vybraným souborovým systémem. Zbývající kroky pak odpovídají postupu použitému v předchozích skriptech.

K dosažení maximální konzistence výstupních obrazů napříč různými distribucemi Linux jsem explicitně specifikoval parametry souborových systémů. Ty byly vytvářeny pomocí příkazu `mkfs`. Minimalizuje se tak riziko odlišného chování či struktury souborového systému. Toto riziko by mohlo vzniknout kvůli rozdílným výchozím hodnotám parametrů nebo verzím nástroje. Drobné odchylky v metadatech mohou způsobit, že jednotlivé instance datové sady nebudou bitově identické. Nicméně, dodržení přesně definovaného postupu generování zajišťuje jejich obsahovou konzistenci a reprodukovatelnost.

6.2 Skripty pro připojování virtuálních obrazů disků

V prostředí Windows jsem vytvořil specializovaný dávkový skript `windows_automount.bat`. Důvodem volby formátu dávkového skriptu (`bat`) byla jeho nativní podpora v systému Windows. Tím se eliminuje potřeba instalovat dodatečné prostředky a závislosti pro spuštění. Hlavní funkcionalitu zajišťuje program třetí strany **OSFMount**[1] verze `3.1.1003`. Pro správnou činnost skriptu je předpokladem mít tento program nainstalován. Jedná se o volně dostupnou aplikaci od společnosti PassMark Software, která se zaměřuje na připojování různých typů obrazů disků. Ačkoliv je primárně navržena s grafickým uživatelským rozhraním, podporuje také spuštění pomocí příkazové řádky. Právě této vlastnosti jsem využil pro její integraci do automatizovaného pracovního postupu tohoto skriptu. Výhodou je jeho schopnost připojit obraz nejen jako logickou jednotku, ale také emulovat fyzické zařízení. Využití je následně možné u forenzních nástrojů, které vyžadují nízkoúrovňový přístup k disku. Pro zajištění korektního připojení jsou předávány parametry `-o rem,ro,physical`. Tím je docíleno připojení definující daný disk jako vyměnitelné médium určené pouze pro čtení, které emuluje fyzický disk. Tyto parametry jsou předávány nástroji OSFMount při každém volání

[1] Program OSFMount je dostupný z: https://www.osforensics.com/tools/mount-disk-images.html.

příkazu pro jeho připojení. Po spuštění skriptu se vyžaduje interakce uživatele pro výběr obrazů. Před každým novým připojením, stejně jako před případným ukončením skriptu, se využívá automatického odpojení všech dříve připojených virtuálních disků spravovaných nástrojem OSFMount.

Obdobný přístup k automatizaci připojování jsem implementoval i pro systém Linux. K vytvoření skriptu `linux_automount.sh` jsem použil jazyk Bash. Narozdíl od předchozího skriptu, tato varianta využívá standardní systémové nástroje dostupné v linuxových distribucích. Svou funkcionalitu skript rozšiřuje o možnost zadávání vstupních parametrů přímo při spuštění z příkazové řádky a možnost automatizované obnovy dat pomocí nástroje PhotoRec a Scalpel. V závislosti na zvoleném režimu skript provádí odlišné akce. V režimu připojení s parametrem `mount` dochází ke stejným úkonům, které jsou po uživateli požadovány jako u verze pro Windows. Pro každý vybraný disk je příkazem `losetup` vytvořeno smyčkové zařízení, které je následně připojeno pomocí `mount` do dočasného adresáře. Ten je definován cestou `/mnt/` a pojmenování adresáře odpovídá názvu zpracovávaného obrazu. Před ukončením nebo při změně výběru obrazu dochází nejdříve k odpojení všech připojených složek příkazem `umount` a uvolnění připojených smyčkových zařízení. Podrobnější informace o použití jednotlivých skriptů, včetně specifikace parametrů a dalších detailů, jsou uvedeny v souboru `README` u těchto skriptů.

6.3 Automatizované vyhodnocení výsledků obnovy

Pro vyhodnocování obnovených souborů a jejich porovnání s původními daty jsem vytvořil automatizovaný validační skript `recovery_validation.py`. Skript jsem implementoval v jazyce Python, protože nabízí rozsáhlé knihovny pro práci s multimediálními daty. Skript využívá modul `argparse` pro zpracování vstupních parametrů. Jednotlivé vstupní parametry a jejich použití jsou detailněji rozepsány v souboru `README`, který se nachází u daného skriptu. Kromě povinných parametrů předpokládá skript existenci adresáře s původními (referenčními) soubory, které slouží jako základ pro porovnání. V současné implementaci je cesta k tomuto adresáři pevně stanovena a odpovídá definované struktuře. Po zpracování vstupních parametrů jsou původní soubory indexovány podle přípon do slovníku. Pro každou z nich je uložena cesta k souboru a vypočtený MD5 hash. Jeho uložením před procesem validace dojde k výraznému zrychlení, protože se nemusí počítat při každém porovnání. Důležitou částí zpracování je načtení informací ze souboru `csv`, který obsahuje seznam disků a odpovídajících očekávaných souborů. Tento soubor je součástí datové sady a vzniká po dokončení jejího generování.

Proces zpracování obnovených souborů řídí funkce `process_recovered_folder()`. Tato funkce postupně iteruje přes všechny podadresáře v zadané složce s obnovenými daty, přičemž každý podadresář reprezentuje jeden analyzovaný diskový obraz. Jak již bylo zmíněno, skript očekává specifickou konvenci pojmenování adresářů obnovených dat, která musí odpovídat názvům disků uvedeným v souboru `csv`. Díky této konvenci lze extrahovat jméno obrazu, které slouží jako klíč pro vyhledání a jednoznačné určení seznamu očekávaných souborů v dříve načtené struktuře. Následně se podle přípon původních souborů vyhledávají odpovídající formáty mezi obnovenými daty. Pro každou kategorii souborů je využita specializovaná třída. Interakce s těmito třídami probíhá prostřednictvím jednotného rozhraní, kterým je metoda `compare`. Tato metoda slouží jako vstupní bod pro validaci souborů. Veškeré vnitřní mechanismy jsem implementoval v interních metodách dané třídy.

Každá z těchto tříd provádí validaci ve **dvou fázích**. První fáze je společná pro všechny typy souborů. Spočívá v porovnání hashů MD5 pro rychlé určení identické shody. Pokud se hashe shodují, další validace není nutná. Druhá fáze je specifická pro daný typ souboru a zahrnuje podrobnější analýzu obsahu. Ve všech třídách jsem použil bloky `try-except`, zejména při operacích s jednotlivými soubory (otevírání, čtení a analýza). V případě chyby je výjimka zachycena a soubor je označen jako poškozený. Jednotlivé třídy, včetně jejich metod použitých k vyhodnocování, jsou popsány dále.

Třída ImageProcessor

Pro obrazová data slouží třída `ImageProcessor`. S využitím knihoven `Pillow`, `OpenCV` a `scikit-image` porovnává obrazy pomocí metriky strukturální podobnosti (SSIM). Tato metoda je založena na hodnocení vizuálních rozdílů mezi dvěma obrazy. Oproti prostému porovnání jednotlivých pixelů, SSIM zohledňuje změny jasu, kontrastu a struktury. Pokud dojde k rekonstrukci, změně nebo poškození těchto vlastností, porovnání pixel po pixelu by mohlo vést ke zkreslenému hodnocení, přestože by byl vizuální výsledek podobný.

Třída AudioProcessor

Tato třída se věnuje audio souborům. Ta využívá knihovny `pydub`, `numpy` a `scipy`. Zvukové signály jsou nejprve převedeny na jednotnou vzorkovací frekvenci a bitovou hloubku. Následně se provádí časová synchronizace signálů pomocí křížové korelace, která umožňuje zarovnat případné posuny. Po zarovnání jsou signály normalizovány a je vypočten absolutní rozdíl mezi odpovídajícími vzorky. Výsledná podobnost je určena jako podíl normalizovaných vzorků, jejichž rozdíl nepřesahuje stanovenou toleranci.

Třída VideoProcessor

Kontrola video souborů ve třídě `VideoProcessor` kombinuje vizuální a zvukovou analýzu. Vizuální část porovnává jednotlivé snímky videa pomocí SSIM, obdobně jako u obrázků. U druhé části je extrahována zvuková stopa a porovnána metodou popsanou výše pro audio soubory. Výsledná podobnost je určena průměrnou hodnotou obou hodnocení.

Třída ArchiveProcessor

U archivů se předpokládá, že i drobná odchylka v obsahu může vést k nefunkčnosti celého archivu. Proto třída `ArchiveProcessor` využívá porovnání na úrovni jednotlivých bajtů, které dokáže detekovat i minimální rozdíly v poškozeném archivu.

Třída DocumentProcessor

Zpracování dokumentových souborů se věnuje třída `DocumentProcessor` a cílem je zhodnocení zachování informačního obsahu. Tyto formáty často obsahují komplexní struktury, do kterých lze zahrnout styly, metadata nebo obrázky. Z obou souborů je extrahován pouze textový obsah (v případě tabulek se extrahuje obsah buněk) a následně je vypočtena procentuální podobnost na základě množiny unikátních řetězců. K tomu se používají specializované knihovny pro jednotlivé formáty: `openpyxl` pro soubory `xlsx`, `python-docx` pro `docx`, `PyPDF2` pro `pdf` a `python-pptx` pro `pptx`. Tato metoda umožňuje posoudit, zda byly obnoveny textové informace bez ohledu na původní formátování.

Následující tabulka 6.2 shrnuje výše zvolené metody, které jsou použity pro porovnání obsahu v druhém kroku validace.

Typ souboru	**Použitá metoda porovnání**
Grafika	Strukturální podobnost obrazu (SSIM)
Video	SSIM pro jednotlivé snímky a křížová korelace zvukové stopy
Zvuk	Časové zarovnání pomocí křížové korelace a porovnání vzorků
Archiv	Binární porovnání na úrovni bajtů
Dokument	Porovnání extrahovaného textového obsahu

Tabulka 6.2: Použité metody pro porovnání obsahu v rámci jednotlivých kategorií souborů.

Nástroj `recovery_validation.py` vytváří výstupní soubor formátu `csv` pojmenovaný jako `recovery_report.csv`, který obsahuje přehled o úspěšnosti obnovení jednotlivých souborů. Každý řádek popisuje jeden zpracovávaný diskový obraz. Ve sloupcích tohoto souboru je možné nalézt následující hodnoty:

- **Disk** – název diskového obrazu;
- **Expected** – počet původně nahraných souborů;
- **Recovered** – počet obnovou nalezených souborů;
- **Summary** – celková procentuální úspěšnost obnovy;
- úspěšnost obnovení pro každý očekávaný soubor.

Výsledný přehled je seřazen podle celkové úspěšnosti. Graficky upravenou ukázku výstupu znázorňuje obrázek C.1.

Kapitola 7

Měření a vyhodnocení výsledků

Vzhledem k různorodosti vybraných nástrojů a jejich přístupu k obnově dat se tato kapitola zaměřuje na jejich praktické porovnání. Cílem je posoudit jejich schopnosti ve specifických podmínkách. V první části se sekce 7.1 zabývá definovanými postupy pro obnovu z vybraných programů. Druhá část v sekci 7.2 porovnává jejich kvalitu s ohledem na vyhodnocené výsledky úspěšnosti obnovy výstupních dat.

Pro účely porovnání byla použita vlastní datová sada, vytvořená pomocí sady automatizovaných skriptů představených v sekci 6.1. Ta zahrnuje následující dva scénáře:

- **Scénář 1** – znázorňuje odstranění souboru bez přepsání obsahu.
- **Scénář 2** – využívá rychlé formátování disku novým souborovým systémem.

Z důvodu vysoké časové náročnosti manuálního testování nebyly použity všechny možné kombinace scénáře 2. Místo toho byl proveden jejich výběr na základě hypotézy, že architektonická rozdílnost souborových systémů významně ovlivňuje míru obnovy dat. Proto nebyly zahrnuty kombinace, ve kterých se nový souborový systém lišil pouze minimálně (např. přeformátování stejným souborovým systémem nebo mezi verzemi systému FAT či Ext). Předpokládalo se, že tyto kombinace by pravděpodobně vedly k vysoké pravděpodobnosti obnovení dat, což je činilo z hlediska testování méně přínosnými. U scénáře 2 bylo z celkových 68 kombinací bez tabulek oddílů vybráno 22 relevantních kombinací.

7.1 Obnova dat pomocí vybraných nástrojů

Tato sekce udává jednotné postupy obnovy, které jsem použil u každého z programů. Nástroje jsem testoval na operačních systémech Windows 10 a Kali Linux. Pro měření bylo manuálně otestováno přibližně 90 diskových obrazů v rámci každého z programů (celkem přes 450 vzorků). Většina z programů totiž neumožňuje automatizované zpracování. Pro účely dalšího zpracování v sekci 7.2 je potřeba, aby byl výstupní adresář z každého procesu obnovy pojmenován stejně jako vstupní zpracovávaný obraz disku. Veškeré zmiňované skripty v této sekci byly již představeny v kapitole 6.

7.1.1 Recuva

Po spuštění programu Recuva jsem zvolil tzv. **Advanced Mode**, který umožňuje přepnutí do pokročilejšího rozhraní. Tím se eliminuje potřeba opakovaného použití průvodce, ve kterém je nutné při každém spuštění nastavovat parametry obnovy. V nastavení jsem

explicitně vybral typ obnovovaných souborů s hodnotou „*All files*“ pro maximální rozsah obnovy. Zároveň jsem aktivoval volbu hloubkového skenování. Bezplatná verze programu Recuva nepodporuje přímé připojení virtuálních diskových obrazů, a proto jsem použil vytvořený skript `windows_automount.bat`. Díky němu není potřeba pořizovat licencovanou verzi, která tuto podporu nabízí. Po úspěšném připojení obrazu do systému ho bylo možné zvolit jako cílové zařízení pro obnovu. Samotné skenování jsem spustil tlačítkem *Scan*. Po dokončení skenování byl zobrazen seznam nalezených souborů včetně odhadovaného stavu jejich obnovitelnosti, jak je patrné z obrázku B.1.

7.1.2 PhotoRec

Pro obnovu dat pomocí nástroje PhotoRec jsem využil skript `linux_automount.sh`. Při spuštění s přepínačem `-p photorec` je zajištěno automatické spuštění nástroje PhotoRec pro všechny specifikované obrazy disku bez nutnosti manuálního připojování každého z nich. Pro spuštění tohoto programu se využívá příkaz `photorec` doplněný o sadu parametrů, které určují jeho chování. Použité parametry uvádí tabulka 7.1, včetně popisu jejich funkce.

Parametr	**Popis**
`/debug`	Aktivace režimu ladění pro podrobnější výstup.
`/log`	Umožnění logování činnosti nástroje do souboru.
`/d $output_dir`	Specifikuje výstupní adresář pro obnovené soubory.
`/cmd $img`	Určení vstupního obrazu disku (vybraný soubor formátu dd).
`wholespace`	Prohledání celého prostoru disku.
`fileopt`	Aktivace možnosti týkající se obnovy souborů s následujícími podvolbami: `everything` – Povolení obnovy všech typů souborů. `disable` – Zákaz interaktivního vyhledávání během obnovy.
`search`	Spuštění procesu vyhledávání a obnovy dat.

Tabulka 7.1: Použité parametry pro spuštění programu PhotoRec.

7.1.3 Scalpel

Nástroj Scalpel jsem testoval automatizovaně pomocí skriptu `linux_automount.sh`, obdobně jak již bylo zmíněno v sekci 7.1.2. K jeho použití jsem musel připravit konfigurační soubor sestavený na základě známých vzorů jednotlivých typů souborů. Základní množinu formátů pokrýval výchozí konfigurační soubor[1], který jsem doplnil o chybějící vzory některých formátů. Tento soubor je dostupný v adresáři `scripts/others/` pod názvem `scalpel.conf`. Spuštění automatizovaného zpracování je v tomto případě možné s přepínačem `-p scalpel`. Přestože byl nástroj Scalpel úspěšně spuštěn v rámci několika měření, nebyl nakonec zahrnut do hlavního vyhodnocení výsledků. Důvodem byla především extrémně vysoká kvantita obnovených souborů, která v některých případech přesahovala i 30 000 položek na jeden diskový obraz. Vysoká četnost byla způsobena značným množstvím falešných nálezů, které vznikaly v důsledku sdílení identických nebo podobných vzorů mezi různými typy souborů. S tím byla úzce spojena také značná velikost výsledných dat, která vedla ke kapacitním problémům.

[1] Výchozí konfigurační soubor pro Scalpel je dostupný z: https://github.com/sleuthkit/scalpel/blob/master/scalpel.conf.

7.1.4 R-Studio pro Linux

Tento nástroj jsem využil v rámci **komerční verze**. Připojení diskového obrazu jsem provedl prostřednictvím funkce *Open Image*, která je dostupná prostřednictvím hlavního panelu, viz obrázek B.6. Výchozí filtr však nenabízí možnost výběru souborů ve formátu `dd`, a proto bylo nezbytné v dialogovém okně manuálně změnit typ souboru na „*All files*“. Následně jsem mohl diskový obraz vybrat a aplikace jej zobrazila v seznamu zařízení. Alternativně bylo možné diskový obraz připojit pomocí skriptu `linux_automount.sh`, čímž došlo k jeho detekci v programu R-Studio jako fyzického pevného disku. Pro zahájení obnovy jsem vybral příslušný disk a spustil skenování pomocí volby *Scan*. Ve všech měřeních jsem ponechal výchozí nastavení a skenování probíhalo nad celým diskovým prostorem. Po jeho dokončení se zobrazil seznam nalezených souborů, jak ukazuje obrázek B.7. U každého z nich byla uvedena odhadovaná pravděpodobnost úspěšnosti obnovy (položka *Recovery chances*). Pro samotnou obnovu bylo třeba označit požadované soubory a proces potvrdit tlačítkem *Recover Marked*. Ve specifických případech došlo k výskytu vícenásobných kopií stejných souborů. K tomuto docházelo, když byly soubory nalezeny jak pomocí metadat souborového systému, tak prostřednictvím techniky vyřezávání dat. V těchto situacích jsem volil variantu s vyšší pravděpodobností úspěšné obnovy.

7.1.5 EaseUS Data Recovery Wizard

K obnově dat pomocí tohoto nástroje jsem využil licencovanou verzi **Professional**, která odstraňuje limit pro množství obnovitelných dat. Program v žádné verzi nenabízí možnost přímého připojení obrazů disků ani specifikaci možností konfigurace obnovy. Z tohoto důvodu jsem k připojení obrazu disku opět využil skript `windows_automount.bat`. Po úspěšném připojení byl obraz automaticky detekován nástrojem jako pevný disk, a to pod názvem „USB Drive“, jak je patrné z obrázku B.4 . Analýzu disku lze spustit kliknutím na tlačítko *Search for lost data*, případně přímým výběrem disku. Program následně automaticky zahájil proces obnovy, který kombinuje rychlé a hloubkové skenování. Výsledky byly po dokončení zobrazeny jako seznam, jak je patrné z obrázku B.5. V postranním panelu se standardně objevují dvě sekce – „USB Drive“ a „Reconstructed“. Pokud to bylo možné, obnova probíhala primárně ze sekce „USB Drive“. Tato volba je doporučena také vývojáři programu. Sekce „Reconstructed“ totiž slouží jako doplňkový zdroj souborů v případech, kdy se původní data nepodařilo nalézt kompletně nebo je bylo potřeba rekonstruovat. Vybrané soubory bylo poté třeba označit a obnovu dokončit kliknutím na tlačítko *Recover All*.

7.1.6 Disk Drill

K měření nástroje Disk Drill jsem využil licencovanou verzi **PRO**. Připojení virtuálních disků jsem provedl přímo v aplikaci pomocí tlačítka *Attach disk image*, které se nachází ve spodní liště programu (viz obrázek B.2). Po výběru disku jsem zahájil analýzu pomocí volby *Search for lost data*. V nastavení jsem aktivoval hloubkové skenování včetně prostoru mimo volné bloky. Po dokončení analýzy byl zobrazen seznam nalezených souborů, jak ukazuje obrázek B.3. Obnova dat se standardně dělila do částí „Deleted or lost“ a „Reconstructed“. Primárně jsem využil první skupinu, zatímco druhá byla použita pouze v případech, kdy v první skupině nebyly nalezeny žádné soubory.

7.2 Vyhodnocení výsledků

Tato sekce se zaměřuje na vyhodnocení výsledků měření obnovy dat, které bylo provedeno v předchozí sekci 7.1. Cílem je analyzovat data získaná z těchto měření a porovnat možnosti jednotlivých nástrojů. Pro analýzu dat byl využit skript `recovery_validation.py`, jehož implementace byla popsána v sekci 6.3. Výsledky jsou podloženy výstupními daty tohoto skriptu (obrázek C.1). Při vyhodnocení byla z definovaných metrik pro porovnání nástrojů (viz sekce 4.1) použita pouze přesnost obnovy. Rychlost obnovy nebyla zahrnuta do hodnocení z důvodu manuálního měření. Ostatní metriky, jako je uživatelská přívětivost rozhraní, nebyly v tomto srovnání zohledněny, protože nemají přímý vliv na primární cíl analýzy. Následující části se věnují prezentaci dosažených poznatků, přičemž jsou uspořádány podle jednotlivých scénářů. Každý scénář využívá buď variantu pouze se souborovým systémem nebo i s tabulkou oddílů. Celkovou úspěšnost obnovy souborů pro jednotlivé nástroje dále ilustrují grafy v příloze C. Z těchto vizualizací je patrné, že nejvíce univerzálním nástrojem se v rámci testování stal program EaseUS Data Recovery (dále jako EaseUS).

7.2.1 Scénář 1 – Obnova smazaných souborů

Z dosažených výsledků vyplynulo, že na souborových systémech FAT16, FAT32, exFAT a NTFS dosahuje většina testovaných nástrojů téměř vynikající úspěšnosti obnovy. Nejvyšší hodnoty byly zaznamenány u nástrojů EaseUS a Disk Drill, které dokázaly obnovit všechny soubory v nepozměněné podobě. PhotoRec dosáhl mírně nižšího skóre (kolem 95%), především kvůli úplnému selhání obnovy archivního formátu `tar`.

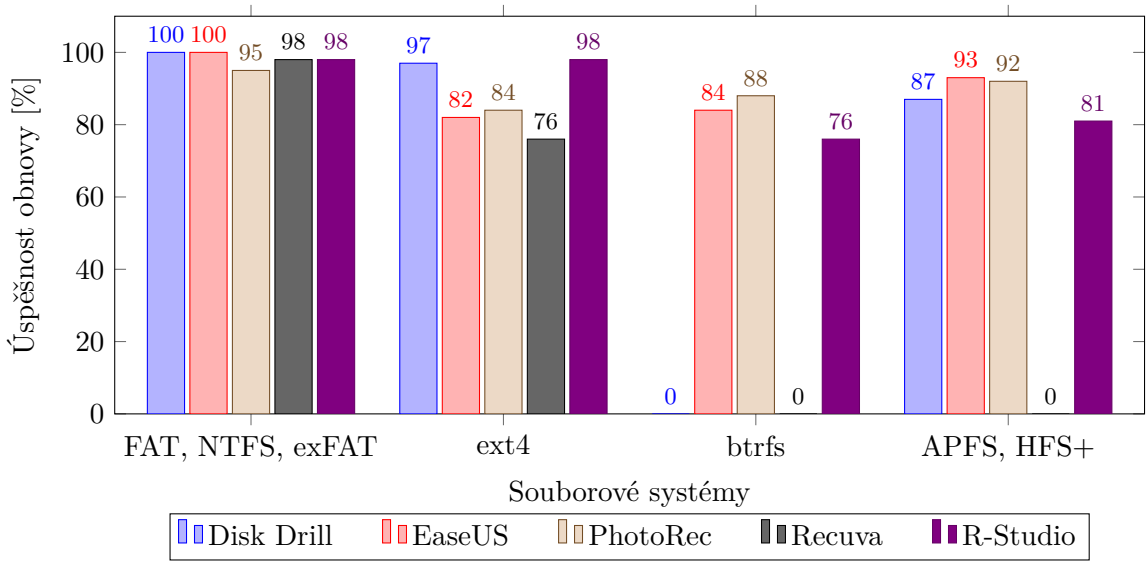
Na linuxových souborových systémech ext2 a ext3 byla celková úspěšnost obnovy u většiny nástrojů výrazně nízká (často pod 10 %). Tento výsledek byl způsoben především vysokým podílem chybějících či nečitelných souborů. Výjimku tvořily programy R-Studio a Disk Drill, které v případě ext3 dosáhly téměř identické obnovy. U ext4 se výsledky znatelně zlepšily. Nejvyšší úspěšnosti (95 %) dosáhly R-Studio a Disk Drill. EaseUS a PhotoRec rovněž poskytly uspokojivé výsledky (82-85 %). Zajímavostí je, že EaseUS dosáhl těchto výsledků i přesto, že ext4 nepodporuje. Recuva byla na tomto souborovém systému nejméně úspěšná. i když ani jeden z testovaných programů dle oficiálních dokumentací nepodporuje btrfs, výsledky ukazují, že EaseUS a PhotoRec dosáhly překvapivých výsledků (okolo 85 %). R-Studio obnovilo data částečně (76 %), avšak výsledky naznačují využití techniky vyřezávání, jelikož bylo nalezeno více než 1400 souborů místo očekávaných 21 souborů. Pokles úspěšnosti byl způsoben především chybějícími a chybně obnovenými formáty videí.

U souborových systémů určených pro macOS (APFS, HFS+) byla úspěšnost obnovy velmi vysoká zejména u EaseUS a PhotoRec (93 %), a to i přes oficiální absenci podpory APFS. R-Studio zde dosáhlo mírně nižší úspěšnosti než předchozí nástroje. Tabulka 7.2 shrnuje optimální nástroje pro dosažení maximální úspěšnosti obnovy dat.

Souborový systém	**Doporučený program**
FAT, NTFS a exFAT	EaseUS, Disk Drill
ext3 a ext4	R-Studio, Disk Drill
btrfs	PhotoRec, EaseUS
APFS a HFS+	EaseUS, Disk Drill

Tabulka 7.2: Doporučené programy pro dosažení nejvyšší úspěšnosti obnovy dat u jednotlivých souborových systémů při scénáři smazání souboru.

Rozšířením je obrázek 7.1, který shrnuje výše zmíněné poznatky.



Obrázek 7.1: Průměrná procentuální úspěšnost obnovy dat pro jednotlivé nástroje v rámci skupin souborových systémů ve scénáři smazaných dat.

7.2.2 Scénář 2 – Obnova po přepsání souborového systému

Druhý scénář představuje výrazně náročnější situaci. Celková úspěšnost obnovy byla obecně nižší než ve scénáři 1. Nejlepších výsledků dosáhly programy EaseUS, PhotoRec a Disk Drill. Ty dokázaly obnovit značnou část dat, zejména při naformátování na souborové systémy FAT32 a exFAT. V těchto případech dosahovala průměrná úspěšnost až 90 %. Program R-Studio vykazovalo úspěšnost v rozmezí 70-80 %, zatímco nástroj Recuva byl schopný obnovit data pouze po formátování na souborový systém FAT32 nebo exFAT.

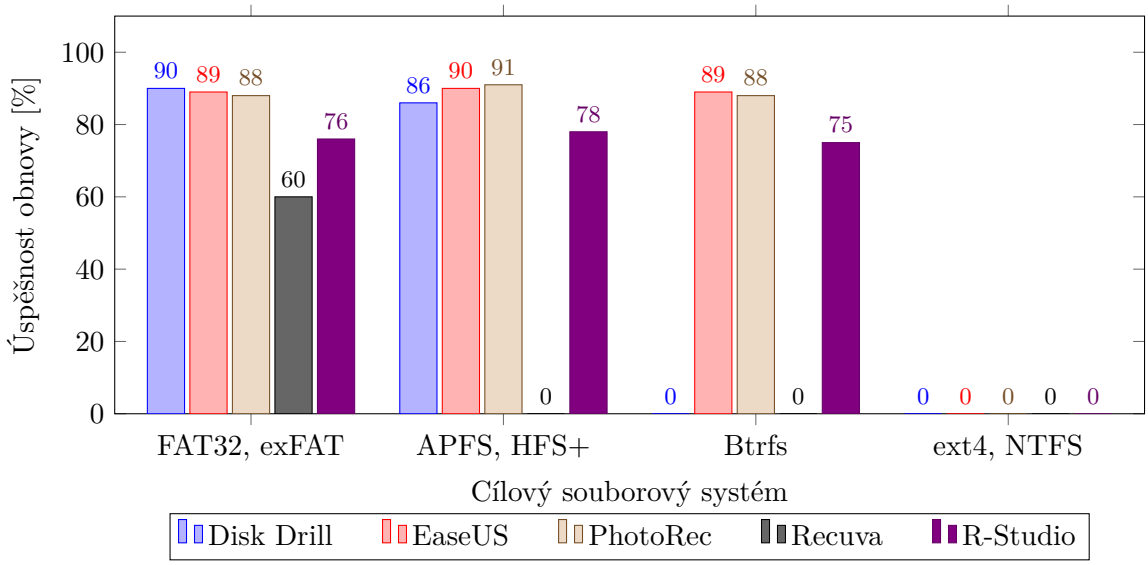
Na souborových systémech určených pro Linux a macOS byly výsledky podobné. Programy EaseUS, PhotoRec a Disk Drill si zachovaly relativně vysokou úspěšnost pohybující se v rozmezí 80-90 %. R-Studio dosahovalo nižších hodnot, přesto se stále pohybovalo v rozmezí 70–80 %. Program Recuva zde zcela selhal, neboť nedokázal obnovit žádná data. Vzájemné přechody mezi systémy APFS a HFS+ byly zvládnuty některými nástroji úspěšně, s průměrnou úspěšností kolem 90 %. I zde byl program Recuva výjimkou, neboť tyto souborové systémy nebyly mezi podporovanými.

U diskových obrazů, kde byl původní souborový systém nahrazen systémem Btrfs, dosáhly programy EaseUS a PhotoRec úspěšnosti mezi 88-92 %. R-Studio zde zaznamenalo průměrně 75 %, zatímco Disk Drill a Recuva nebyly schopny data obnovit vůbec. Zcela zásadním zjištěním v tomto scénáři bylo, že žádný z testovaných nástrojů nedokázal obnovit data po přepsání disku souborovými systémy ext4 nebo NTFS. Bez ohledu na původní souborový systém, typ tabulky oddílů či podporu nástroje byla úspěšnost obnovy v těchto případech nulová. Je však možné, že tento výsledek byl ovlivněn způsobem formátování, konkrétně použitím nástroje `mkfs`. Na základě dalšího vyhodnocení bylo zjištěno, že typ tabulky oddílů (MBR nebo GPT) neměl zásadní vliv na úspěšnost obnovy dat. Jejich rozdíly byly minimální a výsledky často proměnlivé, což neumožnilo jednoznačně určit, který z typů je v tomto ohledu vhodnější. Shrnutí výše zmíněných poznatků je obsaženo v tabulce 7.3.

Nový souborový systém	Programy s nejlepšími výsledky
FAT32, exFAT	Disk Drill, EaseUS
APFS, HFS+	PhotoRec, EaseUS
Btrfs	EaseUS, PhotoRec
ext4, NTFS	Žádný nástroj

Tabulka 7.3: Přehled programů s nejvýše dosaženými výsledky obnovy dat ve scénáři po formátování souborového systému.

Porovnání hodnot úspěšnosti obnovy z nově naformátovaných souborových systémů ukazuje obrázek 7.2.



Obrázek 7.2: Průměrná procentuální úspěšnost obnovy dat pro jednotlivé nástroje ve scénáři po formátování souborovým systémem.

Kapitola 8

Závěr

Cílem této práce bylo provést analýzu a srovnání vybraných nástrojů určených pro obnovu dat z pevných disků. Práce se zaměřila na zhodnocení jejich možností v různých scénářích ztráty dat. Počáteční fáze poskytla teoretický základ pro relevantní souborové systémy a používané tabulky oddílů. Součástí této části byl i popis technik obnovy dat. Následně byly vybrány a popsány konkrétní nástroje pro testování, zahrnující jak volně dostupné, tak komerční varianty dostupné pro různé operační systémy.

V rámci další fáze byl proveden průzkum existujících datových sad. Ten ukázal potřebu komplexnějšího řešení. Hlavním přínosem této práce se proto stalo vytvoření komplexní datové sady pro měření programů. Tato sada je inovativní především v možnosti její automatizované generace pomocí skriptů. Simuluje dva scénáře ztráty dat napříč různými souborovými systémy a tabulkami oddílů. Sada byla navržena tak, aby obsahovala širokou škálu multimediálních dat a umožňovala testování různých kombinací. Pro účely umožnění testování některých programů byly vyvinuty pomocné skripty pro automatizované připojování obrazů disků v operačních systémech Windows a Linux. Zásadní součástí řešení byl také vývoj skriptu pro automatizované vyhodnocení úspěšnosti obnovy, který porovnává obnovená data s původními soubory na základě jejich obsahu a integrity.

Praktické testy provedené s využitím datové sady odhalily značné rozdíly ve schopnostech obnovy nástrojů mezi testovanými. Na základě dosažených výsledků se nástroj EaseUS Data Recovery Wizard projevil jako nejvíce univerzální nástroj pro obnovu dat z většiny testovaných souborových systémů. Naopak nástroj Recuva se ukázal jako velmi omezený a vhodný spíše na souborové systémy pro Windows. Program PhotoRec nedokázal v žádném z testovaných scénářů obnovit formát souboru `tar`, i přesto byl svou celkovou úspěšností srovnatelný s programem R-Studio. Dále se podařilo ukázat, že typ tabulky oddílů neměl v provedených testech zásadní vliv na celkovou úspěšnost obnovy dat. Vytvořené skripty a datová sada mohou sloužit jako základ pro navazující projekty zabývající se testováním a vývojem nástrojů pro obnovu dat. Do budoucna by bylo přínosné rozšířit testovací datovou sadu o další scénáře či souborové systémy. Zároveň by se mohlo otestovat širší spektrum programů, a to i těch méně známých, což by umožnilo rozšířit rozsah srovnání. V kontextu vytvořených skriptů by další vývoj mohl přidat pokročilejší metody pro vyhodnocování úspěšnosti obnovy.

Literatura

[1] Agar, R. *Top 10 Best Data Recovery Software in 2024* online. 2024. Dostupné z: https://www.handyrecovery.com/best-data-recovery-apps.html. [cit. 2024-01-12].

[2] Amissah, F. *Open Source vs. Commercial Software License: What Do You Need?* online. 21. září 2023. Dostupné z: https://www.turing.com/blog/open-source-vs-commercial-software-license/. [cit. 2023-12-25].

[3] Apple. *Uživatelská příručka pro Diskovou utilitu* online. 2023. Dostupné z: https://support.apple.com/cs-cz/guide/disk-utility/welcome/mac. [cit. 2023-12-3].

[4] ArchWiki. *File systems* online. 25. července 2012. revidováno 8.11.2023. Dostupné z: https://wiki.archlinux.org/title/file_systems. [cit. 2023-12-1].

[5] Brouwer, A. *The FAT filesystem* online. 20. září 2002. Dostupné z: https://www.win.tue.nl/~aeb/linux/fs/fat/fat-1.html. [cit. 2023-11-25].

[6] Bundele, A. Comparative study of File systems (NTFS, FAT, FAT32, EXT2, EXT3, EXT4). *IJRASET* online, Srpen 2018, sv. 6, č. 8, s. 183–186. ISSN 2321-9653. Dostupné z: https://www.ijraset.com/fileserve.php?FID=18507. [cit. 2023-11-29].

[7] Burghardt, A. a Feldman, A. J. Using the HFS+ journal for deleted file recovery. *Digital Investigation* online. Elsevier, 2008, sv. 5, s. 76–82. ISSN 1742-2876. Dostupné z: https://doi.org/10.1016/j.diin.2008.05.013. [cit. 2025-04-02].

[8] Carrier, B. *File system forensic analysis*. 2. vyd. Addison-Wesley Professional, 2005. ISBN 0-321-26817-2.

[9] Casey, E.; Nelson, A. a Hyde, J. Standardization of file recovery classification and authentication. *Digital Investigation* online, 2019, sv. 31. ISSN 1742-2876. Dostupné z: https://doi.org/10.1016/j.diin.2019.06.004. [cit. 2025-04-05].

[10] CGSecurity. *PhotoRec* online. 21. května 2023. Dostupné z: https://www.cgsecurity.org/wiki/PhotoRec. [cit. 2023-12-1].

[11] Cleverfiles. *Disk Drill Data Recovery Software* online. 2023. Dostupné z: https://www.cleverfiles.com/data-recovery-software.html. [cit. 2023-12-2].

[12] Craiger, P. a Burke, P. K. *Mac Forensics: Mac OS X and the HFS+ File System* online. 2005. Dostupné z: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=6498824bf271858bcd2a8fc2fbb0da1de7f77367. [cit. 2023-11-30].

[13] EaseUS. *EaseUS Data Recovery* online. 2023. Dostupné z: https://www.easeus.com/datarecoverywizard/free-data-recovery-software.htm. [cit. 2023-12-2].

[14] Fairbanks, K. D. An analysis of Ext4 for digital forensics. *Digital Investigation* online, 2012, sv. 9, s. 118–130. ISSN 1742-2876. Dostupné z: https://doi.org/10.1016/j.diin.2012.05.010. [cit. 2025-03-02].

[15] Ghadi, M.; Laouamer, L. a Moulahi, T. Enhancing digital image integrity by exploiting JPEG bitstream attributes. *Journal of Innovation in Digital Ecosystems* online, Listopad 2015, sv. 2, č. 1, s. 20–31. ISSN 2352-6645. Dostupné z: https://doi.org/10.1016/j.jides.2015.10.003. [cit. 2025-03-16].

[16] Goda, K. a Kitsuregawa, M. The History of Storage Systems. *Proceedings of the IEEE* online. IEEE, Březen 2012, sv. 100, Special Centennial Issue, s. 1433–1440. Dostupné z: https://doi.org/10.1109/JPROC.2012.2189787. [cit. 2023-12-25].

[17] Golden, R. *Scalpel-1.60* online. 13. ledna 2024. Dostupné z: https://github.com/nolaforensix/scalpel-1.60. [cit. 2025-02-13].

[18] Hamm, J. *Extended FAT File System* online. 2009. Dostupné z: https://paradigmsolutions.files.wordpress.com/2009/12/exfat-excerpt-1-4.pdf. [cit. 2023-11-28].

[19] Hansen, K. H. a Toolan, F. Decoding the APFS file system. *Digital Investigation* online. Elsevier, Září 2017, sv. 22, č. 1, s. 107–132. ISSN 1742-2876. Dostupné z: https://doi.org/10.1016/j.diin.2017.07.003. [cit. 2023-12-01].

[20] Kára, J. *Ext4, btrfs, and the others* online. 2009. Dostupné z: https://picture.iczhiku.com/resource/paper/WhkeiZuyatSryccm.pdf. [cit. 2023-11-29].

[21] Lee, S.-Y. a Shon, T. Improved deleted file recovery technique for Ext2/3 filesystem. *The Journal of Supercomputing* online, 2014, sv. 70, č. 1, s. 20–30. Dostupné z: https://doi.org/10.1007/s11227-014-1282-y. [cit. 2025-04-05].

[22] Li, Z.-N.; Drew, M. S. a Liu, J. *Fundamentals of Multimedia* online. 2. vyd. Springer, 2014. ISBN 978-3-319-05289-2. Dostupné z: https://drive.uqu.edu.sa/_/mskhayat/files/MySubjects/20178FS%20Multimedia%20Systems/Fundamentals_of_multimedia_2e.pdf. [cit. 2025-03-15].

[23] Lyle, J. R. A Strategy for Testing Metadata Based Deleted File Recovery Tools. In: *Digital Forensics and Cyber Crime* online. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, s. 104–114. ISBN 978-3-642-35515-8. Dostupné z: https://doi.org/10.1007/978-3-642-35515-8_9. [cit. 2024-01-13].

[24] Microsoft. *How NTFS Works* online. 10. srpna 2009. Dostupné z: https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc781134(v=ws.10)?redirectedfrom=MSDN. [cit. 2023-11-25].

[25] Miguel, P. G. *21 Best File Recovery Software of 2024 Unveiled* online. 2024. Dostupné z: https://thectoclub.com/tools/best-file-recovery-software/. [cit. 2024-01-12].

[26] Na, G.-H.; Shim, K.-S.; Moon, K.-W.; Kong, S. G.; Kim, E.-S. et al. Frame-Based Recovery of Corrupted Video Files Using Video Codec Specifications. *IEEE Transactions on Image Processing* online, 2014, sv. 23, č. 2, s. 517–526. Dostupné z: https://doi.org/10.1109/TIP.2013.2285625. [cit. 2024-01-13].

[27] Nemeth, E.; Snyder, G.; Hein, T. R.; Whaley, B. a Mackin, D. *UNIX and Linux System Administration Handbook*. 5. vyd. Boston: Addison-Wesley, 2018. ISBN 978-0-13-427755-4.

[28] Ni, F.; Wu, X.; Li, W.; Wang, L. a Jiang, S. WOJ: Enabling Write-Once Full-data Journaling in SSDs by using weak-hashing-based deduplication. *Performance Evaluation* online, 2018, 127–128, s. 56–69. ISSN 0166-5316. Dostupné z: https://doi.org/10.1016/j.peva.2018.09.004. [cit. 2024-01-12].

[29] Nikkel, B. J. Forensic analysis of GPT disks and GUID partition tables. *Digital Investigation* online. Elsevier, Září 2009, sv. 6, 1–2, s. 39–47. ISSN 1742-2876. Dostupné z: https://doi.org/10.1016/j.diin.2009.07.001. [cit. 2023-11-03].

[30] Park, B.; Savoldi, A.; Gubian, P.; Park, J.; Lee, S. et al. Recovery of Damaged Compressed Files for Digital Forensic Purposes. In: *2008 International Conference on Multimedia and Ubiquitous Engineering (mue 2008)* online. Květen 2008, s. 365–372. ISBN 978-0-7695-3134-2. Dostupné z: https://doi.org/10.1109/MUE.2008.49. [cit. 2025-03-16].

[31] Povar, D. Forensic Data Carving. In: *Digital Forensics and Cyber Crime* online. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, s. 137–148. ISBN 978-3-642-19513-6. Dostupné z: https://link.springer.com/chapter/10.1007/978-3-642-19513-6_12. [cit. 2023-12-27].

[32] Rodeh, O.; Bacik, J. a Mason, C. BTRFS: The linux B-tree filesystem. *ACM Transactions on Storage* online, Srpen 2013, sv. 9, č. 3, s. 1–32. ISSN 1553-3077. Dostupné z: https://doi.org/10.1145/2501620.2501623. [cit. 2023-11-29].

[33] Shibli, A. A. A Review of JPEG File Carving: Challenges, Techniques, and Future Directions. *Applied Computing Journal* online, Březen 2025, sv. 5, č. 1, s. 372–385. Dostupné z: https://doi.org/10.52098/acj.20255124. [cit. 2025-03-16].

[34] Stellar. *What is the HFS File System?* online. 1. února 2023. Dostupné z: https://www.stellarinfo.co.in/ebook/complete-guide-on-hfs-file-system/features-and-drawbacks. [cit. 2023-11-30].

[35] Suthar, H. a Sharma, P. An Investigation on File Carving Tool Methodologies Using Scenario Based Image Creation. *Indian Journal of Science and Technology* online, 2024, sv. 17, č. 3, s. 215–227. Dostupné z: https://doi.org/10.17485/IJST/v17i3.808. [cit. 2025-03-04].

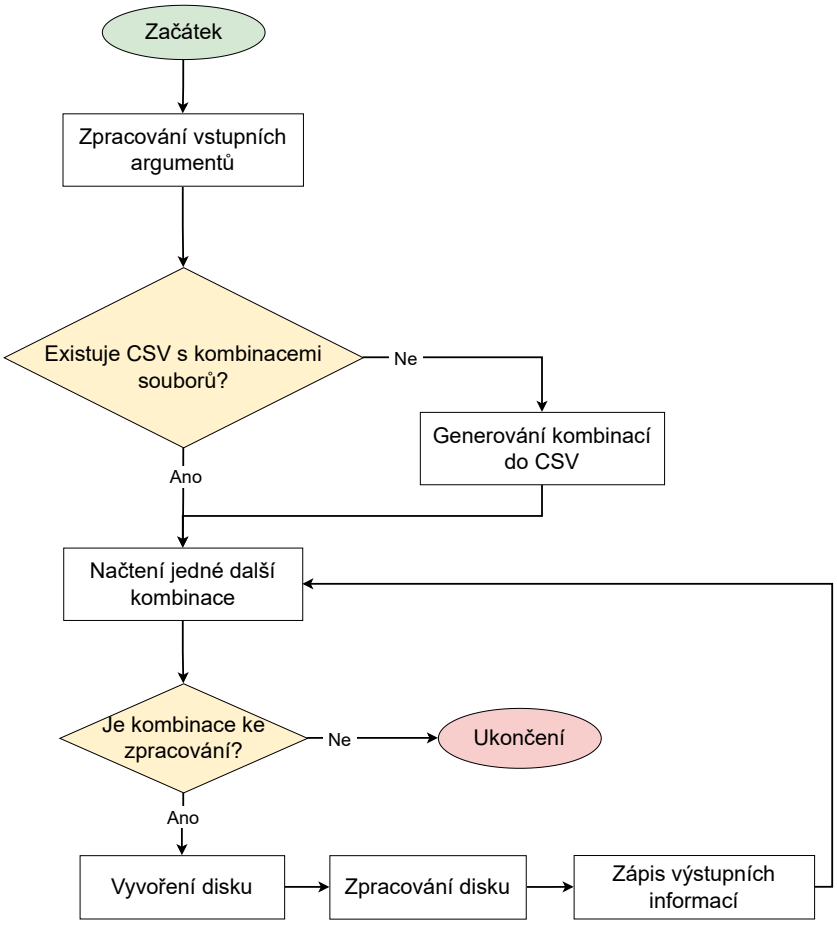
[36] Technology, R.-T. *R-STUDIO* online. 2023. Dostupné z: https://www.r-studio.com/. [cit. 2023-12-2].

[37] Try, A. *Recuva Review* online. 17. listopadu 2023. Dostupné z: https://datarescuetools.com/reviews/recuva/. [cit. 2023-12-2].

[38] Zhang, N.; Jiang, Y. a Wang, J. The Research of Data Recovery on Windows File Systems. In: *2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)* online. 2020, s. 644–647. ISBN 978-1-7281-6698-8. Dostupné z: https://doi.org/10.1109/ICITBS49701.2020.00141. [cit. 2025-03-05].

Příloha A

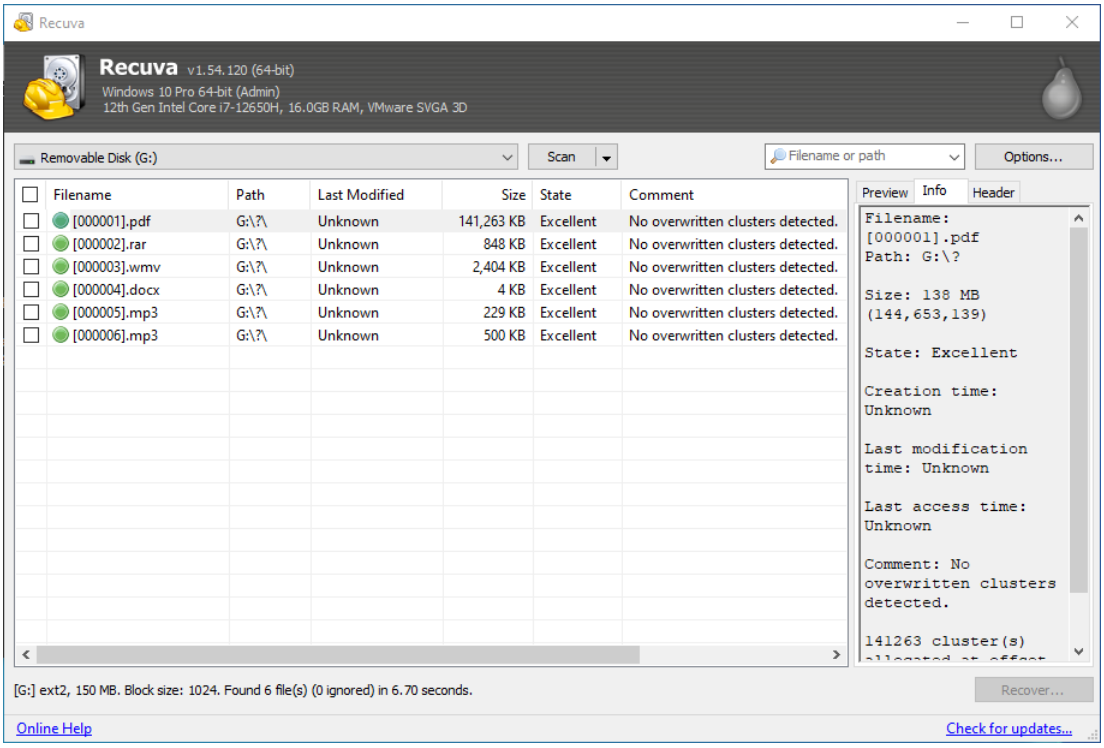
Vývojový diagram



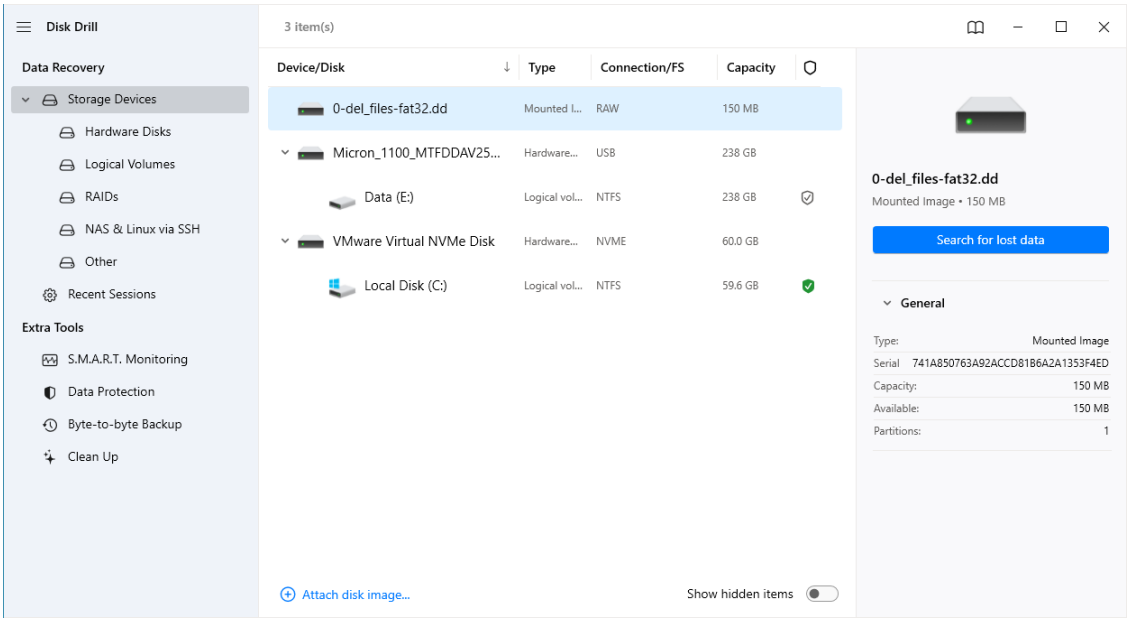
Obrázek A.1: Vývojový diagram pro automatizované generování datové sady.

Příloha B

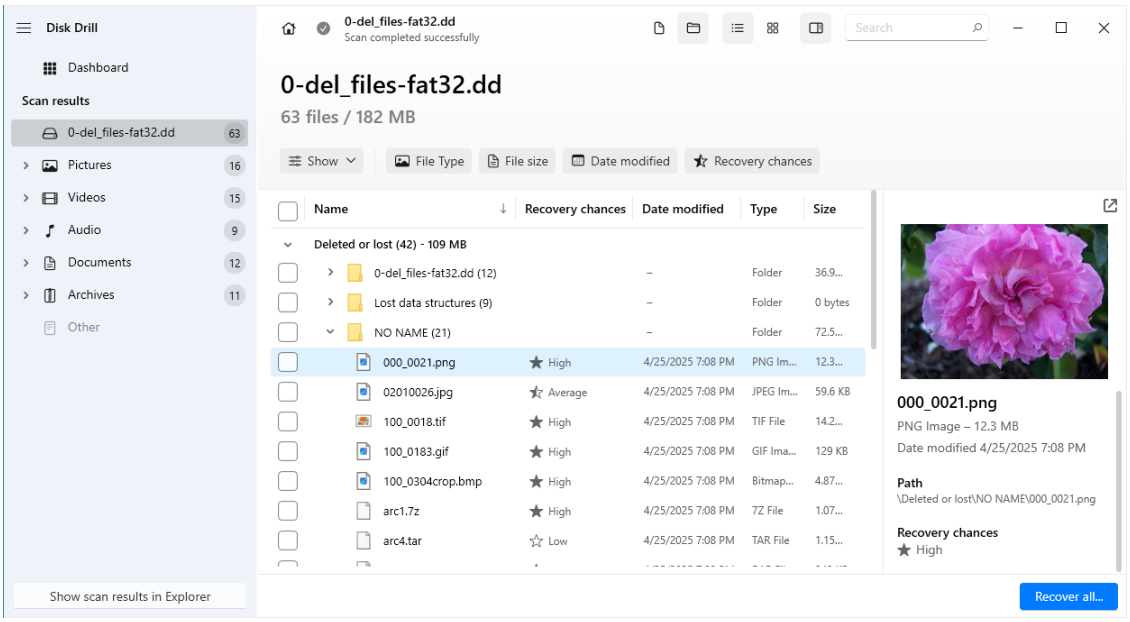
Náhledy forenzních nástrojů použitých při měření



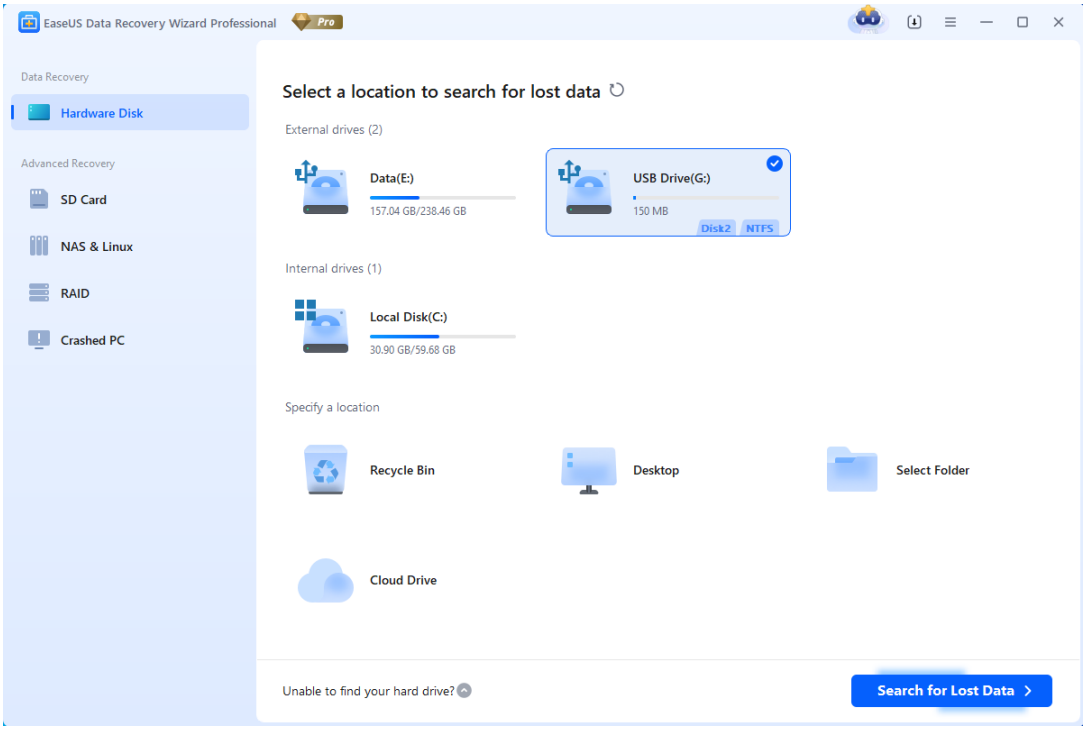
Obrázek B.1: Rozhraní programu Recuva zobrazující výsledky skenování disku.



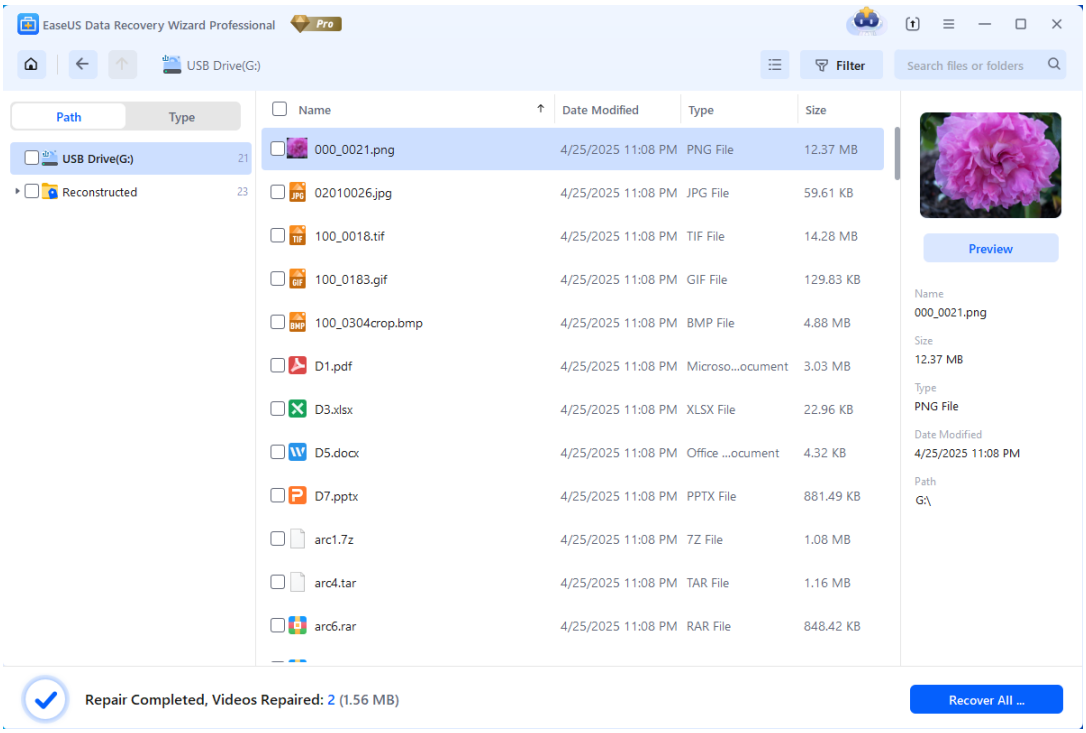
Obrázek B.2: Rozhraní programu Disk Drill při výběru zařízení pro obnovu dat.



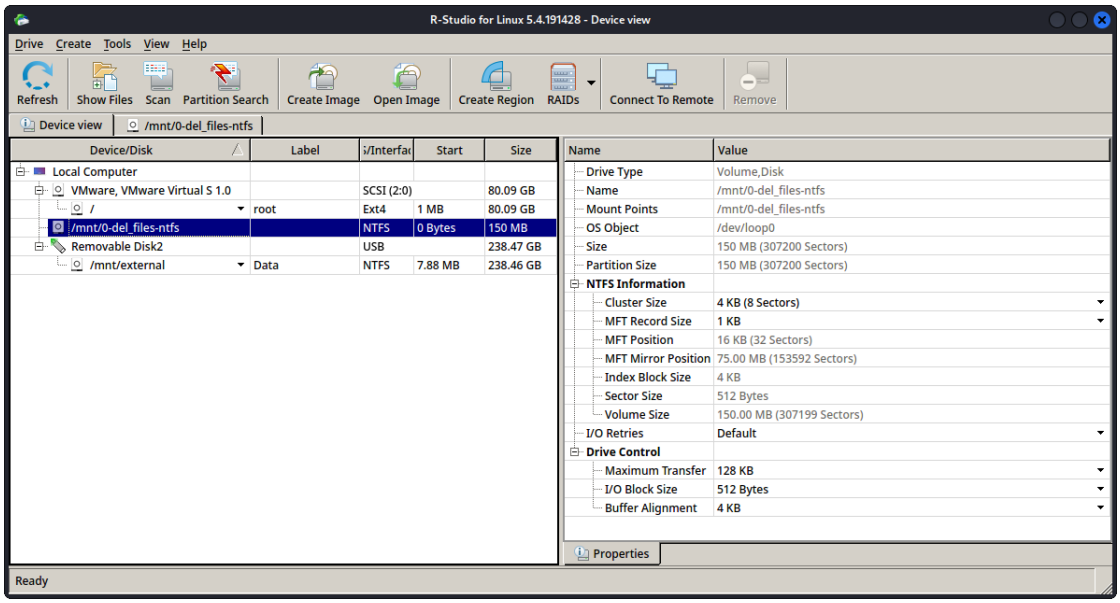
Obrázek B.3: Rozhraní programu Disk Drill zobrazující výsledky skenování disku.



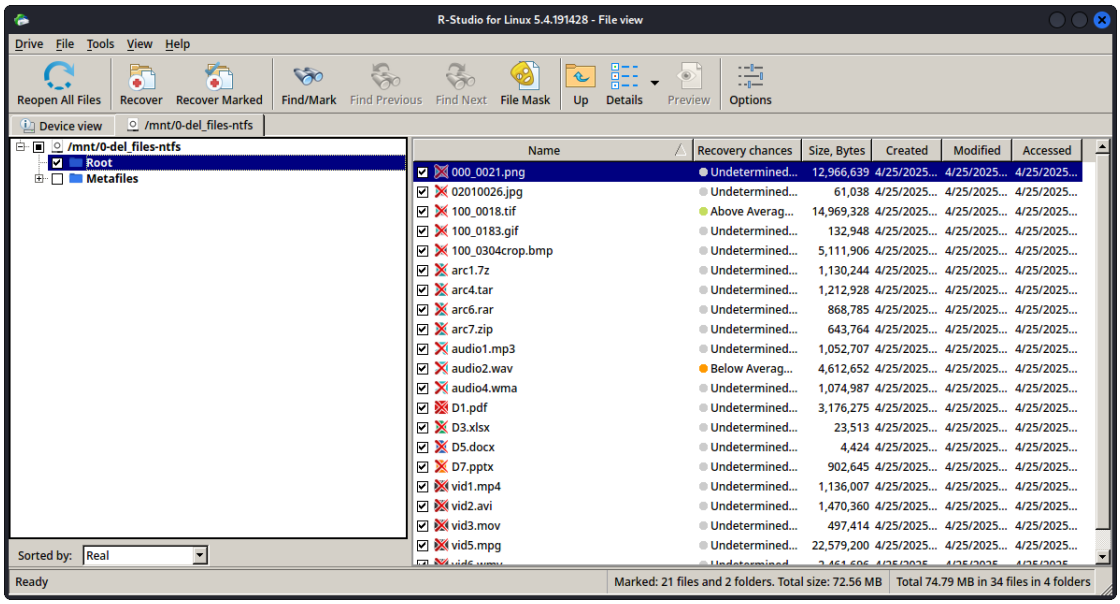
Obrázek B.4: Rozhraní programu EaseUS Data Recovery Wizard při výběru zařízení pro obnovu dat.



Obrázek B.5: Rozhraní programu EaseUS Data Recovery Wizard zobrazující výsledky skenování disku.



Obrázek B.6: Rozhraní programu Disk Drill při výběru zařízení pro obnovu dat.



Obrázek B.7: Rozhraní programu R-Studio zobrazující výsledky skenování disku.

Příloha C

Přehled úspěšnosti obnovení dat

Program: **Disk Drill**

disk	expected	recovered	summary	mov	mp3	mp4	mpg	pdf	png	pptx	rar	tar	tif	wav	wma
0-del_files-exfat	21	23	100	I	I	I	I	I	I	I	I	I	I	I	I
0-del_files-fat16	21	21	100	I	I	I	I	I	I	I	I	I	I	I	I
0-del_files-fat32	21	42	100	I	I	I	I	I	I	I	I	I	I	I	I
0-del_files-ntfs	21	21	100	I	I	I	I	I	I	I	I	I	I	I	I
0-del_files-ext3	21	21	99,71	I	I	I	I	I	I	I	I	I	I	94	I
0-del_files-ext4	21	21	94,91	I	I	I	93	I	I	I	I	I	I	0	I
0-del_files-apfs	21	25	90,44	I	100	I	I	I	I	I	I	M	M	I	99
0-del_files-hfsplus	21	24	85,68	I	100	I	I	I	I	I	I	M	M	I	99
0-del_files-ext2	21	26	8,38	C	M	C	26	M	M	M	2,2	M	M	3,1	0,3

Program: **EaseUS Data Recovery**

disk	expected	recovered	summary	mov	mp3	mp4	mpg	pdf	png	pptx	rar	tar	tif	wav	wma
0-del_files-exfat	21	42	100	I	I	I	I	I	I	I	I	I	I	I	I
0-del_files-fat16	21	42	100	I	I	I	I	I	I	I	I	I	I	I	I
0-del_files-fat32	21	42	100	I	I	I	I	I	I	I	I	I	I	I	I
0-del_files-ntfs	21	21	100	I	I	I	I	I	I	I	I	I	I	I	I
0-del_files-apfs	21	23	95,23	I	100	I	I	I	I	I	I	M	I	I	I
0-del_files-hfsplus	21	22	90,47	I	100	I	I	I	I	I	I	M	I	I	I
0-del_files-ext4	21	23	82,39	I	100	I	9,7	I	I	I	I	M	C	21	I
0-del_files-btrfs	21	21	71,41	I	I	I	I	C	I	C	I	I	I	100	I
0-del_files-ext2	21	29	9,62	C	49	C	0,1	M	C	M	2,2	M	C	3,1	0,3
0-del_files-ext3	21	30	5	C	49	C	0,1	M	C	M	2,2	M	C	6,8	0,7

Program: **PhotoRec**

disk	expected	recovered	summary	mov	mp3	mp4	mpg	pdf	png	pptx	rar	tar	tif	wav	wma
0-del_files-exfat	21	23	95,21	I	100	I	I	I	I	I	I	M	I	I	99
0-del_files-fat16	21	23	95,21	I	100	I	I	I	I	I	I	M	I	I	99
0-del_files-fat32	21	23	95,21	I	100	I	I	I	I	I	I	M	I	I	99
0-del_files-ntfs	21	20	90,44	I	100	I	I	I	I	I	I	M	M	I	99
0-del_files-btrfs	21	23	88,08	I	100	I	50	I	I	I	I	M	I	M	99
0-del_files-ext4	21	23	84,79	I	100	I	61	I	I	I	I	M	M	21	99
0-del_files-ext2	21	46	8,3	M	49	M	26	M	M	M	M	M	M	M	M
0-del_files-ext3	21	47	8,3	M	49	M	26	M	M	M	M	M	M	M	M

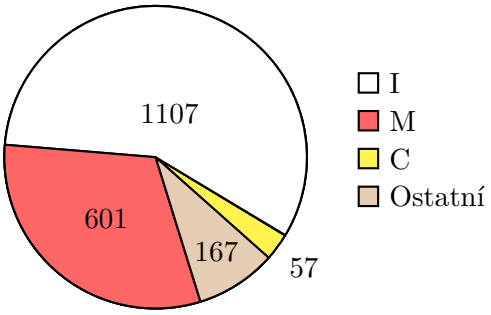
Obrázek C.1: Souhrn výsledků úspěšnosti obnovených souborů, který je výstupem skriptu `recovery_validation.py`. Zobrazuje procentuální vyhodnocení pro jednotlivé soubory i celkovou míru obnovy.

Příloha D

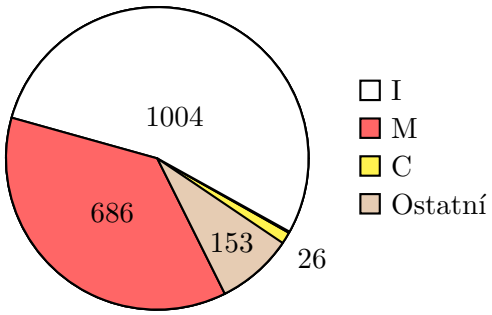
Vizualizace celkové úspěšnosti obnovy souborů pomocí programů

Legenda pro obrázky

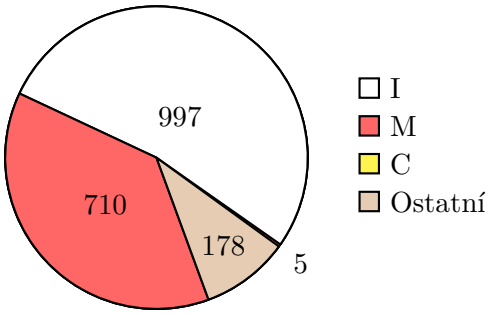
- **Identické soubory (I)**: Soubory, které byly obnoveny v plně identické podobě vůči referenčním datům.
- **Chybějící soubory (M)**: Soubory, které se nepodařilo obnovit.
- **Poškozené soubory (C)**: Soubory, které byly obnoveny, ale následně je nebylo možné otevřít ani zpracovat.
- **Rekonstruované soubory (Ostatní)**: Soubory, které byly poškozeny a program je částečně rekonstruoval do čitelné podoby.



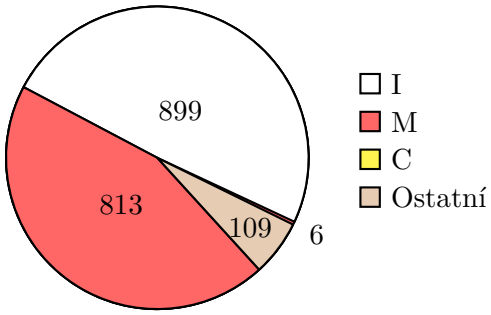
Obrázek D.1: Výsledky obnovy dat pomocí programu **EaseUS Data Recovery Wizard**.



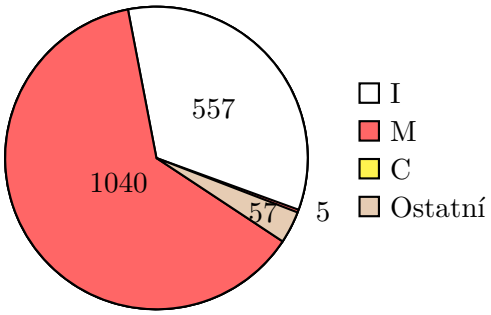
Obrázek D.2: Výsledky obnovy dat pomocí programu **R-Studio**.



Obrázek D.3: Výsledky obnovy dat pomocí programu **PhotoRec**.



Obrázek D.4: Výsledky obnovy dat pomocí programu **Disk Drill**.



Obrázek D.5: Výsledky obnovy dat pomocí programu **Recuva**.