ŠOLSKI CENTER KRANJ
VIŠJA STROKOVNA ŠOLA
INFORMATIKA

DIPLOMSKO DELO

Kranj, avgust 2017 **Andrej Ficko**

ŠOLSKI CENTER KRANJ
VIŠJA STROKOVNA ŠOLA
INFORMATIKA

DIPLOMSKO DELO

Kranj, avgust 2017 **Andrej Ficko**

ŠOLSKI CENTER KRANJ
VIŠJA STROKOVNA ŠOLA
INFORMATIKA

Diplomsko delo višjega strokovnega izobraževanja

REŠEVANJE PODATKOV, KAJ LAHKO STORIMO

Avtor: Andrej Ficko
Ime podjetja: UMBIT, računalniške storitve, d. o. o.
Mentor v podjetju: Boštjan Štrukelj
Mentorica v šoli: Gabrijela Krajnc, univ. dipl. inž.
Lektorica: Tjaša Skaza

Kranj, avgust 2017

ZAHVALA

Zelo mi je pomagal mentor v podjetju s svojimi dolgoletnimi izkušnjami, ki jih je pridobil v karieri računalniškega vzdrževalca in pri podpori uporabnikov. Brez njega bi delu manjkali uporabni nasveti, predstava povprečnega uporabnika pa bi bila zgrešena, saj veliko uporabnikov ne zna odgovoriti na preprosta računalniška vprašanja v anketi.

Nekaj čokolade so si prislužili delavci oddelka za reševanje podatkov podjetja Anni (prevzeto podjetje Kotar) za prikaz strokovnega pristopa k reševanju. Predvsem Dejanu Zvonarju, Urošu Lutarju in Vitu Mlinarcu bi se zahvalil za reševanje osebnega diska s fizično poškodbo. Sodeloval je tudi prijatelj Andrej Glavan. Posodil je vezje svojega diska, ki je bilo na veliko srečo kompatibilno.

Pohvala tudi mentorici na šoli za vso pozitivno energijo, podporo pri izbiri teme, svetovanje o ustreznosti vsebine in pomoč pri obliki. Veliko dela z obliko je imela tudi Kristina Jeram iz šolskega referata s prilagajanjem pri tehnično oblikovnem pregledu, zato se tudi njej zahvaljujem za potrpežljivost.

Brez družine in finančne podpore med študijem mi ne bi uspelo, zato se moram tudi njim zahvalit. Omeniti je tudi treba nekaj odsluženih diskov za raziskavo, ki so jih darovali drugi sorodniki.

POVZETEK

Z vidika, da je lažje preprečiti kot zdraviti, sem bralca najprej seznanil s posledicami izgube podatkov in njihovimi preventivami. Opredeljene so bile pogostejše oblike shranjevanja podatkov v računalnikih za razumevanje omejitev in postopkov pri reševanju. Nato sem bralca vodil skozi prepoznavanje primerov in postopkov reševanja podatkov. Opisani so bili tudi postopki reševanj, ki jih lahko varno izvajamo sami. Na koncu sem razložil delovanje mehanskega hranjenja, njegove okvare in zahtevnosti strokovnega popravila. Vključenih je tudi nekaj teorij in razlogov za nerešljive podatke v praksi.

Ključne besede: reševanje podatkov, slabi sektorji, samostojno izvajanje, hranjenje podatkov, datotečni sistem, trdi disk, SSD-disk, preprečevanje izgube podatkov

ABSTRACT

From the point of view that it is better to be safe than sorry, initially I informed the reader with the consequences of data loss and its prevention. More common forms of computer data storing were defined, in order to understand their limitations as well as rescue procedures. Then I guided the reader through recognizing the cases in addition to the procedures of data rescuing. Furthermore rescue procedures, that we can safely perform ourselves were also described. In the end I explained the mechanical storing operations, their malfunction and consequently the difficulty of expert repair. Additionally, some theories and reasons for rescue failure in practice have also been included.

Key words: data rescue, bad sectors, independent performing, storing data, file system, hard drive, SSD drive, data-loss prevention

Kazalo vsebine

Kazalo slik

UPORABLJENE KRATICE IN SIMBOLI

Kratica/simbol	Razlaga pomena kratice/simbola
B	Bajt
CRC	Ciklično redundantno preverjanje (angl. *Cyclic redundancy check*)
ECC	Koda za popravljanje napak (angl. *Error-correcting code*)
EXT	Razširjen datotečni sistem (angl. *Extended file system*)
FAT	Tabela razporeditve datotek (angl. *File Allocation Table*)
GiB	Gibibajt
GPT	Razpredelnica razdelkov z edinstvenim identifikatorjem (angl. *GUID Partition Table*)
HDD	Trdi disk (angl. *Hard disk drive*)
JPEG	Slikovni format združenih strokovnjakov fotografije (angl. *Joint Photographic Experts Group*)
KiB	Kibibajt
MBR	Glavni zagonski zapis (angl. *Master boot record*)
MiB	Mebibajt
NAND	Logični izraz negiranega IN (angl. *Negative-AND*)
NTFS	Moderen datotečni sistem (angl. *New Technology File System*)
RAID	Vezava neodvisnih diskov (angl. *Redundant array of independent disks*)
SMART	Tehnologija za odkrivanje, analizo in poročanje napak (angl. *Self-Monitoring, Analysis and Reporting Technology*)
SSD	Disk brez premikajočih se delov (angl. *Solid-state drive*)

1 UVOD

1.1 NAMEN DIPLOMSKEGA DELA

Reševanje podatkov je ena dražjih računalniških storitev, saj sta za mehansko popravilo potrebna draga namenska oprema in zelo previdno razstavljanje. Veliko uporabnikov pa ravna napačno, kljub ozaveščenosti o reševanju podatkov se ti premalo zavedajo pomembnosti »pustiti pri miru, zato se zgodi, da podatke nevede prepišejo.

Čeprav si mnogo podjetij prizadeva za varnostno kopiranje vsega, kar se bojijo izgubiti, se zgodi, da kopije zatajijo, ali pa so izgubljene pomembne pretekle spremembe. Tudi v takšnih primerih nastopi reševanje podatkov, od kompleksnosti okvare pa je odvisen strošek izvedbe.

1.2 CILJI DIPLOMSKEGA DELA

Naloga je namenjena vzdrževalcem in serviserjem računalniške tehnologije, ki še niso pridobili ustreznega znanja o reševanju podatkov, a se občasno srečajo s prenosom podatkov in popravilom računalnika z lažjo okvaro. Pa tudi vsem tistim, ki imajo veliko stika z uporabniki in zato lahko pripomorejo k ozaveščanju uporabnikov, sami pa lahko uspešneje rešijo podatke. Če disk še deluje in dovolj zgodaj zaznamo težave, lahko podatke rešimo sami, čeprav imajo strokovnjaki boljše možnosti zaradi drage opreme.

Pri zagotavljanju podpore uporabnikom lahko večkrat naletimo tudi na primer logične izgube podatkov, kar pomeni, da je medij nepoškodovan. S pravilnim postopkom in znanjem lahko podatke rešimo približno tako uspešno kot strokovnjaki. Če pa so podatki življenjskega pomena, se raje obrnimo na strokovnjake, predvsem pri mehanskih okvarah, teh se nikakor ne lotevajmo sami. Glede na vrednost svojih podatkov sami presodimo, ali res želimo tvegati.

1.3 METODE DELA

Na podlagi lastnih izkušenj in posvetovanj z osebjem iz podjetja za reševanje podatkov so predstavljeni postopki v primerih, ki se jih lahko lotimo sami. Razloženi so delovanje medijev za shranjevanje podatkov in mogoče okvare ter stopnje zahtevnosti za popravilo in reševanje. Pri postopkih reševanja je, kolikor je mogoče, uporabljena odprta ali brezplačna programska oprema, da se izognemo nepotrebnim stroškom. Priporočena orodja so med seboj primerjana za lažjo izbiro in samostojno delo.

Nabor je omejen na operacijska sistema Linux in Windows. Prvega smo izbrali zaradi neodvisnosti in prilagodljivosti na strojno opremo, drugega pa zaradi razširjenosti programske opreme in njihove preproste uporabe. Omenjenih je nekaj plačljivih programov, saj lahko olajšajo delo in jih je mogoče dobiti v preizkusni različici. Običajno za reševanje potrebujemo navaden računalnik in vsaj dvakrat toliko neodvisnega prostora na brezhibno delujočem disku za hranjenje rešenih podatkov.

1.4 PREDSTAVITEV PODJETJA

Osebni podatki so naše največje bogastvo. Tega se zaveda tudi podjetje, ki je sodelovalo pri nastanku te naloge. Poleg popravil in zagotavljanja podpore uporabnikom so poskrbeli še za večjo varnost podatkov in ozaveščenost uporabnikov. Ponujali so storitve varnostnih kopij, kot je oblačen prostor za posameznike, ter postavitev samostojnega omrežnega sistema za podjetja z več računalniki. Večkrat so se srečali s poškodbami medija, zato so meje rešljivosti dobro znane. V težjih primerih je bilo treba sodelovati s podjetjem s strokovnim pristopom.

2 PREVENTIVE

»Nekaj 10 let nazaj, ko so trdi diski stali nekaj tisoč evrov, je bilo popravilo trdega diska bistvenega pomena. Če se nam danes trdi disk pokvari, ga enostavno zamenjamo z novim. Novi disk bo zaradi napredka večje kapacitete, cenejši ter hitrejši. Prišli smo do dejstva, da je tudi za domačega uporabnika vrednost podatkov večja od vrednosti trdega diska.« (Šinkovec, 2011, str. 8).

Veliko mehanskih okvar diskov je bilo odkritih, ko so uporabniki na popravilo prinesli nedelujoč računalnik. Takrat so ugotovili, da se računalnik noče prižgati zaradi okvare diska, ki pa je bila posledica padca, naleta ali večkratnega izpostavljanja tresljajem. V nekaterih primerih je bilo mogoče opaziti simptome pred odpovedjo diska, vendar se uporabniki nanje niso ozirali, razen če jih niso ovirali pri delu.

Če predpostavimo, da reševanje podatkov z diska glede na vrsto mehanske okvare stane od 500 evrov do dva tisoč evrov, varnostno kopiranje pa največ 150 evrov letno, lahko potrdimo pregovor Bolje preprečiti kot zdraviti. Okvare, ki jih lahko popravimo sami, so ocenjene na od 50 evrov do 200 evrov, odvisno od velikosti podatkov in potrebnega časa zaradi manjših poškodb.

Najboljša zaščita je nedvomno varnostna kopija, nekaj malega pa lahko prispevamo z organiziranjem podatkov. S shranjevanjem dokumentov na ločen razdelek lahko preprečimo razmetavanje podatkov, saj operacijski sistem in programi med delovanjem pišejo začasne datoteke in dnevnike. Iskanje na neaktivnem razdelku je tako lažje in uspešnejše. Nekaj malega lahko prispevamo tudi z defragmentiranjem, saj tako datoteke niso preveč razkosane.

2.1 VARNOSTNE KOPIJE

Varnostne kopije so najboljša zaščita pred izgubo podatkov, ki nas poleg okvare obvaruje tudi pred virusi, krajo računalnika in lastno napako. Za domačo rabo in hrambo podatkov je bolj priljubljena uporaba oblačnih storitev, predvsem z brezplačnimi ponudniki, v poslovni uporabi pa je pogostejša postavitev omrežnega strežnika za podatke, kamor se shranjujejo posnetki celotnih računalnikov. Tako lahko v primeru napak zelo hitro postavijo računalnik nazaj v operativno stanje, predvsem ker vključuje vse uporabnikove programe, nastavitve in dokumente.

Statistično gledano največ prostora zavzame avtorsko zaščitena vsebina, prenesena z medmrežja, veliko manj pa osebne fotografije in dokumenti, zato je smiselno, da zaščitimo vsaj to. Uporabniki brez izkušenj, ki zaščito vzamejo v svoje roke, kupijo zunanji trdi disk, ki pa je še bolj izpostavljen morebitnim poškodbam, poleg tega še pogosto pozabijo osveževati kopije.

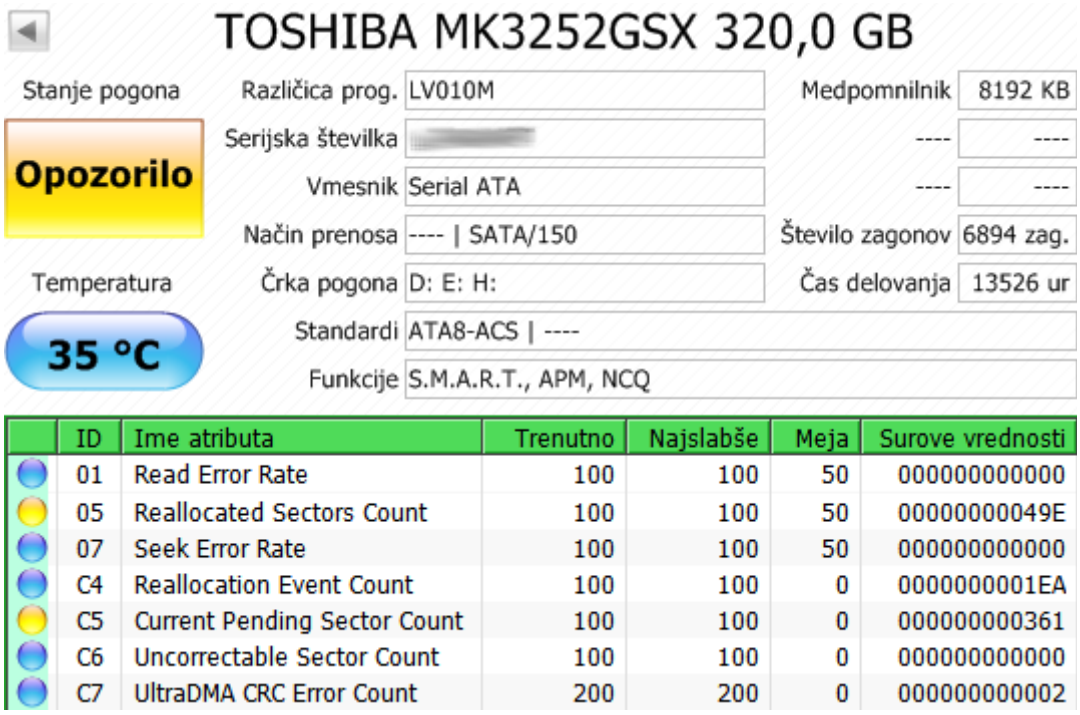
Za podjetje z več računalniki je značilna postavitev sistema NAS, ki omogoča prostor za shranjevanje v notranjem omrežju s protokolom za deljenje podatkov SMB. Poleg varnosti omogoči tudi uporabo tega prostora za odlaganje skupnih podatkov. Najpogosteje smo uporabljali Veeam Endpoint Backup z napravami Synology NAS in diske WD RED v vezavi RAID 1. Skupna cena za 15 računalnikov znaša približno 800 evrov.

Za manjša podjetja je priporočen oblačni program Crash Plan, ki podatke šifrira na stopnji računa. Za stranke smo ga gostili sami, uporabi pa se lahko tudi plačljiv prostor pri proizvajalcu.

2.2 PREGLEDOVANJE NAPAK

Če računalnik deluje počasi ali pa se ustavlja, je to prvi znak za težavo z diskom. Sistem ne javi napake, dokler okvara ne prizadene sistemskega ali programskega opravila. S pomočjo podatkov SMART lahko ugotovimo, ali je to vzrok. Njegovo delovanje je podrobneje opisano v poglavju 5.1.1 na strani 31. Večina operacijskih sistemov privzeto ne preverja zdravja diska. Tisti, ki pa ga, se ravnajo po proizvajalčevih mejah in ponavadi opozorijo, ko je že zelo pozno.

V sistemu Windows za namizno uporabo ni vgrajenega orodja, zato potrebujemo programe tretjih oseb. Najbolj priročen in brezplačen je CrystalDiskInfo. Na Sliki 1 lahko vidimo primerek diska z večjim številom poškodb, parametri z rumeno oznako kažejo okvare sektorjev. Surovi parametri so v šestnajstiški obliki. Če preračunamo, ima disk nekaj tisoč slabih sektorjev. Mnogo mu jih najbrž še ni uspelo zaznati, zato disk ni več primeren za uporabo. Polno pridobivanje podatkov utegne trajati več dni.

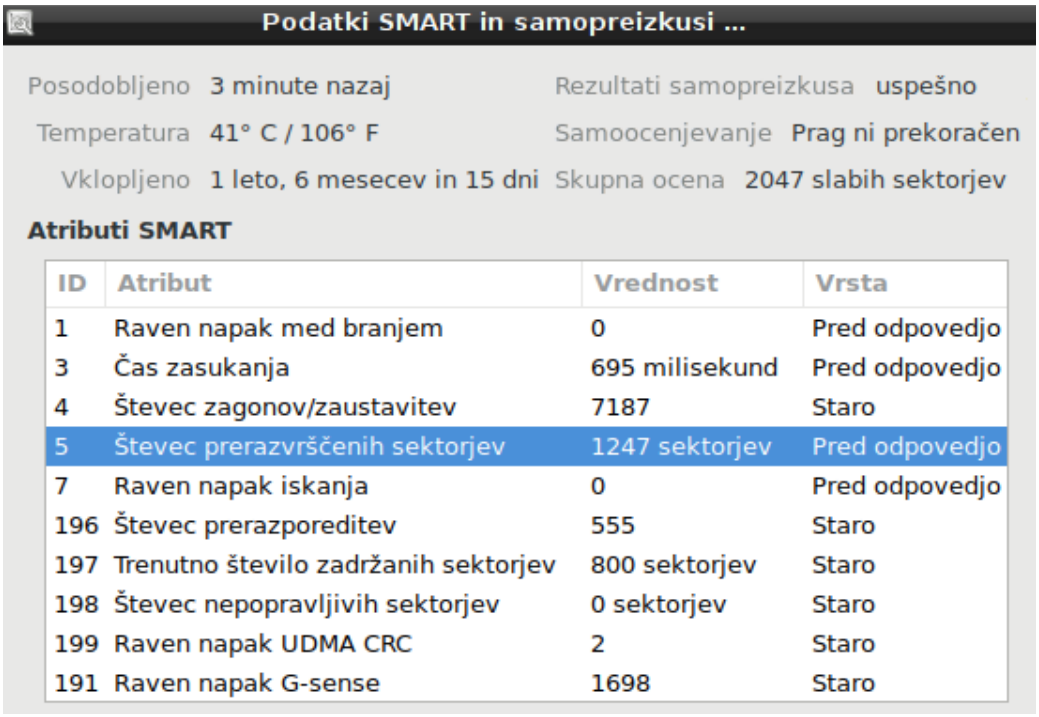


Slika 1: Program CrystalDiskInfo na sistemu Windows

Vir: Prirejeno po: Crystal Dew World, 2017

V namizno okolju Linux se običajno vgradi paket gnome-disk-utility, kličemo pa ga z ukazom gnome-disks. V meniju se imenuje Disks ali Diski v slovenskem prevodu. Med pripomočki omogoča ogled podatkov SMART, surovi podatki pa so že pretvorjeni v berljivo obliko.

Na Sliki 2 lahko opazimo visoko vrednost napak G-sense, ki povedo, da so vzrok za poškodbo tresljaji. Sliki sta prirejeni, da kažeta le ključne podatke, vidnih podatkov je bilo trikrat toliko, število prikazanih pa se razlikuje med modeli. V nekaterih okoliščinah gnome-disks ne more prikazat podatkov SMART. Takrat je priporočen program GSmartControl, ki je na voljo za večino operacijskih sistemov.



Slika 2: Orodje Gnome-disks na sistemu Linux

2.3 POŠKODOVANI PODATKI

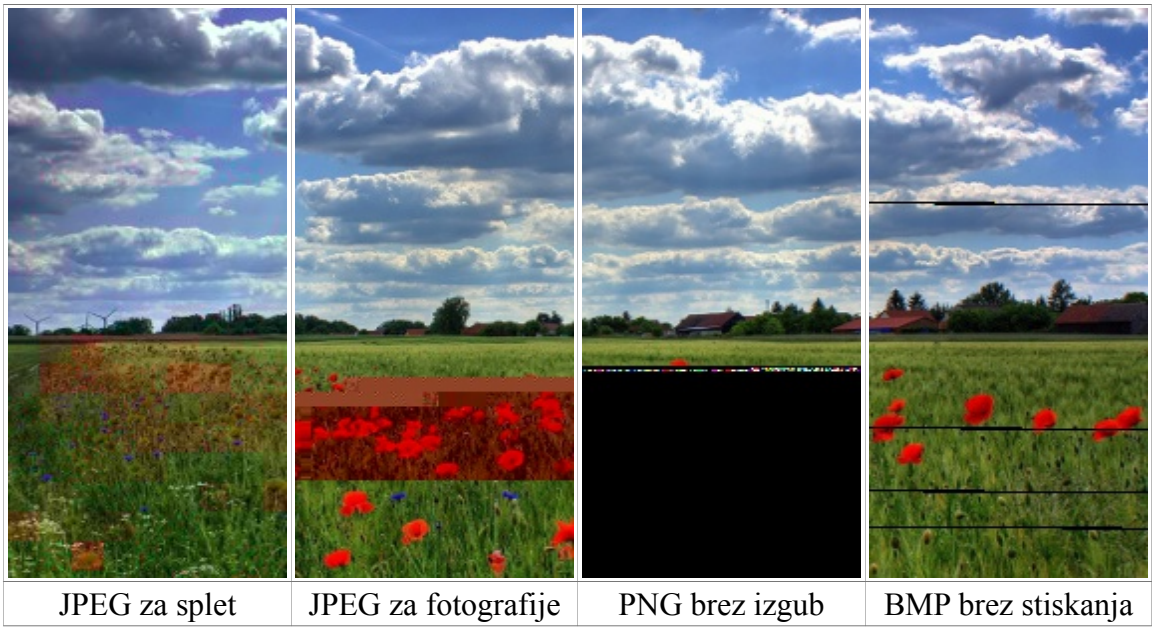
V računalništvu smo si zelo prizadevali, da bi dosegli kar se da veliko učinkovitost pri shranjevanju podatkov. Na žalost s tem izgubimo možnost rekonstruiranja poškodovanih datotek. Običajno velja pravilo, da boljše razmerje stiskanja pomeni večjo občutljivost na napake.

Podatke lahko razdelimo v kategorije glede na vrsto stiskanja. V začetku so prevladovali formati brez stiskanja, neposredno pretvorjeni iz analogne oblike. Motnje in napake so povzročile šumenje v zvoku, sneženje na sliki in izgubo črk v surovem besedilu. Pozneje so izumili algoritme za stiskanje, ki so bili prilagojeni tipom podatkov. Ločijo se na formate z izgubami in brez izgub.

Med najbolj znane formate brez izgub spada ZIP, deluje pa tako, da so vhodni podatki pred stiskanjem enaki pozneje razširjeni vsebini. To pa ne velja za formate z izgubami, ker podatke popačijo. V večini lahko razmerje stiskanja nastavimo. Posledično se to pozna na popačenjih, zato je velikost izhodnih podatkov veliko manjša kot pri tistih brez izgub. Prilagojeni so tipu podatka, kot so MP3 za zvok, JPEG za sliko in H264 za video, da ljudje čim manj opazijo spremembe podatkov glede na stiskanje.

Na občutljivost drastično vpliva velikost trdnega bloka in označevanje ponovnega začetka. Trdni bloki se uporabljajo za stiskanje datotek. Večji ko je, bolj učinkovito so stisnjeni podatki, vendar napaka v bloku pomeni prekinitev toka podatkov. Tako ne more nadaljevati do naslednjega bloka, preskočeni podatki pa so izgubljeni.

Pri branju naključne datoteke pa mora od začetka brati blok, v katerem se nahaja. Podobno deluje označevanje ponovnega začetka pri medijskih datotekah. Slike ponavadi ne uporabljajo ponovnega začetka, pri videih in zvoku pa se začetek priporočeno osvežuje na nekaj sekund, zato da omogoči preskakovanje po posnetku. S simulacijo slabih sektorjev lahko preizkusimo občutljivost na napake. Na Sliki 3 so bili preizkušeni najznačilnejši predstavniki za shranjevanje slik. Vsi razen formata brez stiskanja so dobili en slab sektor.



JPEG za splet	JPEG za fotografije	PNG brez izgub	BMP brez stiskanja

Slika 3: Poškodovani podatki s slabim sektorjem

Pri formatih brez stiskanja napake povzročijo manj škode, saj sosednji podatki med seboj niso odvisni. Pri bolje stisnjenih formatih, kot so PNG, WEBP in GIF, napaka pomeni prekinitev izrisa, nadaljnji podatki pa so neuporabni, ker slonijo na izgubljenem delu.

Formati z izgubami, kot je JPEG, sliko popačijo, od tipa zapisa in poškodovane sekcije pa je odvisna poškodba. Najznačilnejši sta poškodba barvne sheme in zamik slikovnih točk. Pri sliki JPEG za fotografije lahko opazimo, da se barvna shema popravi zaradi ponovnega začetka, vendar je uporaba odvisna od proizvajalca. Pri sliki JPEG za splet pa je poškodba rdeče barve vidna do konca slike, ker se ga ne uporablja zaradi učinkovitosti.

Čisti vektorski zapisi, kot se jih na primer uporablja pri 3D-modeliranju, so bolj poškodovani. Poškodba povzroči neskladnost med podrobnostmi predmetov, pri rekonstrukciji pa preskoči elemente s poškodbami. Mešani vektorski zapisi, kot sta PDF in SVG, združujejo različne formate, stisnjeno, brez stiskanja, z izgubami ali brez. Poškodbe so odvisne od vključene vsebine, ki je bila prizadeta. Če je ogrodje dokumenta nepoškodovano, potem ne vpliva na preostanek dokumenta. Stisnjen vektorski zapis lahko vključuje mešane podatke, vendar so na koncu stisnjeni, kakor DOCX in ODT to storita z ZIP-formatom. V primeru poškodbe se ta poveča, saj stiskanje zahteva dodatno raven izgube, manjša poškodba pa lahko vpliva na večjo količino podatkov.

2.4 VARNO BRISANJE

Včasih ne želimo, da naši podatki pristanejo v napačnih rokah. V preteklosti se je večkrat dogajalo, da je kdo s pomočjo zavrženih diskov pridobil podatke kreditne kartice. Najvarnejši način je, da disk fizično uničimo, vendar to ni potrebno, saj lahko s programi uničimo vso pomembno vsebino (povzeto po: Kovačič, 2009).

Za navadne diske imamo veliko izbire, saj možnost »Wipe« ponuja že skoraj vsak večfunkcijski program. Eden od mnogo programov, ki omogoča tudi čiščenje in vzdrževanje sistema, je CCleaner. Prepišemo lahko le prazen prostor ali pa celoten razdelek, postopek pa lahko traja tudi več ur. Pri izbiri števila prepisov ni treba pretiravati, enkraten prepis ponavadi zadostuje, saj javnosti še niso znane zanesljive metode za reševanje prepisane vsebine. Podrobnosti in teorija sledijo v poglavju 5.5 GLOBINSKA ANALIZA na strani 41. Podatke bolj občutljive narave pa lahko iz varnosti prepišemo nekajkrat, čeprav so možnosti obnove že zelo majhne. Tudi če v prihodnosti iznajdejo zanesljiv način, se bodo v tem času prepisani podatki porazgubili.

Bolj kompleksni so SSD-diski, saj je za pravilno brisanje treba uporabiti ukaz. To brisanje ima tudi pomen vzdrževanja, zato je v idealnih razmerah ta ukaz uporabljen pri vsaki izbrisani datoteki. Razlogi za njihovo neuporabo so nepodprtost vodil, operacijskega sistema, okvara datotečnega sistema ali ročni izklop. Z uporabo orodja za vzdrževanje trdih diskov proizvajalca lahko vidimo stanje in ali omogoča ročno čiščenje. Prav tako lahko v programu CrystalDiskInfo vidimo, ali je na razpolago funkcija TRIM. Za brisanje celotnega diska je priporočen zagonski sistem Parted Magic z orodjem »Disk Eraser« in izbiro ukaza »Internal«.

3 HRANJENJE PODATKOV V DIGITALNI OBLIKI

Poglavje obrazloži programsko delovanje računalnika pri shranjevanju podatkov. Razlage so na stopnji zahtevnosti srednje šole, potrebne pa so za poznejše razumevanje problemov reševanja in popravila logičnih napak. Če sistem naleti na obliko, ki je ne prepozna, lahko medij datoteko bere le kot dolg niz znakov. Brez določanja standardov shranjevanja bi imeli zmešnjavo, kakršna je bila na začetku računalništva. Vsak proizvajalec je ustvaril obliko zase, preživele pa so le najbolj uveljavljene na trgu. Ne glede na to, da obstajajo boljše alternative, so razlogi za prevlado predvsem v podpori.

3.1 RAZPREDELNICA RAZDELKOV

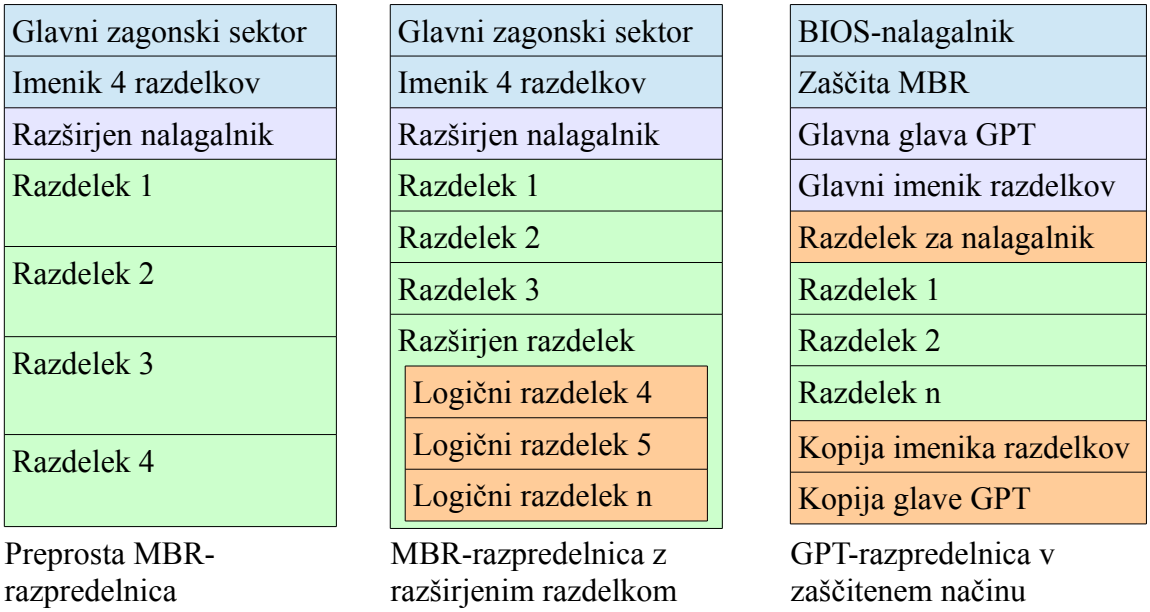
Razpredelnica razdelkov: Hranjenje podatkov se začne z razpredelnico razdelkov, ki določa način razporeditve razdelkov. Ta bolj vpliva na omejitve velikosti, števila razdelkov in definiranje nalagalnika operacijskega sistema. Med njimi sta najbolj znana MBR in GPT, obstajajo pa še drugi, ki so bili ustvarjeni v času razvoja računalnikov za operacijske sisteme, te pa danes redko srečamo.

MBR: Za MBR je značilno, da se je uporabljal že v DOS-obdobju, zato ga v današnjem času vežejo omejitve štirih razdelkov z največjo velikostjo 2 TB. Tako je preprost, da poškodba strukture nima mehanizma za samodejno popravilo.

Razširjen razdelek: Večji in bolj kompleksni sistemi so pri uporabi MBR potrebovali več kot štiri razdelke, zato so v enega od razdelkov vstavili novo razpredelnico, da lahko vsebuje podrazdelke. V orodjih se glavni razdelki imenujejo osnovni ali primarni, podrazdelki pa logični razdelki (povzeto po: https://en.wikipedia.org/wiki/Disk_partitioning).

Razdelek: V razdelek ponavadi spada datotečni sitem, ki je zadolžen za shranjevanje datotek. Poleg razširitve razdelkov lahko vanj postavimo tudi sistema LVM in LDM za logično upravljanje pogonov, ki omogočata dinamično upravljanje in njuno vezavo.

GPT: Zaradi vse bolj zmogljivih strežnikov je bilo treba preseči omejitve. Zanje so izumili GPT-razpredelnico, ki so jo 20 let pozneje skupaj z nalagalnikom UEFI začeli množično uporabljati. Omogoča do 128 razdelkov, seznam teh pa se nahaja na začetku z varnostno kopijo na koncu diska. Podpira jo večina operacijskih sistemov od leta 2008 (povzeto po: https://en.wikipedia.org/wiki/GUID_Partition_Table).



Slika 4: Primerjava najpogostejših struktur razpredelnic razdelkov

Vir: Prirejeno po: Desmond, 2012

Na Sliki 4 je z modro barvo označen prostor, namenjen osnovni razpredelnici MBR, prostor, ki so ga pri GPT namenoma preskočili, da je bil omogočen hibridni način. Način zaščite v MBR-prostoru le označi celoten disk kot en sam razdelek z oznako. To operacijskemu sistemu, ki pozna le MBR, pove, da so na podatki disku v neznani obliki. Ta način je samodejno uporabljen pri sistemih Windows, saj uporabnike opozori, da so na disku podatki.

Vijoličasta barva označuje rezerviran prostor, ki je pri MBR namenjen razširjenemu nalagalniku, pri GPT pa ga uporabi večji seznam razdelkov. Razširjen nalagalnik je zato premaknjen v prvi razdelek, ki je neviden in lahko ima poljubno velikost.

Z zeleno je označen enostaven prostor za razdelke, z rdečo pa mogoči zapleti, kadar iščemo ali prenašamo poškodovano strukturo.

Zagon operacijskega sistema: Skupaj z MBR se večinoma uporablja postopek nalaganja prek programske kode BIOS. Po zagonu strojne opreme BIOS prebere Glavni zagonski sektor na napravi, ki pokaže na mesto razširjenega nalagalnika. Ta nato zažene uporabniški vmesnik za nalaganje na razdelku, označenem za zagon, ki začne postavljati izbrani operacijski sistem od jedra navzgor.
S prihodom sistema UEFI se je ta postopek poenostavil in doprinesel varnost, saj ne potrebuje nalagalnika v razpredelnici in meni zažene sam. To je hkrati povzročilo nekaj nevšečnosti pri popravljanju računalnikov, ker oteži dostop zunanjih zagonskih orodij.

Standard omogoča katerokoli kombinacijo MBR ali GPT z BIOS ali UEFI, vendar je podpora odvisna od operacijskega sistema. Hibridni način v GPT je podoben načinu zaščite, le da vsebuje tudi glavni zagonski sektor, ta pa omogoča zagon prek BIOS. Čeprav je bila združljivost načrtna, je sistemi, kot sta OSX in Windows, ne podpirajo (povzeto po: https://en.wikipedia.org/wiki/GUID_Partition_Table).

3.2 DATOTEČNI SISTEM

Datotečni sistem: Ta določa, kako bomo hranili podatke v razdelku. V osnovi se deli na tabelo datotek z metapodatki in prostorsko shrambo. Datotekam sta določeni velikost in povezava na sektor začetka, včasih je razdeljena na več začetkov. Ko datoteke dodajamo, se te najprej shranijo v tabelo, nato se jim dodelijo metapodatki, šele potem se začne zapisovanje vsebine datoteke. Podrobnejše delovanje med datotečnimi sistemi se razlikuje, saj so tabele drugačne, ponekod tudi ločene za metapodatke. Ob izbrisu se vnos običajno označi za izbrisanega, dokler se seznam ne napolni in ga ne prepiše nova datoteka.

Formatiranje: To je operacija, ki oblikuje podatke tako, da jim določi razpredelnico in razdelke ter povozi obstoječo strukturo. Določi lahko tudi datotečne sisteme v razdelkih, včasih pa se izraz uporablja za ponovno namestitev operacijskega sistema.

Velikost gruče: Velikost gruče določa najmanjšo velikost datoteke, pa tudi faktor, s katerim zaseda velikost na disku. Velikost datotek na disku se poravna na gručo, to pomeni, da bi pri gruči 16 KiB nekaj bajtov veliko datoteko zaokrožila na 16 KiB. Pri velikem številu datotek tako izgubljamo učinkovitost shranjevanja.

Priporočeno je tudi, da velikost gruče ni manjša od fizične velikosti sektorja, saj mora disk prebrati celoten sektor. Pri manjših gručah to pomeni izgubo odzivnosti, prava velikost sektorja pa je lahko drugačna od prikazane. Razlog tiči v podpori, saj večina operacijskih sistemov podpira le sektorje velikosti 512 B, pri modernih diskih pa to več ne zadošča.
Zaradi gostote so velikost spremenili na 4 KiB, zaradi združljivosti pa večina teh diskov uporablja emulacijo na 512 B. Če je trdi disk večji od 2 TiB, potem skoraj zagotovo uporablja 4 KiB. Da ne izgubimo hitrosti, velikost gruče ne sme biti manjša od velikosti sektorja, zato je privzeta velikost sektorja 4 KiB, razdelki se poravnajo po vrednosti MiB, ta pa je postala privzeta poravnava pri Windows Vista (povzeto po: https://en.wikipedia.org/wiki/Advanced_Format).
Podpora za neposredno uporabo sektorjev velikosti 4 KiB je bila leta 2010 vgrajena v Linux in Windows 8. Uporabniki starejših sistemov morajo zato pri nakupu novih diskov biti pozorni na oznako 4Kn, predvsem če so namenjena za poslovno rabo.

Velikost črk datotek: Datotečni sistemi imajo za shranjevanje imen datotek na voljo omejeno tabelo, od nje sta odvisna dolžina in nabor znakov. V začetku računalništva so se uporabljale le velike črke iz tabele ASCI, ki se je med regijami razlikovala. Sistemi, kot je FAT, jo še vedno uporabljajo zaradi združljivosti. Z razširjeno podporo znakov hrani imena v obeh oblikah.
Sodobni sistemi uporabljajo le razširjeno obliko, zato so regijsko neodvisni. Med prehajanjem na modernejše sisteme je nastala težava razlikovanja velikih in malih črk. Če jih ne razlikuje, pomeni, da lahko na isto mesto shrani samo eno od variacij ’datoteka’,’Datoteka’ in ’DATOTEKA’. Vsi sistemi, združljivi s standardom Unix, razlikovanje podpirajo, FAT32 pa na primer še vedno ne razlikuje velikih črk, a upošteva pravilno ime (povzeto po: https://en.wikipedia.org/wiki/Case_sensitivity).

Metapodatki: Vsak datotečni sistem svojim datotekam pripisuje njene lastnosti, shranjene na skupnem seznamu. Med najbolj razširjenimi so velikost, datumi nastanka, spremembe zadnjega dostopa in pravice. Večina ima ta prostor omejen na količino podatkov, nekateri sistemi pa omogočajo pregibne dodatke. NTFS na primer omogoča dodajanje opisa, hrani pa tudi fragmentiranje datotek, a ker ima omejen prostor, presežek shrani v nove vpise, vezane na isto datoteko.

Fragmenti datotek: Z brisanjem datotek nastane prostor nenadzorovanih oblik. Čeprav je na takem disku prostora dovolj, se morajo za uspešno shranjevanje datoteke razdeliti oziroma fragmentirati. Stari sistemi brez te možnosti so v takšnem položaju ponavadi vrnili napako pomanjkanja prostora. Drug zelo pogost vzrok za fragmentacije se zgodi, kadar je datoteka ukleščena med sosednjima in jo je treba povečati. To se najpogosteje dogaja s programskimi dnevniki, saj se ves čas razširjajo. Fragmentiranje vpliva na hitrost branja, zato imajo sistemi, kot so Linux, BSD in OSX, dobre mehanizme za preprečevanje nastajanja fragmentov, česar pa ne moremo trditi za Windows.
V preprostih datotečnih sistemih tabela datotek kaže na začetni kos, konec vsakega kosa pa kaže na začetek naslednjega kosa, dokler ni dosežena skupna velikost datoteke. Ta način je značilen za FAT-družino, izguba ene povezave pomeni izgubo naslednjih blokov. Za branje naključnega dela datoteke mora to brati od začetka. Pri NTFS so vsi podatki o fragmentih shranjeni med metapodatki, zato datoteka pri večjem številu fragmentov ustvari več vnosov v tabelo. Zaradi mašenja tabele to vpliva na odzivnost celotnega razdelka, pri več milijonih fragmentov pa postane zelo neodziven. V sistemih Unix je za fragmente značilna drevesna struktura preusmerjanj z vmesnimi majhnimi bloki za vozlišča. Tako brez oviranja tabele omogoča hitro branje naključnega dela datoteke, izguba teh blokov pa je manj katastrofalna kot pri FAT (povzeto po: Shao, 2016).

Datotečni dnevnik: Sodobni datotečni sistemi, namenjeni poganjanju operacijskega sistema, beležijo dnevnik sprememb v tabeli datotek. Običajno se spremembe najprej zapišejo v dnevnik, nato pa se uveljavijo. Datotečni dnevnik se ponavadi nahaja med prostorom za podatke in je uporabniku neviden. Njegova velikost je omejena, zato samodejno briše stare vnose. V primeru nenadne prekinitve lahko sistem hitreje najde nastale napake in prepreči morebitno izgubo datotek. Z njegovo pomočjo lahko tudi veliko lažje povrnemo izbrisane datoteke, razen če je bilo zahtevano brisanje sledi. Med najbolj znane datotečne sisteme, ki uporabljajo dnevnik, spadajo NTFS, EXT3, EXT4, BTRFS, HFS+ in ZFS.

3.2.1 Pogosti datotečni sistemi

Družina FAT: Razvit je bil za uporabo v sistemu DOS, uporablja pa se še danes, zato je ta družina najbolj podprta med operacijskimi sistemi. Je preprost in robusten, skozi razvoj pa je delovanje ostalo enako. Najopaznejše razlike so omejitve velikosti razdelka, datotek in podpora imen. Danes ga najbolj poznamo v različicah FAT16, FAT32 in exFAT. Najbolj znan je FAT32, in sicer po velikosti gruče in omejitvi datoteke na 2 GiB. Naslednik exFAT popravi omejujoče velikosti datotek in razlikovanje velikosti črk, a je preveč patentno zaščiten in nedokumentiran, zato njegova podpora med različnimi sistemi raste bolj počasi (povzeto po: https://en.wikipedia.org/wiki/File_Allocation_Table).

NTFS: Ustvarjen je bil zato, da nadomesti FAT32, ki se je takrat uporabljal za sistem Windows. Oblikovali so ga z modernimi zahtevami skupaj z združljivostjo Unix, kar pomeni, da lahko shrani datoteke z razlikovanjem velikosti črk, čeprav operacijski sistem Windows uporabnike omejuje ter onemogoča znake in imena datotek, ki jih uporablja za funkcije. Glede razlikovanja velikost črk se obnaša kot sistem FAT. Če je drug sistem ustvaril konfliktne datoteke, bo pri odpiranju prebral naključno, pri brisanju pa odstranil vse (povzeto po: https://en.wikipedia.org/wiki/NTFS).

V prihodnosti ga bo v strežniški shrambi nadomestil ReFS, saj omogoča večje prostore, samodejna popravila in navidezne razdelke. Čeprav zagotavlja določeno stopnjo kompatibilnosti z NTFS, ne omogoča nadgradnje in ne more zagnati sistema Windows.

Družina EXT: EXT2 je bil predstavljen skupaj z različico 1.0 jedra Linux, pozneje je bil nadgrajen na EXT3. Ker je datotečni dnevnik glavna dodana stvar, omogoča nadgradnjo in način združljivosti, tako da vsa orodja za EXT2 delujejo na EXT3. Z dodatnimi razširitvami se je oblikoval še EXT4, ki prav tako omogoča združljivost. S sistemom za neodvisnost povezav se v programih delo z datoteko kliče s potjo, nato pa izvaja prek datotečnega identifikatorja. Če datoteko premaknemo ali izbrišemo in je trenutno v uporabi, jo bo ta program še vedno lahko uporabljal neodvisno (povzeto po: https://en.wikipedia.org/wiki/Ext3).

Neodvisno od prejšnjih razvijalcev je nastal BTRFS, ki jih bo v prihodnosti zamenjal. Dodaja mnogo naprednih funkcij, kot sta protipodvajanje podatkov in logično upravljanje podrazdelkov. Zaradi pregibnega imenika se lahko nadgradi iz sistemov EXT, prejšnji imenik se ohrani v obliki logičnega razdelka, zato lahko pretvorbo tudi razveljavimo.

HFS+: Uporablja se na večini operacijskih sistemov podjetja Apple, njegovo razlikovanje velikosti črk pa je nastavljivo. To v praksi povzroči nekaj preglavic, saj nekatere aplikacije ne podpirajo razlikovanja in zato ne delujejo. Struktura sloni na prvotnem HFS, ki je bil ustvarjen za podporo trdih diskov.
HFS+ so razvili še pred prehodom na procesorje x86 zaradi prostorske neučinkovitosti predhodnika na večjih pogonih. Zaradi takratnih razmer in procesorske arhitekture so nekateri podatki zapisani v drugačnem zaporedju, kakor jih danes bere. Veliko kritik je požel zaradi nedokončane podpore znakov Unicode, saj mora pretvarjati med to in končno obliko (povzeto po: https://en.wikipedia.org/wiki/HFS_Plus).

Veliko potrebnih funkcij ni bilo vgrajenih, zato so jih postopoma dodajali z nadgrajevanjem operacijskega sistema OSX. To je razlog, da večina gonilnikov za druge sisteme ne priporoča pisanja na medij z nameščenim OSX-sistemom. Z vgrajevanjem dnevnika se mu zaradi združljivosti spremeni naziv v HFSJ.
V letu 2017 ga je na aktualnih različicah nadomestil APFS, zato da popravi posledice zastarelosti in utrdi programsko dograjene funkcije iz HFS+.

3.2.2 Podatki

Datoteke: Vsi dostopni podatki so zapisani v obliki datotek, razlikujejo se le po tipu. Najpogostejša tipa sta podatkovna datoteka in mapa. Ponavadi si predstavljamo, da so v datoteki le podatki. A mapa je prav tako datoteka, njeni podatki pa so seznam vsebujočih datotek. Med bolj znanimi tipi je tudi povezava na drugo datoteko, v njej lahko vidimo in urejamo vsebino vezane datoteke ali mape, lahko pa je tudi na drugem pogonu.

Oblike datoteke: Pri zapisanih podatkih se njihovo vrsto shrani z dvema indikatorjema. Najhitreje opazen in prepoznan je s pomočjo končnice, pri sistemih Windows je to edini upoštevani indikator. Drugi indikator znane oblike datoteke se začne na začetku podatkov z glavo. Začne se z opredelitvijo skupine in formata, nadaljuje pa s parametri kodiranja, ki ga uporablja.

Linux privzeto gleda na podatke glave, če je ne prepozna, pa na končnico, zato mnogo sistemskih datotek nima končnice. Izguba podatkov glave ponavadi pomeni, da so podatki neuporabni, mogoče jih je obnoviti po znanem vzorcu ali pa s preizkušanjem pri manj kompleksnih formatih.

Šifrirana vsebina: Šifriranje spremeni in zavaruje podatke z uporabo ključa, ta pa je potreben za branje teh podatkov. Izvaja se lahko na več načinov in stopnjah varnosti, s tem pa se poveča kompleksnost strukture. Najbolje zavarovano šifriranje je vgrajeno v disk, pri njegovi okvari pa povzroča največ težav.

Večina trdih diskov, ki omogoča zaklepanje, zaradi drage strojne zahteve ne šifrira podatkov. Takega gesla ni mogoče ukrasti z zlonamerno kodo in če gesla ne vemo, lahko ukrepamo enako kot pri okvari elektronike. Pri diskih z aktivnim šifriranjem pa je edina možnost pridobiti geslo iz elektronike.

Nekateri zunanji diski z možnostjo varovanja podatkov uporabijo navaden disk in nato z zunanjo elektroniko izvajajo šifriranje. Če takšen disk vzamemo iz ohišja, podatkov ne moremo prebrati.

Da šifriramo podatke na vsakdanji strojni opremi, moramo to storiti na razdelku. Če želimo šifrirati razdelek z operacijskim sistemom, potrebujemo nalagalnik na ločenem razdelku, ki ni nešifriran. Izberemo lahko več stopenj zaščit po dolžini ključa. Daljši in bolje zaščiten ko je, več procesorske moči potrebuje med delovanjem. To deluje tako, da na začetku z našim geslom zaščiti samodejno ustvarjen ključ te dolžine, ki pa je nato uporabljen na celotnem razdelku (povzeto po: Kovačič, 2007).

Ko geslo zamenjamo, ta le ponovno zaščiti glavni ključ, če pa naše geslo ni dovolj varno, tudi daljši ključ ne more bolje zaščititi podatkov. Pozornost je treba nameniti tudi zaščiti nalagalnika, saj njegova okužba pomeni, da nam geslo lahko ukradejo.

4 SAMOSTOJNO REŠEVANJE

Nesreče se nam lahko zgodijo zelo hitro in v takih situacijah je pomembno, da vemo, kako reagirati. Če podatkov ne vidimo več, še ne pomeni, da ne obstajajo, pomeni pa, da smo izgubili povezavo do njih. Ob nezgodi zato ne smemo več zapisovati na disk, ker lahko tako izgubljene podatke prepišemo in naredimo še več škode. Premalo pa se zavedamo, da dandanes skoraj vse, kar naredimo na domačem računalniku, sproži ukaz za zapis, tudi zaustavitev sistema.

Fizični pogon mora biti zanesljiv. Če ima medij slabe sektorje, je počasen ali pa je splošno nezanesljiv, moramo nujno ustvariti posnetek. Če smo prepričani, da strojna oprema deluje brezhibno, imamo le logično težavo s podatki.

Izdelava slike pogona je še vedno priporočena, tudi zato da se zavarujemo pred lastno napako. Koliko so podatki vredni za nas in koliko smo pripravljeni tvegati, presodimo sami. Vedno obstaja možnost spontane okvare, za morebitne nepredvidljive situacije in izgube pa odgovarja vsak sam.

Za neodvisen dostop do podatkov potrebujemo neodvisen operacijski sistem. Za določena opravila je nujna uporaba sistema Linux, predvsem za ustvarjanje posnetka. Na srečo ga ni treba namestiti, saj obstajajo zagonski mediji z vsemi potrebnimi orodji. Trenutno sta priporočeni distribuciji Deft za uporabniku prijazno mešanico orodij za reševanje in Knoppix za polno namizno izkušnjo. Za dolgotrajno delo je bolj priročno, da ga namestimo, če ga pa že imamo, lahko potrebna orodja dodamo.

4.1 STANJE STROJNE OPREME

Preden začnemo, moramo ugotoviti, kakšen je medij, ki ga hočemo reševati. Najpomembnejši dejavnik, da se lahko sami lotimo reševanja z diska, je zaznavanje strojne opreme. Pri NAND-napravah (SSD, ključki, kartice) je najpogostejša mehanska okvara pri krmilniku. Pri vseh razen SSD-diskih lahko začnemo reševanje, če deluje.

Kar zadeva izbrisano vsebino, je pri SSD-diskih v večini primerov potreben strokovnjak (razloženo v poglavju 5.4 na strani 39). V času pisanja je pogostost okvar SSD-diskov znašala približno en odstotek. Vzroka se razdelita na dotrajanost po več letih delovanja in odpoved elektronike po nekaj mesecih uporabe.

Z diski HDD pa je večja težava, najpogostejša je fizična poškodba. Če disk nima zunanjih znakov poškodbe, ne dela čudnih klikajočih zvokov in ga BIOS prepozna, gre najbrž le za poškodbo površine. Včasih so se diski bolj kvarili zaradi mehanske napake in dotrajanosti, danes pa prevladujejo fizične poškodbe zaradi zunanjih vplivov.
Če se disk dovolj dobro odziva in ima zgolj manjše število slabih sektorjev, potem ni sporno kopirati le datoteke, ki jih potrebujemo s programom, kot je Roadkil's Unstoppable Copier. Ni primeren za logične težave, saj je edina sposobnost preskakovanja slabih sektorjev.
Pred 25 leti je celoten disk zadrževal največ 1 GiB podatkov, toliko lahko danes zapišemo na površino, ki jo pokrije las, dolg šest centimetrov in pol.

4.2 ODTIS POGONA

Če je stanje diska vprašljivo in obstaja možnost, da se podatki pokvarijo, medtem ko jih razvrščamo, potem moramo celoten disk prepisati drugam. Predvsem je to pomembno, če disk ni več zanesljiv zaradi mehanskih vzrokov. Če imamo veliko slabih sektorjev, je ta korak obvezen, predvsem ker se lahko nahajajo na pomembnih delih datotečnega sistema. Nadaljnje reševanje podatkov brez izdelave odtisa lahko stanje še poslabša, saj z večkratnim branjem poškodbe povečamo.
Najenostavneje je ciljne podatke shraniti v datoteko, lahko pa uporabimo tudi večji ali enako velik disk. Pri pisanju na disk je dobro, če je bil ta prepisan s prazno vrednostjo, sicer je treba preostale stare podatke naknadno prepisati s prazno vrednostjo. Če tega ne storimo, lahko pri delni obnovi povzročijo zmedo podatkov.
V primeru slabega sektorja ga disk privzeto poskuša prebrati večkrat. Sekvenčno branje je tako zelo upočasnjeno pri večjem številu slabih sektorjev. Zaradi možnosti okvare je pomembno, da uporabne podatke čim hitreje zberemo.

Strokovnjaki uporabljajo naprave in razširitvene kartice, narejene za tovrstna opravila, ker podpirajo dodatne ukaze. Namenska oprema prisili disk, da sekvenčno branje nadaljuje, mesta neuspelih sektorjev pa si zapomni. V naslednjih korakih ponovno prebere celoten disk in postopoma uporablja ukrepe za popravljanje napak na neuspelih področjih. Ta metoda je najbolj priporočena, saj vedno obstaja možnost, da med kopiranjem disk odpove, mi pa želimo prekopirati čim več.

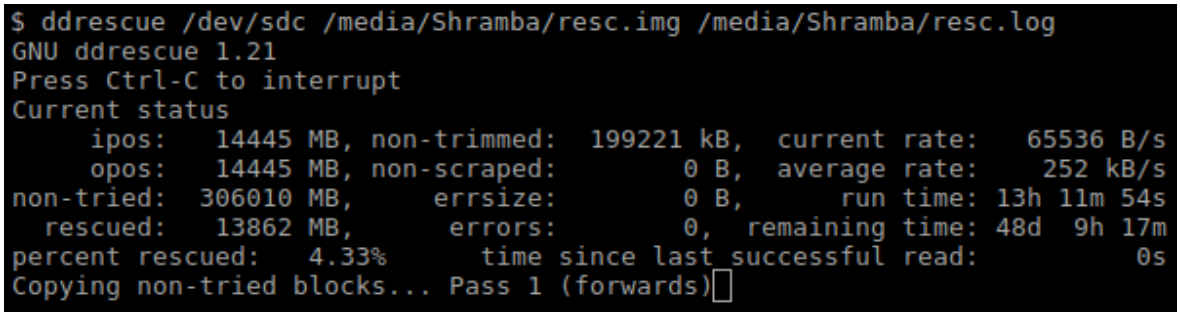
Upravlja tudi napajanje diska, zato da lahko samodejno ponovno zažene neodzivno elektroniko. Na diskih, ki omogočajo ukazne priklope, se lahko priključi in pridobi še večjo prednost nad programskim reševanjem (povzeto po: Shipley in Door, 2012).

Za domačo rešitev zadostuje programski nadomestek, saj uporablja podoben pristop. Čeprav obstaja mnogo orodij za kloniranje na sistemih Windows, ga nikakor ne priporočamo v tem postopku, saj programom ne omogoči prevzema nad vmesnikom.
Najprimernejše za celoten disk je brezplačno orodje GNU ddrescue na sistemu Linux. Večina distribucij ga namešča s paketom gddrescue in zažene z ddrescue. Na voljo je tudi predhodnik, pravilnega prepoznamo po oznaki GNU. Če Linuxa nimamo na voljo, lahko uporabimo živo različico pripravljenih distribucij za reševanje podatkov. Ta ima vgrajena orodja za nadaljnje reševanje in poenostavi vpenjanje samo za branje.

Program uporablja datoteko za načrt, kamor shrani vse uspele, preskočene in neuspele sektorje med kopiranjem, da lahko normalno deluje po prekinitvi in sesutju. Pri zagonu programa samo znova uporabimo isto datoteko z načrtom. Po prekinitvi, ki vključuje ponovni zagon računalnika, je treba preveriti, ali ima disk še vedno enako pot naprave. Pred zagonom poti naprav preverimo v programu gnomes-disks ali gparted (povzeto po: Data Medics, 2015).

- Osnovni ukaz za zagon:
  - `sudo ddrescue [vhod] [izhod] [datoteka_za_načrt].`
- Vhodna naprava v datoteko posnetka:
  - `sudo ddrescue /dev/sdc /media/Shramba/resc.img /home/resc.log`
- Kopiranje brez razreza (ker ta traja relativno dolgo za majhen doprinos):
  - `sudo ddrescue -f -n /dev/sdc /media/Shramba/resc.img /home/resc.log`
- Ponovni preizkus slabih sektorjev (včasih jih še nekaj uspe):
  - `sudo ddrescue -f -A /dev/sdc /media/Shramba/resc.img /home/resc.log`

– Kloniranje naprave v napravo:
  • `sudo ddrescue -f /dev/sdc /dev/sdd /home/resc.log`
– Prepisovanje s prazno vrednostjo, tam, kjer so slabi sektorji (uporabno, če smo kopirali na disk, ki pred tem ni bil izpraznjen):
  • `sudo ddrescue --fill-mode=- -f /dev/zero /dev/sdd /home/resc.log`



Slika 5: Program GNU ddrescue

Vir: Prirejeno po: Diaz Diaz, 2016

Ddrescue prav tako kot strojna oprema deluje z več intervali branja, njegove prikazane informacije vidimo na Sliki 5. Zaradi raznolikosti in omejitev strojne opreme na osnovni plošči ne more prisiliti diska v drugačno delovanje, zato ob napaki branja preskoči večje število sektorjev in jih ponovno preizkusi v naslednjih korakih.

Zazna tudi počasne sektorje, jih preskoči in se po prvem koraku vrne nanje. Počasni sektorji so kategorizirani v tri skupine glede na hitrost in tudi bere jih v treh intervalih. Ko konča trenutno fazo branja, se obrne v nasprotno smer in bere preostale drobce. Če naleti na slabši sektor od trenutno dovoljenega v intervalu, naredi enako kot na začetku.

Zadnji sklop faz se ukvarja s slabimi sektorji in traja najdlje. Branje preskočenega polja zaradi slabega sektorja iz nasprotne smeri označitve se imenuje obrezovanje. Če naleti na slab sektor, spet preneha, saj se slabi sektorji večinoma držijo skupaj. Predzadnja faza se imenuje razrez, saj preizkusi še zadnje neznanke. Preostalo je malo nepreizkušenih podatkov in velika je verjetnost, da bodo sektorji slabi, zato ni nenavadno, da ta korak ročno preskočimo. Zadnji korak je le še ponovno branje slabih sektorjev, ki ga lahko ponavljamo (povzeto po: Diaz Diaz, 2017).

	Zdrav sektor		Slab sektor		Preskočeno počasno polje
	Prebran počasen sektor		Počasen sektor		Označeno za obrezovanje
	Prebran slab sektor		Rešljiv slab sektor		Označeno za razrez

Osnovno stanje: 22 dobrih, 16 počasnih, 12 slabih (2 rešljiva)

Strojna oprema za kopiranje podatkov

Hitro branje: 50 sektorjev, 50 časovnih enot

Branje z nizko korekcijo: 28 sektorjev, 280 časovnih enot (sk. 330)

Branje z visoko korekcijo: 12 sektorjev (2 prebrana), 1200 časovnih enot (sk. 1530)

Programsko kopiranje z ddrescue

Iskanje najhitreje dostopnih podatkov: 11 dobrih, 2 počasna, 1 slab, 131 časovnih enot

Poizkušanje počasnih sektorjev: 9 dobrih, 9 počasnih, 1 slab, 199 časovnih enot (sk. 330)

Obrezovanje slabih sektorjev: 2 dobra, 3 počasni, 1 slab, 132 časovnih enot (sk. 462)

Razrez slabih sektorjev: 2 počasna, 9 slabih, 920 časovnih enot (sk. 1382)

Ponovni poskus nerešenih podatkov: 12 slabih (2 prebrana), 1200 časovnih enot

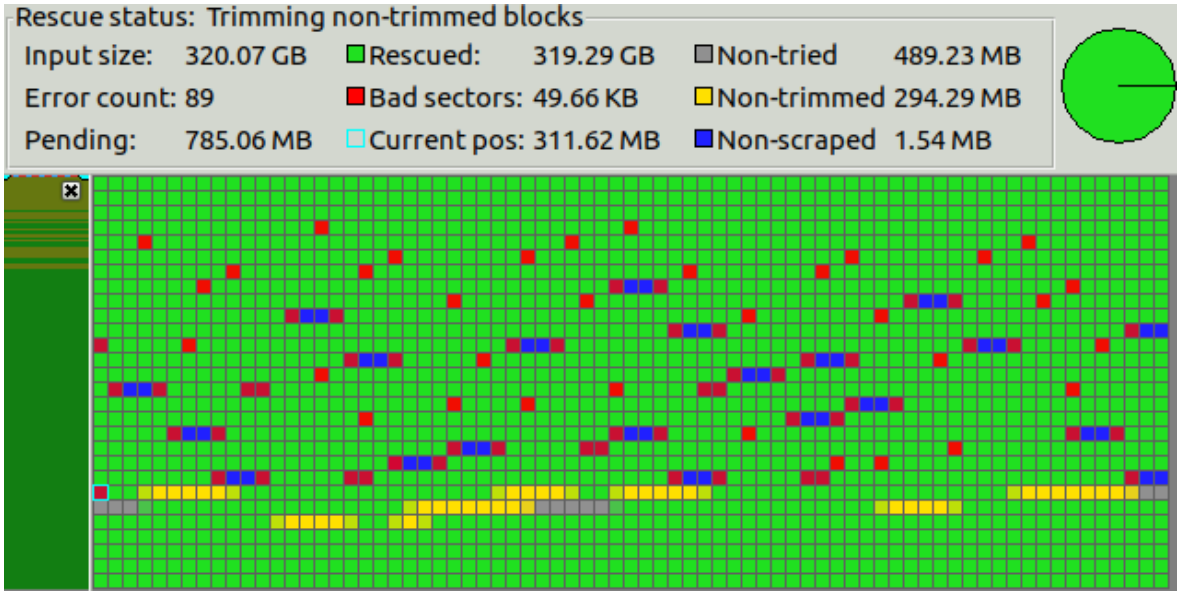
Slika 6: Primerjava postopkov branja na odseku diska.

Na Sliki 6 lahko opazimo prednosti kopiranja s pomočjo strojne opreme, v praksi pa se lahko uporabi še več vmesnih korakov. Medtem ko ddrescue porabi 462 časovnih enot za kopiranje večine podatkov, jih strojna oprema le 330 in še uspešnejša je. V zadnjih korakih pa porabi več skupnega časa, vendar ima boljše možnosti. Podobno raven uspešnosti dosežemo pri ddrescue s ponovnim branjem slabih sektorjev.

Programi za reševalno kopiranje z zaporednim branjem, kot je dc3dd, bi na celotnem disku potrebovali enako časa kakor ddrescue, vendar bi zaradi slabe časovne razporeditve pri tisoč časovnih enotah uspešno prebrali le sedem sektorjev.

Ko kopiranje končamo, preverimo, ali je pisanje končano, z ukazom sync. Če nas takoj vrže na `$:` za izvajanje novega ukaza, je zapisovanje končano. Zakasnjeno pisanje se pojavi, če je naprava, ki jo beremo, hitrejša od izhodne. Tako se prebrani podatki nalagajo v pomnilnik. Ponavadi se pomnilnik prazni takrat, ko vhodni disk dela počasneje, v zadnjih korakih.

Nekatere novejše distribucije že ponujajo ddrescueview za grafični pregled trenutnega stanja. Povežemo ga na datoteko z načrtom, za prikaz aktualnega stanja pa moramo vključiti osveževanje načrta. Na Sliki 7 vidimo prikaz stanja v zelo veliki povečavi, dovolj podrobno, da je viden vzorec branja sektorjev.



Slika 7: Grafični pregledovalnik ddrescueview

4.3 LOGIČNA POŠKODBA

Ko imamo zanesljiv medij ali sliko, lahko na njem zaženemo programsko opremo za reševanje. Posnetke pogona v sistemu Linux vpnemo za branje, na sistemu Windows pa potrebujemo program OSFMount. Programi ga zaznajo kot logičen disk in večina z njim deluje normalno. Pri uporabi diska pa moramo biti previdni, saj s pisanjem hitro prepišemo kakšen del izbrisane ali izgubljene datoteke.

V okoliščinah, kjer je naš glavni sistem Linux, lahko preostanek programskega reševanja, ki zahteva Windows, izvajamo s pomočjo navidezne strojne opreme Virtualbox. Prek ukazov lahko ustvarimo diskovno datoteko z vezavo na fizično napravo.

```
sudo VBoxManage internalcommands createrawvmdk -filename /home/link.vmdk -rawdisk /dev/sdb
```

Namesto povezave na fizično napravo lahko uporabimo tudi navidezno napravo `/dev/loop0`, ki nastane pri vpenjanju posnetkov v Linuxu. Privzeto in priporočeno se posnetki v te naprave vpnejo samo za branje, vendar takšna premostitev dela težave sistemom Windows ob poskusu zapisa podatkov. Mogoče je tudi vpeti posamezne razdelke, `/dev/loop0p1` in `/dev/sdb2`, vendar jih v tej obliki Windows ne zazna.

Vsakodnevna opravila, ki sprožijo zapisovanje na razdelek:

- vklop in izklop operacijskega sistema,
- uporaba večine programov, predvsem spletnih brskalnikov,
- odpiranje map z nalaganjem ogledov sličic ustvari datoteko s predogledi na tistem razdelku (skrite thumbs.db na Windows, .DS_store na OSX, Linux dela mapo .thumbnails v domači mapi),
- vpenjanje pogona, saj Windows samodejno ustvari mape, če ne obstajajo,
- orodja za popravilo sistema (fsck, Check disk, Startup diagnostic),
- defragmentiranje, pazimo na izvajanje po urniku,
- namestitev programa − bodimo pozorni, kam jih nameščamo.

Glavni namen reševanja je pridobitev podatkov, včasih pa je potrebno tudi popravilo sistema, saj lahko le prek njega izvozimo zaščitene stvari, kot so certifikati.

Primeri logičnih reševanj se delijo na:

- Izbrisane ali izgubljene datoteke:
  - na datotečnem sistemu z dnevnikom ali brez dnevnika:
  - na neodvisnem razdelku ali na glavnem z operacijskim sistemom.
- Poškodba strukture, kot se zgodi pri prekinjenem zapisovanju in slabih sektorjih, tako navidezno izgubimo razdelke ali datoteke.
- Delni prepis razdelka, kot je formatiranje.
- Bolj prepisana struktura, kot se zgodi pri tovarniški obnovi in ponovni namestitvi operacijskega sistema čez obstoječe podatke.

4.4 IZGUBLJENI RAZDELKI

Se vam je že kdaj zgodilo, da vas je računalnik vprašal, ali želite disk formatirati? To se zgodi zaradi poškodbe strukture in brez pogleda na preostanek strukture podatki niso vidni. Najznačilnejša okvara se zgodi na prenosnih ključkih s sistemom FAT32. V lažjih primerih ga Linux odpre samo za branje, medtem ko Windows ponuja formatiranje.

Eden pogostih vzrokov, ki ni posledica okvare, je prekinjeno pisanje. Predvsem se to dogaja na sistemih Windows, saj uporabljajo medpomnilnik pri pisanju na počasnejši medij. To pomeni, da so podatki še vedno lahko v pomnilniku in njihov zapis še traja, zato je še vedno priporočeno varno odstranjevanje naprav, saj izvrže napravo po končanju zapisa.

Za ponovno umeščanje razdelkov, ki smo jih izgubili, je zaradi poškodbe razpredelnice ali pa ročnega izbrisa priporočen program Testdisk. Uporaba na prepisanih razdelkih ni priporočena, če hočemo iz njih le pridobiti podatke. Program poišče sledi zgradbe in prikaže domnevno strukturo pred okvaro, s katero zapišemo popravek te strukture.

Na začetku nas program vpraša za disk, če želimo delati s slikovno datoteko, jo moramo podati med zagonom. Ukaz »`sudo testdisk /media/Shramba/resc.img`« bo zagnan na prejšnjem odtisu, nato nas vpraša po razpredelnici razdelkov. Izbrana pomeni, da bo pri popravljanju pretvorjena. Oznaka Intel je uporabljena za MBR, kot to lahko vidimo na Sliki 8.

Izbor razpredelnice razdelkov

```
Please select the partition table type, press Enter when done.
>[Intel  ] Intel/PC partition
 [EFI GPT] EFI GPT partition map (Mac i386, some x86_64...)
 [Humax  ] Humax partition table
 [Mac    ] Apple partition map
 [None   ] Non partitioned media
 [Sun    ] Sun Solaris partition
```

Glavni meni

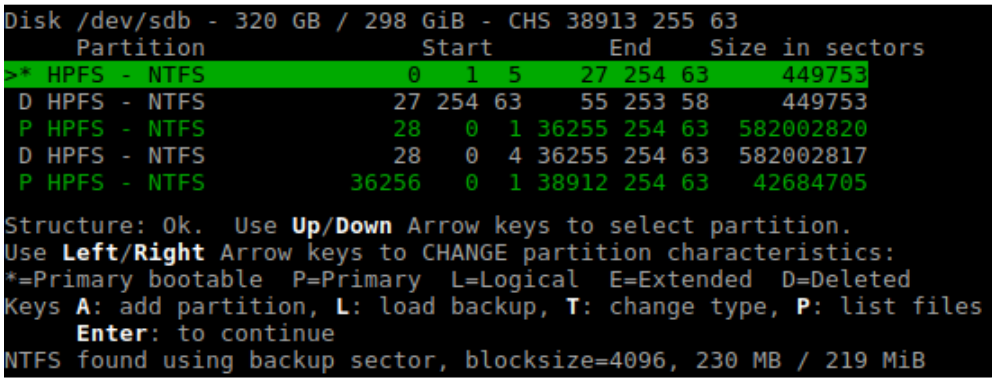
```
>[ Analyse  ] Analyse current partition structure and search for lost partitions
 [ Advanced ] Filesystem Utils
 [ Geometry ] Change disk geometry
 [ Options  ] Modify options
 [ MBR Code ] Write TestDisk MBR code to first sector
 [ Delete   ] Delete all data in the partition table
 [ Quit     ] Return to disk selection
```

Slika 8: Možnosti v orodju Testdisk

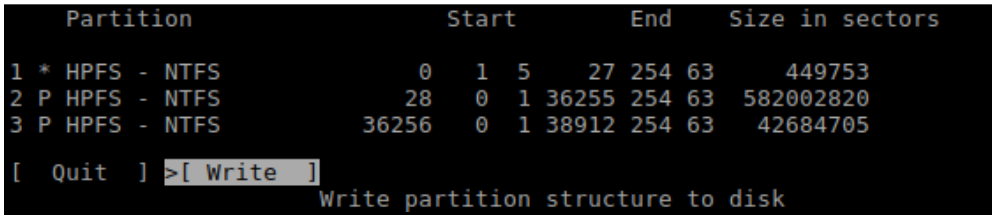
Vir: Prirejeno po: CGSecurity, 2015

Pri iskanju razdelkov poleg fizično prisotnega razdelka najde tudi varnostne kopije datotečnega sistema ali razpredelnice razdelkov. Omogoča tudi popravljanje zagonskega zapisa za medije z operacijskim sistemom. Pri postavljanju strukture je treba s črkami označiti razdelke za popravilo po tipu zagonska, glavna ali logična kot na Sliki 9.

Najdeni razdelki s kopijami in označevanje popravka



Zapis sprememb



Slika 9: Obnavljanje izgubljenih razdelkov z orodjem Testdisk

Vir: Prirejeno po: CGSecurity, 2015

V prikazanem primeru je bil prepisan celoten disk z operacijskim sistemom Windows 7. Prvi razdelek je bil po popravilu strukture preveč poškodovan za prepoznavo. Ta razdelek je skrbel za nalagalnik BOOTMGR, ki je nadomestljiv. Zamenjamo ga z razdelkom iz enakega operacijskega sistem. Različica licence ni pomembna, le da se ujema s servisnim paketom in bitnim sistemom. Nato je treba uporabiti orodje za popravilo zagona. Najdemo ga na zagonskem orodju ERD ali na namestitvenem disku z izbiro popravila sistema. Ta operacija bo storila več sprememb, zato je varnostna kopija priporočena. Če pri formatiranju ni bilo prepisanih tudi drugih pomembnih področij sistema, lahko to zadošča.

4.5 ISKANJE IZBRISANIH DATOTEK

Datoteke lahko izgubimo z brisanjem ali zaradi okvare datotečnega sistema. Preden začnemo, se moramo zavedati, kje se izbrisane datoteke nahajajo − na kateri fizični napravi, v katerem razdelku − in ali je struktura zaznana pravilno.

Če so podatki na delujočem ločenem razdelku, disku ali odstranljivem mediju, lahko preprosto namestimo program in rešene podatke shranimo drugam. Če pa so podatki bili na glavnem razdelku, je potrebna dodatna previdnost.

V bolj enostavnih primerih prek neodvisnega podatkovnega medija, kot je ključek, zaženemo prenosno (angl. *portable*) različico programa. Pri tem moramo paziti na sprožitve zapisovanja, delo s programi in samodejna opravila v ozadju operacijskega sistema.

Najučinkoviteje je, da delo s programi shranimo na neodvisen medij in računalnik izklopimo iz napajanja, saj ob zaustavitvi shrani tudi nekaj manj pomembnih podatkov.

Nadaljujemo z neodvisnim operacijskim sistemom, naj bo to zagonski operacijski sistem, ali pa disk vstavimo v drug delujoč računalnik. Če je bila struktura poškodovana ali prepisana, je to edina možnost, ki nam ostane.

4.5.1 Reševanje s pomočjo delcev datotečnega sistema

V idealnih pogojih uporabimo strukturo datotečnega sistema za uspešnejše reševanje. Z uporabo dnevnika postane vračilo izbrisanih datotek zelo enostavno, pri sistemih brez dnevnika pa je potrebna analiza celotnega razdelka. Včasih se izgubijo imena nekaterih datotek ali map.

Pri običajnem brisanju datotek na delujočem ali rahlo poškodovanem datotečnem sistemu zadostujejo brezplačni programi, ki pa običajno potrebujejo več ročnega dela in nastavljanja. Pri programu Recuva je treba vklopiti sestavljanje strukture map, sicer jih shrani kot posamezne datoteke. Čeprav je samodejno prepoznal strukturo, je prikazal seznam v obliki posameznih datotek.

V huje poškodovanih primerih preostale drobce uporabimo za povezavo mest datotek z njihovimi imeni. Preostanek najdemo s prepoznavo vzorcev zapisa datotek, ki mu rečemo golo reševanje. Boljši in plačljivi programi znajo to opraviti sami, pri brezplačnih pa je potrebna kombinacija programov.

S programom CloneSpy lahko izločimo podvojene datoteke reševanj dveh programov. Z načinom dveh bazenov (angl. *pool*) vsakega vežemo na svojo mapo, ki vsebuje rešene podatke različnih programov. Nato ukažemo, da izbriše tiste datoteke, ki jih je našel program za golo reševanje, saj želimo obdržati imena datotek iz dnevnika. Nekaj zelo podobnih datotek bo treba izločiti še ročno. To je kompromis med brezplačno opremo in reševanjem, kar je mogoče rešiti.

Z enostavnimi primeri izgube smo preizkušali programe. V datotečnih sistemih z najširšo podporo FAT32 in NTFS smo uprizorili lažje poškodbe. V prvem primeru smo izbrisali nekaj datotek brez pisanja in večina programov je delovala dobro ali vsaj zadovoljivo. V naslednjem primeru je bil razdelek formatiran s starim datotečnim sistemom, zato da je prepisal le majhen del strukture. Ta primer je mnogim, tudi plačljivim programom predstavljal težave.
Vsi spodaj omenjeni plačljivi programi so pri danem primeru delovali odlično in so tudi do uporabnika prijazni. Omogočili so analizo posnetkov diska, na voljo pa so na preizkus. Dokler ne kupimo licence, nas omejujejo na skupno količino rešenih podatkov ali na največjo dovoljeno velikost rešene datoteke.

Preizkušenimi programi, ki so omembe vredni:

- Plačljivi, na voljo so tudi v preizkusni različici:
  - **R-TT R-Studio** najbolje zazna drobce datotečnih sistemov, ima najširšo podporo datotečnih sistemov in omogoča rekonstrukcijo vezav RAID. Priporočen tudi za stare različice (R-Tools Technology, 2017).
  - **Runtime GetDataBack Simple** ponuja širok izbor datotečnih sistemov, za polno reševanje deluje dovolj dobro (Runtime, 2017).
  - **GetData Recover My Files** je trenutno omejen le na datotečne sisteme za Windows in OSX (GetData, 2017).
- Brezplačni:
  - **Piriform Recuva** je priporočen brezplačni program, tudi prenosen, za izbrisane datoteke deluje dovolj dobro, pri poškodovanem datotečnem sistemu pa je priporočeno kombiniranje z golim reševanjem (Piriform, 2016).

- **Convar PC™ Inspector File Recovery** je omembe vreden, čeprav je že zelo zastarel in se v bolj kompleksnih primerih slabo izkaže (Convar, 2005).
- **Testdisk** ima sposobnost iskanja po dnevniku v meniju »advanced«, »undelete«, vendar je njegova uporaba zapletena (CGSecurity, 2015).
- **Restoration** je najmanjši brezplačen program za iskanje izbrisanih datotek v datotečnem sistemu NTFS in je primeren le za reševanje posameznih datotek (Kato, 2003).

4.5.2 Golo reševanje

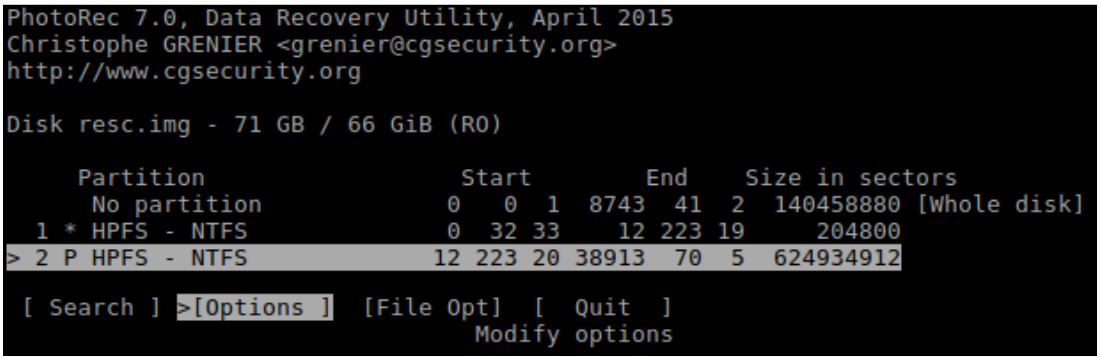
Z golim reševanjem poiščemo uporabne datoteke v danem prostoru − naj bo to celoten disk, razdelek ali pa nezavzet prostor v datotečnem sistemu. Najbolj uporabno je, če želimo najti prejšnje ostanke po formatiranju in delnem prepisu diska. Primerno je tudi za iskanje ostankov izbrisanih datotek po okužbi z izsiljevalskim virusom, ki je zašifriral dokumente, uspešnost pa je odvisna od količine preostalega prostora.
Ker so podatki v tabeli datotek lahko napačni, se nanjo ne zanaša, zato so imena datotek izgubljena. Bolj strokovni programi omogočajo kombinirano reševanje in upoštevajo ostanke dnevnika. Sposobni so zaznati, kateri kosi dnevnika pripadajo sistemu pred prepisom, vendar dobro delujejo le plačljivi.

Priporočen brezplačen program za golo reševanje je Photorec, v distribucijah Linux se namesti v paketu skupaj s testdisk. Med programi, ki lahko prepoznajo tudi oblike datotek, omogoča največji nabor različnih formatov.
V samodejnem delovanju preišče in obdrži le datoteke, ki ne kažejo znakov poškodbe. Če ga zaženemo brez parametrov, nas vodi skozi nastavitve. Enako kot pri programu testdisk lahko uporabimo odtis diska s parametrom v ukazu za reševanje datotek.
Ukaz »`sudo photorec /media/Shramba/resc.img`« bo zagnan na prejšnjem odtisu diska.

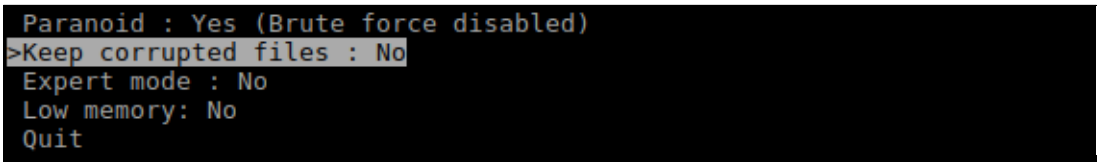
Če želimo poln potencial programa, ga je treba pred iskanjem nastaviti. V dodatnih nastavitvah, kot jih vidimo na Sliki 10, srečamo nekaj izrazov:

- **Paranoid:** Poskusil bo sestaviti fragmentirane datoteke, privzeto je vklopljeno.
- **Brute force:** Poškodovane datoteke bo s surovo silo popravil, potrebuje veliko procesorske moči in časa, privzeto je izklopljeno.
- **Expert mode:** Med postopkom bo uporabnika spraševal za potrditev ugotovljene velikosti gruče, odmik datotečnega sistema in možnost shranitve preostanka neprepoznavnih podatkov, privzeto je izklopljeno.
- **Low memory:** Operacija potrebuje veliko delovnega pomnilnika. To nastavitev se uporabi le v primeru težav, privzeto je izklopljeno.

Izbor razdelka ali celotnega diska za iskanje



Dodatne nastavitve, dostopne prek spodnjega menija



Slika 10: Nastavitve v programu Photorec

Vir: Prirejeno po: CGSecurity, 2015

Ko nas vpraša za vrsto datotečnega sistema, je to zaradi fragmentiranih datotek, zato se ločijo v dve skupini, kot lahko vidimo na Sliki 11. Obseg iskanja omejimo na prazen prostor v primeru izbrisanih datotek in formatiranja, na celoten razdelek pa pri pokvarjeni strukturi. Pred začetkom iskanja nas vpraša po izhodni mapi. Tam bo poskušal najdene datoteke združiti z najdenimi podatki map.

Izbor družine razdelka podatkov vpliva na iskanje fragmentacij

```
To recover lost files, PhotoRec need to know the filesystem type where the
file were stored:
 [ ext2/ext3 ] ext2/ext3/ext4 filesystem
>[ Other     ] FAT/NTFS/HFS+/ReiserFS/...
```

Izbor celotnega razdelka ali pa le prostega prostora

```
Please choose if all space need to be analysed:
 [   Free    ] Scan for file from NTFS unallocated space only
>[   Whole   ] Extract files from whole partition
```

Postopek delovanja

```
Pass 1 - Reading sector    6154304/624934912, 14898 files found
Elapsed time 0h01m21s - Estimated time to completion 2h15m44
txt: 6797 recovered
exe: 2931 recovered
tx?: 2059 recovered
png: 691 recovered
gif: 478 recovered
jpg: 425 recovered
bmp: 132 recovered
others: 476 recovered
  Stop
```

Slika 11: Delo s programom Photorec

Vir: Prirejeno po: CGSecurity, 2015

5 HRANJENJE PODATKOV V MEHANSKI OBLIKI

Pri težavah mehanskega izvora so potrebni dodatna oprema, znanje in predvsem previdnost. Kadar se zgodi hujša poškodba ali okvara, je treba medij odpreti ali razstaviti, kar pa se sme početi le v dovolj čistem okolju. Nekatere manjše mehanske težave ne bodo povzročile napake, vendar jih lahko že začutimo v obliki ponovljive upočasnitve.

5.1 MIKROKRMILNIK

Da lahko prostor za shranjevanje deluje enako ne glede na vrsto in model medija, potrebujemo združene vmesnike. Prek vmesnika se pošiljajo ukazi za branje, zapisovanje in vzdrževanje, zanje pa so zadolženi mikrokrmilniki. Na osnovni plošči komunicira med operacijskim sistemom in medijem. Delujejo lahko v različnih standardiziranih načinih, da pokrijejo zahteve strojne opreme in uporabnikov.
Mikrokrmilnik na napravi skrbi za prenos, integriteto in samodejno vzdrževanje podatkov. Njegovi notranji parametri so prilagojeni za vsak model naprave, skozi delovanje pa se oblikujejo izjeme pri hranjenih podatkih. Z vse večjo zmogljivostjo potrebujejo na mediju tudi vse več prostora za hranjenje teh izjem. Med drugim ga uporabljajo pri popravljanju napak za preusmerjanje slabe površine ter njihovo beleženje.
Poganja jih strojna koda (angl. *firmware*), od nje je odvisna učinkovitost delovanja in samodejnega vzdrževanja. V redkih primerih je tovarniška napaka povzročila poznejšo odpoved celotne serije, ki se je lahko zgodila po nekaj letih ali celo mesecih. Večinoma so proizvajalci izdali popravek strojne kode. Pri nekaterih primerih je bilo mogoče medij po okvari popraviti brez poseganja.

5.1.1 Tehnologija za odkrivanje in popravljanje napak

Pretekle napake in njihovi samodejni popravki so bili včasih skrivnosti posameznih diskov, to, da je nekaj narobe, se je vedelo le po preizkušanju odzivnosti celotnega medija. Na neki način je prikazal stanje, vendar napake niso bile vidne, dokler ni preseglo proizvajalčeve meje.

Od leta 1995 je predpisan standard SMART za predvidevanje odpovedi in zaznavanje napak. Eden izmed naborov ukazov za vzdrževanje je poizvedba trenutnih podatkov. Poleg modela tako pridobimo še temperaturo, število vklopov, čas delovanja in statistiko napak. Vsako napako pri branju, popravljeno ali nepopravljeno, se zapiše v poročilo. V statistiki tako ponavadi dobimo število napak v sektorjih, za vsako vrsto napake v svojem parametru (povzeto po: https://en.wikipedia.org/wiki/S.M.A.R.T.).

Tehnologija SMART je značilna za trde diske HDD in SSD. Uporabljati se je začela z vodilom IDE, na SCSI in SATA pa je vgrajen v standard. Z zmožnostjo prenosa ukazov SCSI na vodila USB, FireWire in ThunderBolt lahko statistiko pridobimo tudi prek adapterjev in nekaterih bolje opremljenih prenosnih medijev.

Statistika pokvarljivosti trdih diskov ponudnika oblačnih storitev Google je pokazala vpliv okoliščin na okvare. Delovna temperatura le rahlo vpliva, večje odstopanje je zaznano le pri temperaturah, nižjih od 20 stopinj Celzija, kjer so bile okvare podvojene. Nenehna obremenjenost prav tako krajša življenjsko dobo, vzdržljivost je odvisna od namembnosti diska. Na podlagi podatkov SMART lahko tudi predvidevamo povečano možnost za okvaro. Če sodimo po statistikah, je kar 40-krat večja možnost za okvaro v naslednjih 60 dneh pri katerikoli zaznani napaki. Predvsem opazni so števec prerazporejenih sektorjev, napake iskanja in napake preizkušanja (povzeto po: Pinheiro et. al., 2007).

5.1.2 Slabi sektorji

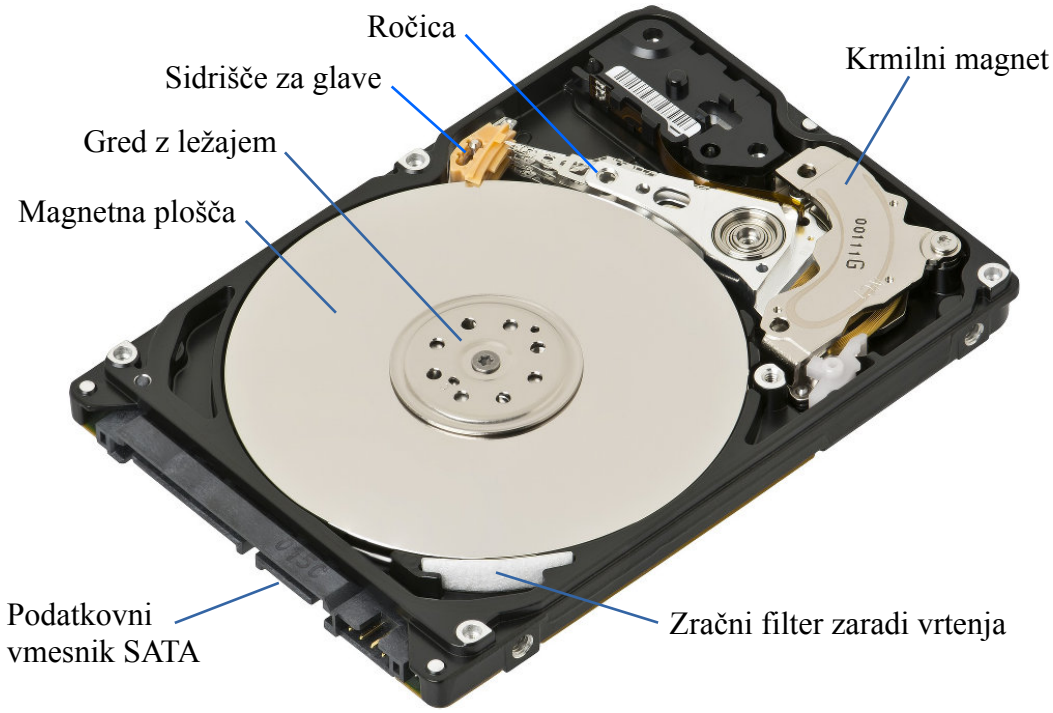
Na vsakem mediju se podatki shranjujejo v sektorjih, ki so lahko različnih velikosti, vendar mora njihovo velikost podpirati operacijski sistem. Najpogosteje lahko zasledimo velikost 512 B, saj ima najbolj pokrito podporo, leta 2011 pa so začeli množično proizvajati diske s sektorji, velikimi 4 KiB, vendar jih večina uporablja način združljivosti z razčlembo na 512 B. Mediji s SMART po koncu sektorja pripišejo še preverjevalnik parnosti ECC, ki lahko odkrije do 200 poškodovanih bitov, njima pa sledi še CRC-povzetek, ki potrdi popravek. Ko mikrokrmilnik prebere podatke, naprej preveri ujemanje z ECC. Če se ne ujema, jo poskusi popraviti. Če mu uspe, podatke potrdi z ujemanjem CRC-povzetka, če pa mu ne uspe, ga poskuša večkrat prebrati z drugačnimi metodami, dokler mu ne uspe ali pa ne doseže dovoljenega časa.

Slabe sektorje lahko delimo po treh kriterijih. Popravljiv je takrat, ko mu je uspelo prebrati vsebino, vendar je za to potreboval več časa. Nepopravljiv je takrat, ko mu kljub naporu ni uspelo v dovoljenem času pridobiti pravilnih podatkov.
V statistiko se lahko nepopravljivi delijo glede na sloj integritete, ki ni uspel, ECC ali CRC, to je odvisno od proizvajalca. Nazadnje pa imamo še uničene sektorje, če iz polja sploh ni mogoče prebrati podatkov. SMART jih kategorizira med nepopravljive, prepoznamo pa jih po zaporedni množičnosti.
Čas, potreben za razvrstitev slabega sektorja, je odvisen od mikrokrmilnika in se lahko razlikuje med modeli. Ko medij naleti nanje, si jih zapomni in jih doda v statistiko. Samodejno ga preusmeri, če je bil popravljiv, če je bil nepopravljiv, pa šele po prepisu. Tako lahko še vedno naletimo na slabe sektorje, ker mu jih bo morda pozneje uspelo prebrati − morda zaradi ugodnejših fizičnih razmer ali pa ker jih beremo z orodjem za reševanje, ki mu bo uspelo poskus zagotovo shraniti.

5.2 KLASIČNI TRDI DISKI (HDD)

Trdi disk z magnetnim zapisom je danes najstarejši še živeči način za hranjenje podatkov. Podatki so zapisani na vrtečih se diskih, čeznje pa se pomika ročica z elektromagnetno glavo. Položaj je krmiljen z zelo močnim magnetom, kot lahko vidimo na Sliki 12.
Delovanje je podobno kot pri gramofonskih ploščah, le da vsebuje več plošč s podatki na obeh straneh in uporablja magnet. Z razvojem so postajali vse manjši in bolj kompaktni. Podatki so razvrščeni na kolobarje z izrazitimi mejami, ki jim rečemo sledi. Nato se podatki razdelijo še na sektorje, ki so razvrščeni od notranjega kroga navzven.

Ko mikrokrmilnik dobi ukaz za določen sektor, premakne glavo na pravilno sled in počaka, da se plošča zavrti v položaj, kjer se ta nahaja. Pri zaporednem branju podatkov to deluje zelo dobro. Če pa mikrokrmilnik pridobiva ukaze za sektorje na različnih mestih, to vključuje veliko čakanja.
Glave za vsako stran plošč so združene v sklop, ta pa se mora na položaju umiriti. Diski z bolj odzivnim krmiljenjem lahko lovijo sektorje bližje trenutnemu zasuku plošče. Glava je ločena na pisalni in bralni del. Pisalni del je na sprednji strani, da pri pisanju s takojšnjim branjem overi zapisano.

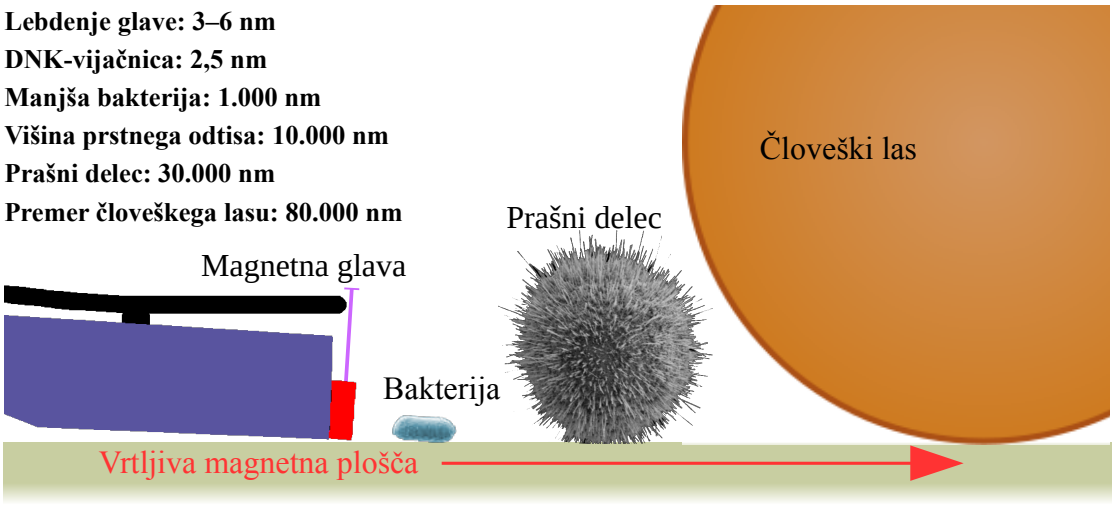


Slika 12: Notranjost trdega diska za prenosne računalnike

Vir: Prirejeno po: Amos, 2013

Plošče so znotraj trdne, zunaj pa prevlečene z magnetno plastjo. Za varnost so še premazane proti kemičnim vplivom iz zraka. Vrtijo se s konstantno hitrostjo, nekateri varčni diski pa prehajajo med dvema hitrostma. Glede na oddaljenost od središča glava z različno hitrostjo lebdi čez površino, posledično so tudi podatki zapisani v različni gostoti. Glede na najpogostejše hitrosti vrtenja se lahko glava čez površino premika s hitrostjo od 20 do 110 kilometrov na uro, zato se je ne sme dotikati in je običajno oddaljena nekaj nanometrov. Zunanji vpliv je lahko usoden, zato ima danes večina diskov vgrajen senzor. Ob majhnih tresljajih prepreči nalet, a se kljub temu površina hitro poškoduje. Na tak način najhitreje dobimo slabe sektorje.

V mirovanju ročica zasidra glavo iz območja podatkov. To je njen privzeti položaj, kadar magnet ni krmiljen. Lebdenje glave se vzdržuje z zračno blazino ob vrtenju plošč. Če se glavi ne uspe pravočasno umakniti, lahko trči. Pri delujoči elektroniki v primeru izpada napetosti ta reagira dovolj zgodaj, da prepreči trk. Težko si je predstavljati občutljivost te tehnologije, zato Slika 13 primerja odmik glave od površine z vsakdanjo nesnago.



Slika 13: Primerjava natančnosti delovanja diska

Vir: Prirejeno po: https://acsdata.com/data-recovery-3tb-seagate-hard-drive/ (1. 7. 2017)

Diski so zaprti proti vdoru prahu, večina ima zaščitne luknjice za izenačevanje pritiska, ki prepuščajo zrak in vlago. Postavljene so na manj dostopnih mestih in označene z opombo, da se jih ne sme prekrivati. Najnovejši patent je popolnoma nepredušno zaprt zaradi vsebnosti helija, s čimer sta dosežena še manjši zračni upor in razmik med ploščami.
Vsak sektor se začne s kratko identifikacijo začetka, nadaljuje se s podatki in konča s kodama ECC in CRC. Včasih so magnetno polje zapisovali v neposredni vrednosti, danes pa zaradi gostote in posledično medsebojne magnetne interference delajo z magnetnimi prehodi. Ko glava bere podatke, reagira na prehod med polji, zato so zapisani podatki preoblikovani tako, da vsebujejo zadostno število prehodov, preprečijo pa tudi uporabo identične kode, kot je postavljena na začetek sektorja (povzeto po: Šinkovec, 2011).

Na sledeh so poleg podatkov tudi enakomerno razporejeni servozapisi, ki služijo za določanje sredine sledi in amplitude sektorjev. Zaradi točno določenih položajev je lahko vrinjen med sektor, ki ga tako razpolovi. Magnetni zapis obstane do deset let, glava pa zazna oslabljen magnetni zapis in ga obnovi. Če se disk segreje, se zapisani podatki lahko začnejo raztezati in rahlo zamikati, pri napakah branja se to upošteva in prebira tudi zamaknjeno na sled. Če pa se disk segreje na zelo visoke temperature, na primer ob požaru, se magnetni zapis izgubi (povzeto po: Šinkovec, 2011).

V primeru slabih sektorjev in če želimo disk še uporabljati, potem je najbolje, da se sektorje preusmeri z mikrokrmilnikom ali pa označi za slabe znotraj datotečnega sistema. Nekateri programi omogočajo popravljanje slabih in počasnih sektorjev s prepisovanjem. Dostopni čas se jim izboljša, vendar rešitev ni trajna, saj se stanje lahko hitro spet poslabša.

5.2.1 Popravljanje klasičnih trdih diskov

Večina hujših okvar zahteva poseganje v notranjost. Zaradi občutljivosti se to dela v izoliranem čistem prostoru, ki ne sme vsebovati delcev prahu ali vlage. V času diskov, velikih manj kot GiB, je bilo mogoče zamenjati elektroniko z identičnega diska, pri čemer ni bilo treba posegati v notranjost. Dandanes pa so prek mikrokrmilnika izredno natančno umerjeni. Pri velikost do nekaj 10 GiB zamik glave postane tolikšen, da mora pri vsakem sektorju popraviti položaj branja; pri nekaj sledeh tudi neuspešno, zato večji diski sledi ne ujamejo. Umerjanje je shranjeno v mikrokodi v ločeni spominski celici, brez nje pa težje pridobimo podatke.
Nekateri diski imajo kontakte za terminalski dostop do tovarniškega načina. Njegova uporaba je nevarna, saj lahko zaradi napačnih ukazov ostanemo brez podatkov. Pazimo, da jih ne zamešamo s kontakti za spremenjen način delovanja. Diski lahko omogočajo oboje, le eno, ali pa nič od tega.

Tudi ko elektronika ne deluje več pravilno, obstaja možnost pridobitve potrebne konfiguracije prek tovarniškega načina. V nasprotnem primeru pa nam ostane le še odstranjevanje in ponovna spojitev čipov s temi podatki na tiskano vezje iz identičnega diska. Paziti moramo še na nadgradnje modelov, saj jih pogosto večkrat izboljšajo. Te spremembe lahko onemogočijo medsebojno kompatibilnost. Čas izdelave je ponavadi viden na zgornji stani diska, podrevizija pa je označena na tiskanem vezju.
Ker se rezervnih delov ne dobi od proizvajalca, jih lahko dobimo le od trgovcev rezervnih delov prek spleta, ki diske razstavijo. Ker je včasih nemogoče dobiti rezervni del točne podrevizije, je vse pogostejše odločanje za univerzalno elektroniko. Pri tem je potrebnega več dela in preizkušanja različnih parametrov. Natančno je treba poravnati zamike sledi, magnetno gostoto pa prepozna sam. Če so sektorji že bili preusmerjeni, to lahko predstavlja dodatno težavo.

V primeru trka je nujno treba odpreti disk, saj iz poškodovane površine nastane prah, ki ogroža preostale podatke. Notranjost je treba očistiti, da ne vsebuje nobenega prahu. Uporaba čiste sobe je obvezna, saj oddaljenost med glavo in ploščo dandanes znaša zgolj nekaj nanometrov.

Kakršnakoli nesnaga v zraku je nekaj tisočkrat večja od tega, pokrije pa lahko več 100 KiB podatkov. Ob delovanju bi glava treščila v te delce, jih metala po površini in razbila na še manjše delce. Delci se lahko zataknejo med glavo in ploščo ter drgnejo po površini.

Pri trku se lahko glava tudi poškoduje. Po tem lahko v nekaterih primerih še bere podatke, vendar zelo upočasnjeno. Pri odmikanju s površine je treba paziti, da dodatno ne drgne po površini, po odmiku s površine pa, da se ne zlepijo skupaj in še bolj poškodujejo. Strokovnjaki pri tem postopku uporabijo posebne glavnike. Če je le mogoče, se naprej uporabijo obstoječe glave, ker je umerjanje novih zelo zahtevno.

Zaradi hujše fizične poškodbe, ki je posledica udarca diska, se lahko ukrivijo plošče, praktične rešitve za to pa ni. V teoriji bi bilo mogoče brati zelo počasi s pomočjo senzorja gibanja, vendar je ta metoda časovno in ekonomsko neupravičena.

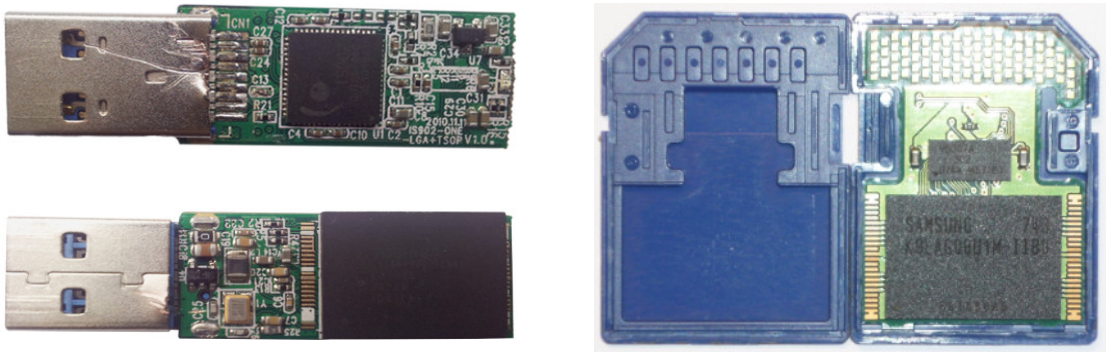
Med bolj zoprnimi okvarami sta servomotor in ležaj. Sneti je treba plošče, te pa morajo obvezno ostati popolnoma poravnane. Vpete so na nosilec z gredjo na sredini. Ohišje diska je rahlo manjšega premera kakor nosilec in nosi servomotor z ležajem. Razmik med ploščami mora biti točen zaradi glav, zasuk pa zaradi servozapisov, zato se menjave ne loteva, dokler ni zares potrebno.

V primeru poplave se diska ne suši, ampak se ga namoči v destilirano vodo. Delci, raztopljeni v vodi, se ob sušenju namreč primejo na površino diska, kovinski pa na magnet.

5.3 MEDIJI S POMNILNIKOM NAND

Polnilnik NAND se je razvil iz pomnilnika EEPROM, ki je znan po obstojnosti in hitrem dostopnem času. Za razliko od njegovih predhodnikov je imel tudi zmožnost večkratnega prepisovanja. Oblika NAND je omogočila tudi pisanje po kosih v večji blok celic.

Za razliko od trdih diskov se lahko celice vežejo v poljubno fizično velikost. Je bolj odporen proti zunanjim vplivom in ima boljši dostopni čas, vendar je pri zaporednem branju veliko počasnejši. Ponavadi se jih uporablja za notranji spomin manjših prenosnih naprav, spominske kartice in USB-ključke. Na Sliki 14 lahko vidimo v notranjost najbolj znanih predstavnikov.



Slika 14: Notranjost ključka USB 3.0 in SD-kartice

Vir: Ravenperch, 2012 in Leva, 2009

Imajo pomanjkljivost pri prepisovanju celic, saj glede na kakovost zdržijo omejeno število prepisov. To je najbolj opazno pri cenenih modelih. Večina jim ima mikrokrmilnik, ki je zadolžen za komunikacijo, že dolgo pa imajo vgrajen mehanizem za izenačevanje obrabe. Prevečkrat prepisane celice označi za slabe in jih ne uporablja več. V potrošniškem rangu deluje tako preprosto, da uporabljene celice, ki se ne prepisujejo, pusti pri miru, industrijski način pa uporabi vse zdrave celice. Če je v uporabi, podatke premakne drugam, tako da vse celice spravi na isto obrabo. Ta način se uporablja tudi v nadaljnjih SSD-diskih. Nekateri ceneni mediji ne uporabljajo izenačevanja obrabe, zato se lahko zgodi, da nekaterih celic ni več mogoče prepisati. Zanje je priporočljiva uporaba datotečnega sistema, prilagojenega za NAND-pomnilnike. Ti sami skrbijo za pravilno obrabo, vendar jih podpira bolj malo operacijskih sistemov.

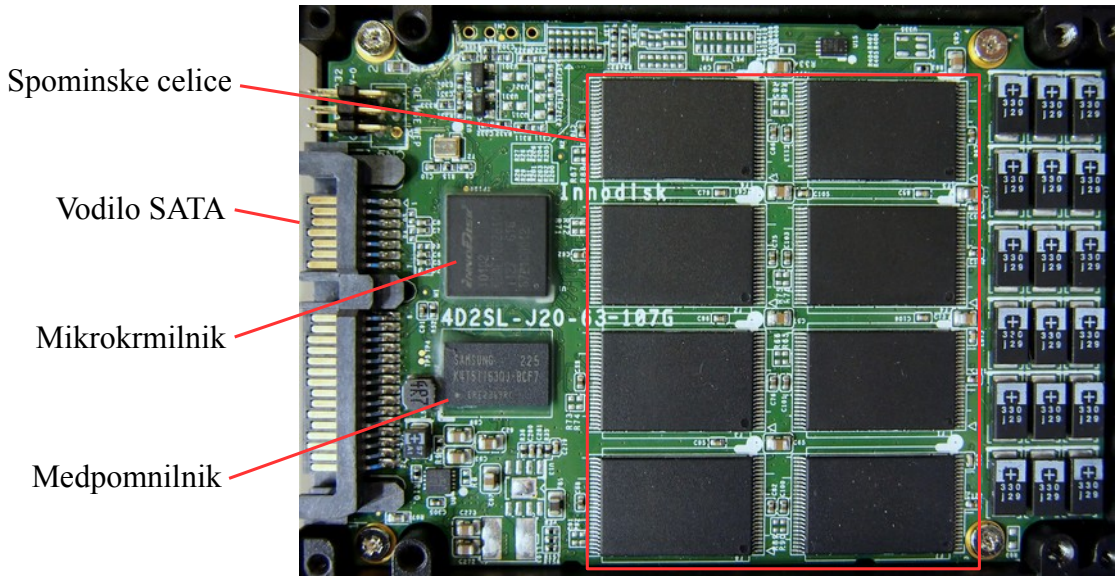
Pred pametnim upravljanjem podatkov, velikih manj kot 1 GiB, je bila pogostejša napaka izraba celic. Danes se dogajajo redkeje, in to v izjemno poceni medijih, saj mikrokrmilnik ni narejen za popravljanje celic. Podatke še vedno lahko preberemo, le nekaterih celic ne moremo več prepisati, zato se lahko na novo shranjene datoteke pokvarijo.

Pri moderni kompleksnosti se pogosteje pokvari mikrokrmilnik. Z nenavadnim obnašanjem lahko predčasno opazimo simptome, ki napovedujejo odpoved. Če ne glede na položaj podatkov naprava deluje počasi ali pa če občasno ni prepoznavna, je to prvi znak.

Po odpovedi lahko poskusimo s hlajenjem medija v zamrzovalniku, le da pazimo na kondenz. Če to ne uspe, je treba zamenjati mikrokrmilnik. Strokovnjaki uporabljajo univerzalnega, mogoče pa je tudi branje vsake spominske celice ločeno.

5.4 DISKI BREZ PREMIKAJOČIH SE DELOV (SSD)

Razvoj NAND-tehnologije je privedel do nove oblike, ki temelji na zmogljivosti računalniškega pomnilnika. S pomočjo izredno majhnih spominskih celic lahko podatke preberemo takoj in simultano. Glede na cenovno območje se celice delijo po velikosti. Najhitrejše se imenujejo SLC, shranijo le en bit, so hitrejše, bolj odporne in dražje. Drugi obliki sta MLC z dvema bitoma in TLC s tremi biti, obe pa sta cenovno dostopnejši.
Znani so bili po spominskem učinku, ker v prepisanih celicah ostaja zgodovina in jih to tudi upočasni. Na Sliki 15 je prikazan modernejši SSD-disk z dodanim medpomnilnikom za samodejno vzdrževanje.



Slika 15: Notranjost SSD-diska
Vir: Prirejeno po: Springer, 2014

Približno tri leta po izidu SSD-diskov so začeli množično vgrajevati funkcije čiščenja in varnega brisanja (trim ali discharge). Ukazi sporočijo sektorje, ki niso več v uporabi, disk pa jih doda na čakalno vrsto za izpraznitev, ko ni obremenjen. To deluje samo na novejših vmesnikih, kot so SCSI, AHCI, NVMe. Še nekaj drugih podpira ukaze, saj so zmožnost podedovali s prepuščanjem ukazov podprtega vmesnika (povzeto po: Gubanovis in Afonin, 2012).

Uporaba teh ukazov je odvisna od operacijskega sistema. Windows 7 ter novejši ga uporabljajo privzeto, in to pri praktično vsem, kjer je mogoče. Večina distribucij Linux privzeto tedensko sporoči vse nezasedene sektorje, lahko pa ročno vklopimo takojšnje poročanje.
Pri poškodovanem datotečnem sistemu zaradi varnosti noben operacijski sistem ne pošilja ukazov. Vsak razlog, da ukazi niso delovali, je že dovolj, da lahko rešujemo z enako zanesljivostjo kot pri trdih diskih. Strokovnjaki na zelo visoki ravni pa imajo v teoriji še več možnosti z zgodovino. Strošek reševanja se sicer zelo podraži, saj gre tukaj za dražjo storitev med strokovnjaki (povzeto po: Gubanovis in Afonin, 2014).

Če želimo, da SSD-disk deluje z optimalno hitrostjo, ga je treba vzdrževati, ker prepisane celice drastično izgubljajo hitrost. Po nekaj letih uporabe brez čiščenja nastanejo celo slabi sektorji. Bolj odporni so postali z dodatnimi tehnologijami, kombiniranimi z izenačevanjem obrabe. Deluje tako, da nove podatke vpiše na sektor, ki je bil najmanj obremenjen, če je mogoče, pa na izpraznjenega. Tako za običajno vzdrževanje zadošča nekaj odstotkov prostega diska. Posledično se podatki zelo zmešajo, tabelo preusmeritev pa pozna le mikrokrmilnik.
Nekaj modelov uporablja medpomnilnik, ki zadrži zapisovanje podatkov. To je uporabno, če se v kratkem ponovno prepišejo, hkrati pa lahko med tem čisti celice. Na ta način se vzdržujejo tudi brez podpore čiščenja sistema, vendar napolnjen medpomnilnik pomeni upočasnitev pisanja. Najpreprostejša metoda za vzdrževanje ima več fizičnega prostora, kakor ga prikaže, tako da ima vedno na voljo očiščen prostor.
Ena izmed kompleksnejših inovacij za čiščenje je tehnika s stiskanjem vsebine. Ta disku omogoči dodaten prostor, a se odzivni čas poveča, njegova surova hitrost pa je odvisna od podatkov. Če so ti stisljivi oziroma ponovljivi, bo hitrost visoka, pri že stisnjenih pa je zapisovanje veliko počasnejše (povzeto po: Layton, 2011).

Če povzamemo, so bili starejši modeli preprosti in odprti za reševanje podatkov, vendar je bila njihova življenjska doba omejena skupaj s padajočo hitrostjo, moderni pa s samodejnim vzdrževanjem hudo otežujejo zmožnosti reševanja podatkov. Funkcija čiščenja je na začetku delovala tako, da se je podatke takoj po tem še vedno dalo prebrati, saj niso bili počiščeni takoj.

Kmalu je čiščenje postalo varnejše, tako da sektorji na čakanju vrnejo prazno vrednost, ker mikrokrmilnik sektorje izbriše s seznama preusmeritev. Pred tem se je pri brisanju diska še po nekaj urah dalo dobiti nekaj podatkov, zdaj pa je lahko dostop onemogočen že po nekaj sekundah. Podatki še vedno so nekje, če pa ga pustimo na napajanju, jih bo počasi brisal. Še večja težava je nastala s samodejnim šifriranjem podatkov, ki ob varnem brisanju celotnega diska še zamenja ključ. Takšne podatke lahko pravilno vrne le še proizvajalec. Če šifriranja ni, lahko strokovnjaki delajo s postopkom mrtvega mikrokmilnika.

Z vso moderno kompleksnostjo v elektroniki je ta najbolj podvržena odpovedi, celice pa redkeje dosežejo kritično mejo. V takem primeru lahko strokovnjaki poskušajo popraviti morebitno programsko napako v mikrokrmilniku. Če pri tem niso uspešni, lahko le še odstranijo vse spominske čipe, preberejo podatke, iz njih naredijo analizo algoritma za izenačenje obrabe in pravilno sestavijo podatke. Tehnologija SSD-diskov se razlikuje med proizvajalci, zato je priporočeno, da ima podjetje za reševanje izkušnje s proizvajalcem vašega diska.

5.5 GLOBINSKA ANALIZA

V teoriji je mogoče pridobiti prepisane podatke s trdega diska, v praksi pa je to skoraj nemogoče. Če želimo v današnjem času dosegati te podatke, se moramo spopasti z dvema težavama: količino podatkov z njihovo fizično gostoto in minimalno celovitostjo sektorja. Jakost magnetnega zapisa lahko odstopa, to se pojavi po prepisu. Če je bil pol prepisan z enakim, bo postal močnejši, če z nasprotnim, pa postane šibkejši od povprečja.

Dokler je bil prostor za zapis enega bita dovolj velik, da motnje s sosednjih polj niso predstavljale nevarnosti slabljenja, sta se na diske zapisovali le dve stanji. Pri tej metodi je bilo dovolj branje absolutne magnetne vrednosti vsakega sektorja, izhodni podatki so bili tako nekaj desetkrat večji.

Leta 2005 pa se je skupaj z diski za prenosnike na trgu uveljavil nov način magnetnega zapisovanja, ki temelji na spreminjanju pola. Logično podatki ostajajo v bitih, vendar so tokrat zapisani tako, da jih medsebojna motnja ne prizadene. Poleg obeh polj je na mestu bita lahko celoten prehod na drugi pol. Bralna glava zazna to spremembo in vrednost bita je pol, na katerega je prešel. To vrednost zadrži za naslednje bite do nove spremembe.

Pri tem načinu shranjevanja postane globinska analiza kompleksnejša. Pri prejšnji tehnologiji je zadostoval AD-ojačevalnik s prilagodljivim krmilnikom, danes pa bi komaj

dobili uporabne podatke. Druga težava nastane, ko imajo zapisani podatki definiran največji dovoljeni presledek med prehodi, zato so podatki sektorja šifrirani, za uspešno dešifriranje pa moramo pravilno prebrati minimalno 96 odstotkov sektorja. Pravilna metoda za branje novega zapisa bi bila z magnetnim mikroskopom, vendar je ta zelo počasna. Za celoten disk bi porabili več let, izhodni podatki pa bi bili približno tisočkrat večji (povzeto po: Šinkovec, 2011).

Študija je pokazala, da je pridobivanje podatkov s to metodo zelo nezanesljivo. Na novem disku, katerega površina je še popolna, je bilo s pomočjo magnetnega mikroskopa v besedilu mogoče prepoznati zgolj nekaj besed, preostalo je bil nesmisel, pri naslednjem prepisu diska pa so rešeni podatki postali popolnoma nerazpoznavni (povzeto po: Wright et. al., 2008).

Če upoštevamo, da je magnetni zapis po disku zapisan z različno gostoto in dotrajanostjo materiala in da so stari podatki imeli različno oslabljeno jakost, lahko zatrdimo, da takšnih podatkov ni mogoče povrniti. Tudi če nam bo v prihodnosti uspelo iznajti način, bo magnetni zapis razpadel do te mere, da odstopanja zaradi prejšnje vrednosti ne bo mogoče zaznati.

V začetkih SSD-diski niso podpirali vzdrževanja in so vse podatke le prepisovali, zato je nastal tako imenovan spominski učinek. Nato smo z varnim brisanjem dobili možnost izbire. Lahko smo žrtvovali hitrost in njegovo življenjsko dobo ter tako zadržali prepisane podatke. Dandanes pa se proizvajalci trudijo diskom dodajati tehnologije, ki jim omogočijo samostojno vzdrževanje in hkrati brisanje starih podatkov.

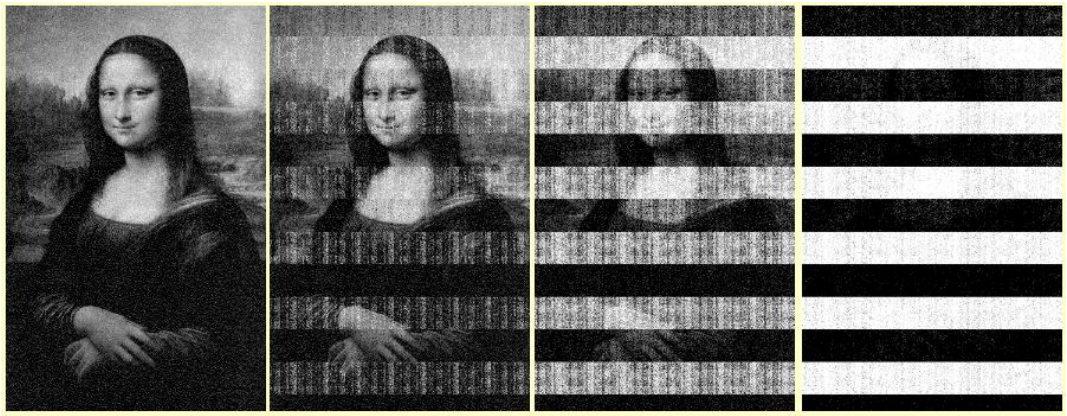
V zlati dobi SSD-forenzike so lahko spominske čipe prebrali z zelo natančnim orodjem in pridobili že nekajkrat prepisane podatke, ker vsebina celic sloni na jakosti, tako kot pri magnetnem zapisu, le da je veliko bolj obstojna. Manj bitov ko je shranjeno v celico, več zgodovine lahko zadrži. Če pa so podatki že bili pobrisani s čiščenjem, potem niti ta metoda ne pomaga. Tudi če se čiščenje ne izvaja, bi bili podatki hudo razmetani.

5.6 POMNILNIK

Včasih je treba rešiti informacije, ki se še niso shranile na trdi disk, na primer zaradi sesutja pred ali med shranjevanjem. Ti podatki so ostali v pomnilniku, do njih pa še lahko dostopamo.

Če ni razlog strojni izpad, lahko izkoristimo ranljivost vmesnika FireWire, saj omogoča neposreden dostop do pomnilnika, vendar ga vse redkeje srečamo. Mogoče je uporabiti tudi orodje za izvoz celotnega pomnilnika, vendar tako tvegamo prepis nekaj podatkov. Ponavadi je poškodovan posnetek spomina programa v praksi neuporaben, vendar lahko s programom Mimikatz pridobimo vsaj gesla.

Mogoča je tudi drugačna metoda. Ob izklopu se namreč podatki ne izgubijo takoj, izgubljajo se linearno. Pod običajnimi pogoji traja največ osem minut, z zamrzovanjem pa lahko te podatke hranimo več ur. Tako je mogoče izvesti napad hladnega zagona (angl. *ColdBoot*), ki z zamenjavo zagonske naprave po kratkem stiku računalnika izvozi pomnilnik, vendar tako tvegamo delni razpad podatkov, kot jih vidimo na Sliki 16. Ti so bili zaseženi prek DDR2-pomnilnika po 5 sekundah, 30 sekundah, 60 sekundah in 300 sekundah. Idealnih razmer ni mogoče doseči, saj ob zagonu osnovne plošče na kratko prekine napajanje pomnilnika in izgubi nekaj bitov. Obstojnost podatkov je odvisna od operativne napetosti pomnilnika, pri novejših je nižja in razpadejo hitreje. Ta napad je mogoče preprečiti z vsiljenim testiranjem pomnilnika ob zagonu, ki prepiše vsebino (povzeto po: Kovačič, 2013).
Pred kratkim so v javnost prišli podatki o novi tehnologiji z dodanim procesorjem na osnovni plošči za šifrirane pomnilnika, to je tehnologija, predstavljena s procesorji AMD Ryzen™ Pro in Epyc. Zdaj je le še vprašanje časa, kako dolgo bo napad pomnilnika še mogoč.



Slika 16: Rekonstrukcija v napadu Cold Boot zasežene slike
Vir: Halderman et. al., 2008, 6

5.7 VEZAVA DISKOV

V RAID-u povežemo samostojne enake diske, da zagotovimo boljšo hitrost ali zanesljivost. Priporočljivo je, da so diski enake velikosti, sicer vezava upošteva velikost najmanjšega. Nekateri krmilniki vezave lahko neizkoriščen prostor uporabijo za kombinacijo vezav. Podobno lahko storimo tudi z razdelki enake velikosti na programski ravni. Strojno prepleteni diski delujejo neodvisno, programske pa upravlja gostitelj in zato potrebuje delovne vire.

RAID 0: To je edina konfiguracija, ki ne vključuje redundance. Zaradi posledic okvar je dobil vzdevek strašljivi RAID. Podatki so med sektorji prepleteni, s pa tem omogočena podvojena hitrost branja in pisanja, zato ne sme biti odstopanj. Velikost vezave se sešteva in je odvisna od najmanjšega diska.
Ob okvari le enega diska nam ostanejo hudo razrezani podatki. Tudi če nam jih uspe rekonstruirati, bi pridobili zelo malo uporabnih podatkov. Razlog je prikazan in razložen v poglavju 2.3 POŠKODOVANI PODATKI na strani 6.

RAID 1: Ta je najpreprostejši, najbolj zanesljiv in deluje, dokler je živ poslednji disk. Vsi podatki so v celoti prekopirani na vse vezane diske. Za razliko od drugih vezav lahko diske pozneje dodajamo. Hitrost branja in odzivni čas sta povečana, saj lahko bere iz kateregakoli diska, hitrost pisanja pa je omejena z najpočasnejšim.

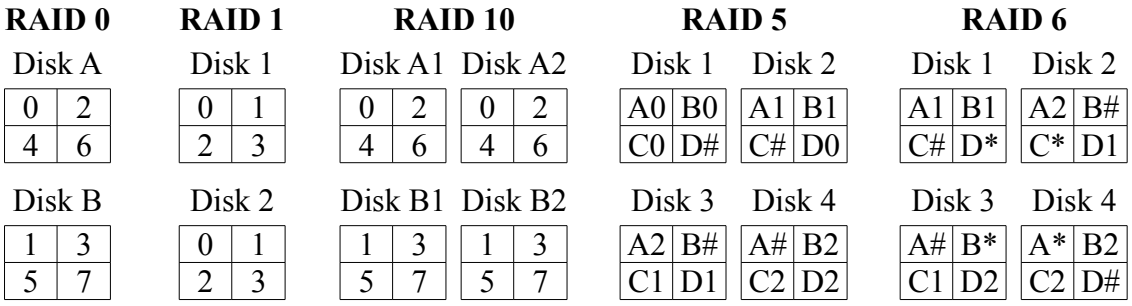
RAID 10: Da ohranimo enostavnost ter združimo varnost s hitrostjo, lahko mešamo polji 0 in 1 v vezavo RAID 10. Nižje ležeče polje 1 podatke varnostno kopira, nato pa se sklopi združijo v polju 0. Pri rekonstrukciji se morajo podatki le prekopirati in operator ne potrebuje računske moči.

RAID s paritetami: Da bi zagotovili hitrost, varnost in izkoristek prostora na istem številu diskov, je bilo treba podatke še bolj preplesti. To stori s pomočjo parnosti, ki prešteje bite na istem mestu med diski in v parnega zapiše podatek za sodo ali liho. V enačbi lahko manjka le en operand, da po izpadu izračuna manjkajoči del. Ta princip se je začel z RAID 2 in 3 z diskom samo za parnost, vendar se ju zaradi omejitev ne uporablja več. Nadaljevalo se je z RAID 4 z boljšo razporeditvijo parnosti na disku. Odzivnost pri obnovi je bila izboljšana, njegova uporaba pa je bila vse redkejša (povzeto po: http://en.wikipedia.org/wiki/RAID).

Danes sta najpogostejša RAID 5 in 6, saj parnost razporedita v bloke po vseh diskih. Zaradi enakomerne obremenitve med obnovo vezavi dovoljujeta skoraj normalno delovanje. Logično uprizorjena razlika med polji je vidna na Sliki 17.

RAID 5 uporablja enojno parnost in dovoljuje izpad enega diska, zato uporabe ne priporočajo pri kritičnih podatkih. Med obnovo je treba prebrati podatke na vseh preostalih diskih, odkritje slabih sektorjev pa že pomeni izgubo. Poleg tega obstaja večja možnost okvare še enega diska, preden je obnova dokončana.

Varnejši je RAID 6, ki pa uporablja dvojno parnost z zahtevnejšim algoritmom in prenese okvaro dveh trdih diskov. Če izpade več podatkov, kot jih vezava zmore obnoviti, je teoretično mogoče rešiti dobre bloke podatkov, podatki v izgubljenih blokih pa ostanejo polovična uganka. Na primer, pri RAID 5 z izgubo dveh parnih blokov ugotovimo le, da so posamezni biti sodi ali lihi. Tako je kombinacija vsakega posameznega bita omejena na dva namesto na štiri.



Slika 17: Prikaz najpogostejših oblik RAID

Vir: Prirejeno po: https://www.prepressure.com/library/technology/raid (4. 7. 2017)

6 SKLEP

Ugotovili smo, da se večina uporabnikov in običajnih kupcev računalnikov ne zaveda posledic izgube podatkov, dokler niso seznanjeni s tem ali dokler podatkov dejansko ne izgubijo. Ena izmed večjih napak kupcev diskov je, da želijo čim več prostora za čim manj denarja, kar pa po okvari obžalujejo. Razlog nerazgledanosti tiči v posploševanju uporabe računalnikov. Takrat se je stanje začelo slabšati. V začetkih računalnika je bil delež tehnično podkovanih oseb večji, ker je bilo delo z računalnikom zapleteno, dandanes pa prevladujejo uporabniki, ki znajo uporabljati zgolj nekaj programov, pa naj bo to za delo ali zabavo.

Sčasoma je bilo mogoče opaziti manjšanje izgub podatkov pri domačih uporabnikih, saj se vse več vsakodnevnih storitev izvaja na oblaku. Čeprav se nemalo ljudi vpraša, ali je varno našo zasebnost zaupati družbenim omrežjem, je realnost ta, da podatki tam ostanejo tudi po okvari našega računalnika.

S povečanim medijskim izpostavljanjem o vdorih in izsiljevalskih virusih se je raven izdelave varnostnih kopij še dodatno dvignila. 80 odstotkov oseb, ki smo jim predstavili pomembnost varnostnih kopij, se je odločilo, da so njihovi podatki vredni storitve, preostali pa računalnik uporabljajo za nepomembna opravila.

Glede na natančnost tehnologije navadnih trdih diskov je zanimivo, kako dobro jim danes uspe preprečevati trke glave na površino. Čeprav je bil pred desetimi leti razmik med elementi večji in gostota zapisanih podatkov manjša, so bile okvare pogostejše. Še zanimivejša pa je statistika večjega nabora diskov, ker brez fizične poškodbe več diskov odpove v času treh mesecev kot pa v letu dni obratovanja. To velja tako za navadne diske kot tudi za SSD-diske, kar nam pove, da brez varnostnih ukrepov nikoli nismo zares varni.

V prihodnosti lahko pričakujemo vse več okvar modernega časa, kot so bolj kompleksni mikrokrmilniki v SSD-diskih in helij namesto zraka v trdih diskih. Čeprav se bodo s tem ukvarjali strokovnjaki, bo te ovire mogoče opaziti pri cenah reševanja.

Pri preizkušanju programov sem ugotovil, da od mnogo razpoložljivih, tudi plačljivih programov le peščica zmore uspešno sestaviti celoten seznam iz drobcev pri rahlo prepisani vsebini, s pomočjo kombinacije brezplačnih programov pa dobimo nepregledne podatke. Lahko pa si pomagamo s preizkusnimi različicami plačljivih programov, da vidimo strukturo in rešimo za nas najpomembnejše podatke.

7 VIRI IN LITERATURA

ACS Forensics. *Data Recovery On A 3TB Seagate Hard Drive* (online). 2016. (citirano 3. 7. 2017). Dostopno na naslovu: https://acsdata.com/data-recovery-3tb-seagate-hard-drive/

Amos, E. *Laptop hard drive exposed* (online). 2013. (citirano 21. 7. 2017). Dostopno na naslovu: https://commons.wikimedia.org/wiki/File:Laptop-hard-drive-exposed.jpg

CGSecurity. *Photorec* (program). 2015. (citirano 23. 7. 2017). Dostopno na naslovu: http://www.cgsecurity.org/wiki/PhotoRec

CGSecurity. *Testdisk* (program). 2015. (citirano 15. 7. 2017). Dostopno na naslovu: http://www.cgsecurity.org/wiki/TestDisk

Convar. *PC Inspector File Recovery* (program). 2005. (citirano 30. 7. 2017). Dostopno na naslovu: http://www.pcinspector.de/default.htm?Language=1

Crystal Dew World. *CrystalDiskInfo* (program). 2017. (citirano 6. 7. 2017). Dostopno na naslovu: http://crystalmark.info/software/CrystalDiskInfo/index-e.html

Data Medics Data Recovery. *How to Clone a Hard Drive With Bad Sectors Using ddrescue* (online). 2015. (citirano 8. 7. 2017). Dostopno na naslovu: https://www.data-medics.com/forum/how-to-clone-a-hard-drive-with-bad-sectors-using-ddrescue-t133.html

Desmond, B. *The difference between booting MBR and GPT with GRUB* (online). 2012. (citirano 14. 6. 2017). Dostopno na naslovu: https://www.anchor.com.au/blog/2012/10/the-difference-between-booting-mbr-and-gpt-with-grub/

Diaz Diaz A. (2017). *GNU ddrescue Manual* (online). 2017. (citirano 11. 7. 2017). Dostopno na naslovu: https://www.gnu.org/software/ddrescue/manual/ddrescue_manual.html

Diaz Diaz, A. *GNU ddrescue* (program). 2016. (citirano 11. 7. 2017). Dostopno na naslovu: https://www.gnu.org/software/ddrescue/

GetData. *Recover My Files* (program). 2017. (citirano 30. 7. 2017). Dostopno na naslovu: http://www.recovermyfiles.com/

Gubanovis, Y. in Afonin, O. *Recovering Evidence from SSD Drives in 2014: Understanding TRIM, Garbage Collection and Exclusions* (online). 2014. (citirano 31. 3. 2017). Dostopno na naslovu: https://articles.forensicfocus.com/2014/09/23/recovering-evidence-from-ssd-drives-in-2014-understanding-trim-garbage-collection-and-exclusions

Gubanovis, Y. in Afonin, O. *Why SSD Drives Destroy Court Evidence, and What Can Be Done About It* (online). 2012. (citirano 31. 3. 2017). Dostopno na naslovu: https://articles.forensicfocus.com/2012/10/23/why-ssd-drives-destroy-court-evidence-and-what-can-be-done-about-it/

Halderman, J. A., et. al. *Lest We Remember: Cold Boot Attacks on Encryption Keys* (online). 2008. (citirano 6. 12. 2016). Dostopno na naslovu: http://citpsite.s3-website-us-east-1.amazonaws.com/oldsite-htdocs/pub/coldboot.pdf

Kato, B. *Restoration* (program). 2003. (citirano 30. 7. 2017). Dostopno na naslovu: http://www.snapfiles.com/get/restoration.HTML

Kovačič, M. *Napad na delovni pomnilnik* (online). 2013. (citirano 6. 12. 2016). Dostopno na naslovu: https://slo-tech.com/clanki/13002/

Kovačič, M. *Šifriranje nosilcev podatkov v okolju Linux in Windows* (online). 2007. (citirano 6. 12. 2016). Dostopno na naslovu: https://slo-tech.com/clanki/07003/

Kovačič, M. *Trajno brisanje podatkov* (online). 2009. (citirano 6. 12. 2016). Dostopno na naslovu: https://slo-tech.com/clanki/09002/

Layton, J. B. *On-the-fly Data Compression for SSDs* (online). 2011. (citirano 3. 7. 2017). Dostopno na naslovu: http://www.linux-mag.com/id/7869/

Leurs, L. *RAID* (online). 2017. (citirano 4. 7. 2017). Dostopno na naslovu: https://www.prepressure.com/library/technology/raid

Leva, F. *SD memory card internals 2 GB* (online). 2009. (citirano 27. 7. 2017). Dostopno na naslovu: https://commons.wikimedia.org/wiki/File:SD_memory_card_internals_2_GB.jpg

Pinheiro, E., et. al. *Failure Trends in a Large Disk Drive Population* (online). 2007. (citirano 28. 6. 2017). Dostopno na naslovu: https://static.googleusercontent.com/media/research.google.com/en//archive/disk_failures.pdf

Piriform. *Recuva* (program). 2016. (citirano 30. 7. 2017). Dostopno na naslovu: https://www.piriform.com/recuva

R-Tools. Technology Inc. *R-Studio* (program). 2017. (citirano 30. 7. 2017). Dostopno na naslovu: http://www.r-studio.com/

Ravenperch, *USB 3.0 Flash Drive PCB* (online). 2012. (citirano 27. 7. 2017). Dostopno na naslovu: https://commons.wikimedia.org/wiki/File:USB_3.0_Flash_Drive_PCB.jpg

Runtime. *GetDataBack Simple* (program). 2017. (citirano 30. 7. 2017). Dostopno na naslovu: https://www.runtime.org/data-recovery-software.htm

Shao, Z. *Lecture 17: Files and Directories* (online). 2016. (citirano 24. 6. 2017). Dostopno na naslovu: http://flint.cs.yale.edu/cs422/lectureNotes/L17.pdf

Shipley, T. G. in Door, B. *Forensic Imaging of Hard Disk Drives- What we thought we knew* (online). 2012. (citirano 31. 3. 2017). Dostopno na naslovu:

https://articles.forensicfocus.com/2012/01/27/forensic-imaging-of-hard-disk-drives-what-we-thought-we-knew-2/

Springer, T. *Embedded World 2014: Innenleben eines Innodisk-Industrial SSD-Laufwerk* (online). 2014. (citirano 1. 8. 2017). Dostopno na naslovu: https://commons.wikimedia.org/wiki/File:Embedded_World_2014_SSD.jpg

Šinkovec, A. *Reševanje podatkov s trdih diskov* (online). 2011. (citirano 28. 6. 2017). Dostopno na naslovu: http://eprints.fri.uni-lj.si/1373/1/Sinkovec1.pdf

Wikipedia. *Advanced Format* (online). 2017. (citirano 22. 6. 2017). Dostopno na naslovu: https://en.wikipedia.org/wiki/Advanced_Format

Wikipedia. *Case sensitivity* (online). 2017. (citirano 20. 6. 2017). Dostopno na naslovu: https://en.wikipedia.org/wiki/Case_sensitivity

Wikipedia. *Disk partitioning* (online). 2017. (citirano 16. 6. 2017). Dostopno na naslovu: https://en.wikipedia.org/wiki/Disk_partitioning

Wikipedia. *ext3* (online). 2017. (citirano 24. 6. 2017). Dostopno na naslovu: https://en.wikipedia.org/wiki/Ext3

Wikipedia. *File Allocation Table* (online). 2017. (citirano 21. 6. 2017). Dostopno na naslovu: https://en.wikipedia.org/wiki/File_Allocation_Table

Wikipedia. *GUID Partition Table* (online). 2017. (citirano 14. 6. 2017). Dostopno na naslovu: https://en.wikipedia.org/wiki/GUID_Partition_Table

Wikipedia. *HFS Plus* (online). 2017. (citirano 25. 6. 2017). Dostopno na naslovu: https://en.wikipedia.org/wiki/HFS_Plus

Wikipedia. *NTFS* (online). 2017. (citirano 23. 6. 2017). Dostopno na naslovu: https://en.wikipedia.org/wiki/NTFS

Wikipedia. *RAID* (online). 2017. (citirano 4. 7. 2017). Dostopno na naslovu: http://en.wikipedia.org/wiki/RAID

Wikipedia. *S.M.A.R.T.* (online). 2017. (citirano 28. 6. 2017). Dostopno na naslovu: https://en.wikipedia.org/wiki/S.M.A.R.T.

Wright, C.,et al. *Overwriting Hard Drive Data: The Great Wiping Controversy* (online). (citirano 1. 8. 2017). Dostopno na naslovu: http://www.vidarholen.net/~vidar/overwriting_hard_drive_data.pdf

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Diplomant: **Andrej Ficko**,

Rojen: **31. 3. 1994** v kraju **Ljubljana**,

i z j a v l j a m,

da sem diplomsko delo z naslovom

REŠEVANJE PODATKOV, KAJ LAHKO STORIMO

izdelal in napisal samostojno.

Kranj, avgust 2017 Podpis študenta

ZAŠČITA DIPLOMSKEGA DELA

A. Copyright **c**: - diplomant.....podpis

Kopiranje in vsakršen drug način razmnoževanja v celoti ali posameznih delov nista dovoljena brez predhodnega pisnega dovoljenja nosilcev te pravice.

B. Glede na Zakon o avtorski in sorodnih pravicah (UL RS št. 21/95, 9/01, 30/01, 85/01, 43/04, 58/04, 94/04, 17/06, 44/06, 139/06, 16/07) in Zakon o industrijski lastnini (UL RS št. 13/92, 13/93, 27/93, 34/97, 75/97) velja še naslednje:

Diplomsko delo – arhivski izvod si je možno ogledati samo v prostorih knjižnice Šolskega centra Kranj, in sicer s pisnim dovoljenjem:

avtorja – diplomanta

diplomant.....podpis avtorja – diplomanta

Če ni avtorjevega – diplomantovega podpisa, je diplomsko delo v knjižnici Šolskega centra Kranj nedostopno za vpogled.

Pojasnilo k točki B:

Lastnoročni podpis avtorja – diplomanta dovoljuje ogled diplomskega dela le v knjižnici Šolskega centra Kranj,

Brez lastnoročnega podpisa avtorja – diplomanta ogled diplomskega dela ni možen.

Kranj, avgust 2017