

**Offline duplikátor disků postavený na platformě
ARM**
Offline Disk duplicator Built on ARM Platform

Marek Odstrčil

Bakalářská práce
2015



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2014/2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Marek Odstrčil**
Osobní číslo: **A12119**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Bezpečnostní technologie, systémy a management**
Forma studia: **prezenční**

Téma práce: **Offline duplikátor disků postavený na platformě ARM**
Téma anglicky: **An Offline Disk Duplicator, Built on the ARM Platform**

Zásady pro vypracování:

1. Specifikujte oblasti použití duplikátoru disků.
2. Specifikujte hlavní požadavky na duplikátor disku.
3. Navrhněte duplikátor disku pro bezpečnou duplikaci disku s použitím platformy ARM.
4. Proveďte sestavení a implementaci duplikátoru.
5. Analyzujte výkon a funkce duplikátoru.
6. Porovnejte Váš duplikátor s produkty dostupnými na trhu.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. UPTON, Eben. Raspberry Pi: uživatelská příručka. 1. vyd. Brno: Computer Press, 2013, 232 s. ISBN 978-80-251-4116-8.
2. MATTHEW, Neil a Richard STONES. Linux: začínáme programovat. Brno: Computer Press, 2008, 829 s. ISBN 978-80-251-1933-4.
3. VÁŇA, Vladimír. ARM pro začátečníky. Praha: BEN – technická literatura, 2009, 195 s. ISBN 978-80-7300-246-6.
4. KAMENÍK, Pavel. Příkazový řádek v Linuxu: praktická řešení. Vyd. 1. Brno: Computer Press, 2011, 224 s. ISBN 978-80-251-2819-0.
5. BÍBR, Ivan. Ubuntu 10.04 CZ: praktická příručka uživatele Linuxu. Vyd. 1. Brno: Computer Press, 2010, 366 s. ISBN 978-80-251-3121-3.
6. LASSER, Jon. Rozumíme UNIXu. Vyd. 1. Praha: Computer Press, 2002, 252 s. ISBN 80-722-6706-X.
7. PINKER, Jiří. Mikroprocesory a mikropočítače. 1. vyd., 1. dot. Praha: BEN – technická literatura, 2008, 159 s. ISBN 978-80-7300-110-0.

Vedoucí bakalářské práce:

Ing. David Malaník, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce:

6. února 2015

Termín odevzdání bakalářské práce:

3. června 2015

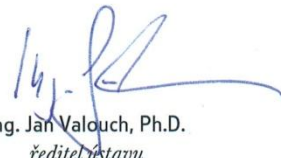
Ve Zlíně dne 6. února 2015



doc. Mgr. Milan Adámek, Ph.D.
děkan



L.S.



Ing. Jan Valouch, Ph.D.
ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

ABSTRAKT

Tato bakalářská práce je zaměřená na zabezpečení důkazů v digitální formě pomocí forenzních postupů s využitím duplikátoru disků. V první části je popsáno, jak duplikátor disků funguje, jaké jsou jeho hlavní vlastnosti, požadavky a v jakých oblastech najdeme jeho uplatnění. V této části jsou také vzaty v úvahu požadavky na duplikátor. V druhé části se nachází souhrn použitých přístrojů, nezbytných k chodu duplikátoru, postup sestavení duplikátoru, analýza výkonu a funkcí duplikátoru v porovnání s duplikátory dostupnými na trhu. Praktická část je opatřena fotodokumentací pro větší přehlednost o jednotlivých částech.

Klíčová slova: zabezpečení, duplikátor, forenzní, digitální data, ARM

ABSTRACT

This thesis is focused to secure evidence in digital formats using forensics procedures by using disk duplicator. The first part describes how disk duplicator works, what are its main features, requirements and in what areas we can find its application. In this section there are also taken into account the requirements for duplicator. The second part is a summary of used equipment, necessary for running a duplicator, a duplicator assembly process, analysis of performance and features compared to the duplicator of duplicators available in the market. The practical part is provided photographs for greater clarity on the individual parts.

Keywords: security, duplicator, forensic, digital data, ARM

Chtěl bych poděkovat všem, kteří mi s vypracováním této bakalářské práce pomáhali a byli mi oporou. Především vedoucímu práce panu Ing. Davidu Malaníkovi, Ph.D. za vedení a pomoc při tvorbě a za zapůjčení všech potřebných pomůcek potřebných k dokončení této práce.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 DUPLIKÁTOR DISKŮ	11
1.1 VÝBĚR SPRÁVNÉHO DUPLIKÁTORU DISKŮ	11
1.1.1 Velikost pracovní stanice duplikátoru.....	12
1.1.2 Frekvence používání	12
1.1.3 Kompatibilita disků.....	12
1.1.4 Možnost kompletního vymazání disku	13
1.1.5 Další zařízení.....	13
2 OBLASTI POUŽITÍ DUPLIKÁTORU DISKŮ	15
2.1 FORENZNÍ ANALÝZA DIGITÁLNÍCH DAT	15
2.2 PŘEDMĚTY ZKOUMANÉ ANALÝZOU	15
2.3 POSTUP PŘI PROVÁDĚNÍ ANALÝZY	16
2.3.1 Identifikace důkazů	16
2.3.2 Sběr důkazů.....	16
2.3.3 Interpretace informací	16
2.4 PŘÍSTROJE POTŘEBNÉ K ZAJIŠTĚNÍ DAT	17
2.4.1 Softwarové vybavení.....	17
2.4.2 Hardwarové přístroje.....	18
2.4.3 Ostatní přístroje	18
3 POŽADAVKY NA DUPLIKÁTOR	20
3.1 NÁROKY NA RYCHLOST PŘENOSU.....	20
3.1.1 Technologie disků	21
3.2 VÝPIS INFORMACÍ DO LOGOVÉHO SOUBORU	22
3.2.1 Hashovací funkce	22
3.2.2 Praktické využití hashovacích funkcí.....	23
3.3 OCHRANA PROTI ZÁPISU NA ZDROJOVÝ DISK.....	23
3.4 VSTUP A VÝSTUP DUPLIKACE	23
3.5 POUŽITÍ LINUXU.....	24
4 NOVÉ TECHNOLOGIE	25
II PRAKTICKÁ ČÁST	26
5 NÁVRH DUPLIKÁTORU DISKŮ	27
5.1 POUŽITÝ HARDWARE	27
5.2 MIKROPOČÍTAČ RASPBERRY PI MODEL B.....	27
5.3 3,2 TFT DISPLEJ.....	28
5.4 KABELÁŽ K PŘIPOJENÍ PERIFERIÍ	29
5.5 ÚLOŽIŠTĚ PRO OPERAČNÍ SYSTÉM.....	29
5.6 CÍLOVÉ A ZDROJOVÉ DISKY	29
5.7 OPERAČNÍ SYSTÉM	30
5.8 EXTERNĚ NAPÁJENÝ ROZBOČOVAČ USB, Y KABEL.....	31
6 SESTAVENÍ A IMPLEMENTACE DUPLIKÁTORU	33

6.1	OŽIVENÍ A INSTALACE OS	33
6.1.1	Nahrání image na SD kartu	33
6.1.2	Spuštění Operačního systému	34
6.2	OŽIVENÍ DISPLEJE	35
6.3	ÚKONY PŘED SPUŠTĚNÍM PROGRAMU DUPLIKACE	35
6.3.1	Automatické spuštění po startu	35
6.3.2	Vypnutí nabídky při vložení média	36
6.4	PROGRAM K DUPLIKACI	36
6.4.1	První možnost kopírování souborů	37
6.4.2	Druhá možnost kopírování image	37
6.5	VZHLED PROGRAMU	38
7	ANALÝZA VÝKONU A FUNKCE DUPLIKÁTORU	40
7.1	TESTOVÁNÍ RYCHLOSTI PŘENOSU	40
7.1.1	Micro sd karta	40
7.1.2	USB Flash Disk	41
7.1.3	Pevné disky	42
7.2	VYHODNOCENÍ RYCHLOSTI PŘENOSU	43
7.2.1	Přenosové rychlosti při duplikaci z flash disku na SD kartu a pevný disk	43
7.2.2	Přenosová rychlost při duplikaci z SD karty na pevný disk SATA	43
7.2.3	Přenosová rychlost při duplikaci pevný disk na pevný disk	43
7.3	TABULKOVÝ PŘEHLED RYCHLOSTÍ	43
7.4	FUNKCE DUPLIKÁTORU	44
8	POROVNÁNÍ S PRODUKTY NA TRHU	45
8.1.1	Levnější varianta	45
8.1.2	Dražší varianta	45
8.2	POROVNÁNÍ S FORENZNÍMI DUPLIKÁTORY	46
8.2.1	Logicube Forencis Falcon	46
8.2.2	Tableau TD2u	47
	ZÁVĚR	49
	SEZNAM POUŽITÉ LITERATURY	50
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	52
	SEZNAM OBRÁZKŮ	53
	SEZNAM TABULEK	54
	SEZNAM PŘÍLOH	55

ÚVOD

Elektronika, jako mobilní telefony, počítače, tablety, televize jsou pro nás v dnešním světě věci, bez kterých bychom si nedokázali představit fungovat. Jsou největším zdrojem přenosu informací. V televizi jsou to zprávy, na počítači a mobilním telefonu komunikace s našimi blízkými prostřednictvím sociálních sítí, ukládání poznámek a mnoho dalších věcí. Důležitým aspektem je ovšem někdy tyto data chránit proti nechtěnému napadení, ovšem někdy se takovéto incidenty stanou a je potřeba je řešit.

Pokud je brán tento čin jako trestný, je potřebné, aby bylo zahájeno pátrání po pachateli a to za pomoci všech možných prostředků služby kriminální policie, která tyto incidenty vyšetřuje. Díky rostoucímu počtu kriminální činnosti v oblasti digitální sféry jsou zapotřebí stále nové prostředky ke sběru důkazů digitálních dat.

Takovýmto zařízením může být duplikátor, o němž je napsána tato bakalářská práce. Tento duplikátor slouží právě pro sběr digitálních důkazů a pro další zpracování pro forenzní účely, jako je předložení důkazů o provedeném trestném činu ve formě poznatků z digitálních dat.

Jelikož jsou tyto duplikátory využívány nejen v kriminální oblasti, ale také v komerční sféře, je jejich překážkou mnohdy jejich cena. Tato práce se proto zabývá návrhem a implementací duplikátoru sestaveného z levných částí založených na mikropočítači RaspberryPi, který má využití i díky své velikosti a ceně v mnoha dalších věcech, jako například multimediální centrum u televize, zabezpečovací ústředna a v neposlední řadě je také používán pro výuku programování ve školách.

I. TEORETICKÁ ČÁST

1 DUPLIKÁTOR DISKŮ

Pro vytvoření kopie pevných disků je možných několik postupů. Pokud ovšem potřebujeme nejrychlejší cestu k vytvoření kopie disku se všemi soubory, i těmi skrytými, je duplikátor disků nejlepší volbou. Je nejrychlejší a nejjednodušší cestou, jak takovouto kopii vytvořit. Takováto zařízení jsou především vhodná pro větší firmy, kde je zapotřebí buďto rychlá záloha systému, nebo vytvoření nové kopie disku bez nutnosti instalace OS. Duplikátory jsou většinou konstruovány pro zápis na více cílových úložišť zároveň pro větší rychlost. K duplikátoru není zapotřebí připojovat jakýkoliv počítač, který by řídil jeho funkci, ani žádný složitý software, který by bylo nutno složitě instalovat a nastavovat. Duplikátory navíc nabízí funkce formátování disků a porovnání integrity dat, což je porovnání shody dat na zdrojovém a cílovém disku. Zdrojový disk je nazván jako master, tento disk obsahuje informace potřebné k duplikaci. [1]



Obr. 1. Duplikátor pro 5 cílových zařízení [1]

1.1 Výběr správného duplikátoru disků

Veškeré duplikátory nabízejí základní funkci duplikace na jedno cílové zařízení. Ovšem při výběru je nutno zohlednit další aspekty. [1]

1.1.1 Velikost pracovní stanice duplikátoru

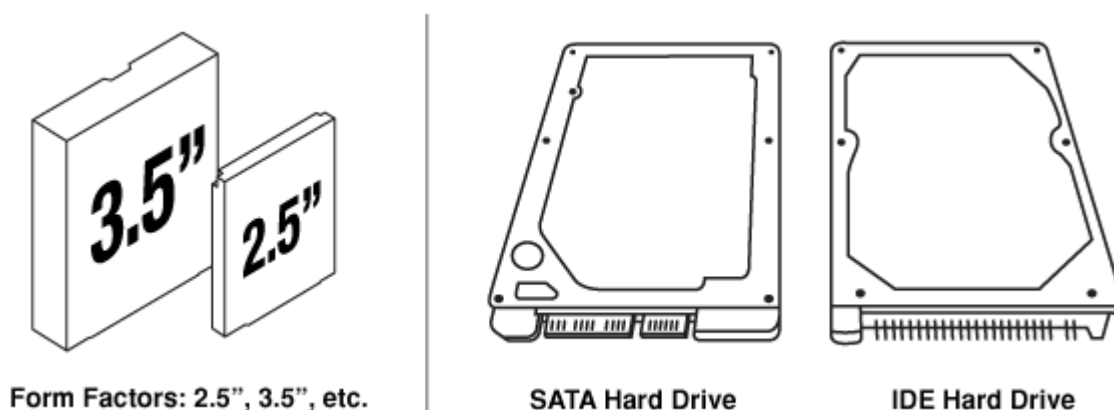
Pro většinu využití postačí zařízení s jedním cílovým diskem, ovšem pro firmy zabývajícími se informačními technologiemi a servisní IT střediska je takovýto přístroj nedostačující. Proto existují duplikátory pro určitý počet cílových zařízení, kde zápis probíhá na všechna zařízení záraz. Toto použití je například vhodné pro sestavení nových počítačů bez nutnosti instalovat na každý individuálně OS se všemi nezbytnými doplňky, jako ovladače a jiné programy nutné k bezproblémovému chodu stanice. [2]

1.1.2 Frekvence používání

Pro občasné používání je výhodnější pořízení jednodušších duplikátorů, které jsou levnější. Hodí se především pro malé podniky. Pro časté nebo trvalé jsou vhodné zařízení s vyšší přenosovou rychlostí a větším množstvím funkcí. Tyto duplikátory se používají především pro požadavky státních potřeb a jako forenzní duplikátory, kde je zvýšený důraz na bezpečnost zdrojového disku. [1]

1.1.3 Kompatibilita disků

Důležitým faktorem je i kompatibilita disků s duplikátorem. Jelikož existuje několik druhů disků, jak do velikosti, tak konektivitou, je nutné podle tohoto kritéria volit zařízení. Pevný disk může mít různé rozměry. Nejčastěji se udávají v jednotkách palců, což značí největší délku zařízení. Nejčastější používané velikosti jsou 2.5" a 3.5". Co se týče konektivity, nejpoužívanějšími zařízeními jsou disky s rozhraním IDE, SCSI a SATA. SCSI jsou disky, které vyžadují vysokou spolehlivost a proto se používají v serverech. Zařízení SATA bylo vyvinuto jako nástupce zařízení IDE, které je nejčastěji používané zařízení v počítačích. [3]



Obr. 2. Ukázka rozlišení velikosti a konektivity disků [1]

1.1.4 Možnost kompletního vymazání disku

Kompletní vymazání dat z disku je proces, kdy je disk několikrát přehrán náhodnými daty, aby bylo zajištěno kompletní a nenávratné smazání dat z pevného disku za účelem zneužití. Tato možnost se využívá u disků, které se vyřazují a jsou na ně kladena přísná kritéria zabezpečení. Tyto kritéria se týkají například státních orgánů, kde jsou na discích uložena citlivá data. [1]

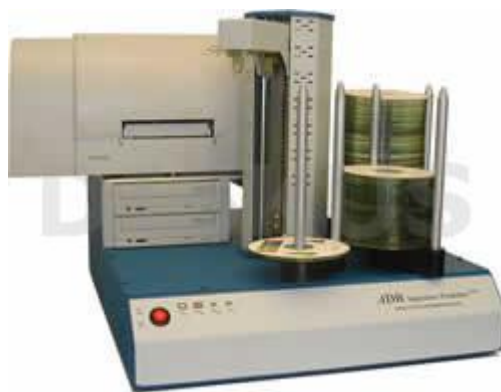
1.1.5 Další zařízení

Dalším kritériem je výběr duplikátorů, které umí vedle pevných disků duplikovat i optické disky, flash disky a paměťové karty. Nejčastějším z těchto duplikátorů jsou duplikátory optických disků, které slouží pro rychlé vytváření kopií nosičů software.



Obr. 3. Duplikátor Flash disků

[4]



Obr. 4. Duplikátor optických disků [5]

2 OBLASTI POUŽITÍ DUPLIKÁTORU DISKŮ

Duplikátory disků mají využití jak v domácnostech, tak malých i velkých podnicích, kde slouží buďto jako zálohovací stanice, nebo jako zařízení pro vytváření kopií pro účely rychlejšího nasazení nebo opravy počítačů v celém objektu. Největší uplatnění má ovšem toto zařízení v IT firmách, kde je při sestavování počítačových sestav nezbytné rychle vytvářet kopie disků pro určité modely s předem instalovaným OS a softwarem. Dalším a velmi významným oborem z hlediska bezpečnosti pro využití duplikátoru disku je forenzní analýza digitálních dat, kde může být jakékoliv digitální médium důkazem, či vodítkem k rozluštění mnoha případů.

2.1 Forenzní analýza digitálních dat

Jedná se o odvětví forenzních věd. Pokud dojde k vraždě nebo vloupání, je nutné prozkoumat místo činu, aby bylo možné sestavit rekonstrukci dané události. Pro tuto rekonstrukci je nutný sběr důkazů, jako pořízení fotografií místa činu, sejmutí otisků prstů a dalších aspektů nutných k nalezení viníka daného zločinu. [6]

Na stejném principu funguje i sběr dat z digitálních médií pro účely sejmutí důkazů, které jsou uloženy na médiích uvedených v předchozí kapitole. Tímto odvětvím se zabývá forenzní analýza digitálních dat. Sběr důkazů digitálních dat je ovšem oproti sběru hmatatelných důkazů velmi náchylný na změnu. Pro tento případ musí být se samotným nosičem nakládáno velmi opatrně a to jak při samotném sběru dat, tak při transportu či archivaci média. Tato věda se neustále přizpůsobuje novým trendům, například s příchodem chytrých mobilních telefonů a tabletů, je nutné prověřovat i tato média. [6]

V minulosti existovaly analýzy počítačů a počítačových systémů, což je zkoumání celého počítače jako celku. Forenzní analýza digitálních dat se oproti analýze počítačů a počítačových systémů zabývá přímo digitálními daty. [6]

2.2 Předměty zkoumané analýzou

Zkoumají se především tato zařízení:

- Počítače.
- Notebooky.
- Mobilní telefony.
- Fotoaparáty.

- Datová úložiště.
- Servery.
- Pevné disky.
- Flash disky.
- Paměťové karty.
- Servery.

Příklady typů dat ke zkoumání:

- Databáze (My SQL).
- Certifikáty, elektronické podpisy.
- Korespondence (Facebook, Skype).
- Komunikace mobilních telefonů (protokol hovorů, SMS).
- Internetové obchody. [7]

2.3 Postup při provádění analýzy

Při postupu je nutno dodržet určitých postupů, je důležité prozkoumat digitální data s cílem určit, co se stalo, jak se to stalo a koho se to týká. Jedná se tedy o metodu podobnou při sběru fyzických důkazů. [6]

2.3.1 Identifikace důkazů

Jedná se o počáteční krok, kde je nutné identifikovat média, na kterých by se mohli nacházet důležitá data. Je nutno podotknout, že digitálním důkazem není médium, ale informace v něm uložené. [8]

2.3.2 Sběr důkazů

V tomto kroku je důležité takzvaně vyextrahovat data na jiné médium, a to za účelem zachování všech dat na důkazním médiu. [6]

2.3.3 Interpretace informací

Jedná se o nejdůležitější část, kde je nutno nalézt data, která jsou vodítkem k předložení důkazů. Další velmi důležitou věcí při prozkoumávání dat je zaznamenávání všech potřebných údajů, jak tedy bylo s daty zacházeno. Tento krok je důležitý například u soudního řízení, kde jsou důležité i ty nejmenší podrobnosti. [8]

Dalším důležitým úkolem při interpretaci je převést tato data do srozumitelného formátu. Tento převod je prováděn z důvodu prezentování před soudem, kde by čistým datům zajisté většina nerozuměla. Důležitými částmi tohoto převodu nejsou technické poznámky, ale závěry: kdo je zodpovědný za data, co bylo odcizeno a jaké jsou následky. [6]

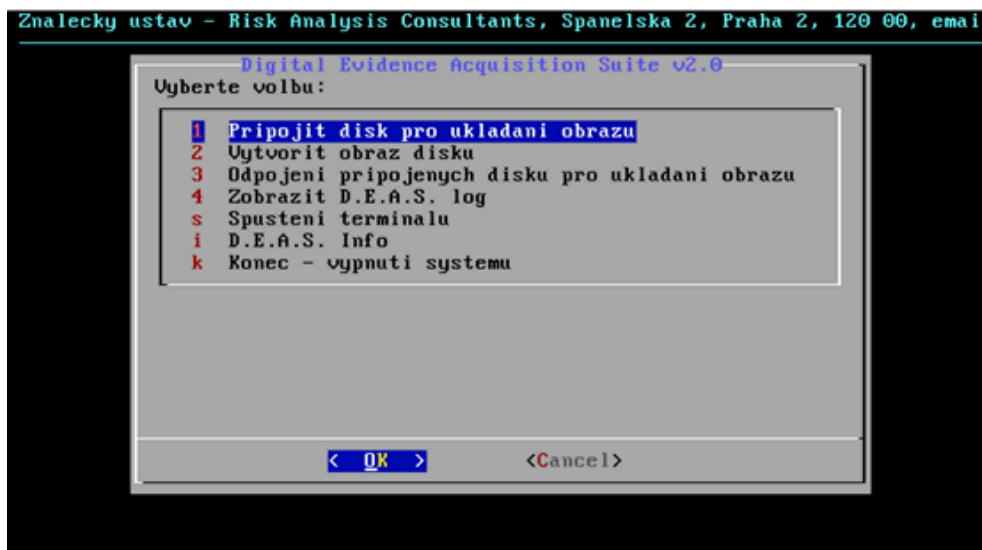
2.4 Přístroje potřebné k zajištění dat

Tyto přístroje lze rozdělit na tři kategorie:

- Softwarové.
- Hardwarové.
- Ostatní. [9]

2.4.1 Softwarové vybavení

Jedná se o softwarové doplňky k hardwarovým přístrojům, mohou nabízet odlišné funkce z hlediska výrobce.



Obr. 5. Distribuce software pro forenzní získávání dat [9]

Na obrázku můžeme vidět příklad softwarového vybavení. Jedná se o Linuxovou distribuci Debian, kde prostředí tvoří jednoduché menu. Toto konkrétní softwarové vybavení je navíc vyvinuto tak, aby u něj vyšetřovatel neprovedl chybu. Jakákoliv chyba, například záměna cílového a zdrojového disku, může znamenat fatální následek při řešení případu.

Jednoduchost je cílená zejména pro obsluhu, kdy není nutné složité školení a není zapotřebí znát další určité parametry programu, které by mohli zmást obsluhu. [9]

2.4.2 Hardwarové přístroje

Existuje velké množství hardwarových přístrojů k zajištění dat při forenzní analýze digitálních dat. Nejpotřebnějším náčiním je především samotný duplikátor a vedle něj blokátor.



Obr. 6. Hardwarové vybavení pro forenzní analýzu dat [9]

Na obrázku můžeme vidět obsah hardwarového vybavení, kde je vedle duplikátoru a blokátoru také další zařízení, jako čtečka paměťových karet a napájecí zdroj. Blokátor disků slouží proti nechtěnému zápisu na zdrojový disk.

Jedná se o nejzákladnější sadu vybavení, používanou pro běžné účely forenzního získávání dat. Celá sada je uložena v odolném boxu, který chrání přístroje před povětrnostními vlivy, pádem, a dalšími vlivy.

Dalším volitelným vybavením může být například redukce na velikostí disků, kompaktní fotoaparát a ostatní rozšířené vybavení. [9]

2.4.3 Ostatní přístroje

Jako nezbytnými potřebami pro získání důkazů mohou být vedlejší přístroje jako kabeláž, nářadí, psací potřeby a jiné nezbytné pomůcky k zajištění materiálu.



Obr. 7. Ostatní vybavení pro zajištění důkazů [10]

3 POŽADAVKY NA DUPLIKÁTOR

Pro účely forenzního duplikátoru je potřeba, aby byl duplikátor dostatečně rychlý, měl dostatečnou konektivitu, výpis všech podstatných postupů při duplikaci a ochranu proti zápisu na zdrojové médium.

3.1 Nároky na rychlost přenosu

Aby bylo možné získat co nejvyšší přenosovou rychlost, je nutností využívání nejmodernějších trendů v oblasti informačních technologií. Nejvyužívanějšími technologiemi pro přenos jsou dnes zařízení s konektivitou USB a SATA.

Tab. 1. Rozdíly ve standardu USB 2.0 a USB 3.0 [11]

Rozhraní	USB 3.0	USB 2.0
Přenosová rychlost	5 Gbit/s (600 MB/s)	480 Mbit (60 MB/s)
Vodiče	Čtyři datové vodiče USB 3.0 + 2 datové vodiče USB 2.0 + 2 vodiče napájení	Dva datové vodiče + dva napájecí vodiče
Úsporné funkce	Několik režimů, úspora i s připojeným zařízením	Pouze odpojeno, připojeno
Napájení	Max 900 mA	Max 500 mA

Z výše uvedené tabulky je patrné, že nový standard má mnohem vyšší přenosovou rychlost, než jeho starší verze. Přibyly zde i nové úsporné funkce a velkým významným vylepšením je proudový odběr, což je důležitý parametr pro možnosti připojení zařízení s větším proudovým odběrem. Například pevný disk o velikosti 2.5" může mít odběr okolo hodnoty 500 mA (podle výrobce). Takovýto disk není schopný standard USB 2.0 rozběhnout a je za potřebí externí napájení, což je další překážkou v používání. [12]



Obr. 8. Rozdíly v konektoru USB 3.0 a USB 2.0 [13]

Tab. 2. Rozdíly ve standardu SATA 3G a SATA 6G [11]

Rozhraní	SATA 3G	SATA 6G
Přenosová rychlost	3 Gbit/s	4,8 Gbit/s
Reálné maximum	300 MB/s	600 MB/s

Z tabulky je opět patrný nárůst přenosové rychlosti novějšího standardu SATA 6G (SATA III) oproti starší verzi SATA 3G (SATA II). SATA II ovšem dostačuje svou rychlostí na maximální rychlost mechanických disků. SATA III přišla v roce 2009 pro potřeby nových pevných disků zvaných SSD, které mají mnohanásobně vyšší přenosovou rychlost než jejich mechanické obdoby. [11]

3.1.1 Technologie disků

Jak již bylo popsáno v předchozí části, na trh se dostávají stále nové technologie. V dnešní době už můžeme zakoupit zařízení nazývané SSD, které oproti svým starším mechanickým diskům disponují nespočetnými výhodami, jako je vyšší přenosová rychlost, žádné mechanické části, vyšší odolnost vůči otřesům či pádům, menší proudový odběr a mnoho dalších parametrů. Tyto SSD disky pracují na principu ukládání dat pomocí NAND čipů. Nevýhodou těchto disků je omezený počet cyklů zápisů a prozatím vyšší cena, než je tomu u mechanických disků.



Obr. 9. Porovnání mechanického a SSD disku [13]

3.2 Výpis informací do logového souboru

Důležitou částí při postupu duplikace je výpis důležitých dat do logového souboru, kde budou zapsány místa, časy, názvy a další nezbytné informace, které budou sloužit pro lepší přehled a zavedení do databázových systémů. Nezbytnou částí je kontrola integrity dat, což je porovnávání dat na cílovém i na zdrojovém disku. Pro tyto metody jsou zejména vhodné hashovací funkce.

3.2.1 Hashovací funkce

Hashovací funkce složí k vytvoření otisku dat, hesel a textů, která nechceme, aby byla rozluštna. Jsou to tedy takzvaně jednosměrné funkce. Vstupem této hashovací funkce může být libovolně dlouhý text, kde na výstupu bude vytvořený hash s pevnou délkou. Tento hash se nazývá jednosměrný, právě z důvodu, že jej nelze zpětně dešifrovat a tudíž z něj nelze vyčíst původní zprávu. Tyto funkce slouží k ověření integrity, což je kontrola správnosti původních dat oproti zkopírovaným. Pokud je změněn například jediný znak v textu, musí být hodnota obou hashů, jak hashe původního textu tak kopírovaného jiná.

Nejpoužívanějšími hashovacími funkcemi jsou SHA a MD5, které se liší počtem bitů.

SHA-x, kde x je počet bitů, například SHA-256 je hash o délce 256 bitů. SHA-0 nebo SHA-1 jsou hashe, které tvoří hash o délce 160 bitů. Naproti tomu MD5 stačí pro vytvoření hashe 128 bitů.

- MD5 hash jméno: b8e6b611bba846b5cd4d665b049fb0d2
- MD5 hash jména: 17050ab9cf668b8a85d6f59fa48ab39b

Z příkladu hashe MD5 se slovy jméno a jména můžeme vidět, že jejich otisky i přes pouhou změnu jednoho písmena jsou naprosto odlišná. Tento hash je reprezentován pomocí čísla v šestnáctkové soustavě, kde nám pro vyjádření jednoho čísla stačí 4 bity, které reprezentují celkem 16 znaků $2^4=16$. Těchto znaků je v hashi MD5 32, výsledkem je tedy násobek délky hashe 32 a násobkem počtu bitů 4 tedy 128 bitů.

- SHA-1 hash jméno: 689905226213AF150238A087625E3C5BD36748B8
- SHA-1 hash jména: 68C9D37E98E935D950F9675987F3DD201BFC984B

Při dalším příkladu můžeme vidět, že rozdíl v hashi je patrný. Avšak první dvě čísla jsou stejná. I přesto, že je pravděpodobnost vygenerování stejného hashe velmi málo prav-

děpodobná, je doporučeno provést oba typy hashů, jak MD5, tak SHA-1, kde je pravděpodobnost již tak malá, že ji můžeme zanedbat. Druhou možností jak této kolizi předejít je hashovací funkce SHA s větším počtem bitů. [14]

3.2.2 Praktické využití hashovacích funkcí

V praxi se vyskytují tyto hashovací funkce zejména pro porovnávání hesel, či dlouhých textů nebo, jak již bylo zmíněno, integrity dat. Například při stažení jakéhokoliv velkého software je možné zjistit, jestli je tento hash na stránkách vydavatele uveden, jestli je soubor neporušen anebo jestli nenese žádné další informace, například škodlivý software. [14]

3.3 Ochrana proti zápisu na zdrojový disk

Ochranou proti zápisu na zdrojový disk může být buďto software, anebo hardware v podobě blokátoru disků. U hardwarové blokace je nutné provedení podle konektivity daného zařízení. Na následujícím obrázku je znázorněn blokátor proti zápisu pro USB a paměťové karty. [9]



Obr. 10. Blokátor USB a paměťových karet [9]

3.4 Vstup a výstup duplikace

Vstupem duplikace je zdrojový disk, který by měl být chráněn proti změně původních dat, jestli je to technicky možné. Druhou vstupní podmínkou je možnost verifikace dat, tedy vytvoření jejich otisku.

Výstupem by měl být disk, který nese obraz zdrojového disku s kontrolním součtem a ostatními daty potřebnými k archivaci.

3.5 Použití Linuxu

Pro duplikátory, jak již bylo popsáno výše je používán OS založený na Linuxovém jádře. Toto řešení je prováděno z bezpečnostního hlediska. Linux je totiž považován oproti převážně užívanému systému Windows bezpečnější. Prvním faktorem ovlivňující tuto skutečnost je, že Linuxové distribuce jsou takzvaně „open-source“, což znamená, že jejich šíření je zadarmo a není nijak omezen zásah do zdrojového kódu tohoto operačního systému. Oproti tomuto je například hojně využívaný OS Windows zajištěn licencí proti jakýmkoliv úpravám OS a jeho dalším šířením.

Z hlediska bezpečnosti je v tomto případě rozumnější volbou OS založený na Linuxu bezpečnější, protože kód OS, který není volně šiřitelný je stálý a je mnohem snazší nalézt v něm jakékoliv mezery v zabezpečení. Pokud je naopak tento kód změněn, je u každého uživatele tento kód individuální a není tak snadné, aby pronikl do dalších zařízení.

Dalším faktorem pro bezpečnost jsou aktualizace. Jelikož většina uživatelů OS Windows nevlastní legální verzi OS a po nainstalování nestahují aktualizace, stává se jejich bezpečnost počítač jako otevřené dveře do bytu. Zatímco skrze fakt, že Linux je OS systém zdarma stažitelný, nebojí se uživatelé stahovaných aktualizací a udržují tímto systém bezpečný.

4 NOVÉ TECHNOLOGIE

Se stále se vyvíjejícími technologiemi stoupá i využití těchto technologií v různých typech průmyslu. Novým trendem ve sféře elektroniky se stává architektura ARM, která je využita v zařízeních jako mobilní telefony, tablety, chytré televize a mnoho další elektroniky význam zkratky ARM nese název Advanced RISC Machine. Zakladatelem této technologie je společnost ARM Holdings, která v dnešní době procesory nevyrábí, ale pouze navrhuje. Nespornou výhodou těchto procesorů je jejich poměrně vysoký výkon vzhledem k ceně a spotřebě elektrické energie.

Rozdílem oproti x86 (procesory vyráběné firmou Intel, který se používá ve většině osobních počítačích, nese ARM naprosto odlišnou instrukční sadu a nemůže pracovat se stejnými příkazy jako x86 procesor, protože tyto příkazy jednoduše nezná. Tím pádem je nutné použití odlišných OS, podporující tyto procesory.

Hlavním rozdílem v ceně těchto dvou procesorů je, že x86 technologie jsou používány dvěma společnostmi kromě Intelu, což jsou výrobci procesorů AMD a VIA, kteří si zakoupili licenci Intelu a mohou s těmito instrukčními sadami pracovat a vyvíjet je. Kdežto ARM tyto licence prodává daleko většímu procentu výrobců, než je tomu u Intelu a tím pádem nejsou výrobci nuceni vyvíjet vlastní řešení procesorů, čímž se sníží výrobní náklady. [20]

Jelikož ARM procesory umějí používat některé verze Linuxového OS, je tudíž možné aby měli zastoupení i v oblasti duplikátorů disků. Jedním z možných zařízení je možné použít mikropočítač postavený právě na architektuře ARM s názvem RaspberryPi, který je schopen pracovat se standardy jako je USB a je tedy schopen pracovat s pevnými disky, flash disky a jinými zařízeními.

II. PRAKTICKÁ ČÁST

5 NÁVRH DUPLIKÁTORU DISKŮ

V praktické části jsem se rozhodl navrhnout a implementovat duplikátor. Popíši, co všechno bude k duplikátoru potřeba před jeho samotným sestavením a na jakém hardwaru bude pracovat. Dále budu popisovat, jak jsem postupoval při samotné realizaci duplikátoru krok po kroku. Jako platforma bude použita deska, jejímž základem je ARM procesor, konkrétně mikropočítač s názvem Raspberry Pi Model B, který mi byl zapůjčen se vším příslušenstvím fakultou.

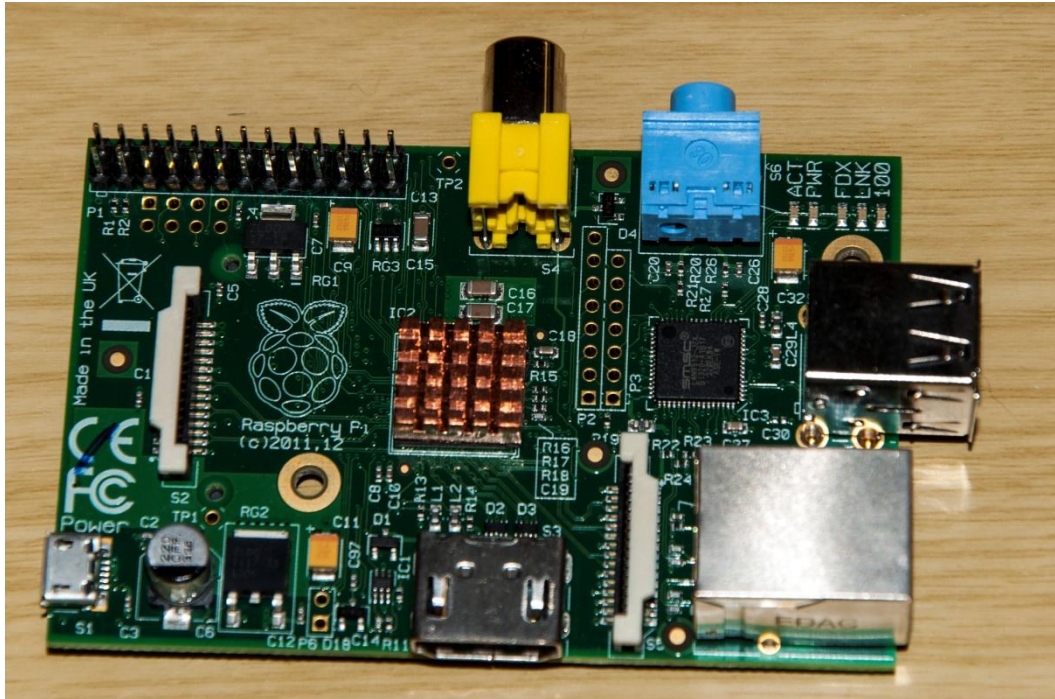
Úkolem praktické části je sestavit a naprogramovat tento mikropočítač tak, aby bylo možné ovládat operační systém pomocí malého dotykového displeje, pomocí kterého se bude spouštět program na duplikaci. Tento program bude provádět samotnou duplikaci. Dalším úkolem je otestovat rychlost přenosu a změřit čas úkonu.

5.1 Použitý hardware

Jak již bylo popsáno v předchozím odstavci, srdcem duplikátoru je mikropočítač založený na platformě ARM. Jedná se o mikropočítač Raspberry Pi britské společnosti Raspberry Pi Foundation, který má primárně sloužit jako interaktivní pomůcka při výuce informatiky ve školství. V domácnosti jej lze připojit k TV, či monitoru jako levnou variantu PC. Jeho velkou výhodou jsou oproti PC velmi malé rozměry a nespočetněkrát nižší spotřeba. Může tedy sloužit jako multimediální centrum, či prohlížeč internetu. [15]

5.2 Mikropočítač Raspberry Pi model B

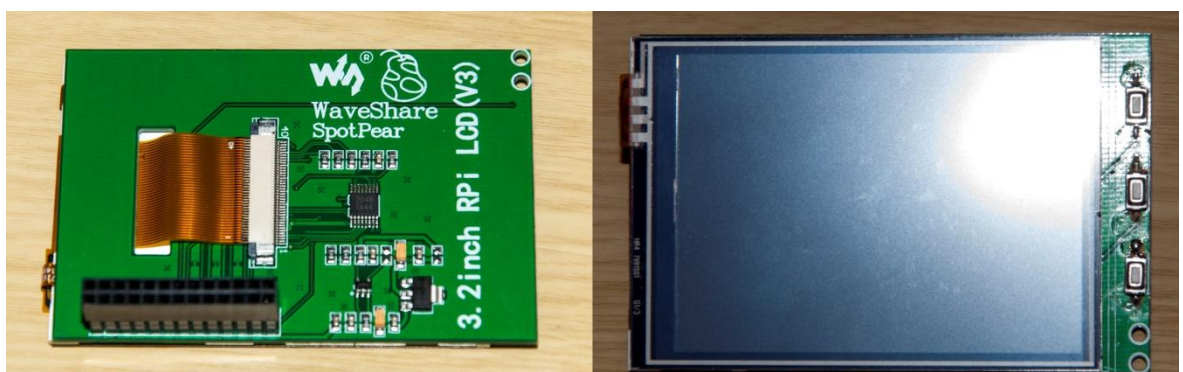
Pro účel duplikátoru mi byl zapůjčen mikropočítač Raspberry Pi model B, který má na desce dva USB porty, jeden RJ45 port pro připojení k síti, HDMI pro zobrazovací jednotku, slot pro paměťové karty typu SD, 3,5 mm výstup pro zvuk. Jako napájení je na desce konektor micro USB, tudíž je možné ho napájet buďto klasickou nabíječkou pro mobilní telefony, či powerbankou (externí baterie). Mikropočítač je poháněn procesorem s taktem 700 MHz, operační paměť nese hodnotu 512 MB SDRAM. Jako operační systém používá různé distribuce Linuxu. V našem případě budeme pracovat s OS Raspbian, který k výrobku dodává výrobce. [16]



Obr. 11. Detail desky Raspberry Pi model B

5.3 3,2 TFT Displej

Jako zobrazovací jednotka bude použit displej od firmy Wave Share s úhlopříčkou 3,2 palců. Na displeji bude zobrazen průběh duplikace. Displej se na desku připojuje pomocí SPI portu, který se nachází na předchozím obrázku v levém horním rohu. Pro tento displej je nutné nainstalovat přímo určený operační systém, který s tímto displejem spolupracuje. Displej je plně dotykový a lze je ovládat i stylusem, či jiným tupým předmětem. Jedná se o rezistivní dotykový displej.



Obr. 12. Displej 3,2 TFT

5.4 Kabeláž k připojení periferií

Jelikož psát cokoli na malém displeji by bylo nepraktické, je nutné k počítači připojit velký monitor, kde je čitelnost značně větší. Pro propojení slouží u Raspberry Pi HDMI kabel, který je v dnešní době u většiny monitorů již standardem. Jako napájecí zařízení slouží síťový adaptér se vstupem 230 V a výstupem 5V s konektorem micro USB. Je možné použít i takzvanou powerbanku, což je prakticky baterie uložena v pouzdře vybavena USB výstupem, která primárně slouží pro dobíjení telefonu.



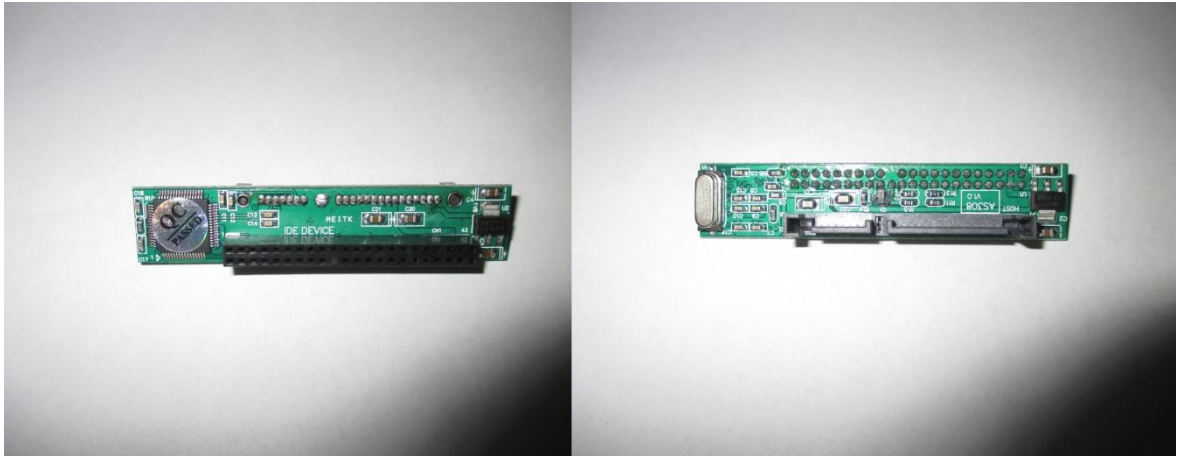
Obr. 13. Síťový adaptér a HDMI kabel

5.5 Úložiště pro Operační systém

Jako médium k uložení OS slouží SD karta, na kterou je nahrán příslušný OS. Taktéž je možné použít microSD kartu, která je dnes používána téměř v každém mobilním telefonu pomocí redukce na klasickou velikost SD karty.

5.6 Cílové a zdrojové disky

Pro tento účel mohou sloužit buďto Pevné disky, SD karty nebo Flash disky. Jelikož většina pevných disků ani SD karet není vybavena portem USB, ale SATA, či IDE portem a mikropočítač touto konektivitou není vybaven a to ani v nejvyšší řadě modelů, je nutno dokoupit redukce micro SD karty na USB, SATA na USB, IDE na SATA nebo diskové rámečky.



Obr. 14. Redukce IDE na SATA

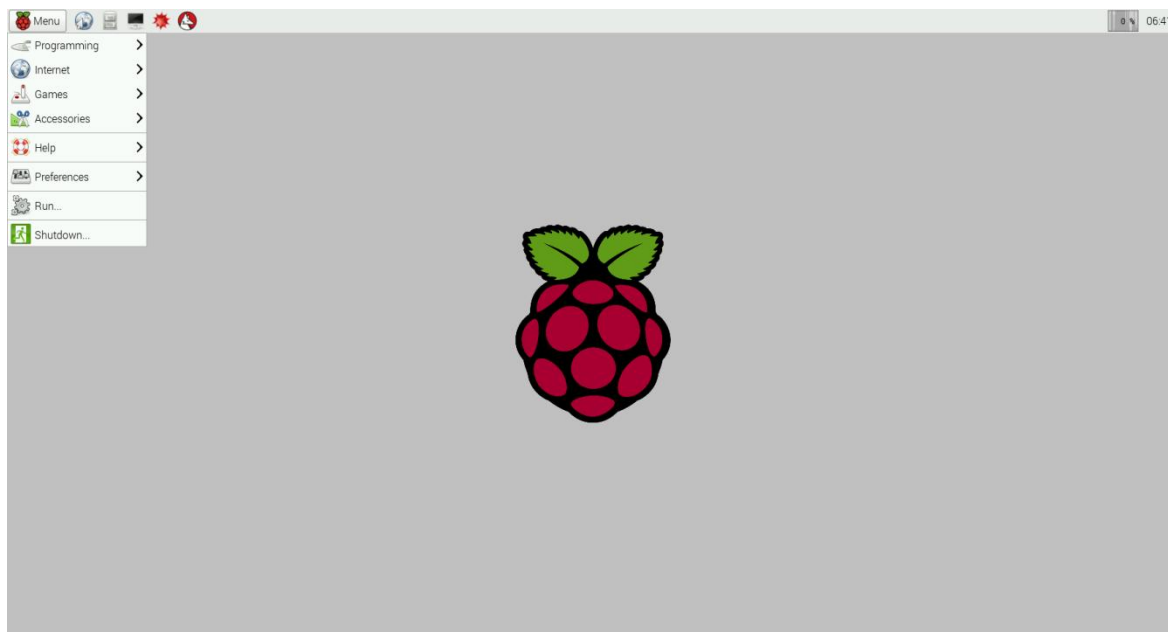


Obr. 15. Redukce SATA na USB

5.7 Operační systém

Duplikátor bude využívat již zmíněnou verzi Linuxu a to Raspbian, který je navrhnut přímo pro daný mikropočítač. Operační systém obsahuje grafické prostředí, tudíž práce v něm je snadná a přehledná. Jak lze vidět z obrázku, na první pohled připomíná OS Windows od světoznámého výrobce Microsoft. Oproti Windows je ale mnohem méně obsáhlejší. Najdeme zde jen několik základních nastavení, jako rozlišení obrazovky, obdoba příkazové řádky ve Windows s názvem Terminal. Žádné rozsáhlejší aplikace jako přehrávač videa hudby a jiné zde nenajdeme. Ovšem je možné tyto aplikace doinstalovat pomocí portálu raspberrypi store, kde je většina aplikací zdarma. Pro využití duplikátoru je použit

upravený OS Raspbian, který má v sobě již nainstalované ovladače na displej, připojený pomocí SPI. Jelikož je Linux takzvaný open source systém, je distribuován zadarmo. Tím se výrazně projevuje cena kompletu. [17]



Obr. 16. Grafické prostředí OS Raspbian [18]

5.8 Externě napájený rozbočovač USB, Y kabel

Jelikož Raspberry Pi neumožňuje na výstupu USB takový proudový odběr, aby bylo možné k němu připojit externí mechanický hard disk, je nutno připojit externě napájený USB HUB nebo Y kabel, který tento problém vyřeší. Problémem ovšem bude snížení rychlosti přenosu z důvodu připojení obou hard disků na jednu linku USB, což v případě této verze Raspberry Pi nevede, protože jsou i tak oba dostupné USB porty připojeny k jedné lince. [19]



Obr. 17. Externě napájený rozbočovač USB

6 SESTAVENÍ A IMPLEMENTACE DUPLIKATORU

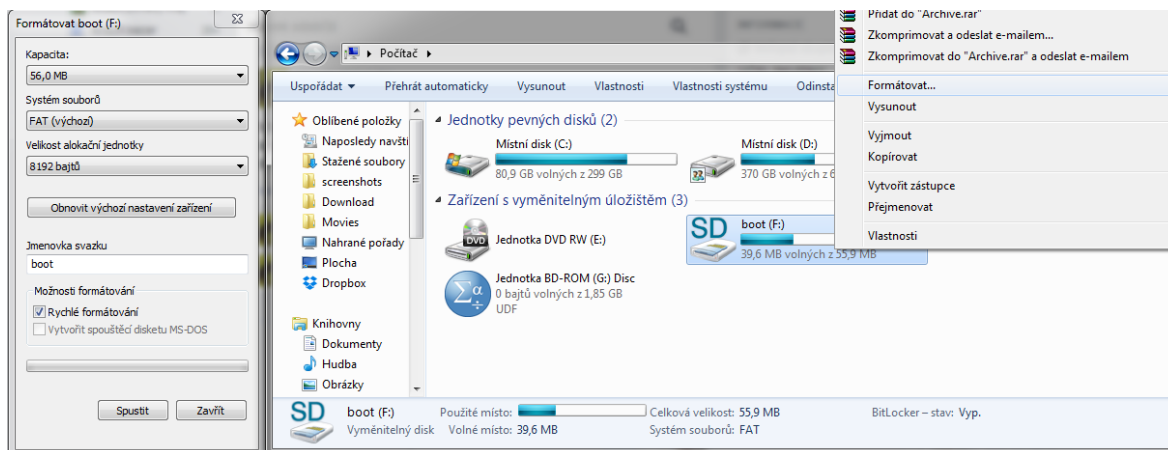
V následující části se budeme věnovat instalaci a oživení systému Raspbian.

6.1 Oživení a instalace OS

Před prvním zapnutím Mikropočítače je nutné nainstalovat OS. Jako médium sloužící k uložení OS bude sloužit SD karta. Pro instalaci je nutné tuto kartu naformátovat a vložit image OS. Pro duplikátor budeme používat OS Raspbian ve verzi 1.4.0, který si stáhneme přímo ze stránek raspberrypi.org v záložce download. Výrobce nabízí instalaci dvěma způsoby. Buďto přímým nahráním image na kartu nebo instalací pro začátečníky, takzvaný „NOOB“. Třetí možností je si zakoupit kartu s již nahreným OS. Instalace pomocí image je ovšem jednodušší oproti instalaci NOOB, tudíž bude zvolena tato metoda. Pro instalaci je nutné mít počítač s osazenou čtečkou karet. Pro zápis a naformátování bude použit program Win32DiskImager, který doporučuje přímo výrobce. [20]

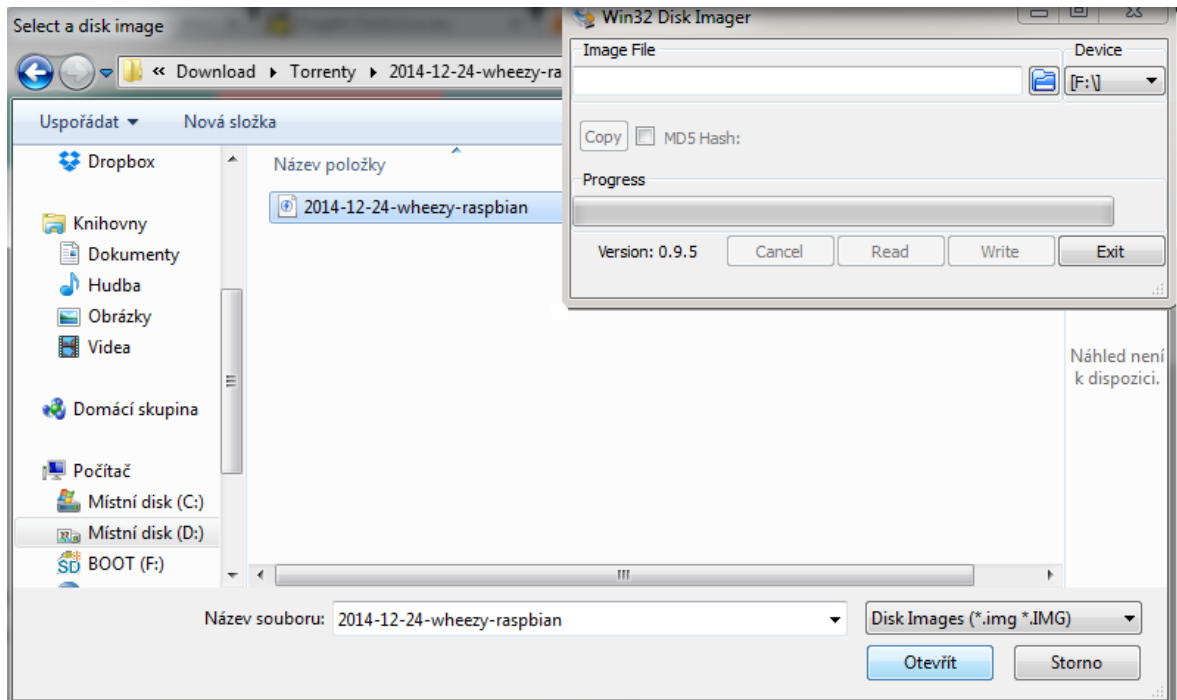
6.1.1 Nahrání image na SD kartu

Nejprve je nutné SD kartu naformátovat, to provedeme pomocí kliknutí pravého tlačítka myši na ikonu SD karty, kde zvolíme položku Formátovat. Necháme vše ve výchozím nastavení a jednotku pojmenujeme boot. Formátování spustíme tlačítkem spustit.



Obr. 18. Formátování SD karty

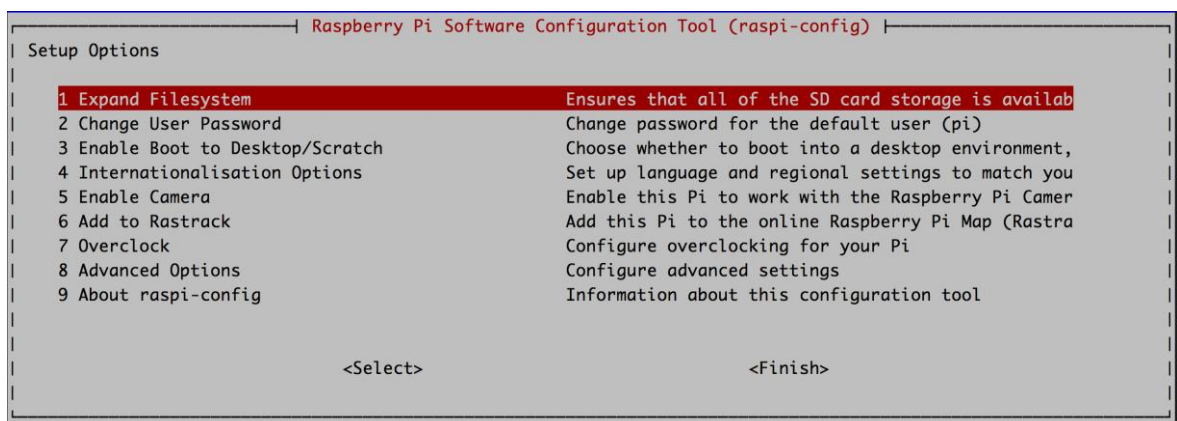
Jakmile je formátování dokončené, můžeme nahrát stažený image na kartu pomocí programu Win32DiskImager. Vybereme image kliknutím na modrou složku. Poté vybereme cílovou jednotku, na kterou bude image zapsán, v našem případě F: a spustíme proces kliknutím na tlačítko Write.



Obr. 19. Zápís image pomocí Win32DiskImageru

6.1.2 Spuštění Operačního systému

Po instalaci můžeme vložit SD kartu do slotu v Raspberry Pi a spustit mikropočítač zapojením adaptéru napájení. Po načtení systému se nám spustí konfigurační okno takzvaný raspi-config, kde je důležité kvůli instalaci updatů nastavit časové pásmo. To nastavíme pomocí položky Internationalisation options, dále vybereme položku Change Timezone, zvolíme Europe a poté Prague. Raspi-config zavřeme vybráním položky Finish. [21]

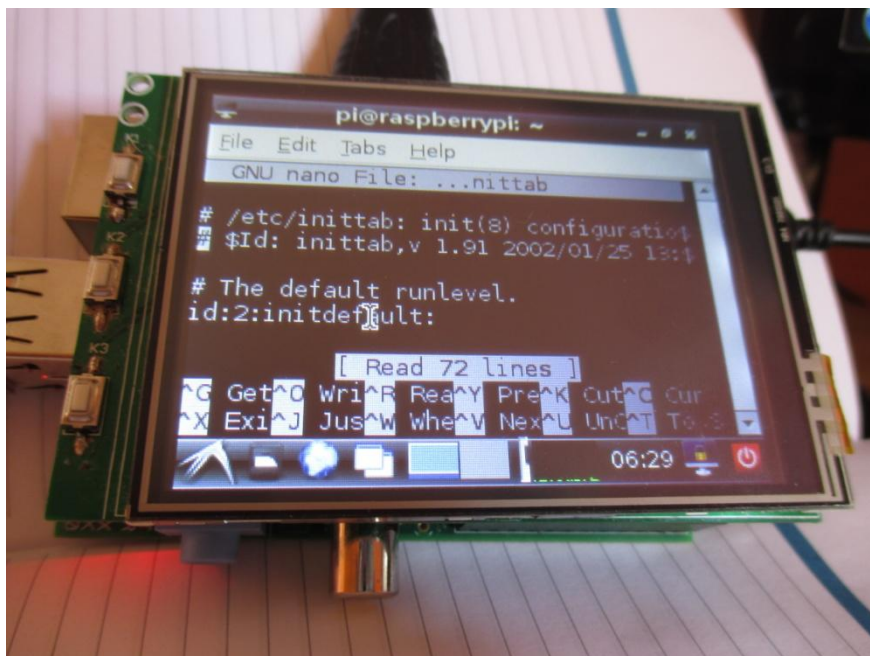


Obr. 20. raspi-config

Operační systém se nám přepne na příkazovou řádku. Grafické rozhraní spustíme příkazem startx.

6.2 Oživení displeje

Pro oživení displeje je možno použít vícero postupů. Jelikož nejsou nainstalovány ovladače pro SPI port, musíme je buďto nainstalovat svépomocí, nebo lze najít již operační systém přímo pro určitý displej, kde jsou už veškeré ovladače součástí OS. Jako zobrazovací jednotka je u duplikátoru použit Displej s úhlopříčkou 3,2“, rozlišením 320x240 obrazových bodů a je plně dotykový. OS systém se instaluje stejně, jak již bylo popsáno v bodě 4.1.1. Pokud je nainstalován OS pro dotykový displej, není možné připojit externí displej skrze HDMI. Pro programování byl tedy zvolen klasický OS. Na obrázku můžeme vidět, jak vypadá OS na dotykovém displeji, pro programování je práce s ním naprosto nevhodná. [22]



Obr. 21. Ukázka čitelnosti dotykového displeje

6.3 Úkony před spuštěním programu duplikace

Před samotným spuštěním programu duplikace je nutné provést několik úprav, nutných k automatickému spuštění programu, velikosti oken a dalších nechtěných faktorů.

6.3.1 Automatické spuštění po startu

Pro to, aby nebylo nutné jakkoliv spouštět program a někde ho hledat, je nejlepší volbou spustit ho ihned po nastartování grafického prostředí OS. Tato operace je provedena spuštěním konfiguračního souboru příkazem `sudo nano`

/etc/xdg/lxsession/LXDE/autostart v aplikaci LXTerminal, což je příkazová řádka OS Raspbian. Na konec souboru je připsán řádek „@sudo lxterminal --geometry=38x11 -e /usr/bin/python /home/pi/Desktop/disk.py“. [23]

Tento příkaz spustí po startu LXTerminal v přizpůsobeném rozlišení k rozlišení obrazovky. Program s názvem disk.py se bude otevírat přes program Python, což je programovací jazyk.

6.3.2 Vypnutí nabídky při vložení média

Aby nevyskakovalo okno pokaždé, kdy je vloženo přenosné médium do USB, je nutné vypnout toto vyskakovací okno. Vypnutí okna lze provést přes File manager, kde nahoře vybereme nabídku Edit, poté Preferences a zvolíme záložku Volume Management, kde zrušíme zaškrtnutí nabídky Show available options for removable media when they are inserted, což vypne vyskakovací okno. Toto vyskakovací okno můžeme také nechat zapnuté, ale pro účely duplikátoru spíše překáží z důvodu překrytí okna programu, kde je spuštěn postup duplikace.

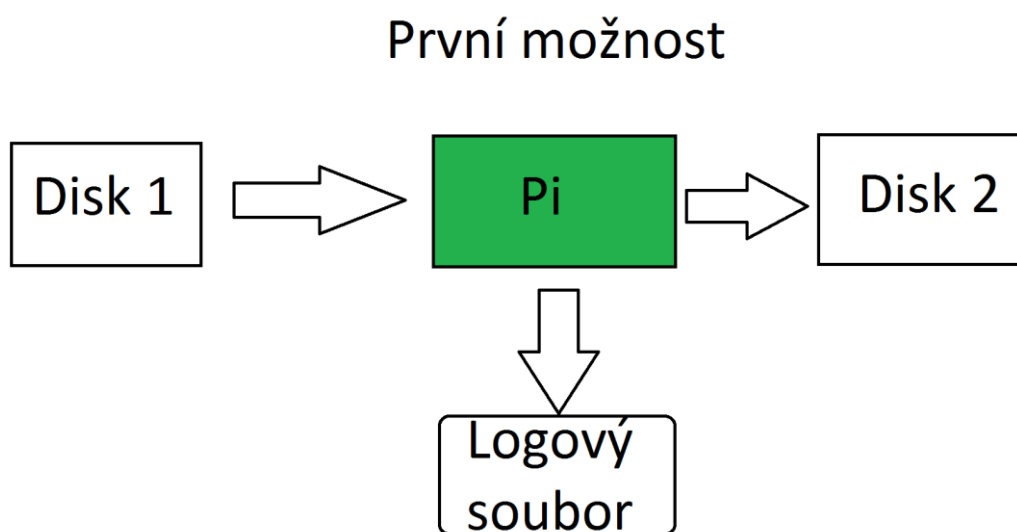
6.4 Program k duplikaci

Program, který se bude skládat ze tří částí, bude spuštěn po nastartování grafického prostředí OS. Bude se skládat z části pro výběr možností, jestli chceme na disk zapsat jeho kopii se zapsáním souboru o čase duplikace na jiné médium. V tomto případě se bude jednat o SD kartu s OS. Druhou možností je nahrání logového souboru image zdrojového disku na cílový, což je jeden soubor s obrazem celého disku, se kterým je poté možno dále jednodušeji pracovat. První možnost, tedy překopírování, je skript, který je napsán v programovacím jazyce Python. Druhou možností je nahrání image, který je prováděn skrze funkci dd.

„Python je dynamický, objektově-orientovaný programovací jazyk, který se může využít v mnoha oblastech vývoje softwaru. Nabízí významnou podporu k integraci s ostatními jazyky a nástroji a přichází s mnoha standardními knihovnamí. Dá se naučit v několika málo dnech. Mnoho pythonýrských programátorů hlásí zvýšení produktivity práce a pocit, že jazyk podporuje vývoj na vyšší kvalitě s čitelnějším kódem.“ [24]

6.4.1 První možnost kopírování souborů

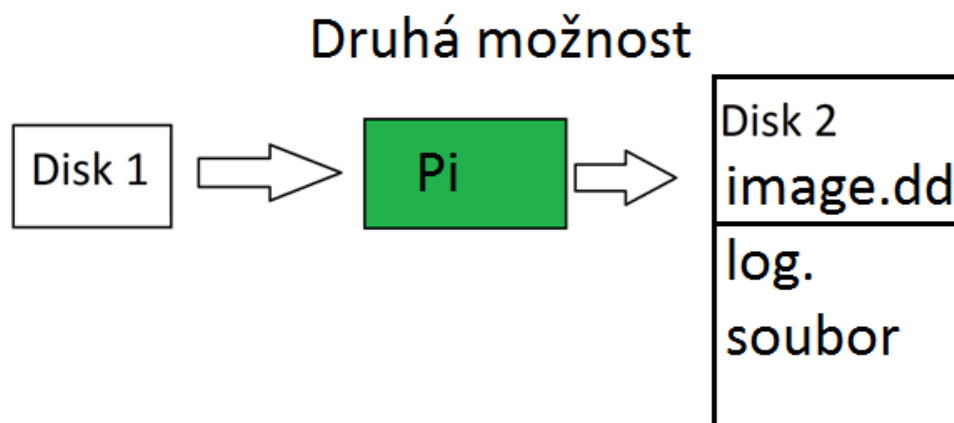
První možnost je přehrání celé bitové kopie disku na cílový disk s vyvedením logového souboru s časem začátku a konce duplikace. Program se skládá z částí pro inicializaci disků (`get_disks`), kopírování (`Copy`) a hlavní části (`Main`), která volá a pracuje se dvěma předchozími částmi. Na displeji je po připojení disků zobrazeno jeho jméno a cesta. Při samotné kopírovací funkci je na displej vypsáno množství přenesených dat v gigabajtech, procentuální postup, čas duplikace a rychlost přenosu. U této funkce lze vybrat ze dvou možností. Buďto po blocích 512 bajtů, anebo 1 megabajt. [25]



Obr. 22. Zapsání kopie na cílový disk s vyvedením logového souboru

6.4.2 Druhá možnost kopírování image

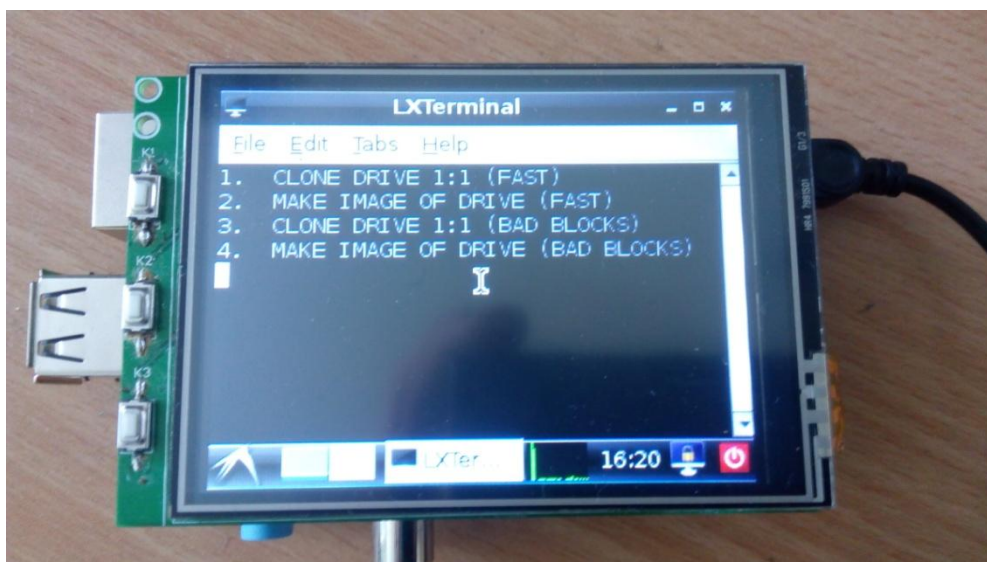
Druhou možností je vytvoření obrazu zdrojového disku na cílový v podobě souboru `image.dd`, který je možno jednoduše přenést na jiná zařízení a dále s ním snadno pracovat. Tato funkce je tvořena stejnými částmi jako v předchozím případě. Rozdíl je v zápisu logového souboru, který se nezapisuje na SD kartu s OS, ale přímo na cílový disk. Zde je nutno podotknout, že cílový disk musí být větší než ten zdrojový z důvodu větší velikosti právě díky logovému souboru.



Obr. 23. Zapsání obrazu a logového souboru na cílový disk

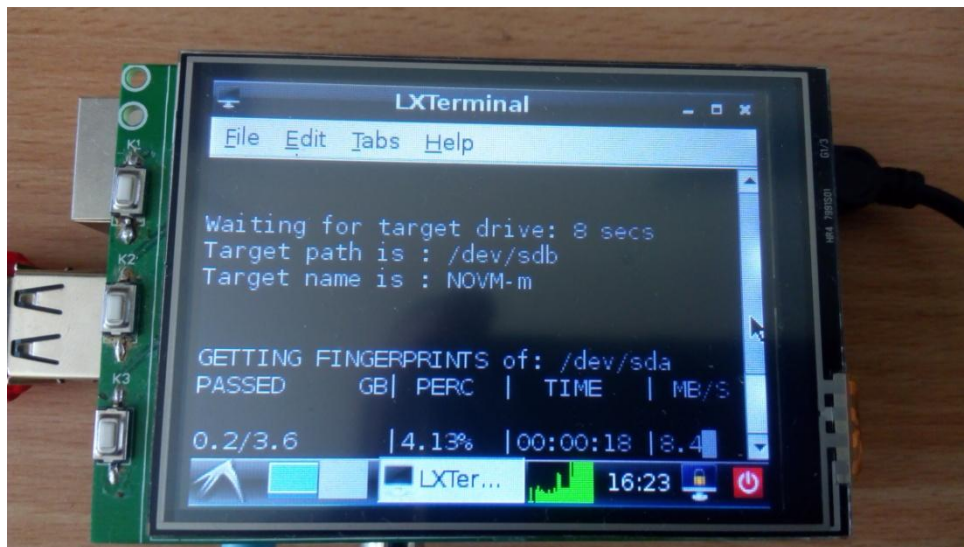
6.5 Vzhled programu

V následujícím obrázku je znázorněn vzhled programu při výběru a při samotném průběhu duplikace. V první části je na výběr ze čtyř možností, které byli popsány v předchozích bodech. Po vybrání určitého typu duplikace nás program vyzve k připojení zdrojového disku a následně cílového disku. Po připojení každého zařízení je zobrazeno jméno zařízení a jeho cesta.



Obr. 24. Vzhled menu pro výběr druhu duplikace

Na dalším obrázku je vidět průběh duplikace, kde je graficky znázorněn postup, rychlost a čas duplikace. Po dokončení je v programu krátce zobrazeno kolik bajtů bylo přeneseno a program se následně sám ukončí.



Obr. 25. Vzhled průběhu duplikace

7 ANALÝZA VÝKONU A FUNKCE DUPLIKÁTORU

V této části bude uvedeno testování rychlosti přenosu dat v závislosti na různých typech zařízení, což jsou především flash disky, paměťové karty a pevné disky o různých velikostech a přenosových rychlostech. Dalším důležitým aspektem je kontrola funkcionality duplikátoru a otestování integrity dat.

7.1 Testování rychlosti přenosu

Testovány byly paměťové karty, pevné disky a flash disky. Zařízení nabízejí různé přenosové rychlosti. U paměťových karet je tento údaj řízen parametrem class, který se dělí na class 2, 4, 6 a 10, kde číslice za slovem class značí, o jak rychlé zařízení se jedná. Číslice především značí rychlost zápisu na médium v megabajtech za sekundu. Class 2 slouží pro uchování videa ve standardním rozlišení, class 4 a 6 v HD rozlišení a 10 v plném HD rozlišení. Jelikož ale bude použita redukce pro USB, rychlost se razantně sníží vlivem maximální přenosové rychlosti USB 2.0, kterým Raspberry Pi disponuje a přenosová rychlost tohoto standardu činí 480 megabitů za sekundu, což teoreticky znamená, že za jednu sekundu přeneseme 60 megabajtů. Prakticky ovšem musíme počítat s menší přenosovou rychlostí a to z již zmíněného důvodu zapojení obou zařízení na jeden USB port skrze USB HUB. [26]

7.1.1 Micro sd karta

Pro testovací účel bude použita microSD karta s označením class 4 výrobce ADATA.



Obr. 26. Detail micro SD karty a redukce na usb

7.1.2 USB Flash Disk

Dalším testovaným zařízením je flash disk, který má výrobcem udanou rychlost zápisu 8 megabajtů za sekundu a rychlost čtení 15 megabajtů za sekundu. Jedná se o zařízení výrobce EMTEC.



Obr. 27. Flash disk

7.1.3 Pevné disky

Posledním testovaným zařízením budou pevné disky s konektivitou SATA a starším IDE o velikostech 120, 200 a 80 gigabajtů.



Obr. 28. Disky s konektivitou SATA a IDE



Obr. 29. Disk zapojený v diskovém rámečku

7.2 Vyhodnocení rychlosti přenosu

Pro vyhodnocení rychlosti je brána průměrná rychlost přenosu mezi zdrojovým a cílovým diskem. Testován byl přenos z USB flash disku na SD kartu a na pevný disk SATA. Poté z SD karty na pevný disk SATA a jako poslední z pevného Disku IDE na pevný disk SATA.

7.2.1 Přenosové rychlosti při duplikaci z flash disku na SD kartu a pevný disk

Při přenosu z flash disku na SD kartu byla průměrná přenosová rychlost 5 megabajtů za sekundu. Takto nízká rychlost byla zapříčiněna nízkou zápisovou rychlostí SD karty. Při přenosu z flash disku na pevný disk SATA činila průměrná rychlost přenosu 10 megabajtů za sekundu. Při kopírování na SD kartu byla přenosová rychlost omezena spíše rychlostí zápisu na SD kartu. U přenosu na SATA disk bylo dosaženo maximální čtecí rychlosti flash disku, která zpomalovala zápis na pevný disk.

7.2.2 Přenosová rychlost při duplikaci z SD karty na pevný disk SATA

Při přenosu z flash disku na pevný disk SATA činila průměrná rychlost přenosu 13 megabajtů za sekundu. Zde je max rychlost ovlivněna rychlostí čtení z SD karty.

7.2.3 Přenosová rychlost při duplikaci pevný disk na pevný disk

Rychlost při kopírování z pevného disku s konektorem IDE na pevný disk se SATA konektorem činila průměrná přenosová rychlost 14 megabajtů za sekundu, tedy přenos 100 gigabajtů by trval přibližně 2 hodiny. Zde byla rychlost omezena jak konektivitou IDE na zdrojovém disku, tak max. přenosovou rychlostí USB.

7.3 Tabulkový přehled rychlostí

Pro lepší přehled v rychlostech jsou rychlosti uvedeny v následující tabulce, rozdělené do sektorů podle zdrojového média.

Tab. 3. Tabulka rychlostí jednotlivých médií

Zdrojové médium	Cílové médium	Rychlost [MB/s]
Flash disk	SD karta	5
	SATA disk	10
SD karta	SATA disk	13
IDE disk	SATA disk	14

7.4 Funkce duplikátoru

Duplikátor je rozdělen do pěti částí, z čehož jedna je hlavní část pro výběr dvou dalších částí zadávána příkazem z klávesnice. Druhá a čtvrtá část slouží pro vytvoření duplicity formou bitové kopie 1:1 s nahráním logového souboru na SD kartu s OS. Třetí a pátá část slouží ke zkopírování obrazu disku na cílový disk s vytvořením logového souboru na cílovém disku. V obou případech je na displeji zobrazen postup přenosu. Do logového souboru je poté zapsán čas začátku, čas konce, celkový čas přenosu, otisk MD5 a SHA1.

8 POROVNÁNÍ S PRODUKTY NA TRHU

Pro porovnání sestaveného duplikátoru byly vybrány dva forenzní duplikátory. Porovnání bude zaměřeno hlavně na konektivitu, cenu, počet připojených disků a přenosovou rychlost. Jako porovnávané modely byly vybrány dva forenzní duplikátory. A to zařízení: Logicube Forencis Falcon a Tableau TD2u. Shrnutí parametrů sestaveného duplikátoru.

Pro shrnutí parametrů byly zohledněny všechny výše popsané parametry. Pro větší rozsah jsou uvedeny dvě cenové kategorie. U levnější je počítáno s co nejnižšími pořizovacími náklady. U dražší varianty je počítáno s komponenty, které byly použity pro tento konkrétní sestavený model.

8.1.1 Levnější varianta

Pro levnější variantu bylo použito téměř nejlevnějších komponent se zachováním stejné nebo lehce omezené funkce. U levnější varianty nebude použit displej a rozbočovač USB. Při použití by bylo ovšem nutno použít externí monitor s konektivitou HDMI a takzvaný ypsilon kabel, který zajišťuje napájení USB. Ceny jsou uvedeny včetně DPH.

Tab. 4. Cena levnější varianty duplikátoru

Levnější varianta	
Předmět	Cena včetně DPH
Raspberry Pi Model B	880 Kč
Y kabel	138 Kč
USB Rámeček	400 Kč
Cena Celkem	1 418 Kč

8.1.2 Dražší varianta

Pro dražší variantu bylo použito všech použitých komponent, které se objevili v této práci. Navíc je tedy oproti levnější variantě přidána redukce SATA na USB, externě napájený USB rozbočovač a displej. Některé komponenty byly u levnější varianty použity dvakrát. U dražší varianty jsou použity jednou a nahrazeny jinými komponenty s lepšími parametry. Týká se to zejména USB rozbočovače pro připojení více zařízení a displeje pro větší mobilitu celé sestavy.

Tab. 5. Cena dražší varianty duplikátoru

Dražší varianta	
Předmět	Cena včetně DPH
Raspberry Pi Model B	880 Kč
3,2 " displej	370 Kč
USB rozbočovač	550 Kč
redukce SATA na USB	600 Kč
USB Rámeček	200 Kč
Cena Celkem	2 600 Kč

8.2 Porovnání s forenzními duplikátory

Pro porovnání byly zvoleny výše uvedené forenzní duplikátory Logicube Forencis Falcon, Tableau TD2u.

8.2.1 Logicube Forencis Falcon

„Logicube je jedním z nejstarších výrobců zařízení pro duplikaci pevných disků a forenzní získávání dat v průmyslu. Je průkopníkem v této oblasti od roku 1993, kdy byla v Kalifornii, USA, založena. Logicube vyrábí zařízení od malých a lehce přenosných produktů pro duplikaci 1 k 1 až po velké systémy určené pro kopírování na více cílů bez rozlišování typu, velikosti nebo výrobce disku. Forenzní řešení potom umožňují práci ve ztížených „polních“ podmínkách, rychlý a autentifikací zajištěný sběr dat, případně extrakci dat z běžných mobilních telefonů, tak i jejich napodobenin založených na čínských čípech.“ [27]

Oproti duplikátoru Raspberry Pi nese Logicube velké množství výhod. Jelikož výrobce cenu modelu neudává, je ovšem patrné, že se nebude pohybovat v takové cenové relaci jako sestavený duplikátor Raspberry Pi. Jeho výhody jsou ovšem obrovské. Přenosová rychlost činí rychlost přenosu až 20 gigabajtů za minutu, má možnost kopírovat ze čtyř zdrojových disků až na pět cílových. Má možnost vzdáleného ovládání skrze http protokol. Podporuje zápis do formátu dd image, ex01 a e01. Má zabudovaný sedmipalcový dotykový displej a má funkci stoprocentní ochrany proti zápisu na zdrojový disk. [27]



Obr. 30. Forezní duplikátor Logicube Forencis Falcon [27]

8.2.2 Tableau TD2u

„Forezní duplikátory Tableau nabízí vytvoření forezní kopie pevných disků a dalších paměťových médií bez jakýchkoliv kompromisů. Ačkoliv je používání přístrojů Tableau velice lehké a intuitivní, mohou nabídnout mnoho užitečných funkcí i pokročilému uživateli. Rychlost a spolehlivost je samozřejmostí.“ [9]

Zařízení Tableau je oproti Logicube patrně horší. Rychlost přenosu je 15 gigabajtů za minutu. Zařízení dále umožňuje stejně jako Raspberry Pi duplikátor výpočet MD5 a SHA1. Podporuje zařízení s konektivitou USB 3.0, SATA a IDE. Oproti sestavenému duplikátoru, který má konektivitu pouze USB 2.0., Tableau umožňuje provádět kopii až na tři cílové disky zároveň. Výrobce cenu modelu neudává.



Obr. 31. Forezní duplikátor Tableau TD2u [9]

ZÁVĚR

Cílem této bakalářské práce bylo seznámit se a shrnout problematiku řešení kriminální činnosti v oblasti kybernetické kriminality s použitím prostředků, využívaných pro řešení tohoto problému. Pro čtenáře by mělo být přínosem také seznámení s novými technologiemi výpočetní techniky a jejím využitím právě v tomto odvětví forezních věd. Další důležitou částí je obeznámení se staršími zařízeními, která jsou dnes stále používána a musí na ně být brán zřetel, jelikož mohou být taktéž vodítkem k řešení případu. Konkrétním prvkem této práce je hardwarové zařízení duplikátor harddisků, u kterého bylo popsáno, jak funguje a proč je důležitou součástí při vyšetřování. Dalším úkolem bylo tento duplikátor sestavit a popsat jak pracuje.

Duplikátor ovšem není jen záležitostí forezních věd, ale je využíván i pro jiné účely, jako pro zálohování, vytváření duplicit CD, DVD, využívaných v komerční, či jiné oblasti. Jelikož ale nejsou tato zařízení pro svůj účel cenově dostupná, je v praktické části popsáno, jak je možné využít levný mikropočítač s procesorem ARM právě pro tento účel.

V teoretické části bylo popsáno, k čemu duplikátor slouží, jaké by měl mít parametry z hlediska nároků na pořízení a v jakých oblastech se využívá. Velmi důležitou částí byl popis využití v kriminalistickém prostředí s vymezením prostředků a metod postupů. V dalších kapitolách bylo popsáno, jaké jsou požadavky a jaké jsou novinky v odvětví výpočetní techniky s uvedením parametrů, které by měly splňovat nejnovější kriminalistické prostředky.

V praktické části je sepsáno přehledné sestavení duplikátoru s fotodokumentací. Na začátku je znázorněno, jaké komponenty byly použity pro duplikátor disků, s jednoduchým vysvětlením jak tyto součásti oživit. Poté je v této části popsán program, který řídí průběh duplikace s analýzou výkonu. Na konci praktické části je uvedeno porovnání parametrů sestaveného duplikátoru s produkty na trhu.

Přínosem této práce bylo sestavení duplikátoru, který je oproti výrobkům uvedeným na trh podstatně levnější a přinejmenším vystačuje pro méně náročné uživatele. Velkou zásluhu na ceně tohoto duplikátoru je použitá technologie ARM procesoru, který je díky své ceně využíván ve většině elektronických zařízení, která jsou pro nás v dnešním světě nepostradatelná.

SEZNAM POUŽITÉ LITERATURY

- [1] SOLOMON, Perry. *Alartec* [online]. [cit. 2015-05-31]. Dostupné z: <http://www.aleratec.com/>
- [2] *Terra* [online]. [cit. 2015-05-31]. Dostupné z: <http://www.terra.cz/>
- [3] VÍTEK, Jan a Petr STÁNSKÝ. *Svět Hardware* [online]. 2009. [cit. 2015-05-31]. Dostupné z: <http://www.svethardware.cz/>
- [4] BARTOŇ, Martin. *Diit* [online]. 2008 [cit. 2015-05-31]. Dostupné z: <http://diit.cz/>
- [5] *Diskus* [online]. [cit. 2015-05-31]. Dostupné z: <http://www.diskus.cz/>
- [6] KADLEC, Josef. *Root* [online]. 2005 [cit. 2015-05-31]. Dostupné z: <http://www.root.cz/>
- [7] LUDMA, Martin. *Mgr. Martin Ludma* [online]. 2007. [cit. 2015-05-31]. Dostupné z: <http://www.martinludma.cz/>
- [8] RISK ANALYSIS CONSULTANTS. *Forenzní zkoumání digitálních důkazů*. 2005.
- [9] *Risk Analysis Consultants* [online]. 2014 [cit. 2015-05-31]. Dostupné z: <http://www.rac.cz/>
- [10] *Elas Brno* [online]. [cit. 2015-05-31]. Dostupné z: <http://www.elasbrno.cz/>
- [11] OBERMAIER, Zdeněk. *PC tuning* [online]. 2009 [cit. 2015-05-31]. Dostupné z: <http://pctuning.tyden.cz/>
- [12] *Images Techtree* [online]. [cit. 2015-05-31]. Dostupné z: http://images.techtree.com.s3-ap-southeast-1.amazonaws.com/ttimages/story/110144_usb-3.jpg
- [13] *All computer sites* [online]. 0011n. 1. [cit. 2015-05-31]. Dostupné z: <http://www.allcomputersites.com/>
- [14] ČERMÁK, Miroslav. *Clever and Smart* [online]. 2010 [cit. 2015-05-31]. Dostupné z: <http://www.cleverandsmart.cz/>
- [15] UPTON, Eben. *Raspberry Pi: uživatelská příručka*. 1. vyd. Brno: Computer Press, 2013, 232 s. ISBN 978-80-251-4116-8.
- [16] *RPishop* [online]. [cit. 2015-05-31]. Dostupné z: <http://rpishop.cz/>

- [17] BÍBR, Ivan. *Ubuntu 10.04 CZ: praktická příručka uživatele Linuxu*. Vyd. 1. Brno: Computer Press, 2010, 366 s. ISBN 978-80-251-3121-3.
- [18] *Distrowatch* [online]. 2015 [cit. 2015-05-31]. Dostupné z: <http://distrowatch.com/>
- [19] PINKER, Jiří. *Mikroprocesory a mikropočítače*. 1. vyd., 1. dot. Praha: BEN - technická literatura, 2008, 159 s. ISBN 978-80-7300-110-0.
- [20] VÁŇA, Vladimír. *ARM pro začátečníky*. Praha: BEN - technická literatura, 2009, 195 s. ISBN 978-80-7300-246-6.
- [21] LASSER, Jon. *Rozumíme UNIXu*. Vyd. 1. Praha: Computer Press, 2002, 252 s. ISBN 80-722-6706-X.
- [22] SANJEEV, Arvind. *DIYhacking. 2014* [online]. [cit. 2015-05-31]. Dostupné z: <http://diyhacking.com/>
- [23] KAMENÍK, Pavel. *Příkazový řádek v Linuxu: praktická řešení*. Vyd. 1. Brno: Computer Press, 2011, 224 s. ISBN 978-80-251-2819-0.
- [24] *Python py* [online]. 2014 [cit. 2015-05-31]. Dostupné z: <http://www.py.cz/>
- [25] MATTHEW, Neil a Richard STONES. *Linux: začínáme programovat*. Vyd. 1. Brno: Computer Press, 2008, 829 s. ISBN 978-80-251-1933-4.
- [26] KUŽNÍK, Jan a Václav NÝVLT. *Technet* [online]. 2010 [cit. 2015-05-31]. Dostupné z: <http://technet.idnes.cz/>
- [27] CATEGORY A.S. *Category* [online]. 2014 [cit. 2015-05-31]. Dostupné z: <http://www.category.cz/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

OS Operační systém

HD High Definition

SSD Solid state disk.

SEZNAM OBRÁZKŮ

<i>Obr. 1. Duplikátor pro 5 cílových zařízení [1]</i>	11
<i>Obr. 2. Ukázka rozlišení velikosti a konektivity disků [1]</i>	13
<i>Obr. 3. Duplikátor Flash disků [4]</i>	13
<i>Obr. 4. Duplikátor optických disků [5]</i>	14
<i>Obr. 5. Distribuce software pro forenzní získávání dat [9]</i>	17
<i>Obr. 6. Hardwarové vybavení pro forenzní analýzu dat [9]</i>	18
<i>Obr. 7. Ostatní vybavení pro zajištění důkazů [10]</i>	19
<i>Obr. 8. Rozdíly v konektoru USB 3.0 a USB 2.0 [13]</i>	20
<i>Obr. 9. Porovnání mechanického a SSD disku [13]</i>	21
<i>Obr. 10. Blokátor USB a paměťových karet [9]</i>	23
<i>Obr. 11. Detail desky Raspberry Pi model B</i>	28
<i>Obr. 12. Displej 3,2 TFT</i>	28
<i>Obr. 13. Síťový adaptér a HDMI kabel</i>	29
<i>Obr. 14. Redukce IDE na SATA</i>	30
<i>Obr. 15. Redukce SATA na USB</i>	30
<i>Obr. 16. Grafické prostředí OS Raspbian [18]</i>	31
<i>Obr. 17. Externě napájený rozbočovač USB</i>	32
<i>Obr. 18. Formátování SD karty</i>	33
<i>Obr. 19. Zápis image pomocí Win32DiskImageru</i>	34
<i>Obr. 20. raspi-config</i>	34
<i>Obr. 21. Ukázka čitelnosti dotykového displeje</i>	35
<i>Obr. 22. Zapsání kopie na cílový disk s vyvedením logového souboru</i>	37
<i>Obr. 23. Zapsání obrazu a logového souboru na cílový disk</i>	38
<i>Obr. 24. Vzhled menu pro výběr druhu duplikace</i>	38
<i>Obr. 25. Vzhled průběhu duplikace</i>	39
<i>Obr. 26. Detail micro SD karty a redukce na usb</i>	41
<i>Obr. 27. Flash disk</i>	41
<i>Obr. 28. Disky s konektivitou SATA a IDE</i>	42
<i>Obr. 29. Disk zapojený v diskovém rámečku</i>	42
<i>Obr. 30. Forenzní duplikátor Logicube Forencis Falcon [27]</i>	47
<i>Obr. 31. Forenzní duplikátor Tableau TD2u [9]</i>	48

SEZNAM TABULEK

<i>Tab. 1. Rozdíly ve standardu USB 2.0 a USB 3.0 [11].....</i>	20
<i>Tab. 2. Rozdíly ve standardu SATA 3G a SATA 6G [11]</i>	21
<i>Tab. 3. Tabulka rychlostí jednotlivých médií.....</i>	43
<i>Tab. 4. Cena levnější varianty duplikátoru.....</i>	45
<i>Tab. 5. Cena dražší varianty duplikátoru</i>	46

SEZNAM PŘÍLOH

Příloha P I Program duplikátoru

PŘÍLOHA P I: PROGRAM DUPLIKÁTORU

```
import os
import commands
import time
import math
import hashlib
from sys import stdout
import subprocess
from threading import Thread

def get_disks():

    paths = commands.getstatusoutput("lsblk -i -l -n -o NAME,SIZE,TYPE -b | grep
'disk' | cut -f1 -d' ' ")
    paths = paths[1]
    paths = paths.split()
    a = 1
    names = []
    for x in paths:
        name = commands.getstatusoutput("sudo blkid /dev/" + x + "*" | grep
LABEL ")
        if not name[1]:
            name = "undefined"
        else:
            name = name[1].split()
            for part in name:
                if "LABEL" in part:
                    name = part
                    name = name[6:]
                    name = name[1:]
            names.append(name)
            paths[a-1] = "/dev/" + x
            a = a + 1
    return paths,names

def get_partitions(disk):
    sizes = []
    biggest_size = 1
    partitions = commands.getstatusoutput("lsblk " + disk + " -i -l -n -o
NAME,SIZE,TYPE -b | grep 'part' | cut -f1 -d' ' ")
    partitions = partitions[1]
    if partitions == "":
        biggest_size_name = "none"
    else:
        partitions = partitions.split()
        for x in partitions:
            x = "/dev/" + x
            size = commands.getstatusoutput("sudo blockdev --getsize64 " + x)
            if size:
                size = size
```



```

        elif not size:
            size = 1
        sizes.append(size[1])
    a = 0
    for x in sizes:
        if int(x) > biggest_size:
            biggest_size = int(x)
            biggest_size_name = "/dev/" + partitions[a]
        a = a + 1
    return biggest_size_name

```

```

def Copy(source,destination,buffer):
    chunksize = buffer
    total_copied = 0
    to_write = commands.getstatusoutput("""blockdev --getsize64 """" + source)
    chunk_size_copied = 0
    time_start = time.time()
    time.sleep(0.1)
    print "\n"
    print "CLONNING..."
    print "TRANSFERED GB| PERC | TIME | MB/S "
    with open(source,'rb') as f:
        with open(destination, "wb") as i:
            while True:
                chunk = f.read(chunksize)
                if len(chunk) == 0:
                    break
                i.write(chunk)
                chunk_size_copied = len(chunk) + chunk_size_copied
                perc = float(int(chunk_size_copied)) / float(int(to_write[1]))
* float(100)

                time_now = time.time()
                time_till_now = time_now - time_start
                perc = float("{0:.2f}".format(perc))
                chunk_size_copied_MB = int(chunk_size_copied) / 1048576
                to_write_MB = int(to_write[1]) / 101048576
                to_write_GB = round(float(to_write[1]) / 1073741824, 1)
                chunk_size_copied_GB = round(float(chunk_size_copied) /
1073741824, 1)

                write_speed = chunk_size_copied_MB / time_till_now
                write_speed = float("{0:.1f}".format(write_speed))
                time_till_now = time.strftime("%H:%M:%S",
time.localtime(time_till_now))
                stdout.write("\r%s" % ( \
str(chunk_size_copied_GB) + "/" + str(to_write_GB) \
+ " " * (13 - (len(str(chunk_size_copied_GB) + "/" +
str(to_write_GB)))) \
+ "|" + str(perc) + "% " \

```

```

        + (7 - (len(str(perc) + "%")) * " " \
        + "|" \
        + str(time_till_now)\
        + " " * (9 - len(str(time_till_now))) \
        + "|" + str(write_speed)))
    stdout.flush()
    f.close()
    i.close()
    print "\n"
    print "Copied " + str(chunk_size_copied) + " / " + str(to_write[1]) + '
bytes'

    if int(chunk_size_copied) == int(to_write[1]):
        print 'Copy successful'
    #return str(chunk_size_copied),str(to_write[1])

def DD_copy(source,destination,buffer):
    buffer = str(buffer)
    source_size = commands.getstatusoutput("""blockdev --getsize64 """" + source)
    source_size = source_size[1]
    source_size_GB = round(float(source_size)/1073741824,2)
    print 'COPYING...'
    print 'TO COPY:'
    print str(source_size) + " BYTES"
    print str(source_size_GB) + " GB" + "\n"
    s = subprocess.Popen(['sudo dd conv=noerror,sync,notrunc bs=' + str(buffer) + ' if=' +
source + ' 2>/home/pi/Desktop/write_error.txt | pv | dd conv=noerror,sync,notrunc bs=' +
str(buffer) + ' of=' + destination + ' 2>/home/pi/Desktop/write_error.txt'],
stdout=subprocess.PIPE, shell=True)
    s.wait()
    return source_size

def Get_fingerprint(source,buffer,to_read):
    chunksize = buffer
    total_copied = 0
    to_write = os.path.getsize(source)
    to_write = str(to_write)
    to_write = to_write.replace('L',")
    to_write = int(to_write)
    if to_write == 0:
        to_write = commands.getstatusoutput('blockdev --getsize64 ' + source)
        to_write = to_write[1]
    #to_write = commands.getstatusoutput("""blockdev --getsize64 """" + source)
#to_write = to_write[1]
    #if "BLKGETSIZE64: Function not implemented" in to_write:
    #    to_write = os.path.getsize(source)
    #    to_write = str(to_write)
    #    to_write = to_write.replace("L", "")
    #    to_write = int(to_write)
    if to_read == "":
        to_read = int(to_write)

```

```

    chunk_size_copied = 0
time_start = time.time()
time.sleep(0.1)
md5 = hashlib.md5()
sha1 = hashlib.sha1()
print "PASSED   GB|PERC | TIME | MB/S "
with open(source,'rb') as f:
    while True:
        if (int(to_read) - chunk_size_copied)< chunksize:
            chunksize = (int(to_read) - int(chunk_size_copied))
        chunk = f.read(chunksize)
        chunk_size_copied = len(chunk) + int(chunk_size_copied)
        md5.update(chunk)
        sha1.update(chunk)
        if len(chunk) == 0 or chunk_size_copied == to_read:
            break
        perc = float(int(chunk_size_copied)) / float(int(to_write)) * float(100)
        time_now = time.time()
        time_till_now = time_now - time_start
        perc = float("{0:.2f}".format(perc))
        chunk_size_copied_MB = int(chunk_size_copied) / 1048576
        to_write_MB = int(to_write) / 101048576
        to_write_GB = round(float(int(to_write))/1073741824,1)
        chunk_size_copied_GB = round(float(chunk_size_copied) / 1073741824,1)
        write_speed = chunk_size_copied_MB / time_till_now
        write_speed = float("{0:.1f}".format(write_speed))
        time_till_now = time.strftime("%H:%M:%S",
time.localtime(time_till_now))
            stdout.write("\r%s" %( \
str(chunk_size_copied_GB) + "/" + str(to_write_GB) \
+ " " * (13 - (len(str(chunk_size_copied_GB) + "/" + str(to_write_GB)))) \
+ "|" + str(perc) + "%" \
+ (7 - (len(str(perc) + "%"))) * " " \
+ "|" \
+ str(time_till_now) \
+ " " * (9 - len(str(time_till_now))) \
+ "|" + str(write_speed)))
            stdout.flush()

    f.close()
    print "\n"
    print "CHECKED \n" \
        + str(chunk_size_copied) + " / " + str(to_write) + ' bytes'
    return md5.hexdigest(),sha1.hexdigest()
def Main():
    time_at_start = time.time()
    os.system('clear')
    mountpoint = "/media/usb"
    while True:
        type_choice = raw_input("1. CLONE DRIVE 1:1 (FAST)\n" \

```

```

+ "2. MAKE IMAGE OF DRIVE (FAST)\n" \
+ "3. CLONE DRIVE 1:1 (BAD BLOCKS)\n" \
+ "4. MAKE IMAGE OF DRIVE (BAD
BLOCKS)\n")
    if type_choice == "1" or type_choice == "2" or type_choice == "3" or ty-
pe_choice == "4":
        break
    else:
        os.system('clear')
        mountpoint = "/media/usb"
        source_path = ""
        target_path = ""
        source_name = ""
        target_name = ""
        paths_first,names_first = get_disks()
        a = 0
        items = 0
        paths_done = paths_first
        names_done = names_first
        print 'Connected drives'
        for x in names_first:
            print x + " : " + paths_first[a]
            a = a + 1
        print "\n"
        time_start = time.time()
        time.sleep(0.1)
        while not source_path:
            time_now = time.time()
            time_till_now = time_now - time_start
            time_till_now = int(time_till_now)
            items = 0
            paths_2, names_2 = get_disks()
            for path in paths_2:
                if path not in paths_first :
                    paths_done.append(path)
                    names_done.append(names_2[items])
                    source_path = path
                    source_name = names_2[items]
                    print "\nSource path is : " + source_path
                    print "Source name is : " + source_name
                    #print paths_done
                    #print names_done
            elif source_path:
                axx = 1
            else:
                stdout.write("\r%s" % ("Waiting for source drive: " +
str(time_till_now) + " secs" ))
                stdout.flush()
                items = items + 1
                time.sleep(0.5)

```

```

print "\n"
time_start = time.time()
while not target_path:
    time_now = time.time()
    time_till_now = time_now - time_start
    time_till_now = int(time_till_now)
    items_1 = 0
    paths_3, names_3 = get_disks()
    for path in paths_3:
        if path not in paths_done:
            target_path = path
            paths_done.append(target_path)
            target_name = names_3[items_1]

            print "\nTarget path is : " + target_path
            print "Target name is : " + target_name
        elif target_path:
            axx = 1
        else:
            stdout.write("\r%s" % ("Waiting for target drive: " +
str(time_till_now) + " secs" ))
            stdout.flush()
            items_1 = items_1 + 1
            time.sleep(0.5)
print "\n"
print "GETTING FINGERPRINTS of: " + source_path
md5,sha1 = Get_fingerprint(source_path,2**17,"")
print "\n"
print "    FINGERPRINTS CREATED"
print "MD5 FINGERPRINT IS:"
print str(md5)
print "SHA1 FINGERPRINT IS:"
print str(sha1)
if type_choice == "1":
    to_read = DD_copy(source_path,target_path,"64k")
    time_at_end = time.time()
    time_elapsed = time_at_end - time_at_start
    time_at_start = time.strftime("%d-%m-%Y|%H:%M:%S",
time.localtime(time_at_start))
    time_at_end = time.strftime("%d-%m-%Y|%H:%M:%S",
time.localtime(time_at_end))
    time_at_elapsed = time.strftime("%H:%M:%S", time.localtime(time_elapsed))
    print "\n" + "MAKING CLONE FINGERPRINT"
    target_md5,target_sha1 = Get_fingerprint(target_path,2**17,str(to_read))
    print "\n" + "source hash is:"
    print str(md5)
    print str(sha1)
    print "clone hash is:"
    print str(target_md5)
    print str(target_sha1)

```

```

with open("/home/pi/Desktop/log.txt","w+") as log:
log.write("TIME_STARTED=" + time_at_start + "\n")
    log.write("TIME_END=" + str(time_at_end) + "\n")
log.write("TIME_ELAPSED=" + str(time_at_elapsed) + "\n")
log.write("source_MD5=" + str(md5) + "\n")
log.write("source_SHA1=" + str(sha1) + "\n")
    log.write("clone_MD5=" + str(target_md5) + "\n")
    log.write("clone_SHA1=" + str(target_sha1) + "\n")
    log.write("original_bytes=" + str(to_read) + "\n")
log.close()
print 'LOG FILE CREATED'
print '/home/pi/Desktop/log.txt'
elif type_choice == "2":
partition = get_partitions(target_path)
print "target = " + partition
if partition == "none":
    sys.exit(0)
x = commands.getstatusoutput("sudo mount " + partition + " " + mountpoint
)

print x[1]
to_read = DD_copy(source_path,mountpoint + "/image.dd","64k")
print "\nIMAGING DONE"
print mountpoint + "/image.dd"
time_at_end = time.time()
time_elapsed = time_at_end - time_at_start
time_at_start = time.strftime("%d-%m-%Y|%H:%M:%S",
time.localtime(time_at_start))
time_at_end = time.strftime("%d-%m-%Y|%H:%M:%S",
time.localtime(time_at_end))
time_at_elapsed = time.strftime("%H:%M:%S",
time.localtime(time_elapsed))
print "\n" + "GETTING TARGET FINGERPRINT"
target_md5,target_sha1 = Get_fingerprint(mountpoint +
"/image.dd",2**17,"")
print "\n" + "source hash is:"
print str(md5)
print str(sha1)
print "image hash is:"
print str(target_md5)
print str(target_sha1)
with open(mountpoint + "/log.txt","w+") as log:
    log.write("TIME_STARTED=" + str(time_at_start) + "\n")
    log.write("TIME_END=" + str(time_at_end) + "\n")
log.write("TIME_ELAPSED=" + str(time_at_elapsed) + "\n")
    log.write("source_MD5=" + str(md5) + "\n")
    log.write("source_SHA1=" + str(sha1) + "\n")
    log.write("image_MD5=" + str(target_md5) + "\n")
    log.write("image_SHA1=" + str(target_sha1) + "\n")
    log.write("original_bytes=" + str(to_read) + "\n")
log.close()

```

```

        print 'LOG FILE CREATED'
        print str(mountpoint) + '/log.txt'
        time.sleep(5)
        y = commands.getstatusoutput("sudo umount " + partition)
        print y[1]
    elif type_choice == "3":
        to_read = DD_copy(source_path,target_path,512)
        time_at_end = time.time()
        time_elapsed = time_at_end - time_at_start
        time_at_start = time.strftime("%d-%m-%Y|%H:%M:%S",
time.localtime(time_at_start))
        time_at_end = time.strftime("%d-%m-%Y|%H:%M:%S",
time.localtime(time_at_end))
        time_at_elapsed = time.strftime("%H:%M:%S", time.localtime(time_elapsed))
        print "\n" + "GETTING TARGET FINGERPRINT"
        target_md5,target_sha1 = Get_fingerprint(target_path,2**17,str(to_read))
        print "\n" + "source hash is:"
        print str(md5)
        print str(sha1)
        print "image hash is:"
        print str(target_md5)
        print str(target_sha1)
        with open("/home/pi/Desktop/log.txt","w+") as log:
            log.write("TIME_STARTED=" + str(time_at_start) + "\n")
            log.write("TIME_END=" + str(time_at_end) + "\n")
            log.write("TIME_ELAPSED=" + str(time_at_elapsed) + "\n")
            log.write("source_MD5=" + str(md5) + "\n")
            log.write("source_SHA1=" + str(sha1) + "\n")
            log.write("clone_MD5=" + str(target_md5) + "\n")
            log.write("clone_SHA1=" + str(target_sha1) + "\n")
            log.write("original_bytes=" + str(to_read) + "\n")
        log.close()
        print 'LOG FILE CREATED'
        print '/home/pi/Desktop/log.txt'
    elif type_choice == "4":
        partition = get_partitions(target_path)
        print "target = " + partition
        if partition == "none":
            sys.exit(0)
        x = commands.getstatusoutput("sudo mount " + partition + " " + mountpoint )
        print x[1]
        to_read = DD_copy(source_path,mountpoint + "/image.dd",512)
        print "\nIMAGING DONE"
        print mountpoint + "/image.dd"
        time_at_end = time.time()
        time_elapsed = time_at_end - time_at_start
        time_at_start = time.strftime("%d-%m-%Y|%H:%M:%S",
time.localtime(time_at_start))
        time_at_end = time.strftime("%d-%m-%Y|%H:%M:%S",
time.localtime(time_at_end))

```

```

time_at_elapsed = time.strftime("%H:%M:%S", time.localtime(time_elapsed))
    print "\n" + "GETTING IMAGE FINGERPRINT"
target_md5,target_sha1 = Get_fingerprint(mountpoint + "/image.dd",2**17,"")
print "\n" + "source hash is:"
print str(md5)
print str(sha1)
print "target hash is:"
print str(target_md5)
print str(target_sha1)
    with open(mountpoint + "/log.txt","w+") as log:
        log.write("TIME_STARTED=" + str(time_at_start) + "\n")
        log.write("TIME_END=" + str(time_at_end) + "\n")
        log.write("TIME_ELAPSED=" + str(time_at_elapsed) + "\n")
        log.write("source_MD5=" + str(md5) + "\n")
        log.write("source_SHA1=" + str(sha1) + "\n")
            log.write("image_MD5=" + str(target_md5) + "\n")
            log.write("image_SHA1=" + str(target_sha1) + "\n")
        log.write("original_bytes=" + str(to_read) + "\n")
    log.close()
print 'LOG FILE CREATED'
print str(mountpoint) + '/log.txt'
time.sleep(5)
y = commands.getstatusoutput("sudo umount " + partition)
print y[1]

```

Main()