UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

DIPARTIMENTO DI INFORMATICA
CORSO DI LAUREA MAGISTRALE IN SICUREZZA INFORMATICA

TESI DI LAUREA IN INFORMATICA FORENSE

REALIZZAZIONE DI FUNZIONALITÀ AVANZATE IN TINY CORE FORENSIC EDITION

Relatore:

Ch.mo Prof. Ugo LOPEZ

Laureando:

Francesca De Benedittis

ANNO ACCADEMICO 2020/2021

Sommario

1. Introduzione

1.1. La scienza forense

Quando si parla di scienza forense si fa riferimento a tecniche e metodologie scientifiche applicate a dei casi di investigazione tradizionali atte a identificare, preservare, acquisire, analizzare e presentare le prove nelle varie fasi di un'indagine giudiziaria. Il termine "forense" indica l'evidenza giuridica, infatti nell'uso moderno questo viene utilizzato come sinonimo di "legale" o comunque alla base del sistema penale. La competenza ed i temi trattati da questa disciplina sono vastissimi. (1)

Possiamo definire la scienza forense come la scienza applicata all'amministrazione della giustizia, che si avvale dell'utilizzo di tecnologie innovative come arma di lotta contro il crimine. Il sequenziamento del DNA, l'esame delle impronte digitali, l'analisi delle tracce biologiche, l'analisi e la datazione di inchiostri, sono tutte tecniche innovative finalizzate alla ricerca, alla raccolta e alla catalogazione di tutti gli elementi rinvenuti sulla scena di un crimine, dunque all'analisi tecnico-scientifica delle tracce raccolte.

La scienza forense è diventata essenziale nella lotta alla criminalità.

Balistica, criminologia, genetica forense, tossicologia forense sono solo alcune delle branche che compongono questa materia e che contribuiscono alla risoluzione di molti delitti e reati, che quasi certamente resterebbero impuniti se non fosse possibile utilizzare queste tecniche investigative.

1.2. *L'informatica forense*

Lo sviluppo delle tecnologie informatiche e di internet degli ultimi decenni ha influenzato la vita quotidiana di ciascun individuo e il mondo del lavoro.

La digitalizzazione dei processi produttivi della quasi totalità delle aziende ha determinato da un lato la necessità di archiviare e conservare dati e informazioni necessari allo svolgimento delle attività lavorative, dall'altro la diffusione dei cosiddetti "crimini informatici".

L'informatica forense è quella branca della scienza digitale che studia l'identificazione, la raccolta, l'acquisizione, la conservazione, l'analisi, la documentazione, la valutazione e la presentazione del dato digitale, al fine di risolvere casi di criminalità di diverso tipo.

1.2.1. Cenni storici

Possiamo far risalire il concetto di informatica forense agli inizi degli anni '80, quando la crescente diffusione dei primi Personal Computer porta all'aumento del numero di crimini cosiddetti informatici.

Nel 1984 l'FBI crea il CART (Computer Analisys Response Team) con il compito di fornire assistenza e supporto tecnico agli uffici nella ricerca e nel sequestro di prove informatiche, quindi nella conduzione delle indagini.

Durante gli anni'90 si assiste ad una crescita esponenziale della richiesta di indagini forensi in campo digitale e ciò porta allo sviluppo di nuovi software forensi, uno dei primi fu il The Coroner's Toolkit (TCT), una suite di programmi gratuiti di sicurezza informatica in grado di funzionare con diversi sistemi operativi correlati a UNIX (FreeBSD, OpenBSD, BSD / OS, SunOS / Solaris, Linux e HP-UX).

Fino a quel momento, tuttavia, la scienza digitale forense era priva di linee guida ufficiali e dunque di standard operativi riconosciuti; solo nel 2004 entrò in vigore la Convenzione sulla criminalità informatica, allo scopo di

accordare le normative vigenti nelle diverse nazioni, poi nel 2005 viene pubblicato lo Standard ISO 17025, che definisce i requisiti che un laboratorio deve soddisfare per garantire dati e risultati affidabili per specifiche prove.

In epoca più recente, gli scandali consumati in materia di violazione della privacy e sicurezza informatica sono moltissimi.

Il sito Wikileaks nel 2010, rese pubblici molti documenti segreti del governo americano dando vita alla più grossa diffusione di documenti riservati della storia. (2)

Nel 2018 la società britannica Cambridge Analytics ottiene illecitamente ed utilizza i dati personali di 87 milioni di utenti Facebook, senza il loro consenso, per attività di profilazione psicologica di massa, dando vita al più grosso scandalo politico di tutti i tempi. (3)

Ancora nel 2018 Google Plus, il social network di Google a causa di un bug espone i dati personali dei suoi utenti, costringendo la società a chiudere la versione del social dedicata al grande pubblico. (4)

Tra le fughe di informazioni più recenti c'è quella relativa al gigante della tecnologia Microsoft. Nel 2020 ha inavvertitamente esposto 250 milioni di record di assistenza clienti per più di tre settimane. Questo è successo in quanto Microsoft ha archiviato tali dati in cinque database Elasticsearch che sono però stati configurati in maniera errata. (5)

Con queste premesse, oggi l'informatica forense è diventata uno dei più efficaci mezzi di investigazione, al fine di ottenere prove da utilizzare in tribunale per giungere alla risoluzione di attività criminali.

L'informatica forense è diventata uno degli strumenti per combattere non solo i crimini strettamente legati all'informatica, come pedopornografia, cyberbullismo, frodi informatiche, ma risulta efficace anche in contesti differenti, quali il diritto civile, amministrativo, tributario, ecc.

L’indagine informatica forense è finalizzata alla raccolta e analisi delle prove digitali che potranno essere utilizzate durante la fase processuale e si compone di una fase preliminare, di ricerca dei dati e di una fase successiva, di analisi, in cui le informazioni raccolte vengono analizzate.

Una digital evidence o prova digitale è un documento informatico che contiene la rappresentazione di atti, dati e fatti giuridicamente rilevanti, che si presenta sotto forma di contenuti digitali (contenuti audio-visivi, registrazioni, testi etc.). (6)

Partendo dal presupposto che la prova digitale, di per sé è sensibile a ogni forma di manomissione, potendo essere facilmente alterata, corrotta o distrutta, una cattiva condotta nella fase della ricerca, può compromettere il suo utilizzo in sede dibattimentale.

Per questo, la fase della raccolta è senza dubbio la fase più delicata dell’intero iter investigativo, la più importante per un tecnico, in quanto un errore in fase di acquisizione comporta l’invalidazione della prova e quindi di tutto il suo contenuto, anche se essenziale all’interno di un’indagine.

Per questo, il modus operandi più adeguato e da preferire, se attuabile, è quello di estrapolare in maniera selettiva i dati dal dispositivo in cui sono memorizzati, così da lavorare sulla copia, preservando l’integrità del device.

Qualora non fosse possibile ottenere una copia dei dati, i tecnici incaricati dovranno preoccuparsi in primo luogo di isolare il device, per proteggerlo da eventuali manomissioni esterne.

L’acquisizione di una prova deve avvenire in maniera ripetibile, salvo impossibilità tecnica, ciò significa che deve essere possibile ripetere l’operazione ed ottenere il medesimo risultato. Se la ripetizione non fosse possibile, per non compromettere la validità processuale, la prova dovrebbe essere acquisita esclusivamente in contraddittorio tra le parti. Durante la

fase di analisi, invece, viene effettuato il confronto tra i dati raccolti e i risultati ottenuti nel corso delle indagini.

Per poter utilizzare gli esiti ottenuti dalle operazioni forensi compiute in questa fase in ambito processuale sarà necessario descrivere le tecniche e gli strumenti utilizzati, nonché gli eventuali processi necessari alla conclusione delle indagini. (7)

2. Stato dell'Arte

2.1. *Sistemi operativi forensi*

Alcuni dei supporti più utilizzati nell'informatica forense sono i sistemi operativi open source come ad esempio Linux. Questi sistemi consentono a programmatori sparsi nel mondo di collaborare ad un unico progetto migliorandolo e aggiornandolo continuamente.

Ancora più utili sono le distribuzioni Linux, ovvero dei sistemi operativi installabili basati su Kernel Linux personalizzabili attraverso l'installazione dei diversi applicativi esistenti o realizzati su misura.

Ma quando possiamo considerare un software "forense"? In generale, quando abbiamo la certezza che non venga alterato per nulla il reperto che si sta analizzando ed inoltre quando si hanno le garanzie relative alla catena di custodia. Allo stato dell'arte sono presenti diverse distribuzioni che posseggono tali requisiti; purtroppo però il progetto per alcune di queste è stato abbandonato (Iritaly, DEFT, Helix-Knoppix). Analizziamo nel dettaglio le distribuzioni che ancora sono utilizzate:

- CAINE
- Tsurgi
- Tiny Core Forensic Edition

2.1.1. CAINE

Con la parola CAINE si fa riferimento all'acronimo di Computer Aided Investigative Environment, è una distribuzione italiana Linux live basata su Ubuntu. Nasce da un'idea di Giancarlo Giustini nel 2008 sviluppata per il progetto Digital Forensics del Centro di Ricerca Interdipartimentale per la Sicurezza (CRIS) dell'Università di Modena e Reggio Emilia. Attualmente

il progetto è manutenuto da Nanni Bassetti. In questo progetto è presente un'interfaccia grafica molto semplice da utilizzare e permette il supporto dei quattro passi dell'investigazione, inoltre fornisce una compilazione semi automatica del report di analisi finale. (8) L'ultima versione di CAINE è la 11.0 ed è basata su Ubuntu 18.04 LTS ("Bionic Beaver") con supporto a lungo termine, infatti saranno garantiti aggiornamenti di sistema attraverso i repository di Ubuntu fino ad aprile 2023. Come le altre distribuzioni, CAINE si concentra sulla digital forensics combinando diversi tool in un comune package. (9) Una cosa molto interessante di CAINE è che tutti i dispositivi su cui vengono effettuate attività di analisi forense sono bloccati in modalità solo lettura, questo per evitare di "inquinare" della fonte di prova. Per poter modificare il file e quindi intervenire direttamente è necessario che l'operatore lanci esplicitamente un comando di sblocco o attraverso linea di comando o attraverso l'interfaccia grafica.

2.1.2. Tsurugi

Tsurugi è una distribuzione forense italiana che fornisce strumenti e supporto durante le attività di perizia informatica svolte da esperti d'informatica forense come quelli che operano nel settore delle Forze dell'Ordine o dai diversi consulenti informatici forensi presenti sul territorio. (10) Nasce nel 2018 da Giovanni Rettare e Marco Giorgi e viene presentata ufficialmente durante la conferenza AvTokio in Giappone. Come ogni distribuzione forense basata su Linux, anche Tsurugi è un ambiente composto da un sistema operativo Linux e può essere lanciato in maniera "live", senza così pregiudicare niente nel sistema preesistente sul PC, oppure può essere installato sul disco di un computer personale o anche su di una macchina virtuale. All'interno del sistema operativo troviamo

software differenti che consentono l'acquisizione forense, l'analisi forense ed ogni altra attività che si possa svolgere generalmente su di un personal computer.

Le versioni disponibili sono 3:

- **Tsurugi Lab:** è una piattaforma che consente acquisizioni e analisi forensi, basata su Linux Ubuntu Mate LTS con Kernel 4.18.5 a 64 bit, distribuita come ISO puo è essere avviata direttamente su DVD o pendrive ed è installabile su PC o macchina virtuale;
- **Tsurugi Acquire:** è un sistema dedicato unicamente alla fase di acquisizione forense, al più consente di eseguire limitatamente dei triage o delle verifiche di copie forensi; è basata su Linux Ubuntu Mate LTS con Kernel 4.18.5 a 32 bit in modo da renderla compatibile anche con dispositivi obsoleti;
- **Bento:** è una raccolta di tool per i sistemi operativi Linux, Windows e Mac, possono essere utilizzati su sistemi accesi e avviati per attività di analisi forense sul campo oppure di incident response.

Come in CAINE anche in Tsurugi i dispositivi sono tutti connessi in modalità "read-only" per prevenire eventuali scritture su hard disk o più genericamente dispositivi di memorie di massa (chiavi USB, schede di memoria, un tempo i floppy disk, etc...) oggetto di investigazioni; questo viene effettuato attraverso l'implementazione di un write blocker implementato a livello di kernel.

2.1.3. Tiny Core Forensic Edition

Tiny Core Forensic Edition nasce da un progetto di tesi dell'Università degli studi di Bari di Sara Ruffo, è una mini-distribuzione di 186 MB, il cui

core gira interamente in RAM ed ha un tempo di boot estremamente ridotto.

Per rendere questo strumento molto interessante nel campo dell'informatica forense oltre a dotarlo di molte applicazioni utili per l'analisi e l'acquisizione, si è provveduto all'implementazione di un Write Blocker chiamato BlockDev. Questo strumento è utilizzato nell'ambito di analisi forensi e permette la protezione dei dati presenti su supporti di memorizzazione oggetto dell'attività forense. I Write Blocker prevengono il rischio di modifiche sul supporto digitale oggetto dell'investigazione per evitare di invalidare potenziali prove involontariamente, assicurandone la connessione in read-only.

Il BlockDev utilizzato è di tipo Software based ovvero che agisce a basso livello quindi al di sotto del livello del filesystem e dei degli altri driver del dispositivo, assicurando quindi l'immodificabilità dei dati presenti sui supporti che si andranno ad analizzare.

Questo blocco ovviamente non è irreversibile; infatti attraverso un comando specifico è possibile riconvertire la partizione nuovamente in scrittura. (11)

Come già anticipato, per rendere questa distribuzione forense, nel progetto di tesi da cui si è partiti è stato necessario includere ad ogni avvio anche una serie di software in grado di supportare al meglio l'acquisizione e l'analisi forense di memorie.

Le applicazioni installate sono le seguenti:

- **Foremost**: è un programma di recupero-dati forense per Linux usato per recuperare i file utilizzando le strutture di dati, le intestazioni e i piè di pagina attraverso un processo conosciuto come file carving. Anche se scritto per le forze dell'ordine, è liberamente disponibile e può essere utilizzato come strumento generale di recupero dati. (12)

- **DdRescue:** è uno strumento di recupero-dati. Permette la copia dei dati da un file o da un dispositivo di memorizzazione ad un altro recuperando solo i blocchi necessari; infatti è in grado di recuperare le porzioni “buone” dei file nonostante si riscontrino errori di lettura. (13)
- **Dd:** diminutivo di Data Dump, è un comando Unix che permette la copia bit a bit di dati in blocco e opzionalmente consente di effettuare anche delle conversioni. (14)
- **Nmap:** Network Mapper è un software gratuito ed open source che esegue le scansioni delle reti e anche security auditing. Nasce con l'idea di poter scansionare in maniera rapida reti molto grandi, ma in realtà è in grado di funzionare molto bene anche verso singoli host. (15)
- **Rhash:** Recursive Hasher è una utility per console che permette di calcolare e verificare i valori hash dei file. Supporta diversi algoritmi come SHA1, SHA256, SHA512, HAS-160, MD4, MD5, Tiger, CRC32, DC++ TTH, ED2K, AICH, GOST R 34.11-94, RIPEMD-160, Snefru-128/256, BitTorrent BTIH, Whirlpool e EDON-R 256/512. (16)
- **Sleuthkit**: è una libreria e una collezione di utility open source basate su Unix e Windows per estrarre dati da unità disco e altre memorie in modo da facilitare l'analisi forense dei sistemi informatici. (17)
- **Volatility3:** è il framework più utilizzato al mondo per l'estrazione di artefatti digitali da campioni di memoria volatile (RAM). Le tecniche di estrazione vengono eseguite in modo completamente indipendente dal sistema in esame, ma offrono visibilità sullo stato di runtime del sistema. (18)

- **Netcat:** è un programma open source a riga di comando che legge e scrive dati attraverso le connessioni di rete, utilizzando il protocollo UDP o TCP. (19)
- **Ewfacquire:** è una utility che permette l'acquisizione di dati multimediali da una fonte memorizzandoli in formato EWF (Expert Witness Compression Format). Fa parte del pacchetto libewf, libreria che supporta il formato di compressione ewf. (20)

3. Analisi e Progettazione

3.1. *Definizione del problema*

L'obiettivo di questo progetto di tesi è stato quello di ampliare la versione base di Tiny Core Forensic Edition mediante la realizzazione e l'integrazione di funzionalità avanzate.

Per prima cosa, il sistema operativo necessita di integrare un'interfaccia grafica minimale, in modo da renderlo più semplice da usare.

Inoltre, come precedentemente spiegato, essendo tutte le partizioni bloccate in sola lettura, per una eventuale estrazione forense, potrebbe essere necessario sbloccare i drive per poterne fare una copia, o memorizzare delle informazioni ricavate dagli svariati tipi di acquisizioni forensi. Per rendere più intuitiva questa operazione si vuole realizzare una piccola utility grafica per gestire il BlockDev già implementato.

Infine, si vuole realizzare un'applicazione per acquisire file e informazioni in modo forense dagli spazi aziendali condivisi di SharePoint. Verrà quindi implementato un tool grafico in grado di estrarre informazioni dai siti di SharePoint disparati previa conoscenza delle credenziali di accesso.

Questa esigenza nasce per indagare su eventuali violazioni dei parametri RID (riservatezza, integrità e disponibilità) negli spazi aziendali di Microsoft SharePoint.

3.2. *Gui Tiny Core FE*

L'obiettivo iniziale consiste nell'integrare alla distro "Tiny Core Forensic Edition v1.0" un'interfaccia grafica per poter svolgere le funzioni base del sistema operativo in modo più pratico e veloce. Uno dei requisiti principali è quello di mantenere il sistema Light-Weight; per questo motivo

occorre rispettare dei limiti di dimensione della ISO e requisiti di memoria e velocità minimali, per permettere la compatibilità con tutti i dispositivi presenti. Senza dover reinventare la ruota si cercherà, quindi, di individuare le estensioni che rendono diversa la versione base "Core" (11 MB) dalla versione grafica "Tiny Core" (16 MB). Queste, successivamente, verranno prima testate e poi installate a partire dalla ISO Forense già implementata, senza implementare dal principio funzioni già esistenti e usufruibili.

3.2.1. Requisiti

Nel sistema operativo Tiny Core Forensic Edition si vuole:

- Integrare un'interfaccia grafica per l'utilizzo;
- Integrare un'applicazione per la gestione dei file;
- Consentire sempre l'utilizzo del terminale a linea di comando;
- Definire un wallpaper di sfondo per contraddistinguere il sistema.

3.3. *App BlockDev Gui*

Per aumentare l'usabilità del sistema, si vuole realizzare un applicativo in grado di disabilitare o riabilitare il blocco in scrittura su un dispositivo. Nel sistema base questa funzionalità è consentita solo tramite comandi su terminale. L'applicazione grafica da realizzare eseguirà questa operazione tramite dei pulsanti che identificheranno il dispositivo, in modo da consentire più facilmente il blocco o lo sblocco delle partizioni desiderate, semplificando quindi il processo di estrazione forense.

Per realizzare l'applicazione BlockDevGui si è partiti come spunto dall'app MountTool preesistente in Tiny Core. Questa consente di montare i dispositivi riconosciuti automaticamente dal sistema in maniera molto intuitiva attraverso l'utilizzo di pulsanti, come spiegheremo immediatamente.

All’avvio dell’applicazione sono presenti nella finestra tanti bottoni per quanti dispositivi sono stati riconosciuti dal sistema più quello di Refresh; questo consente l’identificazione (e quindi la relativa visualizzazione di nuovi pulsanti) di partizioni inserite successivamente. Inizialmente questi bottoni appaiono di colore rosso in quanto la relativa partizione risulta essere solo collegata al sistema; al click però la partizione selezionata verrà montata correttamente (in sola lettura grazie alla presenza del BlockDev) e il bottone cambierà il colore in verde; al successivo click la partizione verrà smontata colorando il pulsante di rosso e così via.

La nostra applicazione come il MountTool sarà molto intuitiva ed utilizzerà anch’essa dei bottoni. L’idea è quella di mostrare all’avvio solo le partizioni, che sono state già precedentemente montate, ed un pulsante di Refresh. La finestra dell’applicazione deve essere gestita in modo dinamico mediante la funzione di autoresize, per permettere l’adattamento automatico delle dimensioni in base al numero dei pulsanti da mostrare. Il colore dei bottoni dipenderà dal tipo di montaggio: rossi per identificare che la partizione è in sola lettura, verdi se invece è in scrittura. Il click sul pulsante intercetterà lo stato della partizione selezionata e ne convertirà il montaggio nello stato opposto a quello in cui si trova.

Quindi, se una partizione è in sola lettura, il bottone relativo sarà rosso e al click la partizione verrà convertita in scrittura facendo diventare il pulsante verde. Anche nella nostra applicazione il pulsante di Refresh sarà in grado di aggiornare la schermata eliminando o inserendo partizioni che sono state smontante/montate successivamente all’avvio dell’app.

Nonostante la presenza di questo pulsante, si è pensato di intervenire anche qualora si dovesse inavvertitamente cliccare su una partizione che è stata precedentemente smontata ma che risulta ancora visualizzata (magari a causa di un mancato refresh appunto). Il click su questa partizione porterà

ad un aggiornamento della schermata anziché all'utilizzo dei comandi di blocco/sblocco.

Inoltre, prima di abilitare la partizione in scrittura, onde evitare disabilitazioni accidentali del BlockDev, verrà sempre mostrata una finestra di conferma utente.

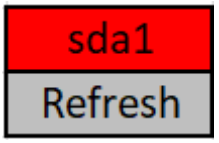
3.3.1. Requisiti

- L'applicazione sarà dotata di un'interfaccia grafica;
- I drive saranno identificati mediante un pulsante rosso, se montato in sola lettura, e verde, se montato in scrittura;
- Verrà richiesta sempre un'ulteriore conferma di montaggio in scrittura tramite una apposita finestra;
- L'applicazione sarà integrata in Tiny Core FE con possibilità di avvio tramite icona wbar desktop.

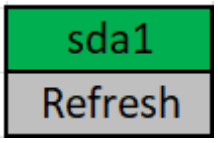
3.3.2. Casi d'uso

- Avvio applicazione con nessun dispositivo montato: l'applicazione mostrerà solamente il pulsante di refresh quando nessun dispositivo è montato al sistema.

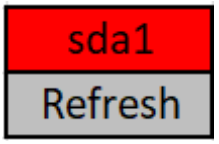
- Avvio applicazione con sda1 montato o refresh dopo montaggio: l'applicazione mostrerà il pulsante rosso sda1 e quello di refresh.



- Click sul pulsante sda1 (rosso): il dispositivo montato in sola lettura verrà montato in scrittura, previa autorizzazione utente nella finestra di conferma.



- Click sul pulsante sda1 (verde): il dispositivo montato in scrittura verrà montato in sola lettura. L'operazione sarà immediata senza la conferma dell'utente.



3.4. *App SharePoint Forensic Downloader*

La rivoluzione informatica e digitale degli ultimi decenni, come detto, ha coinvolto la vita quotidiana di ciascun individuo e, di conseguenza, il mondo del lavoro.

Le aziende grandi e piccole, alle prese con i processi di informatizzazione e digitalizzazione delle tecnologie di lavoro, hanno innovato e modificato i processi produttivi, manifestando esigenze nuove e diverse rispetto al passato.

Condividere uno o più documenti tra i dipendenti, progettare un sito web interno all'azienda e accessibile solo al personale, creare un'applicazione in cui le persone interne all'azienda possono comunicare e connettersi con le risorse esterne, realizzare uno spazio di lavoro digitale riservato e controllato, avere a disposizione uno spazio di archiviazione illimitato sono alcune delle necessità che il mondo del lavoro ha manifestato negli ultimi tempi.

Proprio per venire incontro a queste esigenze e soddisfarle, Microsoft nel 2003 ha sviluppato e commercializzato una piattaforma di collaborazione

con l'obiettivo di connettere le persone e consentire la condivisione e la comunicazione nella stessa azienda.

Inizialmente viene rilasciato come strumento utile alla sola gestione dei documenti, successivamente è diventato però un sistema in grado di gestire i diversi contenuti. Questa piattaforma ha consentito per la prima volta di creare delle reti intranet aziendali in modo da facilitare la condivisione e la comunicazione all'interno dell'azienda. (21)

È un programma completamente integrato con il pacchetto Microsoft Office che gira lato server e permette quindi la creazione e la distribuzione di particolari siti web. Una funzionalità interessante è quella che più utenti situati in posti differenti possono visualizzare lo stesso documento in contemporanea. La modifica, al contrario, è possibile solo da un utente per volta, ma quelli connessi al file hanno accesso alle modifiche in maniera live. L'autenticazione avviene inserendo un nome utente e password in fase di login.

Nel panorama informatico forense a valle di indagini aziendali potrebbe essere necessaria l'acquisizione delle informazioni archiviate in uno SharePoint. Allo stato dell'arte manca un tool che faccia questo in modo forense.

3.4.1. Requisiti

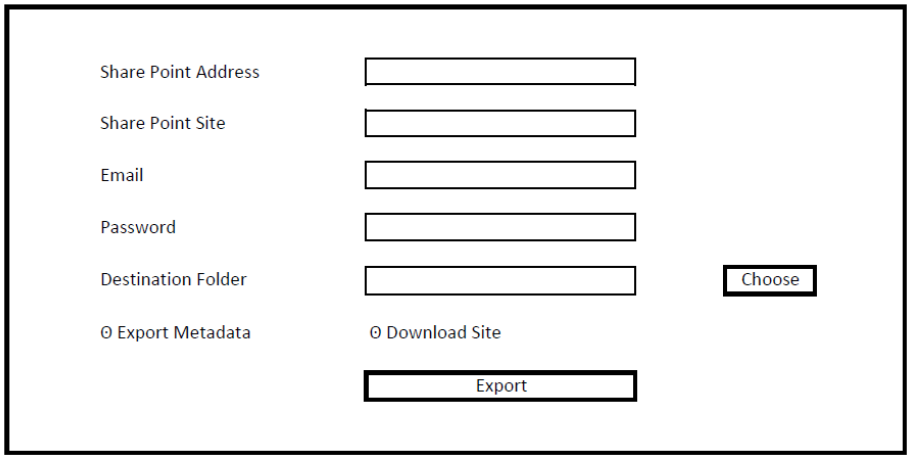
Si vuole realizzare un'applicazione che dovrà soddisfare i seguenti requisiti:

- Consentire il login ad un qualsiasi SharePoint aziendale fornendo informazioni quali indirizzo, sito, e-mail e password;
- Effettuare il download forense dell'intero contenuto del sito;
- Il contenuto del sito deve rispettare l'alberatura;
- Di tutti i file presenti sul sito verrà scaricata ogni singola versione, qualora queste siano presenti;

- Il blocco dati estratto sarà archiviato in un file zip;
- Al fine di rendere l'estrazione forense verrà generato un file di testo contenente il doppio hash (MD5 e SHA256) come sigillo digitale dell'archivio;
- Scaricare i metadati delle singole cartelle e dei singoli file presenti nell'intero sito;
- Generare un file Excel che contenga l'alberatura dell'intero SharePoint;
- L'applicazione sarà dotata di una interfaccia grafica;
- L'applicazione sarà integrata in Tiny Core FE con possibilità di avvio trami icona wbar desktop.

3.4.2. Interfaccia Grafica

Di seguito la bozza iniziale di quella che sarà poi l'applicazione finale



Nella finestra verranno gestiti i seguenti oggetti:

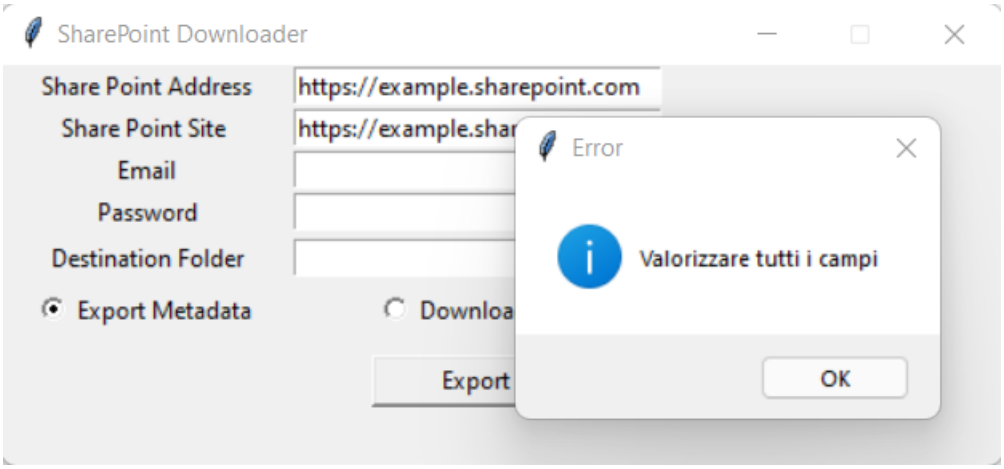
- SharePoint Address: text box in cui dovrà essere inserito l'indirizzo dello SharePoint che si vuole acquisire;
- SharePoint Site: text box in cui dovrà essere inserito il sito relativo all'indirizzo SharePoint che si vuole acquisire;

- e-mail: text box in cui dovrà essere inserita la mail per poter accedere all'indirizzo SharePoint precedentemente specificato;
- Password: text box di tipo password, quindi con i caratteri nascosti, in cui dovrà essere inserita la password per poter accedere all'indirizzo SharePoint;
- Destination Folder e Choose: text box in cui attraverso il pulsante Choose sarà possibile selezionare il percorso desiderato in cui salvare gli export desiderati, mediante una finestra grafica di Explorer;
- Export Metadata e Download Site: radio button esclusivi; a seconda di quello selezionato l'utente deciderà se scaricare un file txt contenente l'elenco dei metadati relativi a tutti i file e le cartelle presenti sullo SharePoint ed un file xlsx con relativa alberatura del sito (Export Metadata) oppure l'intero contenuto del sito comprese le differenti versioni dei file qualora presenti (Download Site);
- Export: pulsante che avvierà l'esportazione della modalità desiderata.

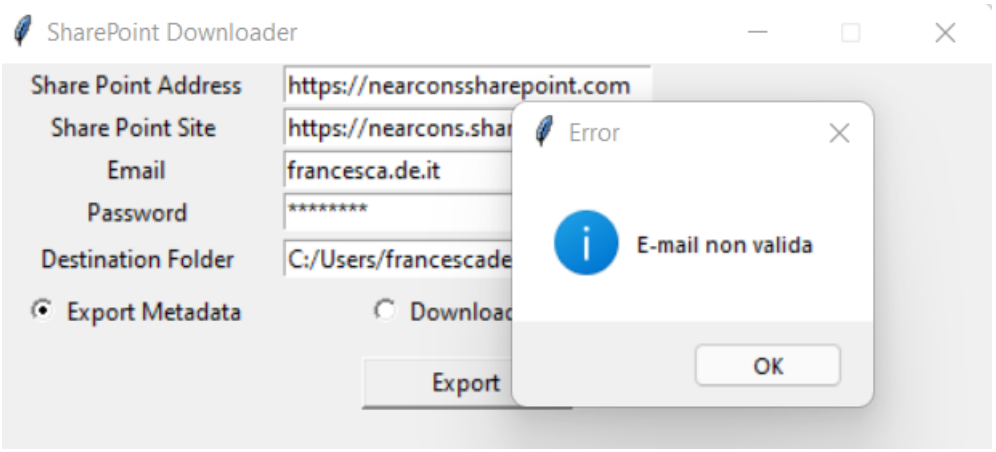
3.4.3. Eccezioni Gestite

Come in tutte le applicazioni esistenti è stato necessario gestire delle eccezioni che potrebbero verificarsi a causa di errori dovuti agli utenti o al sistema. Le principali gestite sono state:

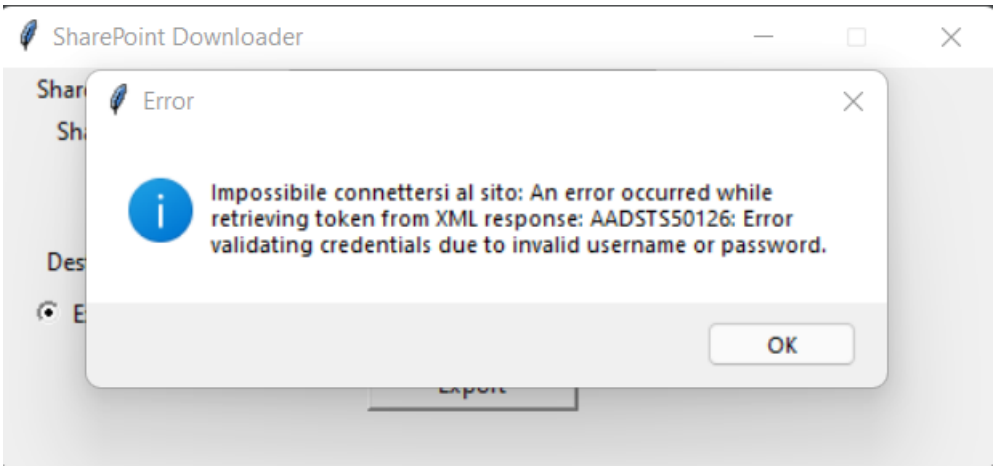
- mancanza di input fornito dall'utente: qualora non venga compilato anche solo uno degli input necessari all'applicazione per funzionare viene mostrato il messaggio "Valorizzare tutti i campi";



- errore di compilazione della mail: qualora la mail inserita dall'utente abbia un formato non valido (gestito attraverso la regular expression: "^[a-zA-Z0-9_+&*-]+(?:\.[a-zA-Z0-9_+&*-]+)*@(?:[a-zA-Z0-9-]+\.)+[a-zA-Z]{2,7}$")" viene visualizzato il messaggio " E-mail non valida";



- impossibilità di connessione al sito di SharePoint: sia che le credenziali fornite dall'utente siano errate, sia che il nome del sito inserito sia inesistente o per eventuali problemi di connessione potrebbe essere impossibile raggiungere il sito e quindi effettuare il download degli elementi desiderati, in questo caso verrà visualizzato il messaggio "Impossibile connettersi al sito:" più la relativa spiegazione che ritorna direttamente da SharePoint;



3.4.4. Casi d'uso

Perché l'applicazione funzioni correttamente sarà necessario essere connessi ad una rete internet ed inoltre bisognerà essere in possesso delle credenziali (e-mail e password) per poter accedere al sito interessato. Adesso analizzeremo i due casi d'uso per espletare delle funzioni messe a disposizione dall'applicazione che vogliamo realizzare.

3.4.5. Download Forense File

All'avvio dell'applicazione verrà mostrata una finestra contenente tutti gli input che dovranno essere forniti all'applicazione per funzionare correttamente. Una volta compilati indirizzo, sito, e-mail e password attraverso il bottone Choose l'utente potrà selezionare il percorso che ritiene più opportuno in cui i file scaricati saranno salvati. Dovrà inoltre fornire un input relativo ai radio button presenti nella schermata. Selezionando il radio button "Download Site" l'utente potrà scaricare l'intero contenuto del sito con le diverse versioni dei file qualora questi fossero presenti.

Dopo aver premuto il pulsante export bisognerà attendere qualche minuto perché nella cartella precedentemente selezionata compaia il file zippato

più un file di testo in cui saranno presenti i doppi hash relativi all'archivio estratto.

3.4.6. Download Forense Metadati

Alla stessa maniera dopo aver fornito i dati precedentemente indicati dopo la selezione del radio button Export Metadata sarà possibile scaricare un file di testo contenente tutti i metadati presenti relativi ad ogni singolo file e cartella presenti sullo SharePoint. Si potrà accedere ad informazioni quali nome, data e ora di creazione di file e cartelle, utenti che li hanno creati, percorso relativo, utenti con cui sono condivise queste informazioni, quante versioni sono presenti di un determinato file e da chi sono state create. Inoltre, verrà scaricato anche un file Excel che contiene l'intera alberatura del sito.

4. Implementazione

Dati i requisiti analizzati nel capitolo precedente, si è scelto di realizzare delle funzionalità avanzate per ampliare la distro Linux forense minimale Tiny Core FE.

4.1. *Gui Tiny Core FE*

Per poter dotare il sistema di un'interfaccia grafica sono state affrontate due fasi. La prima per identificare le estensioni necessarie e la seconda per eseguire il remaster del sistema.

4.1.1. Tiny Core Gui Tcz

Come definito nei requisiti andremo a individuare le estensioni grafiche di Tiny Core per installare sistema operativo forense.

Il sito ufficiale di Tiny Core mostra nel dettaglio i pacchetti che occorre installare per ottenere l'interfaccia grafica a partire dalla versione "Core" (Tiny Core = Core + Xvesa.tcz + Xprogs.tcz + aterm.tcz + fltk-1.3.tcz + flwm.tcz + wbar.tcz).

In una prima fase si è deciso di installare live i seguenti pacchetti per verificare la compatibilità con il sistema "Tiny Core Forensic Edition v1.0" utilizzando i seguenti comandi (dopo aver connesso a Internet il SO):

```
tce-load –wi [nome_pacchetto].tcz
```

Dopo varie prove e ricerche si è deciso di installare nel dettaglio le seguenti estensioni, dato che hanno riportato maggiori compatibilità e performance:

- **Xorg-7.7.tcz**: server X di Linux che sostituisce Xvesa, include Xprogrs e presenta maggiore compatibilità con le varie periferiche hardware presenti;
- **aterm.tcz:** applicativo del terminale grafico usufruibile su gui;
- **wbar.tcz**: barra delle applicazioni presente nel desktop grafico;
- **flwm_topside.tcz:** gestione delle finestre grafiche in ambiente grafico, versione avanzata di flwm;
- **xf86-video-intel.tcz:** driver grafico che include la compatibilità con Xorg e permette una corretta visualizzazione con la maggior parte delle schede video.

Dopo aver testato il corretto funzionamento della nuova distro eseguita in modo frugale si è provveduto ad eseguire un remaster del sistema (22).

4.1.2. Ezremaster ISO Gui

Dopo aver indentificato le "App" necessarie per donare a *Tiny Core Forensic Edition* un'interfaccia grafica si è provveduto ad eseguire il remaster della ISO.

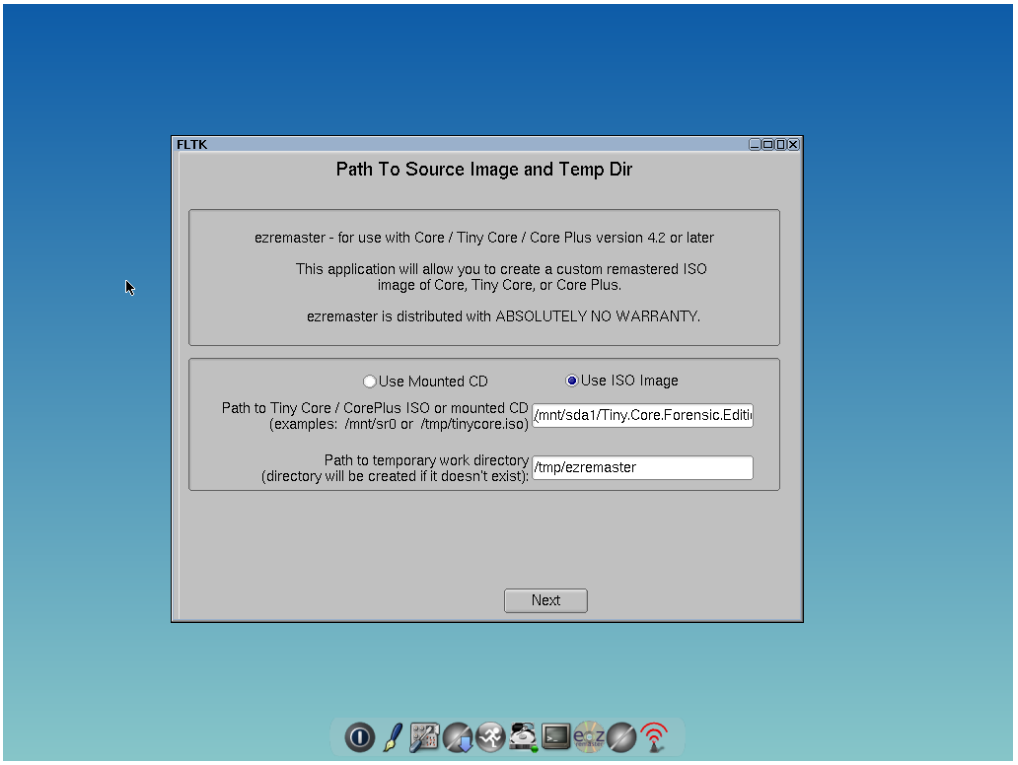
Il Remastering è il processo di editing dell'immagine di sistema e permette la produzione di una nuova immagine ISO.

Tale operazione è stata eseguita attraverso una utility con interfaccia chiamata Ezremaster. Dato che questa è un'applicazione GUI usufruibile solo con interfaccia grafica, tutti i remaster eseguiti sono stati realizzati utilizzando la ISO di "Core Plus" che presenta tale utilità già attiva e preinstallata. (23)

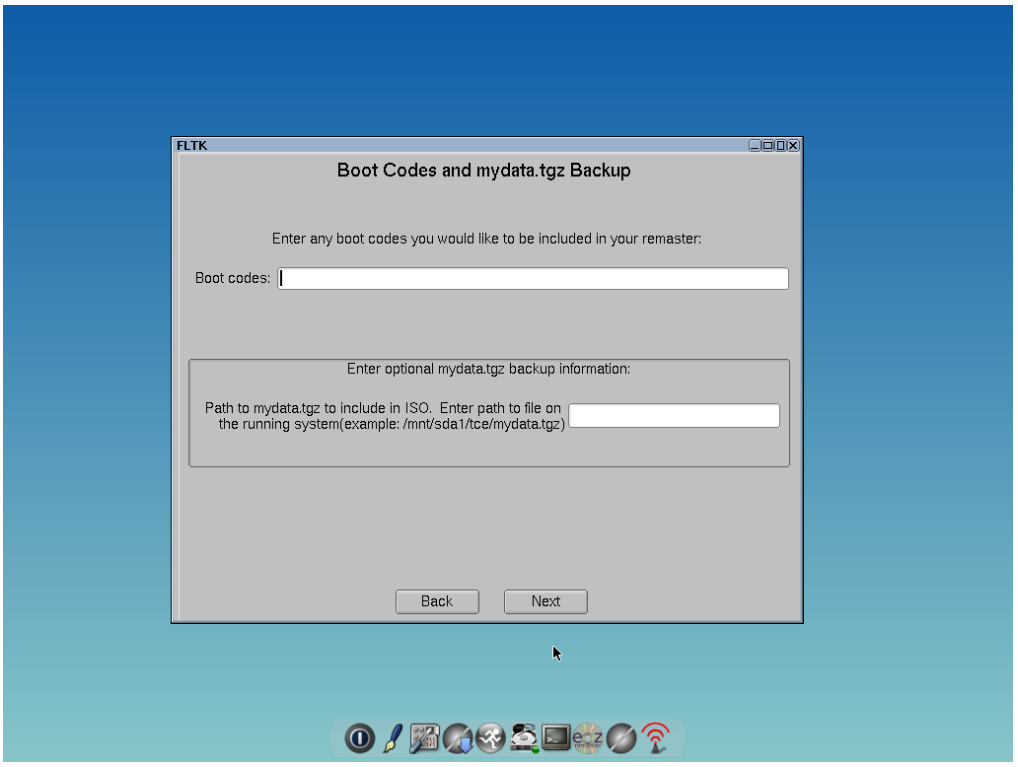
http://tinycorelinux.net/11.x/x86/release/CorePlus-current.iso

Ezremaster permette di rimasterizzare una ISO partendo da quella già montata nel sistema o da un file ISO diverso. Nel nostro caso come base è stata selezionata quella di “Tiny Core Forensic Edition v1.0”. (11)

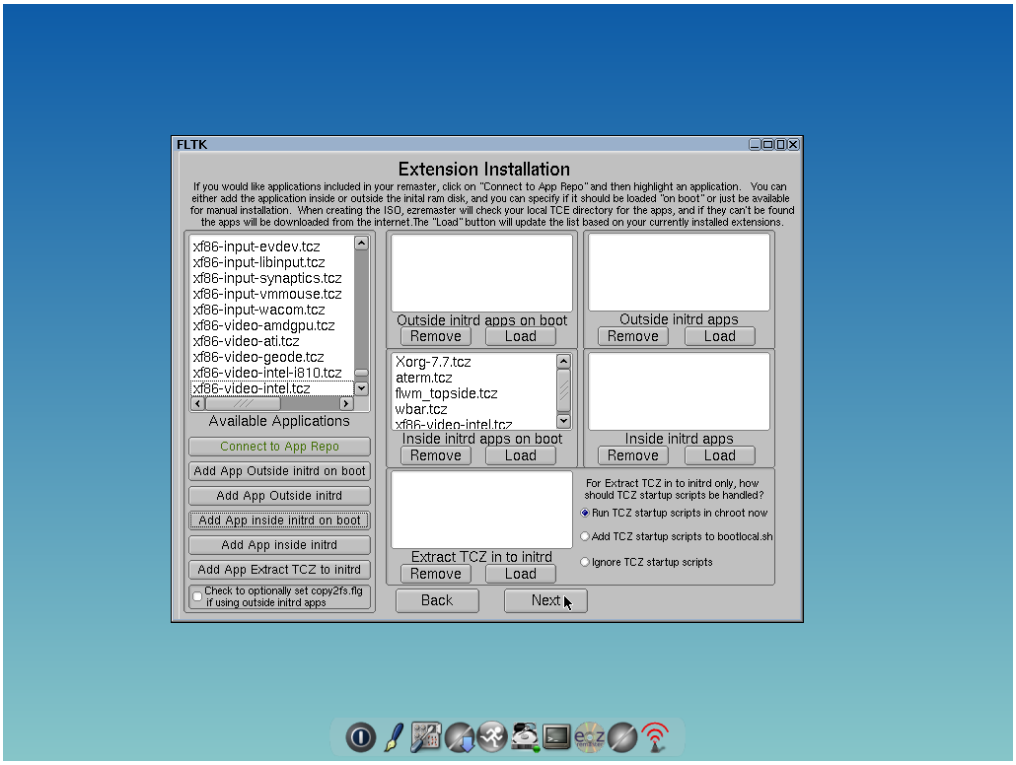
https://github.com/SaraRuffo/Tiny-Core-Forensic-Edition---Tesi/releases/tag/v1.0



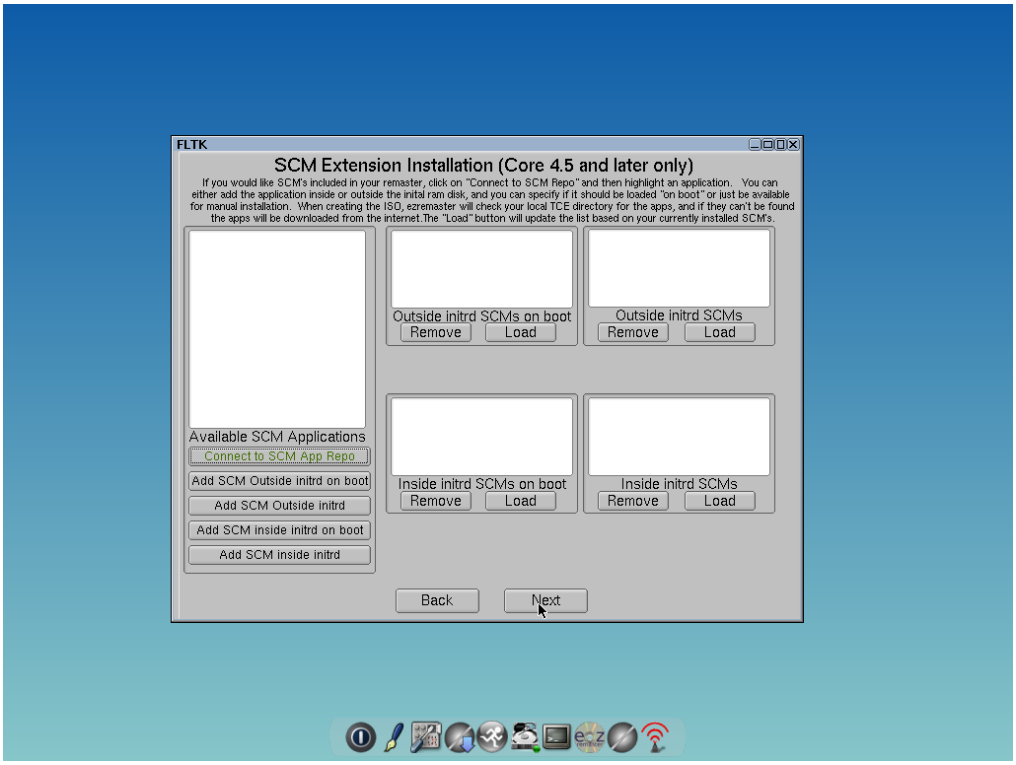
Non è stato selezionato nessun boot code in quanto già presente nel sistema di partenza.



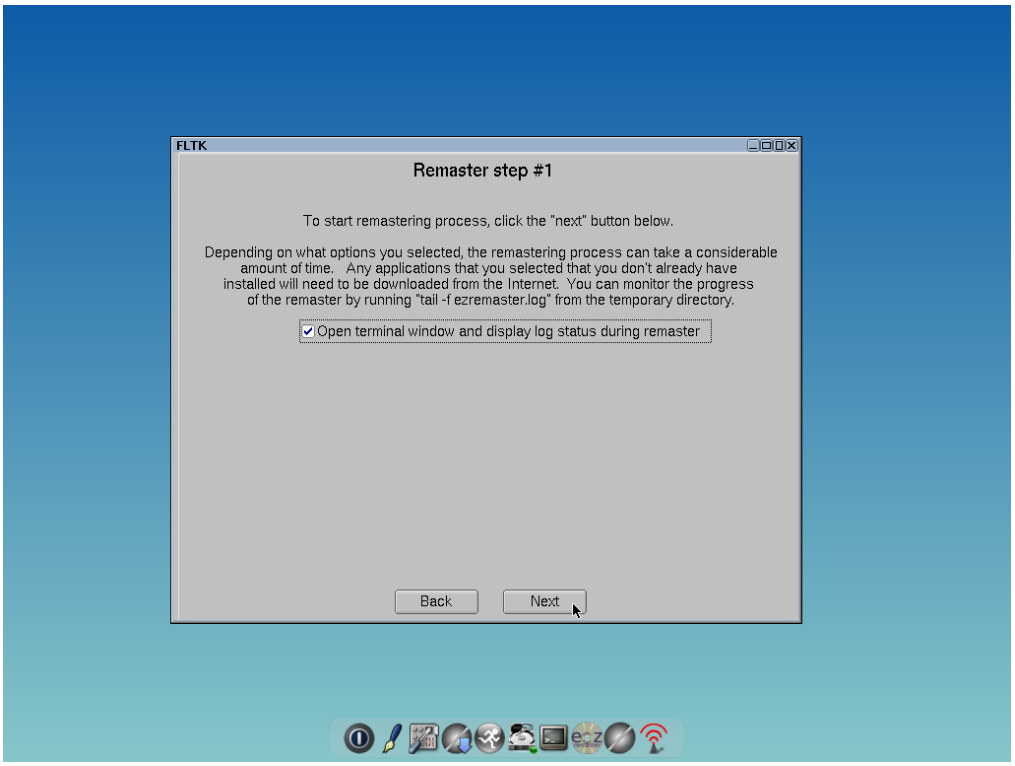
Tutti i pacchetti da rimasterizzare sono stati prelevati direttamente dal repository ufficiale utilizzando la funzione “Connect to App Repo” che visualizza l’elenco completo di tutti i TCZ ufficiali. Sono stati selezionati quelli precedentemente elencati e inclusi nella nuova ISO mediante la funzione “Add App inside initrs on boot”.



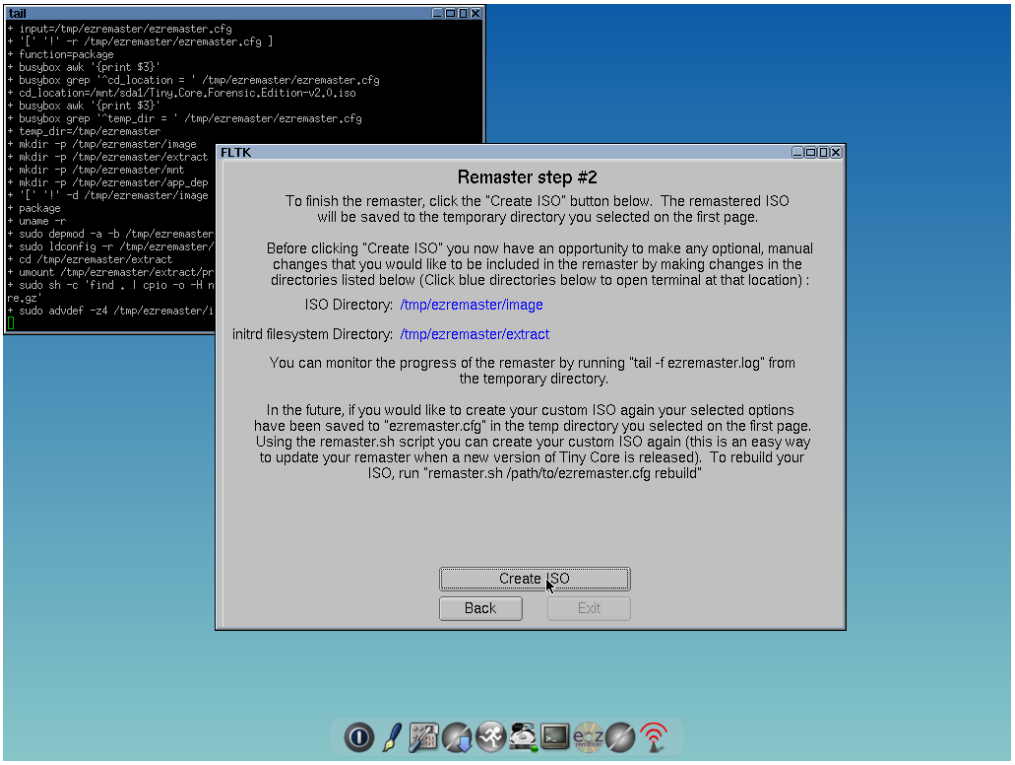
Non è stata selezionata alcuna estensione.



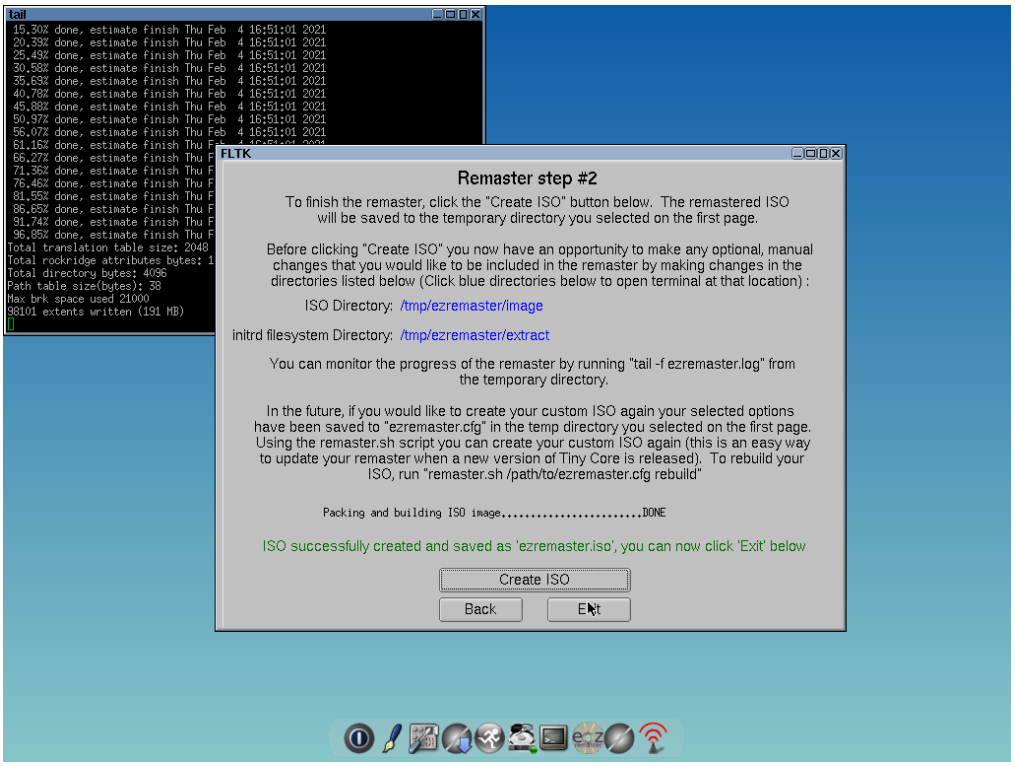
È stata spuntata la selezione della visualizzazione del processo nel terminale e si è proseguito con la schermata successiva.



Si è selezionata la funzione di creazione ISO.



Dopo qualche minuto, l'operazione di remaster è terminata.



Concluso il processo si è provveduto a prelevare ed archiviare esternamente la nuova ISO dalla directory /tmp/ezremaster. Infine la nuova ISO è stata rinominata con il nome “Tiny.Core.Forensic.Edition-v2.0”.

4.2. *BlockDev Gui*

Per realizzare l’applicazione è stato utilizzato Python, linguaggio di programmazione dinamico orientato agli oggetti.

Oltre ad essere un linguaggio fortemente utilizzato, un ulteriore vantaggio è quello di poter avviare un sorgente .py in maniera molto semplice in quanto Tiny Core ha già preinstallato il compilatore utilizzabile per molti tipi di esecuzioni software.

Oltre a definire l’interfaccia grafica come stabilito nei requisiti, è stato necessario interrogare il sistema operativo per identificare i dispositivi montati. La sintassi del comando è:

```
partitions = psutil.disk_partitions()
```

Se il dispositivo è montato in sola lettura e si vuole montare in scrittura, bisognerà far eseguire al sistema i seguenti comandi:

```
sudo blockdev --setrw [device]
sudo mount –o remount,rw [device]
```

Nel caso si volesse ripristinare il blocco in scrittura e quindi il montaggio in sola lettura bisognerà eseguire:

```
sudo blockdev --setro [device]
sudo mount –o remount,ro [device]
```

Python per impartire dei comandi di SO utilizza il seguente codice:

```
os.popen("comando")
```

Per realizzare l'app sono state utilizzate diverse librerie quali os, PsUtil, tkinter e functools. Per la libreria PsUtil sono state eseguite delle ulteriori operazioni per renderla utilizzabile dal compilatore installato in TinyCore FE.

4.2.1. Python3.6 PsUtil

Una libreria molto interessante per visualizzare e gestire informazioni e comandi di sistema è PsUtil. Tutte le funzioni richiamate restituiscono i dati in forma di liste.

Quindi, per visualizzarle in una determinata maniera o estrarne solo alcune occorre manipolarle a seconda dei diversi obiettivi.

Questa libreria non è presente nel repository ufficiale, quindi per la sua integrazione al sistema si è dovuta eseguire una particolare procedura.

Reperito l'archivio psutil-5.6.3.tar sono stati necessari alcuni comandi innanzitutto per compilarlo in TinyCore e successivamente per trasformarlo in formato tcz.

La libreria è stata prelevata:

http://www.tinycorelinux.net/10.x/x86_64/tcz/src/python3.6-psutil/ (24)

Una volta generato il tcz è stato archiviato per un successivo ezremaster.

Tale operazione richiede una serie di passaggi. Infatti, il codice sorgente Python della libreria deve essere prima compilato e poi installato.

Successivamente si dovrà compattare la destinazione di installazione tramite l'utility squashfs-tool per generare il pacchetto tcz da includere nel remaster al fine di essere caricato ad ogni avvio del sistema frugale.

La sintassi dei comandi eseguiti è la seguente:

```
#estrazione e compilazione libreria
cp /mnt/sda1/cd psutil-5.6.3 /tmp
tce-load -wi compiletc.tcz
tce-load -wi python3.6-dev.tcz
mkdir /tmp/python3.6-psutil
cd /tmp/psutil-5.6.3
python3.6 setup.py build
sudo python3.6 setup.py install --root=/tmp/python3.6-psutil
#generazione tcz
tce-load -i squashfs-tools.tcz
cd /tmp
mksquashfs python3.6-psutil python3.6-psutil.tcz
#copia python3.6-psutil.tcz
cp /tmp/python3.6-psutil.tcz /mnt/sda1/
#installazione python3.6-psutil.tcz
cd /tmp
mv -v python3.6-psutil.tcz /etc/sysconfig/tcedir/optional
cd /etc/sysconfig/tcedir
echo python3.6-psutil.tcz >> onboot.lst
tce-load -i python3.6-psutil.tcz
```

4.2.2. Python3.6 Tkinter

La libreria Tkinter, indispensabile per realizzare grafica in python è presente nel repository ufficiale. Dopo un'analisi si è riscontrato che la stessa è già installata nella ISO base di Tiny Core Forensic Edition. Non si è eseguita quindi alcuna operazione oltre a quella di verifica (25).

4.2.3. Python3.6 Blockdevgui

Il risultato della fase di programmazione è il file blockdevgui.py che verrà successivamente incluso in Tiny Core avviabile tramite il comando

```
python 3.6 blockdevgui.py
```

4.2.4. Script Eseguibile Blockdevgui

Il codice Python del blockdevgui implementato è stato inserito nella directory /usr/local/bin con il nome blockdevgui.py al fine di renderlo eseguibile mediante uno script. Il risultato è stato inserito nel medesimo percorso con il nome blockdevgui.sh realizzato con il seguente codice:

```
#!/bin/sh
cd /usr/local/bin
python3.6 blockdevgui.py
```

Per rendere eseguibile lo script è stato necessario lanciare il seguente comando dal terminale dal percorso /usr/local/bin:

```
chmod +x blockdevgui.sh
mv blockdevgui.sh blockdevgui
```

Adesso è possibile avviare l'utility da terminale tramite il comando blockdevgui indipendentemente dal percorso in cui ci si trova.

4.2.5. Wbar Link Blockdev

Per integrare una icona direttamente nella wbar grafica di TinyCore è stata analizzata la guida del wiki ufficiale che ha permesso di realizzare un link grafico nella barra mediante l'esecuzione dei seguenti comandi (26):

```
#creazione directory
cd /tmp/tce
mkdir ondemand
#copia icona
cp /mnt/sda1/icona.png /tmp/tce/ondemand
cd /tmp/tce/ondemand
mv icona.png BlockDev.img
#creazione script icona
cat > BlockDev.sh
#!/bin/sh
```

```
blockdevgui CTRL+C
chmod +x BlockDev.sh
mv BlockDev.sh BlockDev
```

Si precisa che icona.png è una semplice immagine prelevata dal web.

4.2.6. Creazione blockdevgui.tcz

Per creare il tcz della nuova app grafica realizzata è stato necessario creare un ambiente di lavoro provvisorio che mantenga i nomi originali dei percorsi in cui sono contenuti i file che dovranno essere reinstallati a partire dal tcz ad agni avvio live del sistema operativo.

La sintassi dei comandi eseguiti è la seguente:

```
#creazione ambiente di lavoro
mkdir /tmp/blockdevgui
cd /tmp/blockdevgui
mkdir –p usr/local/bin
mkdir – t tmp/tce/ondemand
cp /tmp/tce/ondemand/BlockDev /tmp/blockdevgui/tmp/tce/ondemand
cp /tmp/tce/ondemand/BlockDev.img  /tmp/blockdevgui/tmp/tce/ondemand
cp /usr/local/bin/blockdegui  /tmp/blockdevgui/local/bin
cp /usr/local/bin/blockdegui  /tmp/blockdevgui/local/bin
#generazione tcz
tce-load -i squashfs-tools.tcz
cd /tmp
mksquashfs blockdevgui blockdevgui.tcz
#copia blockdevgui.tcz
cp /tmp/ blockdevgui.tcz /mnt/sda1/
```

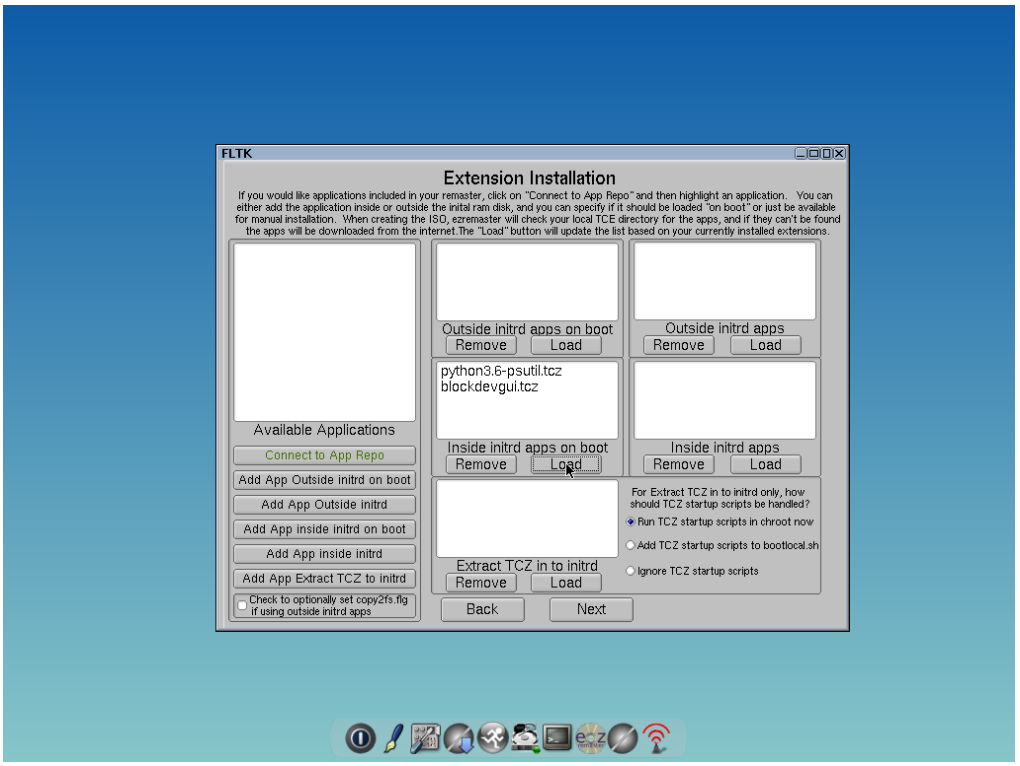
4.2.7. Ezremaster Tcz esterni

Create e realizzate le estensioni “python3.6-psutil.tcz” e “blockdevgui.tcz” si è provveduto ad un nuovo remaster. Per includere tcz esterni è stato necessario eseguire determinati passi. Per prima cosa è stato avviato il sistema operativo Core Plus e dal terminale sono stati eseguiti i seguenti comandi:

```
#psutil
```

```
cp /mnt/sda1/python3.6-psutil.tcz /tmp
cd /tmp
mv -v python3.6-psutil.tcz /etc/sysconfig/tcedir/optional
cd /etc/sysconfig/tcedir
echo python3.6-psutil.tcz >> onboot.lst
tce-load -i python3.6-psutil.tcz
#blockdevgui
cp /mnt/sda1/ blockdevgui.tcz /tmp
cd /tmp
mv -v blockdevgui.tcz /etc/sysconfig/tcedir/optional
cd /etc/sysconfig/tcedir
echo blockdevgui.tcz >> onboot.lst
tce-load -i python3.6-psutil.tcz
```

Rispetto alla procedura descritta in precedenza i TCZ da includere sono stati selezionati in automatico cliccando sul pulsante Load presente nella sezione “Inside initrd apps on boot” dopo avere eseguito i comandi illustrati precedentemente.



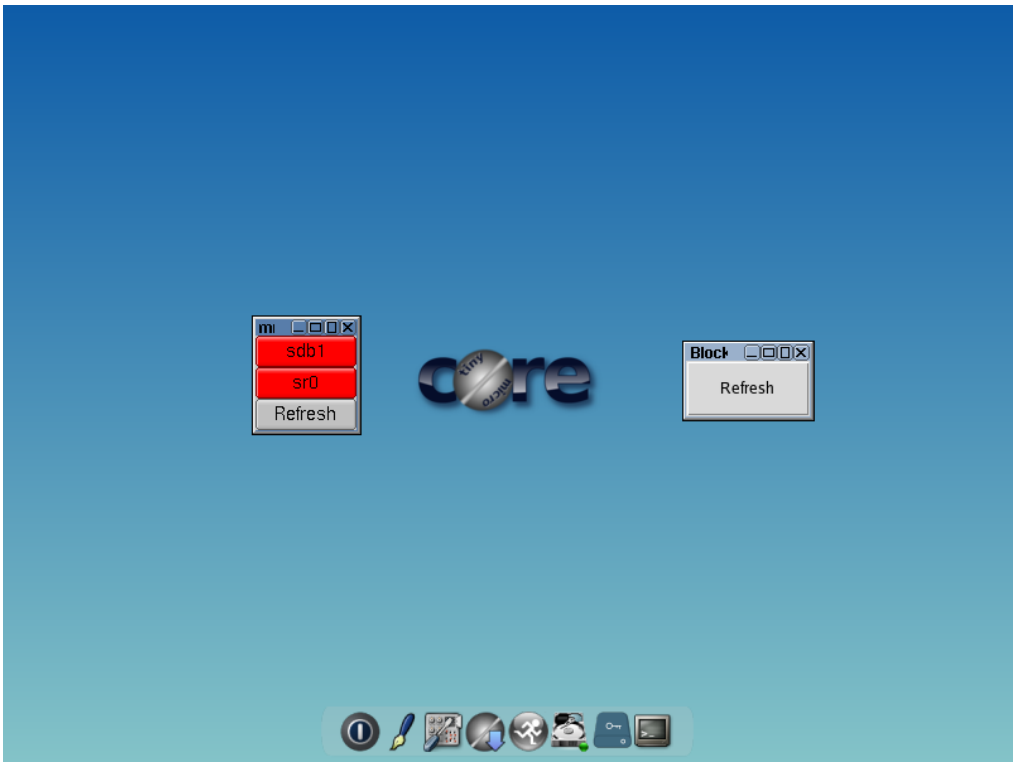
Terminato il processo si è provveduto a prelevare ed archiviare esternamente la nuova ISO dalla directory /tmp/ezremaster. Infine la nuova ISO è stata rinominata con il nome “Tiny.Core.Forensic.Edition-v2.1”.

1.1.1. Esempio di funzionamento

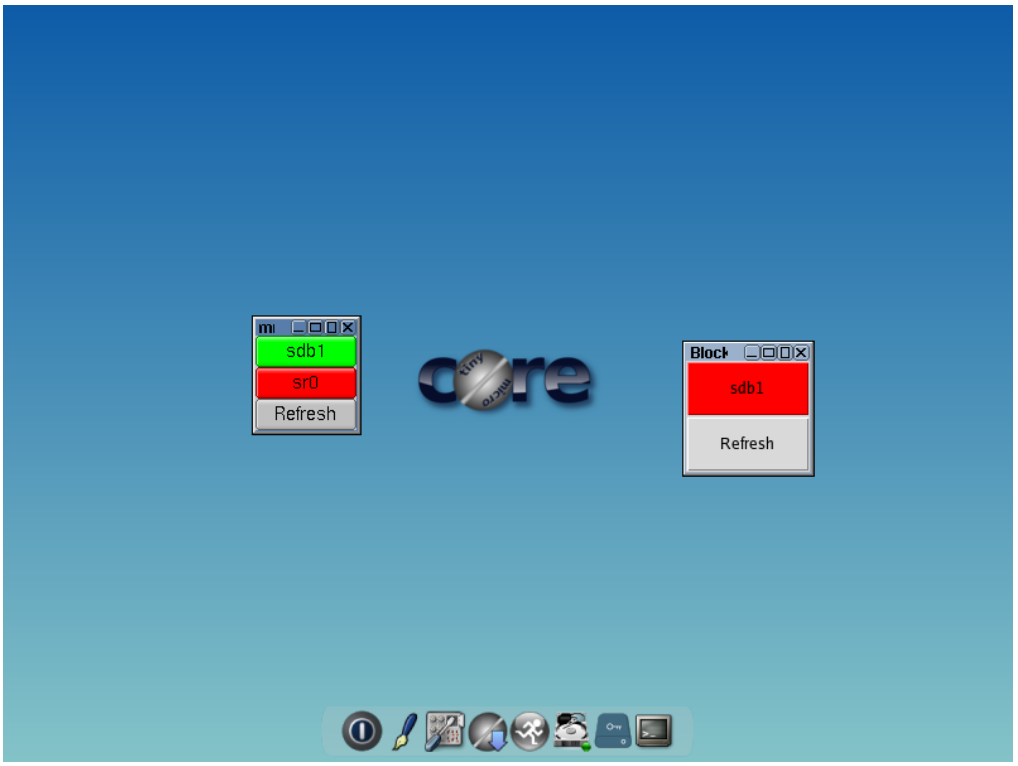
Di seguito, attraverso degli screenshot, si illustra qual è la grafica dell'applicazione e le modalità di utilizzo secando i casi d'uso definiti nella fase progettazione.

Per avviare l'app bisogna cliccare sull'apposita icona posta nella barra delle applicazioni.

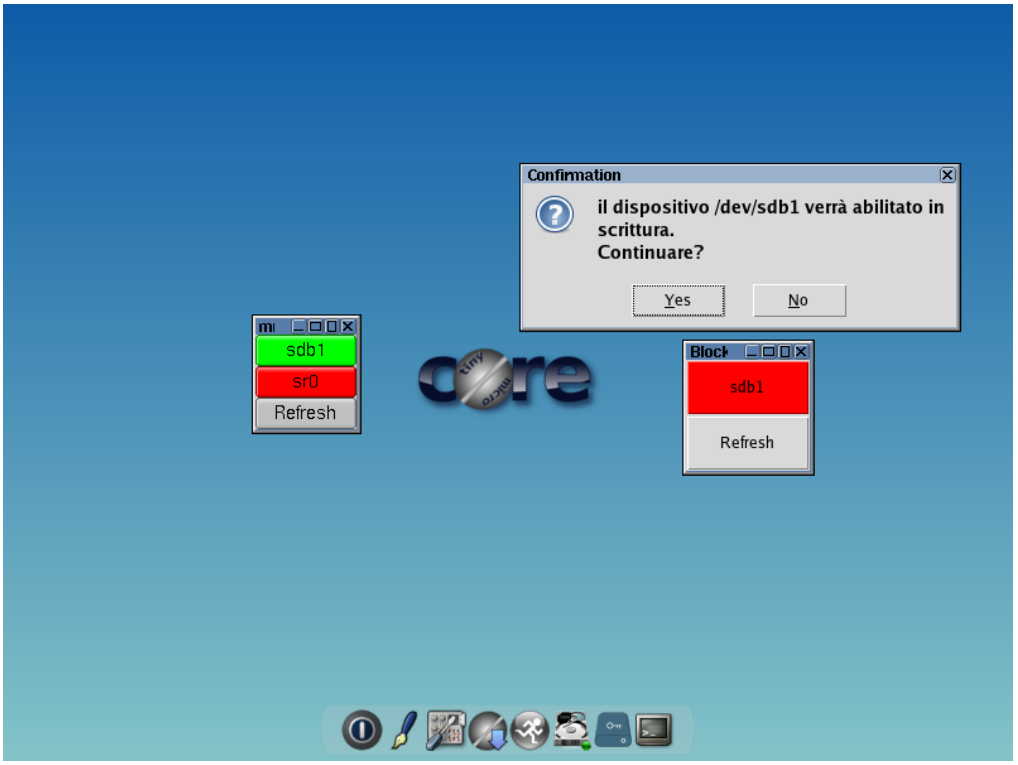
All'avvio dell'applicazione, se nessun dispositivo è stato ancora montato verrà visualizzato esclusivamente il pulsante di refresh.



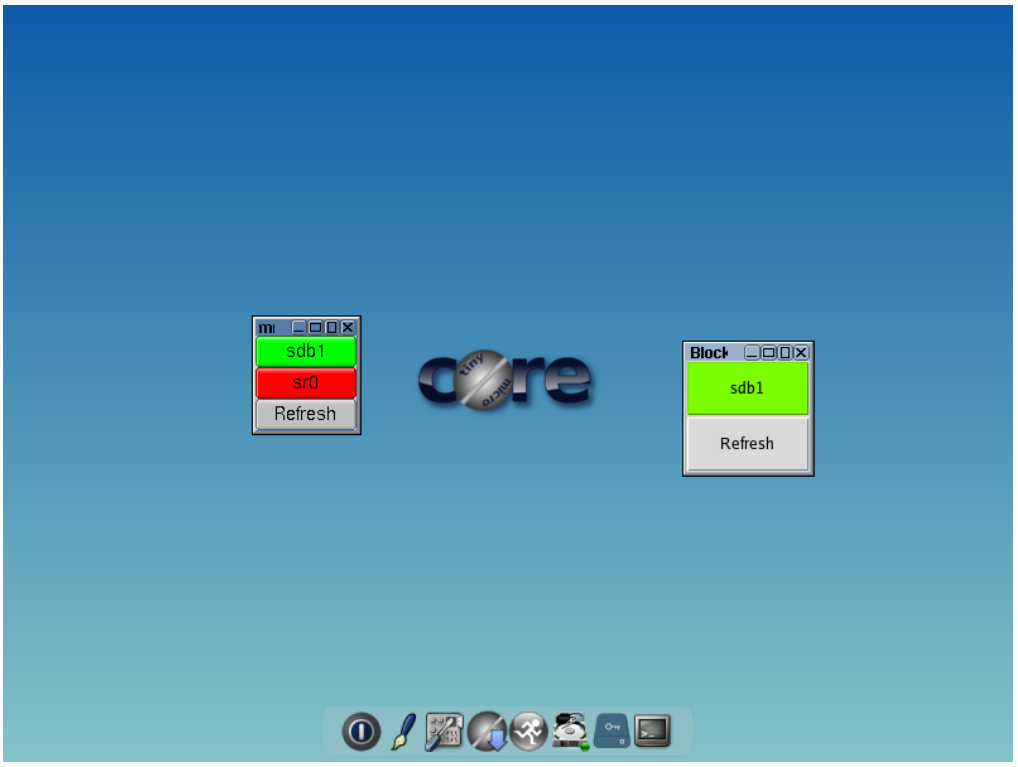
Successivamente al montaggio di una o più partizioni e in seguito al refresh sarà possibile visualizzare tutte le partizioni correttamente montate. Come si evince dall'immagine seguente la partizione sda1 risulta essere in sola lettura in quanto il colore del pulsante è rosso.



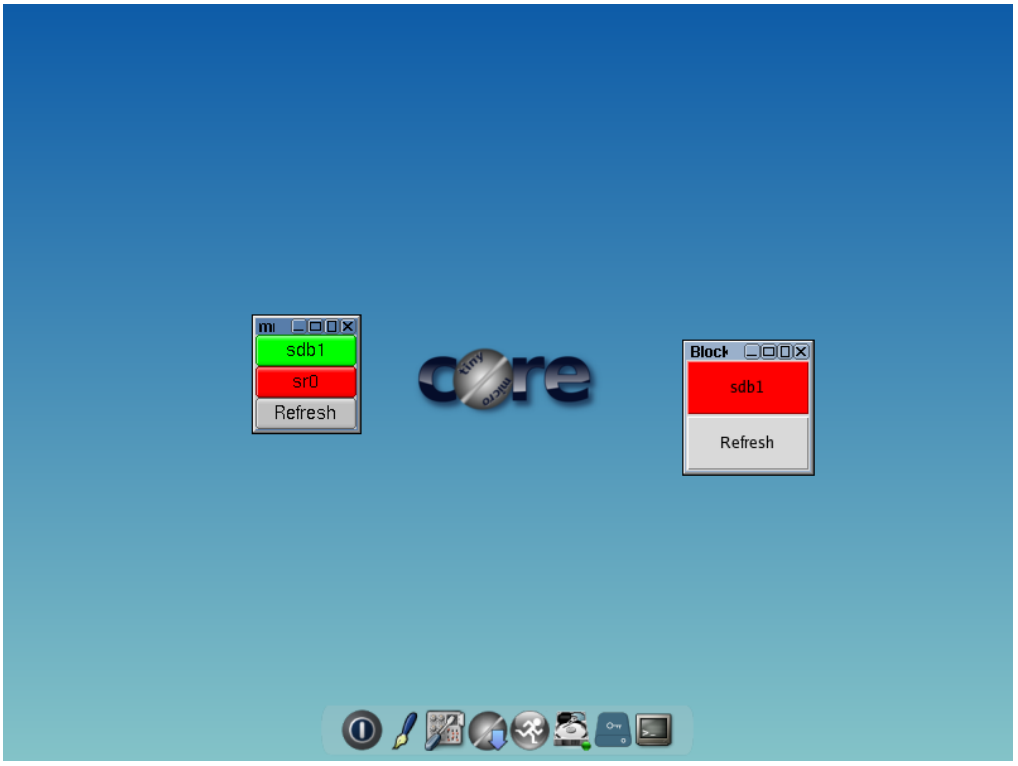
Dopo aver cliccato sul pulsante, verrà mostrato il messaggio di conferma montaggio in scrittura, per evitare montaggi accidentali che comprometterebbero il dispositivo.



Successivamente alla conferma del messaggio, il dispositivo verrà montato in scrittura e il bottone diventerà verde.



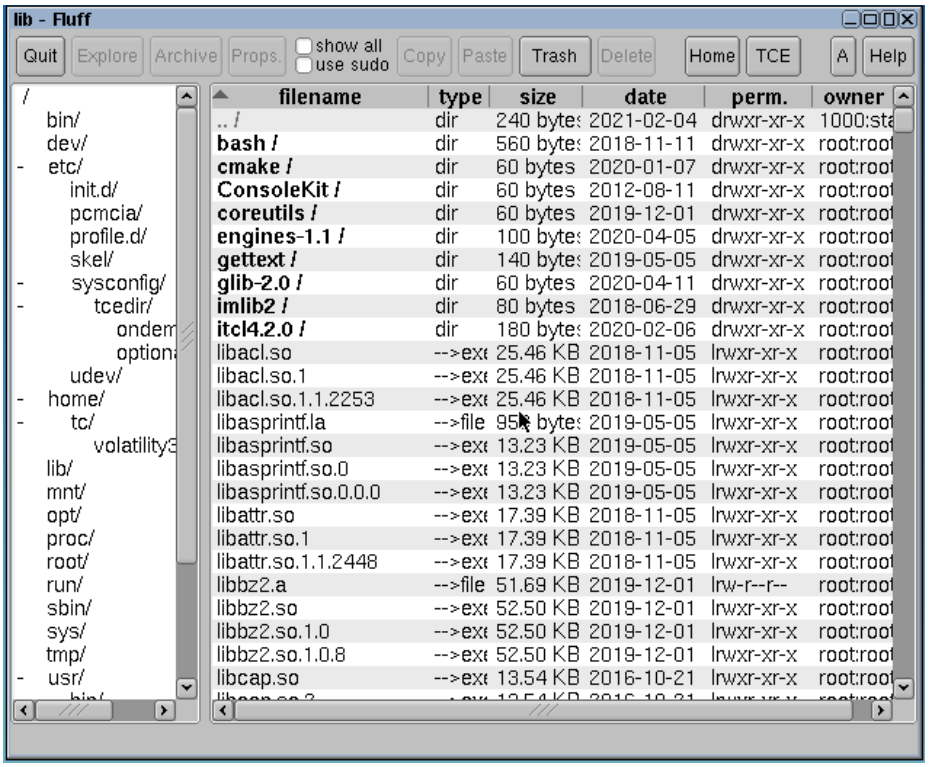
Se invece il drive è verde, quindi montato in scrittura, cliccando sul pulsante si riattiverà il blocco scrittura e il pulsante tornerà di nuovo rosso.



1.2. *Fluff File Manager*

Sin dai primi test di utilizzo della nuova distro realizzata si è sentita subito l'esigenza di integrare un file manager per poter navigare tra i file presenti nelle memorie in modo pratico. Inoltre, un file manager è fondamentale per eseguire operazioni tra i file, in questo caso risulta di particolare importanza la copia.

Fluff è un file manager grafico già presente nel repository ufficiale di Tiny Core. Permette di eseguire tutte le operazioni base come la gestione di file e directory, di aprire un percorso direttamente nel terminale e di eseguire i comandi da amministratore. La sua grafica minimale lo rende perfettamente idoneo per le nostre esigenze, soprattutto per non appesantire eccessivamente il sistema. (27)



1.3. *Wallpaper*

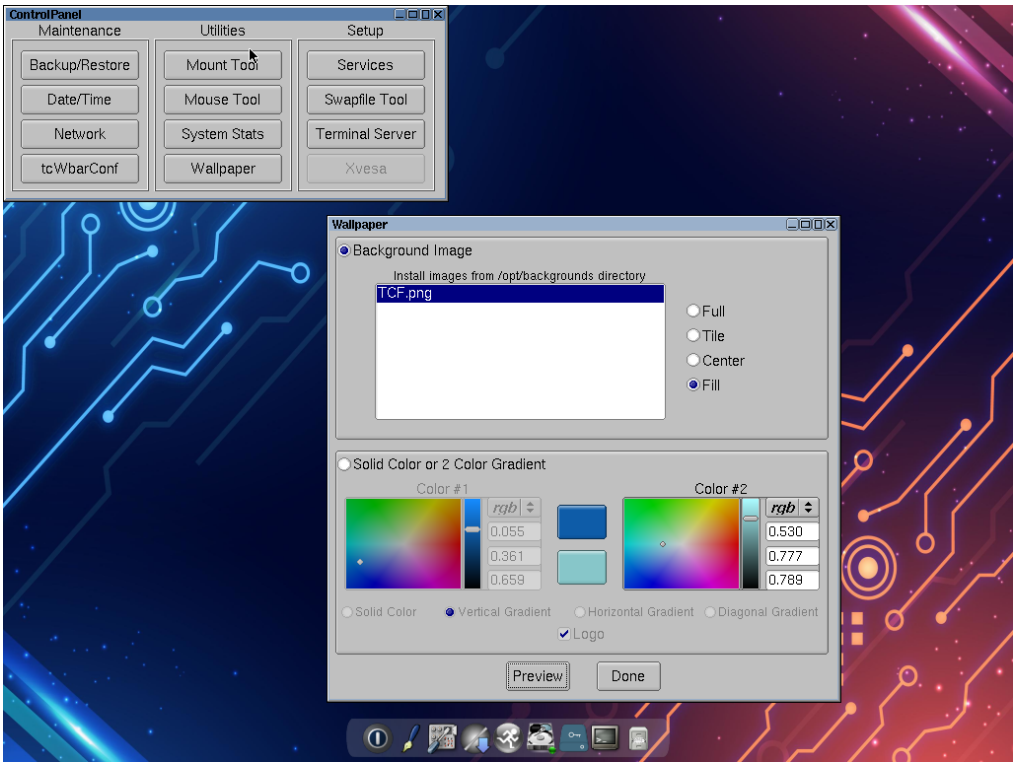
Per “marchiare” questo sistema si è deciso di inserire un wallpaper personalizzato per il desktop.

Con un tool esterno è stato creato il seguente sfondo per il desktop da inserire come nuovo look alla distro.

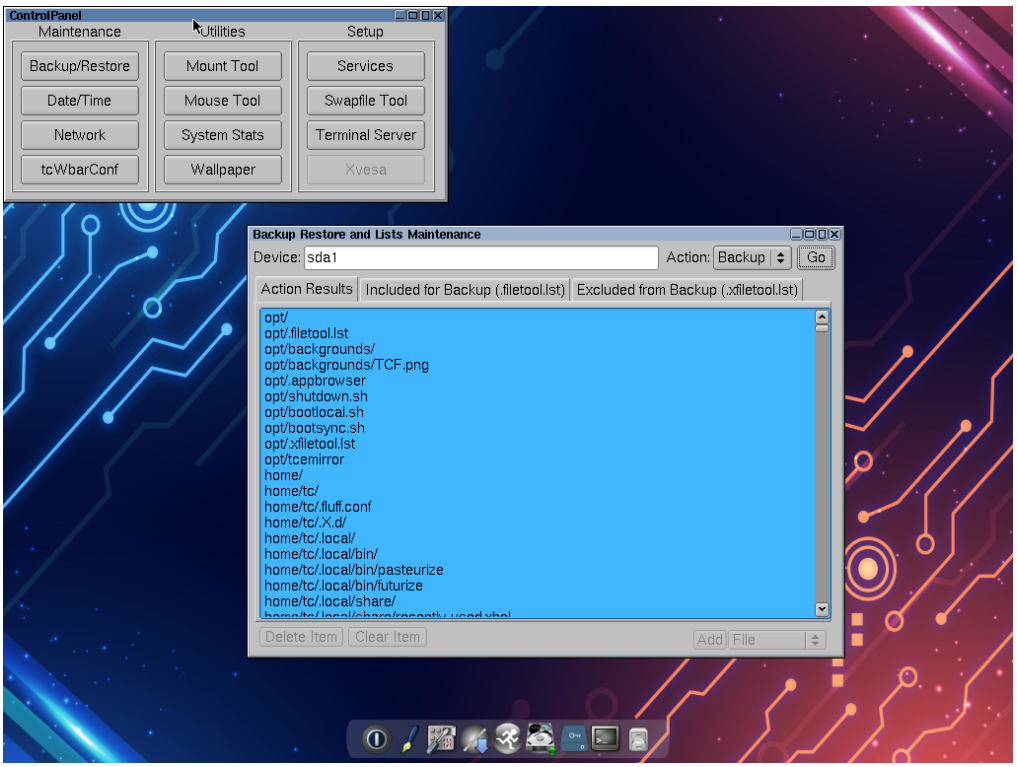
Il file è stato copiato nella apposita directory mediante il seguente comando:

```
cp /mnt/sda1/TCG.png /opt/backgrounds
```

Successivamente è stato impostato lo sfondo mediante le funzioni grafiche del sistema operativo:



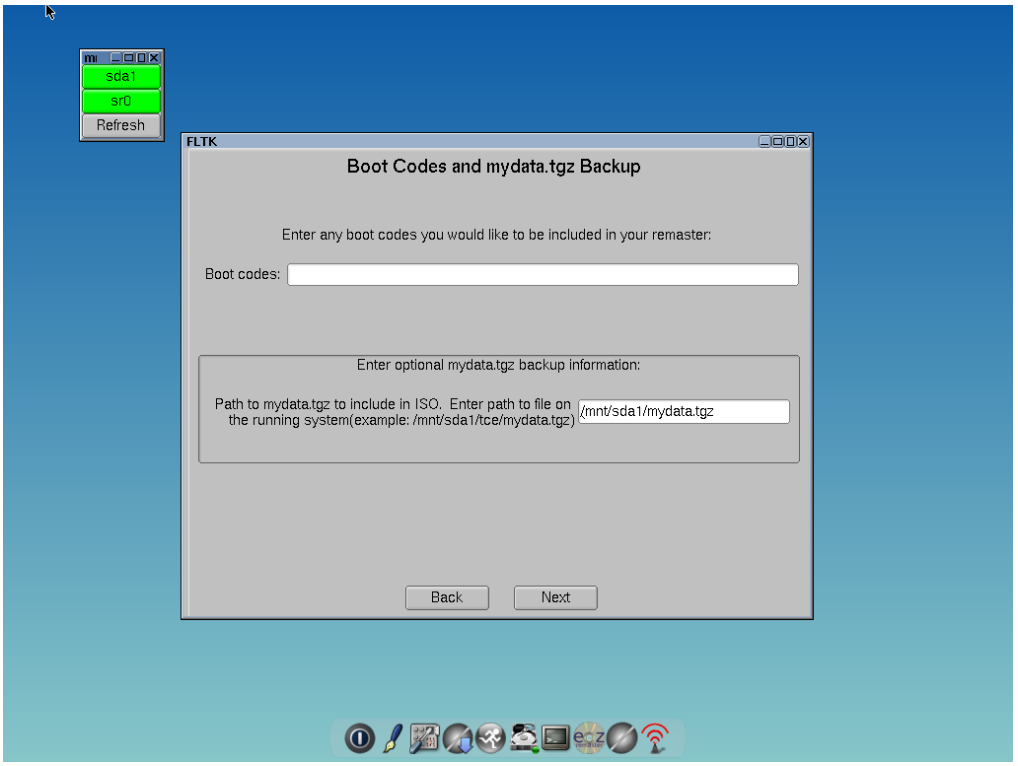
Infine, è stato creato un file di backup per rendere persistente lo sfondo ad ogni riavvio utilizzando l'apposita funzionalità del sistema operativo.



1.3.1. Ezremaster Fluff + Wallpaper

Partendo dalla schermata iniziale di ezremaster è stata selezionata la ISO "Tiny.Core.Forensic.Edition-v2.1".

Proseguendo nella finestra successiva per includere il wallpaper è stato sufficiente includere il file di backup eseguito nella fase precedentemente illustrata.



Successivamente essendo Fluff un'App ufficiale il processo di inclusione nella ISO è stato identico a quello illustrato nella fase dell'integrazione GUI.

Terminato il processo si è provveduto a prelevare ed archiviare esternamente la nuova ISO dalla directory /tmp/ezremaster. Infine questa è stata rinominata con il nome "Tiny.Core.Forensic.Edition-v2.2".

1.4. *SharePoint Downloader*

Per la realizzazione dell'applicazione, come per il blockdevgui, è stato scelto il linguaggio di programmazione python.

Oltre a definire l'interfaccia grafica come stabilito nei requisiti, è stato necessario implementare una funzione che consenta la connessione allo SharePoint attraverso le credenziali fornite dall'utente. La funzione che si occupa di questo è la seguente:

```
ctx, users, error = connect(site, email, password)
```

questa prende in input indirizzo del sito, mail e password ed effettua poi la connessione al sito. In output restituisce gli oggetti ctx di tipo Client Context, il quale rappresenta il contesto contenente tutti gli oggetti e le operazioni possibili in SharePoint, users che è la lista di tutti gli utenti con profilo di accesso valido ed error invece è un oggetto di tipo tupla in cui sono contenuti tutti i possibili errori qualora dovessero verificarsi.

È stata separata la logica attraverso due funzioni principali:

```
export_log(ctx, users, destination)

download_site(ctx, destination)
```

La prima è quella che si occupa dell'estrazione del file di testo contenente tutti i metadati e del file xlsx contenente l'alberatura del sito.

La seconda è quella che si occupa dell'estrazione del file zip contenente tutti i file con le relative versioni (se esistenti) e le cartelle presenti all'interno dello SharePoint aziendale.

Entrambe le funzioni al loro interno richiamano la funzione

```
root = get_library_root_folder(ctx)
```

che è quella che permette il riconoscimento della root dalla quale partire per estrarre in maniera ricorsiva tutti i file e le cartelle. Questo viene fatto attraverso le funzioni:

```
download_folders_and_files(root, ctx, dest)
```

```
export_folders_and_files(root, ctx, users, dest, worksheet)
```

Sono state differenziate in quanto la prima si occupa del download mentre la seconda è una semplice lettura per estrapolare le informazioni necessarie per formare entrambi i file che verranno poi scaricati.

Per realizzare l'app sono state utilizzate diverse librerie quali os, shutil, tkinter, office365, re, hashlib e xlsxwriter. Per le librerie non incluse nel sistema sono state eseguite delle ulteriori operazioni, come vedremo in seguito, per renderle utilizzabile dal compilatore installato in TinyCore FE.

1.4.1. Python3.6 Sharepoint Downloader

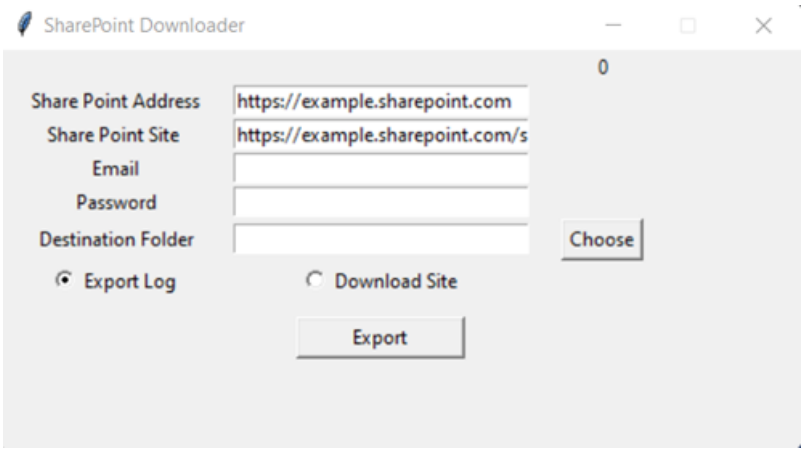
Il risultato della fase di programmazione è il file sharepoint.py che verrà successivamente incluso in Tiny Core avviabile tramite il comando:

```
python 3.6 sharepoint.py
```

1.1.1. Integrazione in TinyCore FE

L'applicativo realizzato viene rilasciato in formato open source ed essendo programmato mediante il linguaggio python sarà facilmente usufruibile su qualsiasi tipo di sistema operativo che utilizzi il compilatore.

Di seguito viene mostrata l'interfaccia grafica dell'applicazione in esecuzione sul sistema operativo Windows X.



Si vuole però integrare questa applicazione forense realizzata nella distro Tiny Core FE.

1.1.2. Librerie utilizzate

Innanzitutto, occorre aggiungere le librerie Python, utilizzate dal programma, e convertirle in formato tcz.

Le librerie utilizzate sono:

- **certifi 2021.10.8**: fornisce la collezione accuratamente curata da Mozilla di certificati radice per convalidare l'affidabilità dei certificati SSL e verificare l'identità degli host TLS. È stato estratto dal progetto Requests.

  La libreria è stata prelevata da:

  https://pypi.org/project/certifi/ (28)

- **chardet 4.0.0:** rilevatore universale di codifica dei caratteri.

  La libreria è stata prelevata da:

  https://pypi.org/project/chardet/ (29)

- **idna 3.3:** fornisce il supporto per Unicode Technical Standard 46, Unicode IDNA Compatibility Processing.

  La libreria è stata prelevata da:

  https://pypi.org/project/idna/ (30)

- **Office365-REST-Python-Client 2.3.11:** Microsoft Graph per Python.

  La libreria è stata prelevata da:

  https://pypi.org/project/Office365-REST-Python-Client/ (31)

- **requests 2.27.1:** è una semplice, ma elegante, libreria HTTP.

  La libreria è stata prelevata da:

https://pypi.org/project/requests/ (32)**urllib3 1.26.9:** è un client HTTP potente e facile da usare per Python.

La libreria è stata prelevata da:

https://pypi.org/project/urllib3/ (33)

- **XlsxWriter 3.0.3:** è un modulo Python per scrivere file nel formato XLSX di Excel 2007+.

  La libreria è stata prelevata da:

  https://pypi.org/project/XlsxWriter/ (34)

La sintassi dei comandi eseguiti è la seguente:

```
#estrazione e compilazione libreria XXX
tce-load -wi compiletc.tcz
tce-load -wi python3.6-dev.tcz
tce-load -i squashfs.tcz
cp -r /mnt/sda1/XXX /tmp
cd /tmp/XXX
python3.6 setup.py build
sudo python3.6 setup.py install --root=/tmp/XXX
#generazione tcz
cd /tmp
mksquashfs python3.6-XXX python3.6-XXX.tcz
#copia python3.6-XXX.tcz
cp /tmp/python3.6-XXX.tcz /mnt/sda1/
#installazione python3.6-XXX.tcz
cd /tmp
mv -v python3.6-XXX.tcz /etc/sysconfig/tcedir/optional
cd /etc/sysconfig/tcedir
echo python3.6-XXX.tcz >> onboot.lst
tce-load -i python3.6-XXX.tcz
```

1.1.3. Script Eseguibile Sharepoint

Il codice Python dell'app implementata è stato inserito nella directory /usr/local/bin con il nome sharepoint.py al fine di renderlo eseguibile

mediante uno script. Il risultato è stato inserito nel medesimo percorso con il nome sharepoint.sh realizzato con il seguente codice:

```
#!/bin/sh
cd /usr/local/bin
python3.6 sharepoint.py
```

Per rendere eseguibile lo script è stato necessario lanciare il seguente comando dal terminale dal percorso /usr/local/bin:

```
chmod +x sharepoint.sh
mv sharepoint.sh sharepoint
```

Adesso è possibile avviare l'utility da terminale tramite il comando sharepoint indipendentemente dal percorso in cui ci si trova.

1.1.4. Wbar Link Sharepoint

Per poter avviare l'app direttamente dalla wbar grafica di TinyCore, anche in questo caso, è stato realizzato un link grafico nella barra mediante l'esecuzione dei seguenti comandi (26):

```
#creazione directory
cd /tmp/tce
mkdir ondemand
#copia icona
cp /mnt/sda1/icona.png /tmp/tce/ondemand
cd /tmp/tce/ondemand
mv icona.png SharePointDW.img
#creazione script icona
cat > SharePointDW.sh
#!/bin/sh
SharePointDW CTRL+C
chmod +x SharePointDW.sh
mv SharePointDW.sh SharePointDW
```

Si precisa che icona.png è una semplice immagine prelevata dal web.

1.1.5. Creazione sharepoint.tcz

Per creare il tcz di SharePoint Downloader è stato definito un ambiente di lavoro parallelo utilizzando gli stessi nomi originali dei percorsi che TinyCore utilizza per eseguire in genere i programmi. Ad ogni avvio questi vengono reinstallati in base al numero dei tcz inclusi nel remaster della ISO.

La sintassi dei comandi eseguiti è la seguente:

```
#creazione ambiente di lavoro
mkdir /tmp/sharepoint
cd /tmp/ sharepoint
mkdir –p usr/local/bin
mkdir – t tmp/tce/ondemand
cp /tmp/tce/ondemand/ sharepoint /tmp/ sharepoint /tmp/tce/ondemand
cp /tmp/tce/ondemand/ SharePointDW.img  /tmp/ sharepoint /tmp/tce/ondemand
cp /usr/local/bin/ sharepoint /tmp/ sharepoint /local/bin
cp /usr/local/bin/ sharepoint /tmp/ sharepoint /local/bin
#generazione tcz
tce-load -i squashfs-tools.tcz
cd /tmp
mksquashfs sharepoint sharepoint.tcz
#copia sharepoint.tcz
cp /tmp/ sharepoint.tcz /mnt/sda1/
```

1.1.6. EzRemaster SharePoint Downloader

Come già visto per la realizzazione di questa applicazione sono state utilizzate molteplici librerie convertite in tcz, oltre al file principale sharepoint.tcz. Per poter includere tcz esterni è stato necessario eseguire determinati passi.

Innanzitutto, le estensioni create e da includere sono:

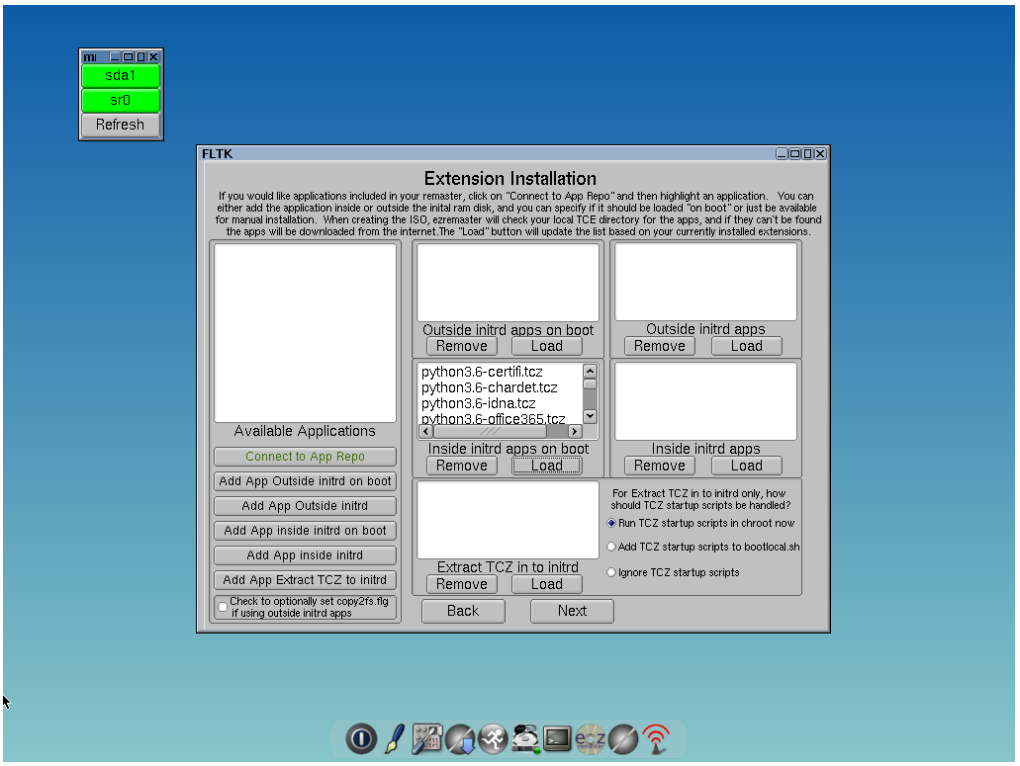
- python3.6-certifi.tcz
- python3.6-chardet.tcz
- python3.6-idna.tcz
- python3.6-office365.tcz

- python3.6-py3dns.tcz
- python3.6-pydns.tcz
- python3.6-requests.tcz
- python3.6-sharepy.tcz
- python3.6-urllib3.tcz
- python3.6-validate_email.tcz
- python3.6-xlsxwriter.tcz
- sharepoint.tcz

Dopo aver avviato la ISO di Core Plus sono stati eseguiti i seguenti comandi per includere un tcz esterno:

```
#XXX.tcz     nomepacchetto.tcz
cp /mnt/sda1/XXX.tcz /tmp
cd /tmp
mv -v XXX.tcz /etc/sysconfig/tcedir/optional
cd /etc/sysconfig/tcedir
echo XXX.tcz >> onboot.lst
cd /etc/sysconfig/tcedir/optional
tce-load -i XXX.tcz
```

Rispetto alla procedura descritta in precedenza i tcz da includere sono stati selezionati in automatico cliccando sul pulsante Load presente nella sezione “Inside initrd apps on boot” dopo avere eseguito i comandi illustrati precedentemente.



Terminato il processo si è provveduto a prelevare ed archiviare esternamente la nuova ISO dalla directory /tmp/ezremaster. Infine questa è stata rinominata con il nome “Tiny.Core.Forensic.Edition-v2.3”.

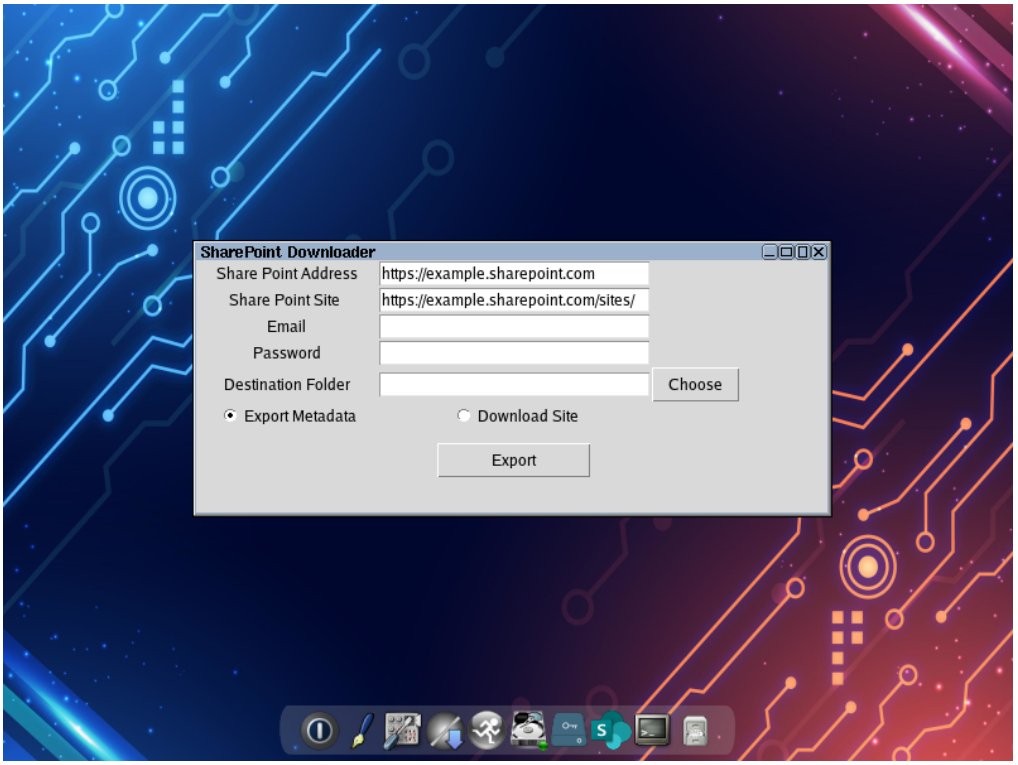
1.1.1. Esempio di funzionamento

Mediante i seguenti screenshot sarà illustrato qual è la grafica dell'applicazione e le modalità di utilizzo secando i casi d’uso definiti nella fase progettazione.

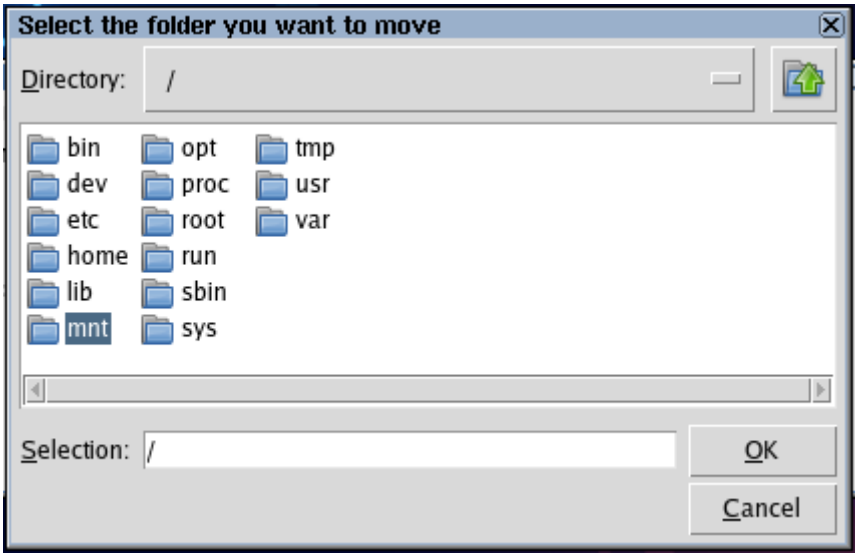
Nell' esempio riportato di seguito si fa riferimento ad un sito esempio.

1. **Avvio applicazione**: Per avviare l’app bisogna cliccare sull'apposita icona posta nella barra delle applicazioni.

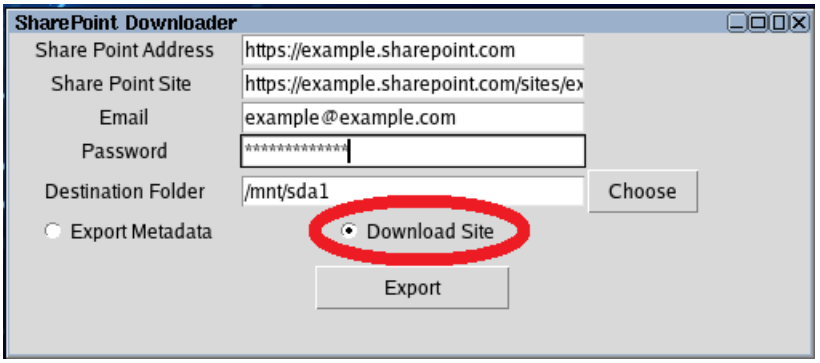
Viene quindi visualizzata l'interfaccia grafica dell'applicazione.



Si ricorda che occorre disabilitare il blockdev sul dispositivo di destinazione dell'export che tramite il tasto Choose sarà selezionabile attraverso la seguente finestra:



2. **<u>Download Forense File</u>**: nella finestra bisognerà inserire tutti gli input necessari all'applicazione per funzionare correttamente.

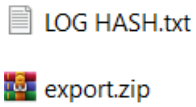


Una volta compilati indirizzo, sito, e-mail e password attraverso il bottone Choose viene selezionato il percorso in cui i file estratti saranno salvati. La scelta della funzione di "Download Site" è stata selezionata attraverso il radio button "Download Site".

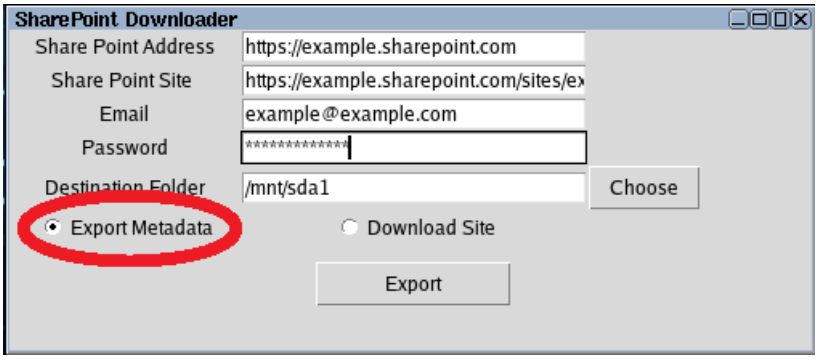
L'operazione è stata avviata premendo il pulsante Export.

Dopo qualche minuto di attesa l'operazione di esportazione sarà conclusa.

Nella cartella precedentemente selezionata vedremo come output il file zippato più un file di testo in cui saranno presenti i doppi hash relativi all'archivio estratto.

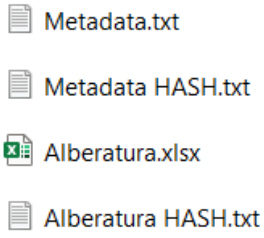


3. **<u>Download Forense Metadati:</u>** Compilando alla stessa maniera gli input iniziali è stato selezionato invece il radio button Export Metadata.



Anche in questo caso l'operazione è stata avviata premendo il pulsante Export.

L'output sarà il seguente:



Nel dettaglio, il file di testo Metadata.txt conterrà tutti i metadati presenti relativi ad ogni singolo file e cartella presenti sullo SharePoint. Di seguito un estratto del file dei Metadata:

Nome Cartella: Documenti Condivisi
Percorso Cartella: /sites/testNearcons/Documenti Condivisi
Data Creazione Cartella: 2021-11-07T00:03:53Z
Data Ultima modifica Cartella: 2021-12-15T18:34:27Z

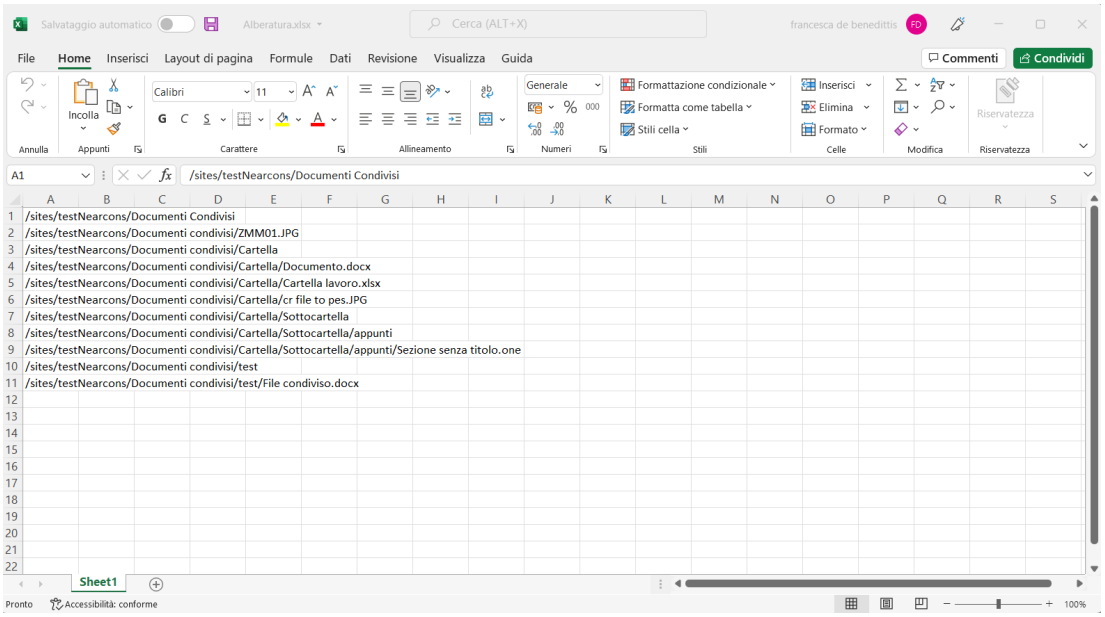
Nome File: ZMM01.JPG
Percorso File: /sites/testNearcons/Documenti condivisi/ZMM01.JPG
Data Creazione File: 2021-12-15T18:34:27Z
Data Ultima modifica File: 2021-12-15T18:34:27Z
File creato da: francesca de benedittis
Versione 1.0 creata da francesca de benedittis

Nome Cartella: Cartella
Percorso Cartella: /sites/testNearcons/Documenti condivisi/CartellaData Creazione Cartella: 2021-12-15T18:32:58Z
Data Ultima modifica Cartella: 2021-12-15T18:34:06Z
Cartella creata da: francesca de benedittis

Nome Cartella: test
Percorso Cartella: /sites/testNearcons/Documenti condivisi/testData Creazione Cartella: 2021-12-10T17:16:38Z
Data Ultima modifica Cartella: 2021-12-10T17:16:41Z
Cartella creata da: francesca de benedittis
Cartella condivisa con: Angelo Peluso, marco carani

Nome File: File condiviso.docx
Percorso File: /sites/testNearcons/Documenti condivisi/test/File condiviso.docx
Data Creazione File: 2021-12-10T11:50:21Z
Data Ultima modifica File: 2022-01-12T08:52:02Z
File creato da: francesca de benedittis
File condiviso con: Sara Bevitore, Angelo Peluso
Versione 6.0 creata da Sara Bevitore
Versione 5.0 creata da francesca de benedittis
Versione 4.0 creata da francesca de benedittis
Versione 3.0 creata da francesca de benedittis
Versione 2.0 creata da francesca de benedittis
Versione 1.0 creata da francesca de benedittis

L'esempio evidenzia le informazioni quali nome, data e ora di creazione di file e cartelle, utenti che li hanno creati, percorso relativo, utenti con cui sono condivise queste informazioni, quante versioni sono presenti di un determinato file e da chi sono state create. Il file xlsx conterrà l'intera alberatura del sito.



Ed infine saranno presenti due differenti file di testo che conterranno l'hash di entrambi i file estratti come per il Download Site.

2. Risultati

Il processo ha portato alla creazione di quattro versioni:

- Tiny.Core.Forensic.Edition-v2.0 di dimensioni 195,6 MB che integra la sola interfaccia grafica;
- Tiny.Core.Forensic.Edition-v2.1 di dimensioni 196,2 MB che integra l'interfaccia grafica più l'applicazione BlockDevGui realizzata;
- Tiny.Core.Forensic.Edition-v2.2 di dimensione 198,6 MB che integra l'interfaccia grafica, l'applicazione BlockDevGui e un file manager in quanto si è pensato potesse rendere ancora più semplice la navigazione tra le cartelle e infine un wallpaper per il desktop di personalizzazione.
- Tiny.Core.Forensic.Edition-v2.3 di dimensione 198,6 MB che integra l'interfaccia grafica, l'applicazione BlockDevGui, un file manager, il wallpaper per il desktop e infine l'applicazione SharePoint Downloader realizzata.

Il sistema Tiny Core Forensic Edition potrà essere installato su una memoria USB in modo da ottenere un'unità flash USB avviabile (35).

I file ISO sopra elencati, compresi i codici sorgente python degli applicativi realizzati sono stati caricati al seguente link:

https://github.com/SaraRuffo/Tiny-Core-Forensic-Edition---Tesi/releases/tag/v2.3

3. Sviluppi futuri

L'obiettivo di questo progetto di tesi è stato quello di ampliare il progetto iniziale di Tiny Core Forensic Edition, realizzando delle funzionalità avanzate quali dotazioni di interfaccia grafica del sistema operativo, realizzazione grafica del BlockDev gui, implementazione dello SharePoint Downloader, uno dei primi esperimenti di tool open source di cloud forensics e integrazione di ulteriori utility.

Dal punto di vista dell'acquisizione di memorie oltre al BlockDev gui realizzato sarebbe utile integrare un tool di acquisizione grafico. Invece, relativamente all'applicazione SharePoint Downloader potrebbe essere interessante integrare oltre all'intero contenuto del sito anche le liste, ed inoltre si potrebbe provare ad aggiungere il login automatico fornito da Microsoft in modo da poter avere la possibilità di effettuare l'autenticazione a due fattori o quella direttamente impostata dall'utente. Inoltre, un ulteriore sviluppo molto interessante potrebbe essere quello di espandere il campo di acquisizione forense anche sulla piattaforma Google Workspace, strumento molto utilizzato in diversi settori, da diversi utenti.

4. Bibliografia

1. **wikipedia.** [Online] https://it.wikipedia.org/wiki/Scienza_forense.

2. **https://it.wikipedia.org/wiki/WikiLeaks.** [Online] https://it.wikipedia.org/wiki/WikiLeaks.

3. **https://www.garanteprivacy.it/home/docweb/-/docweb-display/docweb/9121352.** [Online] https://www.garanteprivacy.it/home/docweb/-/docweb-display/docweb/9121352.

4. **https://www.wired.it/internet/social-network/2018/10/09/google-plus-chiude-vulnerabilita/.** [Online] https://www.wired.it/internet/social-network/2018/10/09/google-plus-chiude-vulnerabilita/.

5. **https://www.affde.com/it/biggest-data-breaches.html.** [Online] https://www.affde.com/it/biggest-data-breaches.html.

6. **https://www.defensis.it/perizie_informatiche/acquisizione_di_prove_digitali.htm.** [Online] https://www.defensis.it/perizie_informatiche/acquisizione_di_prove_digitali.htm.

7. **https://informaticaforense.org/informatica-forense/.** [Online] https://informaticaforense.org/informatica-forense/.

8. **http://nazionlinux.altervista.org/caine-11-0/.** [Online] http://nazionlinux.altervista.org/caine-11-0/..

9. **feedlinux.** [Online] https://www.feedlinux.com/tag/caine.

10. **Checco, Paolo Dal.** [Online] https://www.dalchecco.it/tsurugi-linux-distribuzione-forense/.

11. **Ruffo.** [Online] https://github.com/SaraRuffo/Tiny-Core-Forensic-Edition---Tesi/releases/tag/v1.0.

12. **Wikipedia.** [Online] https://en.wikipedia.org/wiki/Foremost_(software).

13. **gnu.** [Online] https://www.gnu.org/software/ddrescue/ddrescue_it.html.

14. **Wikipedia.** [Online] https://it.wikipedia.org/wiki/Dd_(Unix).

15. **nmap.** [Online] https://nmap.org/.

16. **rhash.** [Online] http://rhash.sourceforge.net/.

17. **wikipedia.** [Online] https://en.wikipedia.org/wiki/The_Sleuth_Kit.

18. **github.** [Online] https://github.com/volatilityfoundation/volatility3.

19. **wikipedia.** [Online] https://it.wikipedia.org/wiki/Netcat.

20. **linux.die.** [Online] https://linux.die.net/man/1/ewfacquire.

21. **https://intranet.ai/risorse-utili/articoli/microsoft-365-e-sharepoint/cosa-e-sharepoint-e-7-funzionalita-fondamentali-nel-2022/.** [Online] https://intranet.ai/risorse-utili/articoli/microsoft-365-e-sharepoint/cosa-e-sharepoint-e-7-funzionalita-fondamentali-nel-2022/.

22. **TinyCoreLinux.** [Online] http://tinycorelinux.net/.

23. **CorePlus.** [Online] http://tinycorelinux.net/11.x/x86/release/CorePlus-current.iso.

24. **psutil.** [Online] http://www.tinycorelinux.net/10.x/x86_64/tcz/src/python3.6-psutil/.

25. **tk.** [Online] http://tinycorelinux.net/3.x/tcz/tk.tcz.

26. **TinyCoreWiki.** [Online] https://www.linuxsecrets.com/tinycorelinux-wiki/wiki:start.html.

27. **Fluff.** [Online] http://tinycorelinux.net/3.x/tcz/fluff.tcz.

28. **certifi.** [Online] https://pypi.org/project/certifi/.

29. **chardet.** [Online] https://pypi.org/project/chardet/.

30. **idna.** [Online] https://pypi.org/project/idna/.

31. **office365-REST-Python-Client.** [Online] https://pypi.org/project/Office365-REST-Python-Client/.

32. **request.** [Online] https://pypi.org/project/requests/.

33. **pypi.** [Online] https://pypi.org/project/urllib3/.

34. **XlsxWriter.** [Online] https://pypi.org/project/XlsxWriter/.

35. **Rufus.** [Online] https://rufus.ie/.