МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВОЛИНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ЛЕСІ УКРАЇНКИ
Навчально-науковий інститут неперервної освіти

На правах рукопису

БУЛАТЕЦЬКИЙ ВІТАЛІЙ ВІКТОРОВИЧ

ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ ЗАСОБІВ ОПТИМІЗАЦІЇ СИСТЕМНОГО РОЗДІЛУ ОПЕРАЦІЙНОЇ СИСТЕМИ WINDOWS 10

Спеціальність: 122 Комп'ютерні науки
Освітньо-професійна програма Комп'ютерні науки та інформаційні технології
Робота на здобуття освітнього ступеня «магістр»

Науковий керівник:
ГРИШАНОВИЧ
ТЕТЯНА ОЛЕКСАНДРІВНА,
кандидат фізико-математичних наук

РЕКОМЕНДОВАНО ДО ЗАХИСТУ
Протокол № ___
засідання ради навчально-наукового
інституту неперервної освіти
від _____ 2021 р.

Директор ННІНО _____ О. Дикий

ЛУЦЬК – 2021

ЗМІСТ

ВСТУП

Сучасна найпоширеніша операційна система Windows останніх версій присутня у кількох редакціях, від Starter до Enterprise (в залежності від версій набір може дещо відрізнятись). Проте загалом такі збірки операційної системи носять універсальний характер і призначені для широкого кола користувачів та апаратних засобів, а, відповідно, призначені для виконання широкого спектру завдань з різними вимогами, тому часто містять у собі надмірну кількість файлової інформації. Самі операційні системи є динамічно змінюваними, і нарощують кількість системних файлів у процесі експлуатації, що пов’язано з постійним їх оновленням, виправленнями, встановленням додаткових програмних засобів, пристроїв та їх драйверів, створенням резервних копій, збоїв тощо. Проте розмір системного розділу є скінченним, а, отже, користувач рано чи пізно стикається з браком вільного місця у ньому, що веде до гальмування роботи системи, неможливості розгортання останніх актуальних оновлень та виправлень, збоїв у роботі. Тому виникає потреба у корегуванні налаштувань та періодичній очистці (англ. clean up – прибиранні) системного розділу від непотрібної або надмірної файлової інформації, завантажених і встановлених пакетів оновлень та виправлень, невикористовуваних драйверів пристроїв, застарілих резервних копій, тимчасових файлів, файлів, пов’язаних з невикористовуваними функціями операційної системи тощо. Тобто, виникає потреба в оптимізації системного розділу. [1]

Мета і завдання дослідження: дослідити механізми та засоби очистки та налаштування операційної системи для його здійснення та розробити програмний продукт призначений для оптимізації системного розділу переважно шляхом вивільнення простору на ньому для операційних систем лінійки Windows 10. Для цього необхідно виконати наступний ряд завдань:

- дослідити структуру системного носія, файлів і папок операційної системи та їх роль в її функціонуванні;

- проаналізувати умови та підібрати методи для зменшення розміру, або видалення деяких системних файлів та папок, визначити налаштування для оптимізації їх розміру;
- сформувати вимоги до програмного засобу;
- визначитися з інструментарієм розробки та шляхами реалізації обраних методів;
- реалізувати та протестувати кожен із обраних методів на предмет ефективності та коректності;
- сформулювати перелік рекомендацій та вимог до використання створеного програмного продукту.

Об’єкт дослідження – операційна система Windows 10, зокрема структура системних папок та файлів, окремих налаштувань операційної системи та їх призначення.

Предмет дослідження – набір методів та засобів оптимізації системного розділу та шляхів їх реалізації.

Публікації.

1. Булатецький В. В., Булатецька Л. В., Пруц Г. С. Методи та засоби вивільнення простору системного розділу ОС Microsoft Windows 10. Комп’ютерно-інтегровані технології: освіта, наука, виробництво, 2018. № 32. С. 85–89.
2. Технології проміжного коду в корпоративних інформаційних системах: Текст лекцій нормативної навчальної дисципліни “Платформи корпоративних інформаційних систем” / Булатецький Віталій Вікторович, Булатецька Леся Віталіївна. – Луцьк : Східноєвропейський національний університет імені Лесі Українки, 2018. – 48 с.

РОЗДІЛ 1
АНАЛІЗ ШЛЯХІВ, МЕТОДІВ ТА ЗАСОБІВ ОПТИМІЗАЦІЇ СИСТЕМНОГО РОЗДІЛУ ОПЕРАЦІЙНОЇ СИСТЕМИ WINDOWS 10

1.1. Організація розділів операційної системи Windows 10

В процесі розвитку операційної системи Windows, починаючи з самих перших (Windows 3.1, Windows 95…) і аж до Windows XP, розробники притримувались позиції, згідно якої, на відміну від *nix-систем, всі файли та папки, як системні, так і користувацькі повинні бути розміщені на одному первинному розділі, який повинен бути активним та завантажувальним. Такий підхід в першу чергу був зумовлений особливістю операційної системи MsDos, яка є основою операційних систем Windows аж до Windows Millennium (1999р) включно. MsDos могла повноцінно працювати лише з одним первинним розділом на одному жорсткому диску, розміченому за стандартом MBR (Master Boot Record). Починаючи з початку 90-х паралельно велась розробка операційних систем типу Windows NT, які вже не були надбудовою над MsDos, і, відповідно, не мали пов’язаних з нею обмежень, що стосувались розділів на жорсткому диску. Навіть сучасна Windows 11 базується на коді, що став основою Windows NT. Проте, з міркувань сумісності з попередніми версіями операційних систем на рівні файлової структури, розробники і надалі притримувались принципу – все на одному розділі. В цьому були свої переваги та недоліки. Одним розділом легше маніпулювати, робити його резервні копії, відновлювати, використовувати єдину файлову систему для усіх даних, використовувати декілька операційних систем (кожна на своєму розділі) тощо. Але в цьому був і основний недолік – у разі незворотного пошкодження такого розділу, користувачі зазнавали втрат власних даних. Частина OEM-виробників, почала використовувати декілька розділів, застосовуючи власні (не вмонтовані в операційну систему) програмні засоби для реалізації механізмів відновлення та підвищення стійкості операційної системи від пошкоджень, які розміщували на інших розділах. Проте, лише з появою Windows Vista, а точніше, із появою

підтримки в ній нового стандарту GPT (GUID (*Globally Unique IDentifier*) Partition Table) розмітки накопичувачів, з'явилась можливість розподілити по розділам (створити їх тепер можна було аж до 128) системні та користувацькі дані.[2] Але, на жаль, розробники і далі за замовчуванням залишають користувацькі та системні дані на одному розділі, залишаючи користувачам можливість та право вибору самостійно переносити власні папки на інші розділи, хоча в процесі установки (починаючи з Windows Vista) створюється, або використовується більше ніж один первинний розділ, але призначені вони в основному для підвищення стійкості самої операційної системи.

Так, наприклад, Windows 10 при своїй установці створює за замовчуванням 4 розділи на нерозміченому накопичувачі [3]:

- Recovery – зарезервований операційною системою розділ, де зберігаються файли, необхідні для скидання операційної системи до початкового стану, і її завантажувач.
- EFI System (Hidden from disk management) – використовується для запису завантажувачів операційних систем, які (завантажувачі) пов'язані через UEFI (сучасна заміна базової системи введення-виведення(BIOS)).
- MSR – захищений розділ Microsoft до 128 Мб створений для резервування частини простору з можливим подальшим використанням операційною системою Windows, встановленої в іншому окремому розділі.
- Розділ позначений літерою C без мітки: – основний системний розділ.

Оскільки, в більшості випадків, звичайні користувачі не приділяють значення місцю розміщення власних даних (або просто не мають навиків їх переміщення) виникає потреба періодичної оптимізації системного розділу шляхом його очистки (прибирання) як від непотрібних користувацьких файлів, так і від файлових даних самої операційної системи, які утворюються в процесі експлуатації та оновлення, про що описано у вступі. І хоча операційна система містить у собі засоби для прибирання (cleanmgr та ін.), їх виявляється недостатньо. Не завжди, або не вчасно і не ефективно вони спрацьовують в автоматичному режимі, або користувачі не мають до них доступу. Тому

користувачі часто використовують стороннє програмне забезпечення, розробляють самостійно такі програмні продукти, або здійснюють прибирання та налаштування вручну.

А для коректної очистки системного розділу варто знати структуру файлових об’єктів на системному розділі.

1.2. Вміст системних каталогів операційної системи Windows 10 та їх призначення

Надалі приведемо перелік основних системних каталогів досліджуваної операційної системи та їх призначення, де %systemroot% – системна змінна, що позначає каталог Windows у корені системного розділу (те саме що і %windir%, не плутати з %systemdrive%, яка позначає корінь системного розділу). Поза всякими сумнівами – це найважливіша папка, розміщена на активному розділі жорсткого диска. У ній зберігаються всі виконувані файли операційної системи, драйвери, модулі та ін. Варто відзначити, що в деяких випадках дана папка може мати й іншу назву, якщо користувач її замінив в процесі установки операційної системи, тому якщо її назва наперед невідома для розробника, в шляхах можна використовують саме %systemroot%, або %windir%.

Оскільки нас цікавлять папки порівняно великого розміру та папки розмір яких суттєво впливає на зайнятий простір системного розділу, і з якими можна працювати для вивільнення, то лише на них акцентуємо увагу.

%Systemroot%\INF\ – включає в себе всі встановлені на комп’ютері INF-файли, що стосуються наперед визначених стандартних пристроїв – як реальних, так і віртуальних. Дана папка прихована.

%Systemroot%\Installer\ – зберігає більшість з пакетів установників Windows, які коли-небудь інсталювали на комп’ютер (детальніше у другому розділі). Якщо необхідно встановити або перевстановити якусь програму, що інсталюється за допомогою MSI-пакетів, і при цьому немає її інсталяційного диска, але раніше вже встановлювали цю програму, то можна спробувати знайти

відповідний інсталяційний файл в цій папці. За замовчуванням ця папка прихована, а пакети, які в ній містяться, мають числові номери (а не реальні назви, які їм відповідають), тому найпростішим способом визначити приналежність пакета є вкладка «Додатково» діалогу «Властивості пакета». [4]

%Systemroot%\LastGood\ (або \LastGood.TMP\) – якщо в системі присутня ця папка, то вона може зберігати резервні копії різних файлів, які пов'язані з останнім вдалим запуском системи.

%Systemroot%\System32\DllCache\ – містить копії всіх системних файлів Windows (не тільки бібліотек, а й програм) і необхідна при пошкодженні або несанкціонованій системою зміні оригінального системного файлу для його переустановлення. Даний каталог стиснутий і прихований.

В папці %Systemroot%\System32\drivers\ знаходяться файли драйверів працюючих пристроїв. Проте існує також споріднена папка %Systemroot%\System32\DriverStore\ – яка є сховищем драйверів, в тому числі застарілих версій та непрацюючих в даний момент пристроїв.

В папці %Systemroot%\system32\config\ містяться файли, які відповідають за системний реєстр:

- файл AppEvent.Evt є журналом подій додатків, використовуваного реєстром Windows;
- файл SAM містить розділ реєстру HKLM\SAM;
- файл SecEvent.Evt є журналом подій безпеки, використовуваного реєстром Windows;
- файл SECURITY містить розділ реєстру HKLM\SECURITY;
- файл software містить розділ реєстру HKLM\Software;
- файл SysEvent.Evt є журналом подій системи, що використовується реєстром Windows;
- файл system містить розділ реєстру HKLM\System.

Папка %Systemroot%\system32\DriverStore\ – являє собою сховище драйверів пристроїв. Причому тут містяться драйвери пристроїв які працюють в даний момент, так і драйвери пристроїв, які в даний момент не працюють, або

тимчасово відсутні у системі і підключаються по мірі необхідності. Особливої уваги заслуговує підпапка FileRepository, в якій містяться як актуальні драйвери пристроїв, так і попередні (застарілі) їх версії, які призначені для «відкату», тобто повернення їх у систему у разі, якщо актуальний драйвер пристрою працює некоректно, на відміну від попередньої своєї версії. [5,6]

%Systemroot%\SysWOW64\ – аналог папки system32 для 64-розрядних версій операційних систем. Саме до цієї папки звертається в першу чергу операційна система в пошуках штатних утиліт та інтерпретаторів різноманітних скриптів, сценаріїв тощо.

%Systemroot%\WinSxS\ – призначена для збереження резервних копій системних файлів під час здійснення оновлень операційної системи. Причому, така папка містить також жорсткі посилання на такі системні файли, які можуть бути розміщенні за іншими шляхами у системі, і, які в даний момент навіть можуть використовуватись операційною системою. Підпапка \Backup використовується для тимчасового збереження файлів розпакованих оновлень під час їх установки і в подальшому може бути використана для відміни таких оновлень. [7,8]

В корені системного розділу розміщена папки \Program Files, \Program Files (x86), призначені для збереження файлів 64- та 32- розрядного встановленого програмного забезпечення відповідно, в тому ж числі і UWP (плиткових) застосунків разом із їх резервними копіями попередніх та видалених версій.

За шляхом %SystemDrive%\System Volume Information\ розміщено приховану системну папку, яка містить інформацію, яку використовують бази даних служби індексування вмісту, які прискорюють пошук файлів, службу тіньового копіювання розділів для резервного копіювання та бази даних служби відстеження розподілених посилань, які використовуються для відновлення ярликів і посилань.

В корені системного розділу можуть міститись також системні файли великого розміру, такі як hiberfil.sys – пов’язаний з режимом гібернації, та

pagefile.sys і swapfile.sys – пов’язані із віртуальною дисковою пам’яттю операційної системи (swap-файли, або файли підкачки). Такі файли створюються при увімкнених відповідних режимах роботи: режим гібернації – передбачає збереження даних з оперативної пам’яті у файл для можливості повного вимкнення живлення з подальшим швидким відновленням стану системи при увімкненні живлення; віртуальна дискова пам’ять – дозволяє за рахунок файлів розширити об’єм оперативної пам’яті до необхідного у випадку її нестачі для запуску тих, або інших процесів.

І нарешті %SystemDrive%\Users\, яка містить файли користувачів, зареєстрованих (в тому числі і локально) у системі, як файли налагодження програмного забезпечення для кожного конкретного користувача, кеші, тимчасові файли працюючих пакетів, так і файли документів користувача, як то текстові файли у вигляді документів різного формату, зображення, відео, звукові файли, файли завантажені з мережі, так і синхронізовані папки хмарних сховищ користувачів, якщо такі використовуються.

Це перелік далеко не усіх системних файлів та папок, які містяться на системному розділі, але вищеописані – це ті з них, які займають порівняно великі об’єми і найбільше вливають на розмір вільного простору на системному розділі.

Визначившись із причинами зменшення вільного простору в системі, локалізувавши на розділі системні файли та папки великого розміру, необхідно підібрати способи і, відповідно, інструменти зменшення їх розміру.

В переважній більшості сама операційна система містить такий інструментарій і надає механізми вирішення проблеми. Проте на даному етапі, такі інструменти є ще досить далекі від ідеальних, існують переважно у вигляді консольних утиліт операційної системи, недоступні безпосередньо через графічний користувацький інтерфейс для звичайного користувача, вимагають використання ключів та параметрів у своїй роботі, а, отже, і додаткових знань. Крім того такі консольні утиліти можуть виконуватись в окремих середовищах, для функціонування яких необхідне використання інтерпретатора командного рядка (cmd), або оболонки з розширеними можливостями PowerShell.

1.3. Інтерпретатор командного рядка, як основний інструмент для роботи з консольними утилітами операційної системи

Незважаючи на простоту використання графічного інтерфейсу користувача (GUI) Windows, інтерфейс командного рядка залишається корисним способом виконання багатьох завдань обслуговування, конфігурації та діагностики. Багато важливих інструментів доступні тільки з командного рядка. І хоча термін «bat-файл» може видатись пов'язаним із старою операційною системою MS-DOS, bat-файли та програмні скрипти залишаються потужними інструментами, що надають ефективний спосіб інкапсулювати в собі загальні функції управління. Утиліти командного рядка, bat-файли та скрипти, засновані на WSH (Windows Script Host) разом надають повний набір будівельних блоків, з яких можна створити високорівневі утиліти для виконання складних завдань. [9]

Основою для відображення роботи інтерпретатора командного рядка є вікно, яке реалізує командний інтерфейс операційної системи у вигляді, так званого запрошення командного рядка (Command Prompt), в якому власне і виконуються консольні застосунки. Тому часто таке вікно називають консоллю інтерпретатора командного рядка операційної системи (рис. 1.1).

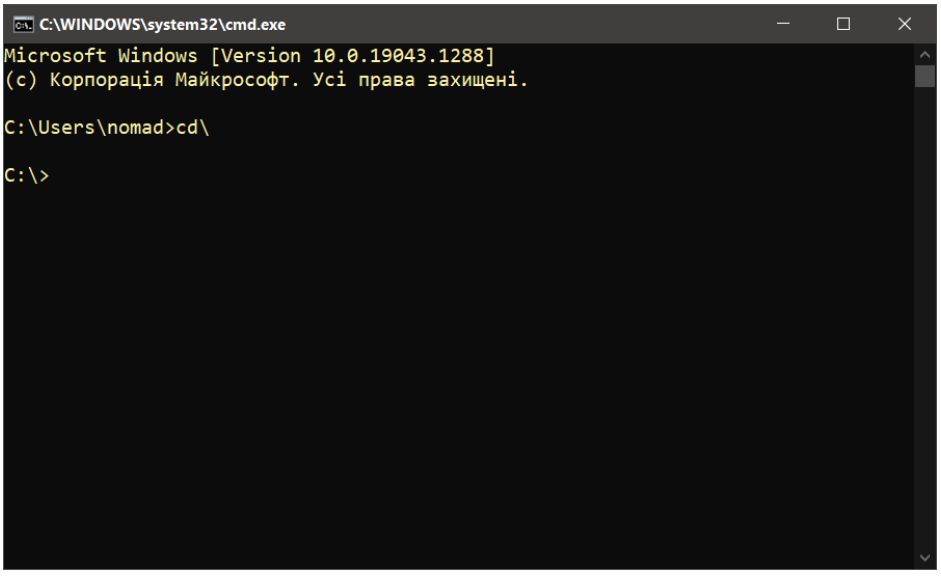

Рисунок 1.1 - Консоль інтерпретатора командного рядка.

Таке вікно призначене як і для введення команд вручну, в інтерактивному режимі, так і для відображення перебігу виконання консольних програм та команд. При виконанні неконсольних команд викликаються вікна графічного інтерфейсу поточної виконуваної команди, або програми.

1.3.1. Розширення імені для запуску файлів

У багатьох файлів Windows є так зване розширення імені файлу – суфікс, що слідує після крапки у назві файлу, який зазвичай складається з трьох латинських символів. Це розширення призначене для визначення типу файлу. Деякі з розширень позначають виконувані файли програм різного виду, це такі розширення, як .exe, .bat і .vbs (таб.1).

Таблиця 1. Типові розширення виконуваних файлів

Розширення	**Тип програми**
.bat, .cmd	Пакетний файл, скрипт із набору команд.
.com	16 розрядна програма MsDos
.exe	GUI-програма Windows, консольна програма або програма MS-DOS (Windows визначає тип програми за вмістом файлу).
.js	Скрипт, написаний мовою JavaScript.
.msc	Консоль керування Windows (Microsoft Management Console).
.vbs	Скрипт, написаний мовою VBScript.

Для виконання команди не обов’язково вводити ім’я програми повністю, достатньо ввести ім’я без розширення. У разі введення команди без розширення операційна система Windows вважає, що це програма, і вона додає розширення файлу, коли здійснюється пошук програми, і якщо така програма знайдена, то вона запускає її. В складі команди запуску програми, як правило, вводиться список аргументів, вказаний в командному рядку після імені програми, і програма повинна відповідним чином обробляти ці аргументи. [10]

1.3.2. Запуск команд із підвищеними привілеями

Деякі утиліти командного рядка (а іноді і програми з графічним інтерфейсом) вимагають для своєї коректної роботи підвищених адміністративних привілеїв (права для користувачів та груп задаються через інструментарій User Account Control). Щоб запустити програму командного рядка з підвищеними привілеями, необхідно запустити її з консолі командного рядка, яка сама має підвищені права.

В такому режимі необережні дії адміністратора можуть призвести до пошкодження системи. Це також стосується і запуску з підвищеними привілеями віконних (GUI) програм Windows. Тому варто використовувати такий запуск лише для спеціальних завдань, які потребують адміністративних привілеїв, і після завершення виконання цих завдань консоль повинна бути закрита.

1.3.3. Bat-файли (пакетні файли)

Хоча в операційній системі існує маса потужних інструментів для створення власних корисних програм, проте часто достатньо використати мову bat-файлів. Bat-файли (пакетні файли) дозволяють отримати доступ до тисяч готових програм командою рядка, які можна знайти для Windows.

Bat-файл на найпростішому рівні це просто список команд для вікна Command Prompt (консолі), які набрані в одному файлі, рядок за рядком, і цьому файлу присвоєно розширення *.bat або *.cmd. Коли вводять ім’я bat-файлу в запрошенні командного рядка, Windows шукає файл з таким ім’ям у поточній директорії і потім у шляхах папок, що позначені змінною оточення PATH. Windows обробляє кожен рядок bat-файлу як команду, і запускає їх одну за одною, ніби їх вводять одну за одною вручну. Навіть на такому самому простому рівні bat-файл суттєво підвищує рівень автоматизації роботи користувача.

Крім цього простого сценарію використання є кілька команд, які можна використовувати для написання найпростіших «програм» у bat-файлі, щоб виконувати різні дії залежно від того, що було введено користувачем у командному рядку, або залежно від результату виконання попередніх команд. Ці

команди значно покращилися з часів MS-DOS, тому написання корисних bat-файлів у Windows набагато простіше, ніж це було раніше в старій ОС MS-DOS. Зокрема, оператори IF та FOR були значно розширені за функціоналом. Можна надавати запрошення введення для користувача, можна маніпулювати рядками та іменами файлів для виконання арифметичних обчислень, можна створювати підпрограми в одному файлі тощо. [10]

Використовуючи інтерпретатор командного рядка та пакетні файли, запускаючи їх з підвищеними привілеями, користувач здатен отримати доступ до інструментарію операційної системи, а отже маніпулювати вмістом системних папок та системними файлами, регулюючи об’єм вільного простору системного розділу.

Проте не всі команди або інструментарій доступні через інтерпретатор командного рядка. Також використання штатного інструментарію може бути суттєво ускладнене за певних умов. Наприклад неможливість коректного виконання пакетного файлу із застосунків, створених у різноманітних середовищах програмування, оскільки такі застосунки можуть бути вже 64-розрядними, вони вимагають відповідного інтерпретатора команд, тоді як навіть в 64-розрядній операційній системі cmd – є 32-розрядним транслятором, розміщеним у системній папці \System32\ і його 64-розрядна версія не використовується за замовчуванням, або відсутня. В таких випадках використовують інші транслятори або середовища на кшталт PowerShell.

1.4. PowerShell, як один із засобів маніпулювання об’єктами операційної системи

Microsoft розробили особливе робоче середовище командного рядка, яке дістало назву Windows PowerShell (WPS). Воно встановлене як стандартний аксесуар починаючи з Windows XP. WPS найчастіше виглядає та діє як вікно командного рядка, але насправді це дуже особлива сутність, яка дає доступ до деяких дуже потужних інструментів програмування.

З одного боку, більшість команд Windows PowerShell (які правильно називати cmdlet) генерують потоки об’єктів, а не текст. Об’єкти це спроба представлення у комп’ютері речей із реального світу. Вони мають властивості, які описують атрибути речей, які вони представляють, та методи, які дозволяють керувати речами. Наприклад, об’єкт представляє певний файл на жорсткому диску, і він має властивості Name (ім’я), Size (розмір) і LastWriteTime (останній час записи), і методи на кшталт Delete (видалити), Edit (редагувати) і Open (відкрити). Windows PowerShell працює з об’єктами новим, незвичайним і, зрештою, дуже потужним способом.

Якщо ввести команду dir у звичайному вікні Command Prompt, то середовище інтерпретації команд обробить dir і у відповідь поверне блок тексту лістингу, де будуть імена файлів і папок поточного каталогу. Команда dir була спеціально запрограмована для друкування текстової інформації про файли в текстовому вигляді. І це все, що вона може робити.

У системі WPS можна ввести dir, і це також призведе до виведення списку файлів, але неявно відбувається зовсім інше. У WPS це просто коротке посилання на Get-Childitem cmdlet, який при найпростішому використанні генерує потік об’єктів File; кожен об’єкт представляє один із файлів у папці, і у кожного такого об’єкта є властивості та методи (наприклад, ім’я та розмір). Коли об’єкт (будь-якого виду) потрапляє у вікно запрошення WPS, система WPS друкує рядок тексту про об’єкт з найбільш важливими його властивостями. Для об’єкта File це включає ім’я файлу, розмір та дату створення. Таким чином, коли ввести dir, WPS генерує потік об’єктів File і він представляється як зручний, розділений табуляціями список файлів.

Кінцевий результат виглядає так, як і в робочому оточенні командного рядка, але все відбувається абстрактнішим способом. Cmdlet не звертає уваги форматування: він просто генерує набір об’єктів File. І вікно WPS перетворить будь-який список об’єктів на читабельний текстовий лістинг. Файли, облікові записи користувачів, служби Windows, і т. д. – будь-який об’єкт, який викликає

відповідний cmdlet, система WPS перетворює у своїй консолі на зручний лістинг тексту.

Додатково WPS включає повнофункціональну об’єктно-орієнтовану мову програмування, яка має доступ до платформи програмування .NET компанії Microsoft. Це означає, що скрипти WPS можуть виконувати складніші обчислення та обмінюватися інформацією з іншими комп’ютерами та мережевими («хмарними») сервісами.

WPS навіть дозволяє робити складні операції з об’єктами без програмування. Можна використовувати символ каналу | для направлення потоків об’єктів від одного cmdlet до іншого, і це дозволяє виконувати дуже складні, специфічні дії з інструментами, які окремо дуже прості та за природою виглядають звичайними.

WPS це повноцінна мова програмування зі змінними, циклами, підпрограмами, визначеними користувачем об’єктами тощо. Можна використовувати це в запрошенні введення команди або у файлах скриптів. Також можна створити ярлики (які називаються псевдонімами, aliases) для команд і скриптів, які найчастіше використовуються, щоб їх можна було простіше використовувати надалі.[9,10]

Крім того через PowerShell можна створювати графічні користувацькі інтерфейси для написаних скриптів. Для цього можна скористатись як окремим командами (рис. 1.2) так і сторонніми розробками, наприклад PowerShell Studio 2021 (рис. 1.3) і, звичайно, засобами Microsoft Visual Studio. [11]

В нашому випадку нас цікавлять можливості PowerShell, які дозволяють обійти обмеження по розрядності при виконанні пакетних файлів із створюваних застосунків та можливість маніпулювання UWP-застосунками та звертаннями до магазину застосунків Microsoft. Такі застосунки через PowerShell можна відмикати, видаляти, встановлювати, відновлювати, тим самим заощаджуючи вільний простір на системному розділі.

Проте деякі маніпуляції із системними папками вимагають написання пакетних файлів та скриптів, які звертаються до служб та інших засобів

операційної системи для коректної очистки системних папок. Такі звертання можна здійснювати через роботу із локальними груповими політиками. Але не всі версії операційної системи мають змогу це зробити, особливо редакції початкового рівня, такі як Starter або Home. Іншим способом роботи з такими об’єктами є робота через системний реєстр операційної системи.

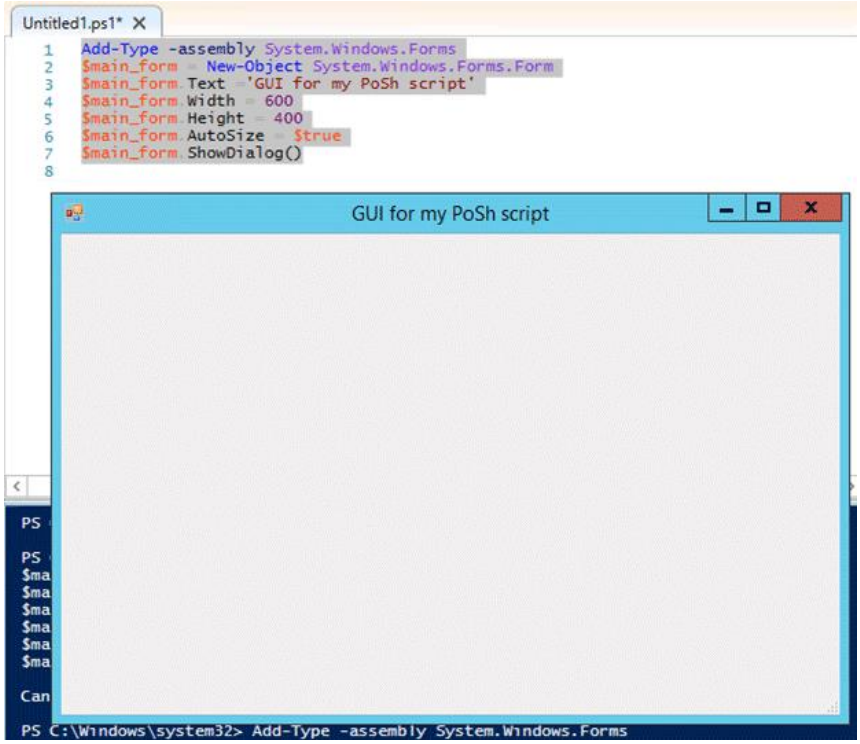

Рисунок 1.2 - Приклад формування графічного користувацького інтерфейсу безпосередньо в PowerShell.

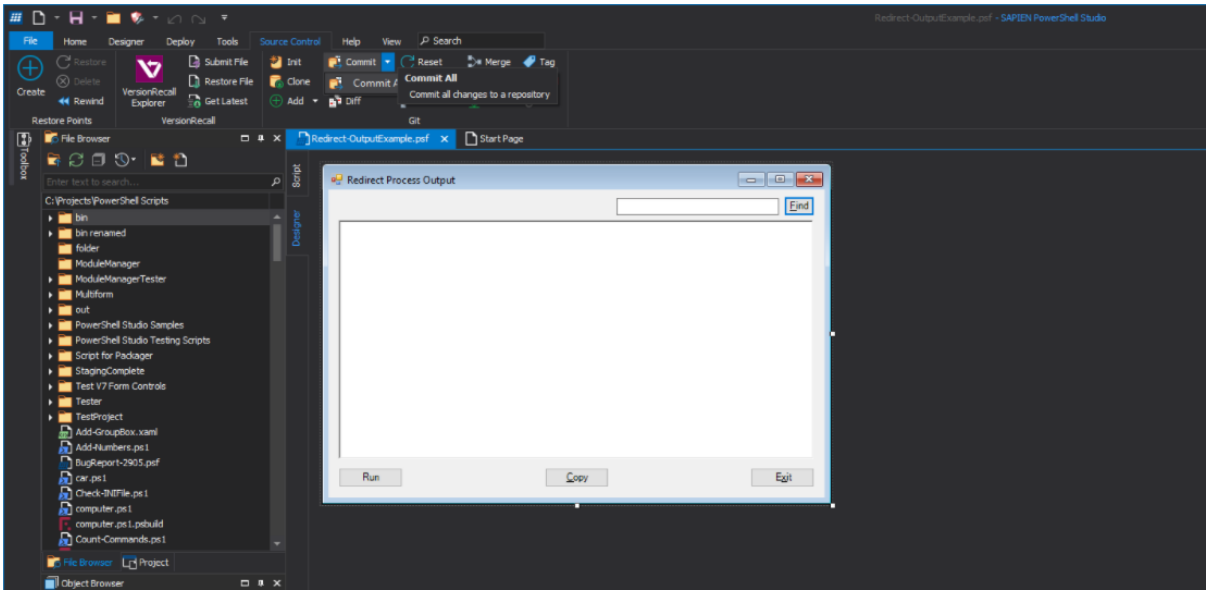

Рисунок 1.3 - Фрагмент вікна середовища PowerShell Studio.

1.5. Реєстр операційної системи Windows 10 та способи його редагування

Реєстр Windows 10 – основне сховище операційної системи, побудоване у вигляді ієрархічної бази даних. В ній зберігається кожен параметр, кожне налаштування операційної системи, інформація про встановлене на комп'ютері програмне забезпечення та функціонуючі апаратні компоненти, дані про всі облікові записи користувачів. У цій базі даних записані асоціації файлів та системні політики. Кожна програма, що працює на ПК, обов'язково звертається до реєстру, наприклад, з метою отримання інформації про систему.[12]

Логічно реєстр являє собою набір ієрархічних дерев (кущів), кожне з яких має власне призначення:

- HKCR (HKEY_CLASSES_ROOT) – містить інформацію про відомі типи документів та їх асоціації з програмами для відкриття;
- HKCU (HKEY_CURRENT_USER) – зберігається налаштування параметрів користувача, що увійшов до системи;
- HKLM (HKEY_LOCAL_MACHINE) – записані відомості про апаратну конфігурацію комп'ютера (перелік обладнання, режими його функціонування, версії драйверів);
- HKCC (HKEY_CURRENT_CONFIG) – містить дві гілки, у першій з яких зберігається конфігурація програмного забезпечення, у другій – параметри системи (більшість ключів розділу, доступні для редагування вигляду графічного інтерфейсу Windows);
- HKU (HKEY_USERS) – містяться налаштування кожного облікового запису завантаженої операційної системи. [13]

Фізично файли такої бази даних Windows розташовуються у різних підпапках системного каталогу. За кожен кущ відповідає окремий файл (за тим же шляхом розміщена і його резервна копія) [13]:

HKEY_LOCAL_MACHINE\SYSTEM:

%systemroot%\system32\config\system

%systemroot%\sysWoW64\config\system

HKEY_LOCAL_MACHINE\SAM:

%systemroot%\system32\config\sam

%systemroot%\ sysWoW64\config\sam

HKEY_LOCAL_MACHINE\SECURITY:

%systemroot%\system32\config\security

%systemroot%\ sysWoW64\config\security

HKEY_LOCAL_MACHINE\SOFTWARE:

%systemroot%\system32\config\software

%systemroot%\ sysWoW64\config\software

HKEY_USERS\UserProfile:

%systemdrive%\Users\%username%

HKEY_USERS.DEFAULT:

%systemroot%\system32\config\default

%systemroot%\ sysWoW64\config\default.

Кожен кущ містить розділи, які в свою чергу містять ключі, яким можна присвоювати ті чи інші значення, в залежності від типу ключів. Передбачено наступні основні типи ключів:

- REG_SZ – рядковий,
- REG_EXPAND_SZ – розширений рядковий,
- REG_BINARY – двійковий, містить необроблені двійкові дані.
- REG_DWORD – числовий, містить ціле число, зазвичай служить як перемикач, де 0 вимкнено, а 1, відповідно, включено,
- REG_LINK – рядковий, вказує шлях до файлу, це символічне посилання у форматі Юнікод,
- REG_MULTI_SZ – багаторядковий, визначає масив рядків,
- та інші, менш поширені типи [13].

Працювати з реєстром, а фактично його редагувати (вносити зміни) можна кількома способами:

- через редактор реєстру (інструмент схожий до провідника, де замість папок та файлів – кущі, розділи та ключі): викликається командами regedit, або regedt32 з командного рядка;
- за допомогою .reg-файлів: текстових файлів, які асоційовані з редактором реєстру, з наперед визначеним синтаксисом введення куща, розділу, ключа та його типу і значенням, містять відповідний заголовок та один і більше рядків із записами про зміни параметрів реєстру;
- за допомогою команди командного рядка REG, яка допускає роботу в пакетних файлах, а отже придатна для автоматизації процесів [14].

1.6. RunDll32, як інструмент виклику функцій системних бібліотек

В процесі застосування визначених методів за допомогою обраних засобів час від часу може виникати необхідність використання вмонтованих в операційну систему штатних інструментів. І хоча звичайний користувач, може скористатись графічним інтерфейсом операційної системи для виклику таких інструментів, такий підхід не дозволить автоматизувати такий процес, а, отже, використати його при проектуванні програмних систем. Тому часто для опису виклику таких інструментів у пакетних файлах та різноманітних скриптах використовують виклики функцій системних бібліотек за допомогою утиліти rundll32.

Така утиліта завантажує та виконує 32-розрядні бібліотеки динамічного компонування (DLL). Настроювання параметрів для rundll32 відсутні. Довідкові відомості надаються для конкретної бібліотеки DLL, яка виконується за допомогою команди rundll32 з підвищеними привілеями (від імені адміністратора). Rundll32 може викликати лише функції з бібліотеки DLL, які були явно написані для виклику за допомогою rundll32. [15]

У 64-бітних версіях ОС сімейства Windows присутні 2 варіанти програми rundll32.exe:

64-бітна версія, розташована в %SystemRoot%\System32\;

32-бітна версія, розташована в %SystemRoot%\SysWOW64\.

Для запуску програми rundll32 використовується синтаксис командного рядка такого вигляду:

rundll32.exe <бібліотека_dll>,< функція> <необов'язкові_параметри_функції>.

Наприклад, для виклику вікна регіональних настройок операційної системи через командний рядок (рис. 1.4):

rundll32.exe shell32.dll,Control_RunDLL intl.cpl,,0

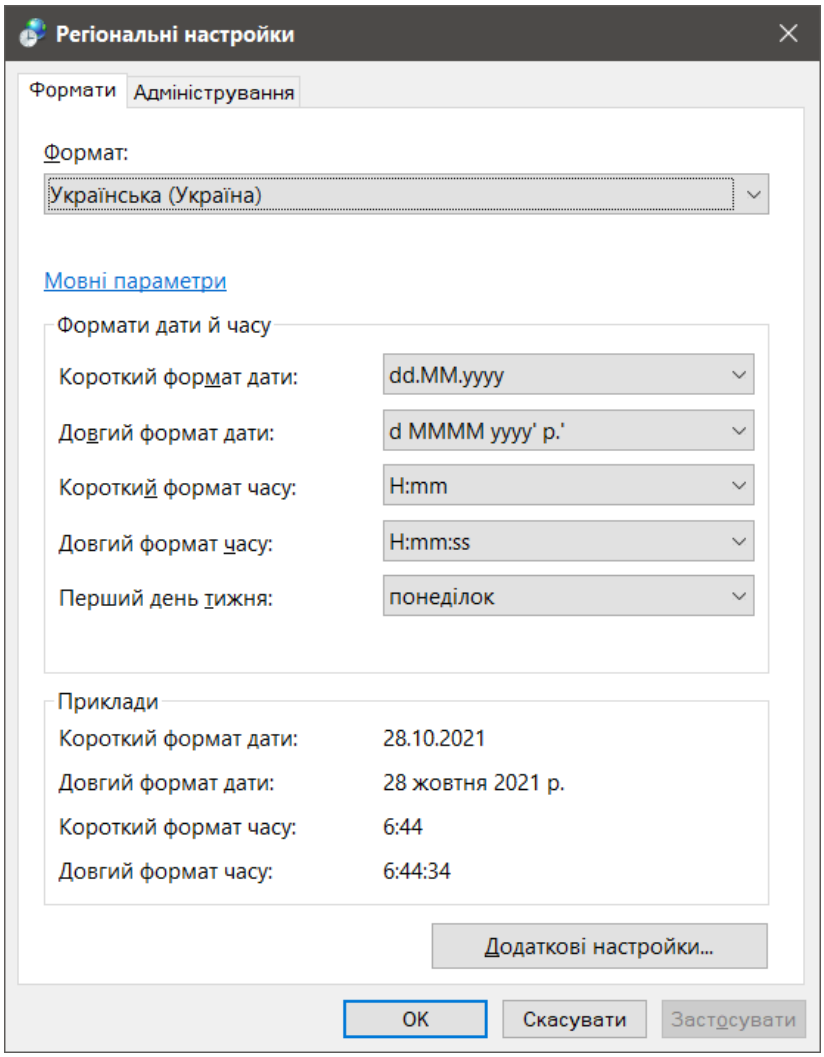

Рисунок 1.4 - Виклик вікна «Регіональні настройки» за допомогою утиліти rundll32.

Отже, маючи набір методів та засобів, можливо реалізувати комплексне обслуговування системного розділу на предмет його очистки (прибирання) з метою вивільнення вільного простору на ньому. Це можна здійснювати як в ручному режимі, що може зайняти досить тривалий час, або реалізувати алгоритми використання вище описаних методів з використанням системних або власних засобів для застосування цих методів на практиці у вигляді окремого програмного забезпечення, яке дозволить прискорити та автоматизувати процес оптимізації системного розділу.

1.7 Огляд сторонніх розробок для очистки та оптимізації системного розділу Windows 10

Існує велика кількість різноманітних програмних продуктів як комерційного так і вільного типу для реалізації методів оптимізації системного розділу, шляхом його очистки (прибирання). За досить тривалий період різними виробниками було розроблено цілий ряд таких готових рішень, проте жодне із них повністю не охоплює усі можливі варіанти очистки системного розділу. Проте, існує цілий ряд програмних продуктів, які досить ефективно реалізують частину таких методів.

Для порівняння було розглянуто штатну консольну утиліту DISM [16] та три безкоштовних, проте досить ефективних програмних засоби:

PatchCleaner (рис. 1.5) [4].

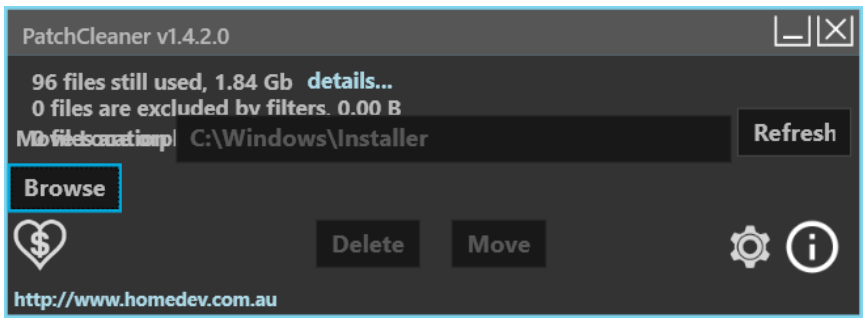

Рисунок 1.5 - Головне вікно PatchCleaner.

Wise Disk Cleaner X (рис 1.6.) [17].

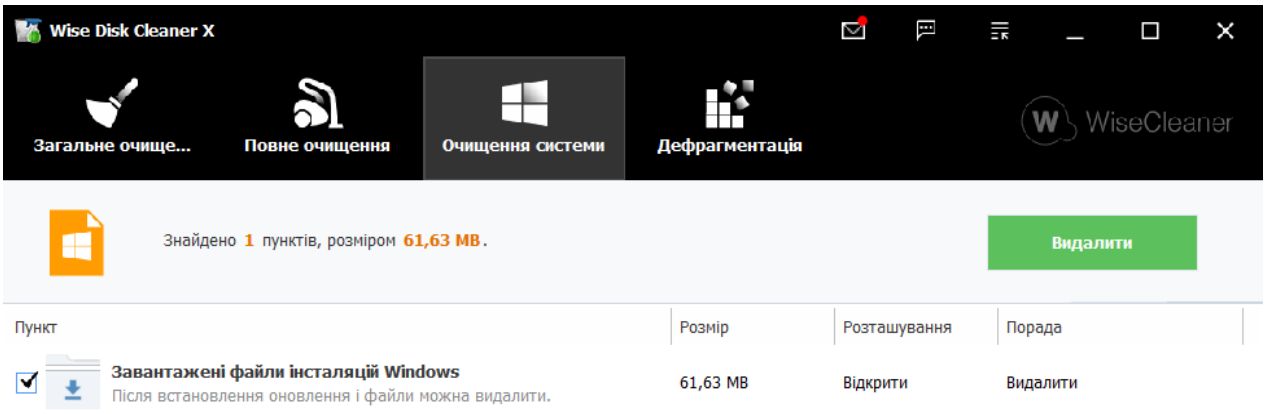

Рисунок 1.6 - Головне вікно Wise Disk Cleaner X.

DISM++ (рис 1.7.) [18].

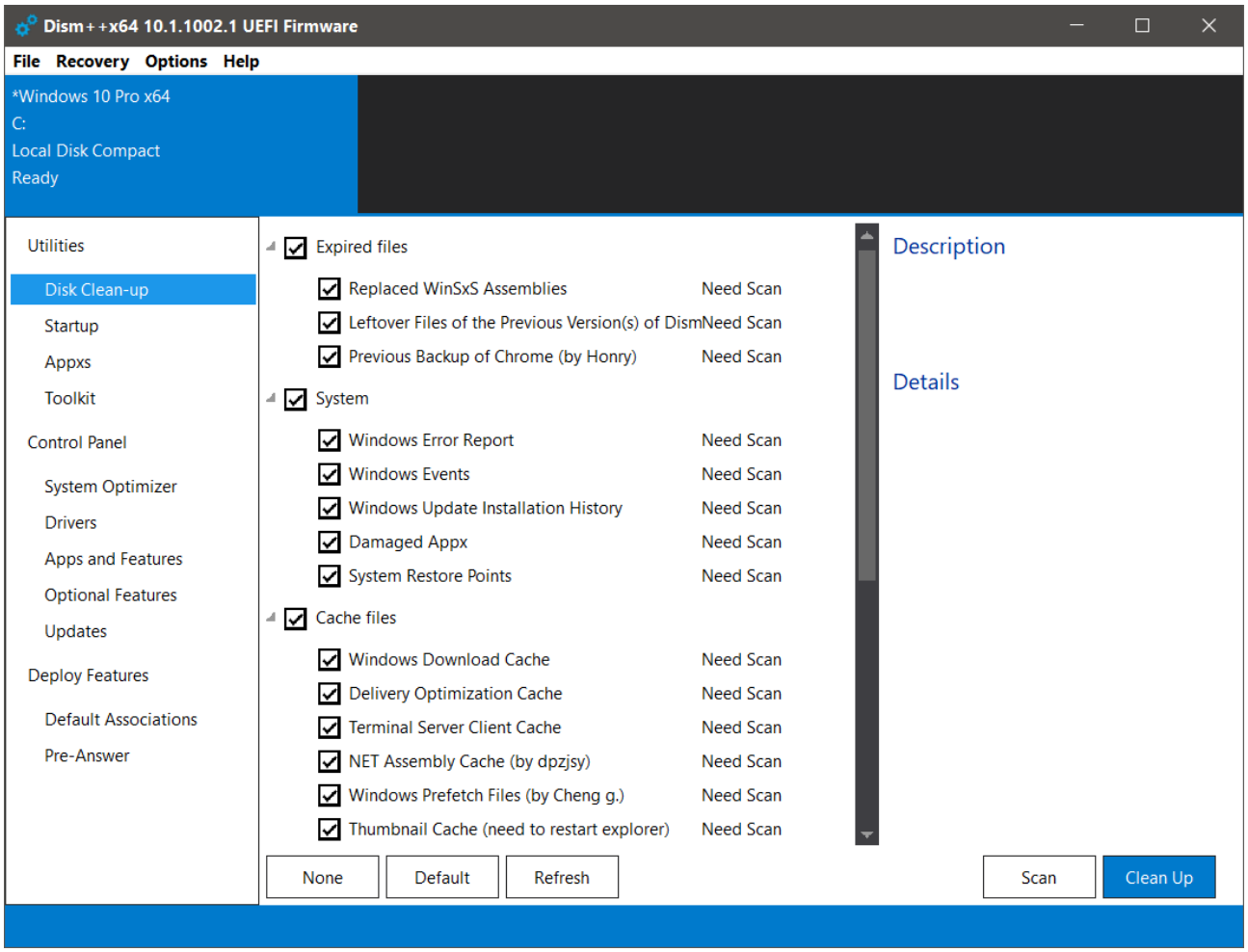

Рисунок 1.7 - Головне вікно DISM++.

Для порівняльного аналізу таких розробок було сформовано порівняльну таблицю (табл. 2).

Таблиця 2. Порівняння програмного забезпечення
по досліджуваному функціоналу:
«+» – наявна функція, «-» – відсутня, «+/-» – частково присутня.

Каталоги для очистки, або інші методи вивільнення простору	PatchCleaner	Wise Disk Cleaner	DISM	DISM++
%systemroot%\installer\	+	-	-	-
%systemroot%\installer\$PatchCache$	-	-	-	-
%systemroot%\installer*.tmp-	-	-	-	-
%systemroot%\winSxS\	-	+/-	+	+
%systemroot%\winSxS\Backup	-	+	-	+
Стиснення системних файлів	-	-	-	-
%systemroot%\system32\DriverStore	-	-	-	+
%systemroot%\system32\DriverStore\ \FileRepository	-	-	-	+
Керування гібернацією	-	-	-	-
Керування віртуальною пам’яттю	-	-	-	-
Видалення тимчасових файлів	-	+	-	+
Керування UWP	-	-	-	+
Видалення файлів пакетів оновлень	-	+/-	+	+
System Volume Information	-	-	-	-

Деякі з цих інструментів вузько спеціалізовані і поки єдині, які зустрічаються серед безкоштовних розробок (наприклад PatchCleaner), а інші мають розширений функціонал, який може стосуватись не тільки очистки системного розділу: Wise Disk Cleaner X та DISM++. Деякий функціонал не може

забезпечити жодна із розглянутих програм, а лише штатні вузькоспеціалізовані системні утиліти, або взагалі лише послідовність дій користувача від імені адміністратора.

Отже, на практиці, варто поєднувати у процесі оптимізації системного розділу декілька програмних засобів, які за своїми функціональними можливостями перекривають один одного та дозволяють максимально збільшити об’єм вивільненого простору на системному розділі та реалізувати налаштування операційної системи, що теж вивільняє простір, або не допускає його зменшення в майбутньому, в процесі експлуатації системи.

РОЗДІЛ 2
РОЗРОБКА СИСТЕМИ ДЛЯ ОПТИМІЗАЦІЇ СИСТЕМНОГО РОЗДІЛУ WINDOWS 10 «OSP»

2.1. Постановка задачі, призначення та вимоги до програмного засобу «OSP»

Системний розділ (%systemdrive%), де міститься сама операційна система, з часом заповнюється інформацією, яка не завжди є критично важливою, проте викликає дефіцит вільного дискового простору, що впливає як на швидкодію операційної системи, так і на функціонал та, навіть, на можливість подальшого безперебійного отримання оновлень та виправлень. Тому виникає потреба здійснити очистку (cleanup – прибирання) такого розділу, а також внесення змін в налаштування операційної системи з метою мінімізації втрат вільного простору в подальшій експлуатації системи. Такі операції, часто доводиться здійснювати не одноразово, а періодично, оскільки з часом системний розділ повторно загромаджується умовно непотрібною файловою інформацією, а також в процесі різноманітних оновлень налаштування системи можуть теж змінюватись та скидатись до значень за замовчуванням, які знову викликають втрату вільного простору на накопичувачі.

Маніпуляції для оптимізації системного розділу, шляхом вивільнення простору на ньому та здійснення налаштувань системи для збільшення цього простору можна здійснювати у ручному режимі, використовуючи штатні інструменти, доступні звичайному користувачеві, приховані від звичайного користувача утиліти, сторонні утиліти та безпосередні дії над об’єктами операційної системи. Такі маніпуляції досить тривалі, деякі з них вимагають дотримання послідовності їх виконання, отримання доступу до суміжних засобів та об’єктів, пошуку сторонніх засобів тощо. Іншим шляхом оптимізації системного розділу є розробка програмного засобу, який дозволив би автоматизувати такий процес, прискоривши його виконання, та передбачити можливі нюанси, які можуть виникнути в процесі оптимізації.

Тому розроблюваний програмний продукт повинен:

- містити єдине інтегроване середовище з графічним інтерфейсом користувача для легкого доступу до засобів оптимізації звичайному користувачеві;
- враховувати особливості роботи програмного забезпечення в операційні системі (середовище виконання, організації процесів тощо);
- давати змогу взаємодіяти з системними засобами операційної системи доступними для звичайного користувача та прихованими від нього;
- допускати завантаження, встановлення та роботу сторонніх засобів оптимізації;
- містити в собі реалізацію додаткових інструментів, які полегшують здійснення процесу оптимізації;
- давати можливість вибіркового застосування методів оптимізації за потребою користувача;
- містити у собі лаконічні та, водночас, вичерпні коментарі, зрозумілі користувачеві під час оптимізації.

Такий програмний продукт повинен інсталюватись в операційну систему і не викликати труднощів при своєму запуску та роботі, містити коротку інструкцію для користувача.

2.2. Вибір моделі розробки програмного засобу «OSP»

Розглянувши існуючі моделі розробки програмного забезпечення та врахувавши особливості призначення та функціонування майбутнього програмного продукту, за основу було обрано інкрементну модель, оскільки в програмному засобі можна чітко виділити окремі компоненти та складові частини, які можна розробляти та вдосконалювати окремо одна від одної.

Така модель передбачає стадійну послідовність розробки від версії до версії, в кожній із яких вводиться певне поліпшення та вдосконалення, шляхом внесення нових компонентів, поки не буде охоплено увесь запланований

функціонал. Розробка програмного забезпечення ведеться ітераціями з циклами зворотного зв'язку між етапами. Міжетапні коригування дозволяють враховувати реально існуючий взаємний вплив результатів розробки на різних етапах, час життя кожного з етапів розтягується на весь період розробки (рис. 2.1).

Рисунок 2.1 - Ітераційна модель життєвого циклу розробки.

На початку роботи визначаються всі основні вимоги до системи, поділяються на більш і менш важливі. Після чого виконується розробка системи за принципом розростання, так, щоб можна було використовувати дані, отримані в ході розробки ПЗ. Кожен інкремент повинен додавати системі певну функціональність. Життєвий цикл даної моделі характерний при розробці складних і комплексних систем, які містять велику кількість різнофункціональних компонент. Основна перевага такої моделі у можливості незалежної розробки та тестування кожної із компонент, незалежно одна від одної. До недоліку можна віднести підвищення складності та іноді навіть порушення загальної структури при додавання нових компонент.

2.3. Загальний опис проєкту

Програмна розробка повинна являти собою інтегроване графічне середовище у вигляді віконної форми, яка містить інструменти керування типу вкладок, кожна із яких представляє ту або іншу компоненту розробки, що стосується окремого об’єкту, або групи об’єктів, які підлягають оптимізації. Кожна із вкладок повинна містити елементи керування (з прив’язаними до них коментарями) через які здійснюється запуск того або іншого інструмента для здійснення оптимізації. Кожен із інструментів запускається окремим процесом та може працювати при необхідності паралельно, незалежно один від одного. Діаграму послідовностей програмної системи зображено на рис. 2.2.

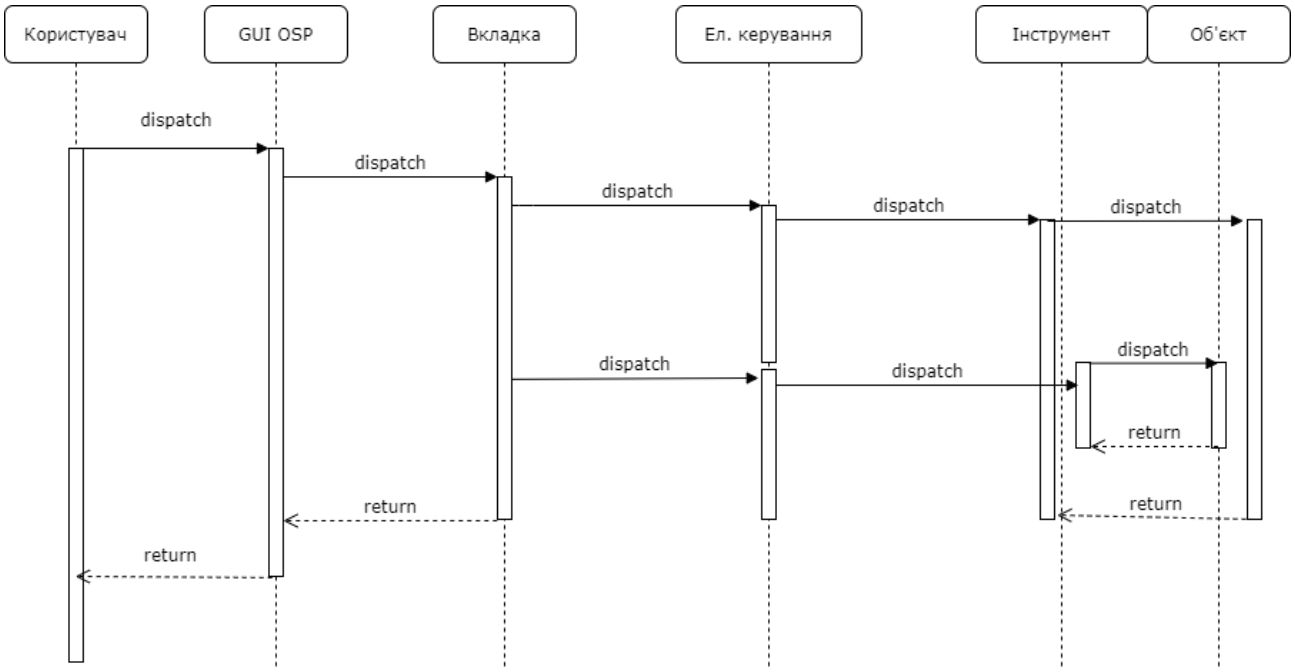

Рисунок 2.2 - Діаграма послідовностей програмної розробки

2.4. Обґрунтування вибору інструментальних засобів розробки

Враховуючи одну із вимог, що стосується середовища виконання та організації процесів для розробки інтегрованого середовища та запуску окремих сценаріїв незалежно одного від одного, було обрано платформу .NET, зокрема

середовище програмування Microsoft Visual C# 2010 Express Edition, що має можливість створення проектів з консольними застосунками та застосунками з візуальним інтерфейсом, що ґрунтується на Web, Windows Form або WPF (Windows Presentation Foundation) та ін. Оскільки планується розробка локального застосунку, то немає потреби використовувати Web інтерфейс, оскільки основні переваги його розкриваються при використанні клієнт-серверної моделі при розробці системи. WPF, хоч і більш універсальний інтерфейс, заснований на форматі на базі XAML, проте набір елементів керування є досить обмежений, а також користувач при запуску може стикнутись з проблемами пов’язаними з відсутністю деяких бібліотек, які встановлюються в процесі розгортання Microsoft Visual Studio і можуть бути відсутні в операційній системі за замовчуванням. Враховуючи такі особливості, за основу графічного інтерфейсу розроблюваної системи було обрано Windows Form (рис. 2.3).

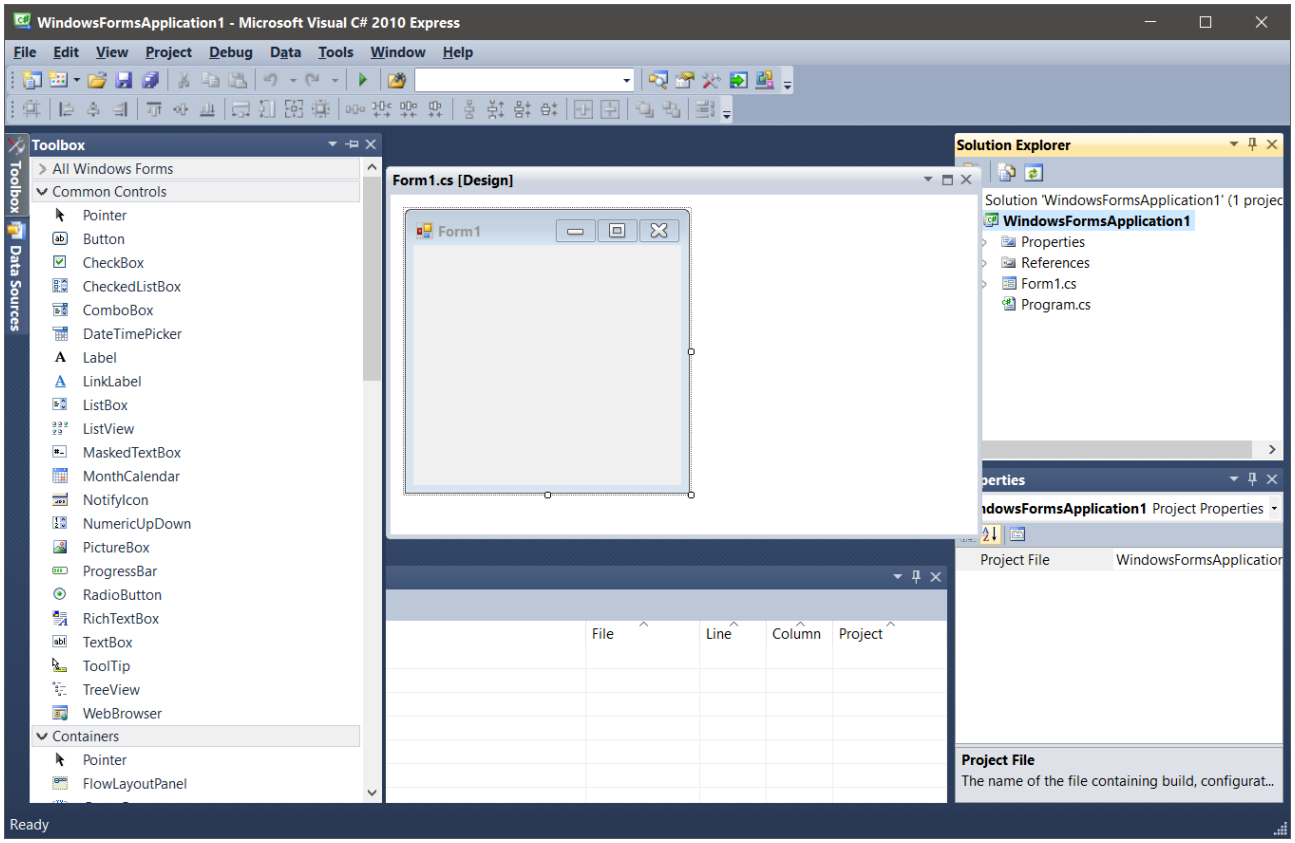

Рисунок 2.3 - Інтерфейс середовища розробки Microsoft Visual C# 2010 Express Edition, проект на базі Windows Form

Саме середовище розробки та розроблювані застосунки здатні працювати в 64-розрядних операційних системах Windows, зокрема в досліджуваній нами Windows 10 x64. Виконання відбувається у CLR (Common Language Runtime) з використанням JIT-компіляції. Середовище CLR надає для кожної підтримуваної комп'ютерної архітектури один або кілька JIT-компіляторів, а єдиний набір інструкцій MSIL можна компілювати і виконувати в будь-якій підтримуваній архітектурі. Коли компілятор C# створює код MSIL (Microsoft Intermediate Language – проміжна мова, або проміжний код, іноді просто IL), одночасно створюються метадані. Метадані містять опис типів в коді, включаючи визначення кожного типу, сигнатури кожного члена типу, члени, на які є посилання в коді, а також інші відомості, що використовуються середовищем виконання під час виконання. MSIL і метадані містяться в переносимому виконуваному (PE) файлі, який ґрунтується на форматах Microsoft PE і COFF, що раніше використовувалися для виконуваного контенту, але при цьому розширює їх можливості. Виконуваний код включає в себе посилання для зв'язування динамічно завантажуваних бібліотек, таблиці експорту і імпорту API функцій, дані для управління ресурсами і дані локальної пам'яті потоку (TLS). В операційних системах сімейства Windows NT формат PE використовується для EXE, DLL, SYS (драйверів пристроїв) та інших типів виконуваних файлів (рис. 2.4, 2.5). [19].

Виконуваний код включає в себе посилання для зв'язування динамічно завантажуваних бібліотек, таблиці експорту й імпорту API функцій, дані для управління ресурсами і дані локальної пам'яті потоку (TLS). В операційних системах сімейства Windows NT формат PE використовується для EXE, DLL, SYS (драйверів пристроїв) та інших типів виконуваних файлів.

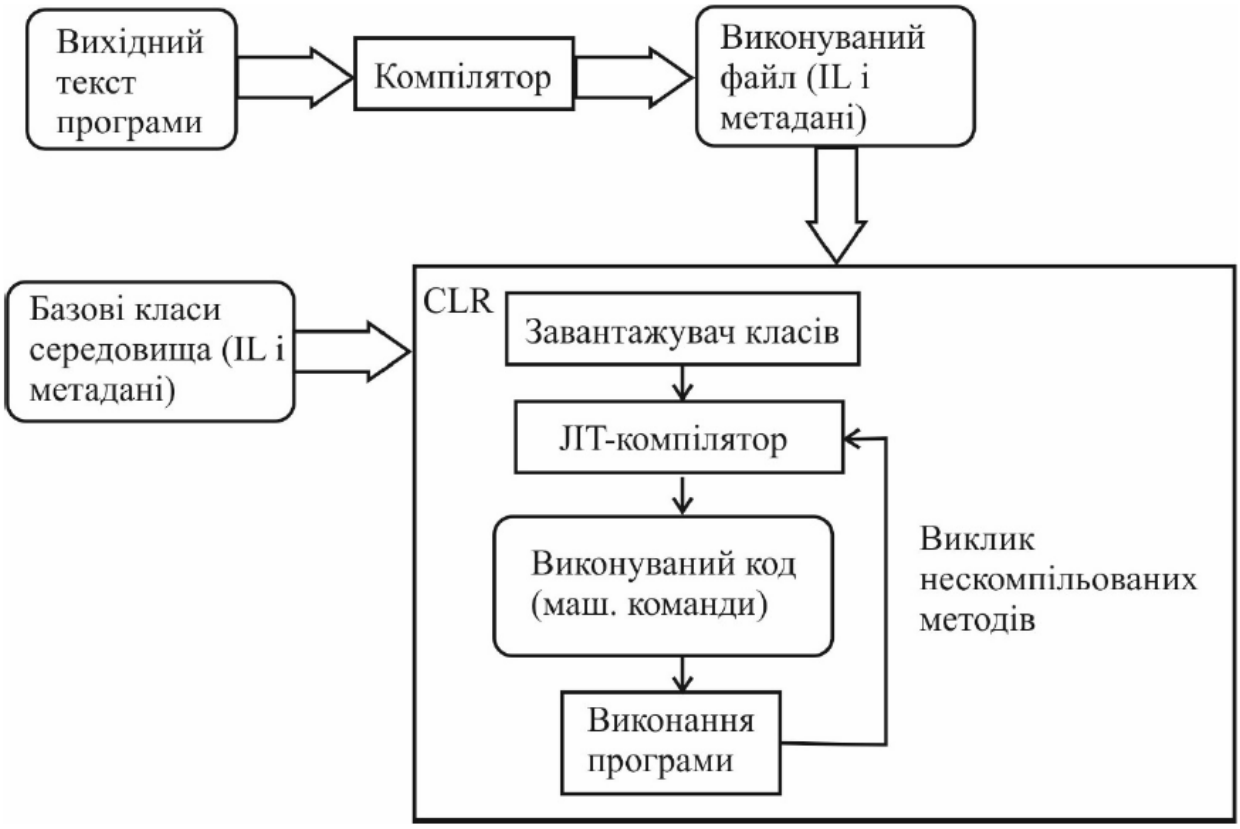

Рисунок 2.4 - Схема виконання програми в .NET

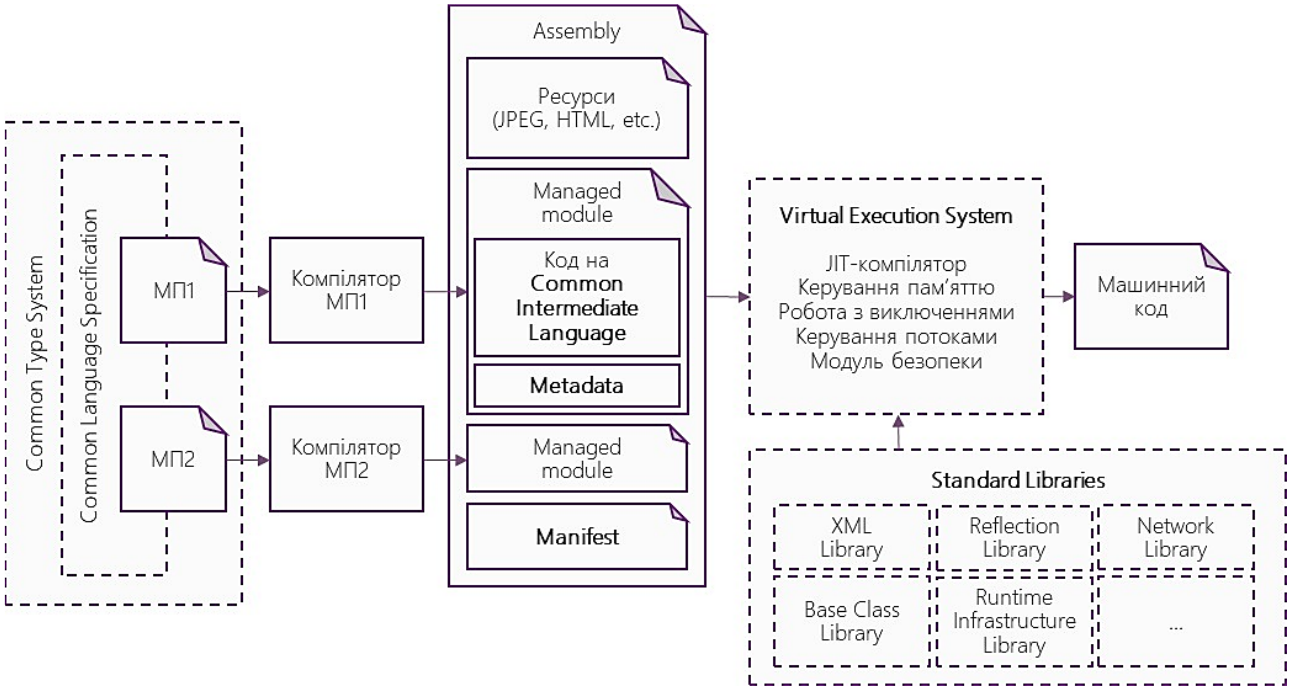

Рисунок 2.5 - Загальна схема перетворення мови програмування в машинний код

Цей формат файлів, що дозволяє розміщувати код MSIL або машинний код, а також метадані, дозволяє операційній системі розпізнавати образи середовища CLR. Наявність у файлі метаданих поряд з MSIL дозволяє коду описувати себе. Це усуває потребу в бібліотеках типів і у мові IDL (Interactive Data Language – мова програмування для аналізу даних). Середовище виконання знаходить і видобуває метадані з файлу в міру необхідності при виконанні.

При JIT-компіляції мова MSIL(IL) перетвориться в машинний код під час виконання додатка за вимогою, коли завантажується і виконується вміст збірки. Оскільки середовище CLR надає JIT-компілятор для кожної підтримуваної архітектури процесора, розробники можуть створювати набір збірок MSIL, які можуть компілюватися за допомогою JIT-компілятора і виконуватися на різних комп'ютерах з різною архітектурою. Подібна схема компіляції в міру необхідності може істотно знизити розміри породжуваного програмного коду і скоротити час підготовки збірки для виконання, разом з цим, в середовищі CLR може бути виконана і повна компіляція збірки перед початком виконання (pre-JITing) [19].

Для виконання засобів, що реалізують засоби очищення (прибирання) та налагодження операційної системи з одного боку найпростішим, а з іншого найефективнішим способом організації кожного із цих процесів є виконання окремих пакетних та інших сценаріїв, причому кожен в окремому процесі. Тобто фактично відбуватиметься передача керування процесами самій операційній системі, в якій такий механізм передбачено заздалегідь і який найкраще справиться з поставленою задачею і не викликатиме конфліктів. Основними інструментами для задіяння можливостей операційної системи є виконання скриптів (в широкому розумінні цього слова) та виклику функцій системних бібліотек, шляхом використання інтерпретатора командного рядка (cmd), робочим середовищем Windows PowerShell (WPS), редактором системного реєстру (regedit) та різноманітними системними утилітами типу RunDLL32. Більшість цих інструментів має власний користувацький інтерфейс, або використовує графічний інтерфейс операційної системи, через який можливе

повернення результатів виконання відповідних скриптів, команд та утиліт (рис. 2.6).

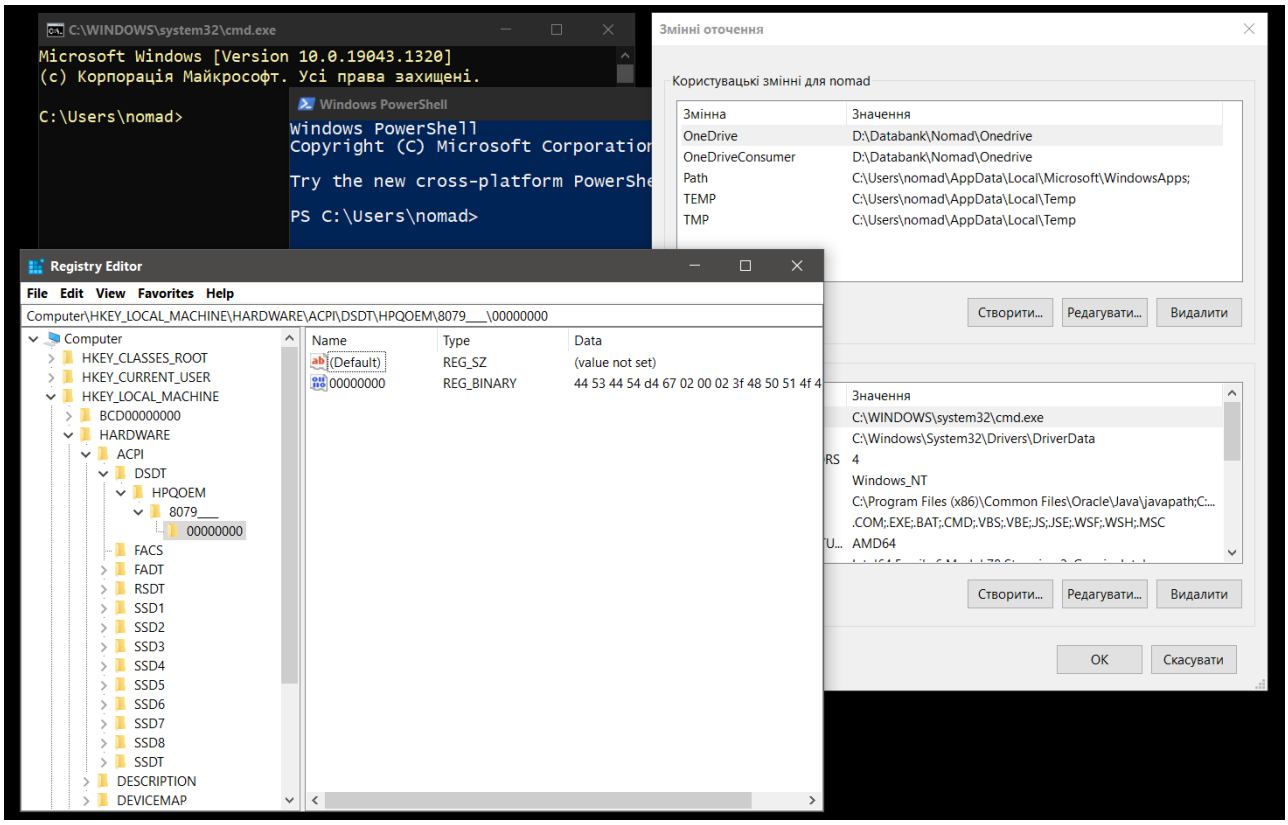

Рисунок 2.6 - Приклади користувацьких інтерфейсів cmd, PowerShell, regedit та виконання утиліти *rundll32.exe sysdm.cpl,EditEnvironmentVariables*.

2.5. Особливості програмної реалізації та основні режими роботи «OSP»

За основу розроблюваного програмного засобу було обрано застосунок створений у середовищі розробки Microsoft Visual C# 2010 Express Edition. Інтерфейс застосунку базується на Windows Form і являє собою вікно фіксованого розміну та позиції, яке містить окремі вкладки (TabControl - TabPage), кожна із яких відповідає за той чи інший реалізований метод оптимізації системного розділу.

У кожній вкладці (TabPage) згруповані окремі елементи керування Button, RichTextBox, PictureBox та LinkLabel, які дозволяють викликати зовнішні

об’єкти (команди, скрипти, пакетні файли, додатки, утиліти), містять описову інформацію щодо об’єктів, які викликаються та посилання на пов’язані з ними ресурси, якщо такі існують.

Розглянемо кожен із методів очистки та налагодження, та особливості їх реалізації. Кожен із них пов’язаний з тим чи іншим об’єктом файлової структури операційної системи.

Installer. Порівняно великий об’єм (декілька гігабайт) займає на системному розділі каталог %systemroot%\Installer (він є прихованим), у якому міститься набір файлів з розширенням .msi та .msp, а також папки іменовані у вигляді кодів з розширенням .tmp-. Цей системний каталог зберігає необхідні для видалення або оновлення програм інсталятори, інсталятори різних програмних компонентів, патчі, файли не сталих системних оновлень й інші дані, які необхідні системі та встановленому програмному забезпеченню для власного функціоналу. Вони використовують для установки службу «інсталятор Windows» і використовуються, коли програма оновлюється або коли її, навпаки, видаляють через «Панель керування» – «Програми та засоби». Також вони можуть використовуватися для функції «Виправити установку». Відповідно, при їх знищенні можуть виникнути проблеми при виконанні таких завдань. [20,21]

Очищення цієї папки варто проводити наступним способом. Операційна система Windows містить список поточних установників і патчів, до яких можна отримати доступ через виклики WMI (Інструментарій управління Windows) [22]. Актуальний список можна вивести виконавши в середовищі командної оболонки PowerShell наступну команду (рис. 2.7):

```
get-wmiobject Win32_Product | Sort-Object -Property Name |Format-Table IdentifyingNumber, Name, LocalPackage -AutoSize
```

а, отже є можливість написання сценарію для автоматизації очищення каталогу Insaller. Все, що знаходиться в папці, але не в списку, вважається «сиротинськими» файлами і може бути переміщено або видалено без шкоди функціоналу системи. Іншим способом може бути використання готового рішення – безкоштовної утиліти PatchCleaner [4].

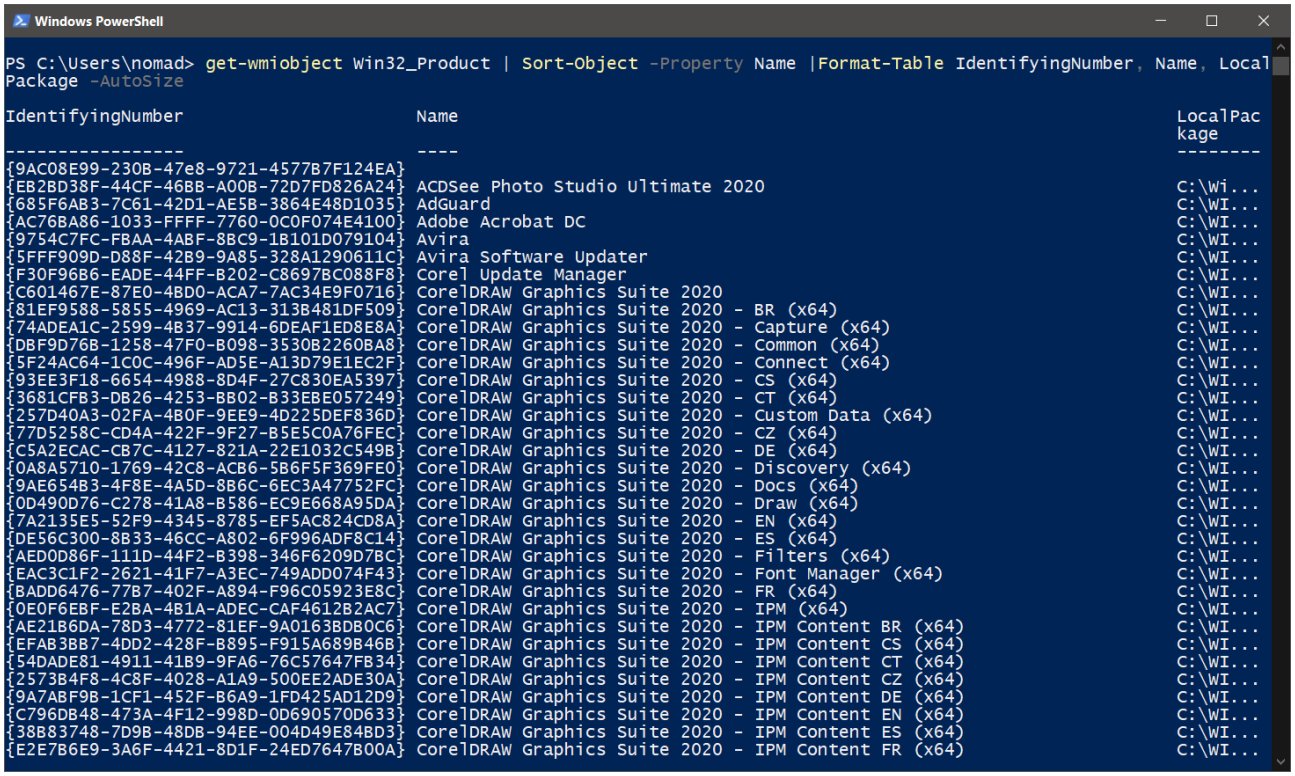

Рисунок 2.7 - Виклик списку встановлених пакунків через WMI за допомогою PowerShell

Папки з розширенням .tmp- є тимчасовими в вище вказаному каталозі, тому після завершення встановлення, або оновлення їх можна позбутися звичайним способом, якщо вони не зникають самі через деякий час.

Особлива папка в цьому каталозі %windir%\Installer\$PatchCache$. Ця папка може з’явитись в процесі оновлення системи і залишиться по його завершенню. В ній зберігаються базові версії файлів, які змінюються при оновленні ПЗ за допомогою msp-патчів і використовуються вони для того, щоб не просити дистрибутив при кожному оновленні. Якщо установник цих файлів не знайде, то попросить вставити диск або вказати шлях, звідки була встановлена програма [23]. Після завершення оновлення потреба в ній відпадає. А при наступному оновленні вона створюється знову автоматично і автоматично заповнюється необхідними пакетами.

Для реалізації очистки вище вказаного каталогу створюється серія пакетних файлів, які містять наступні команди для його вибіркової очистки:

```
del /f /s /q %windir%\Installer\$PatchCache$
rd /s /q %windir%\Installer\$PatchCache$

powershell rd %windir%\Installer\*.tmp-
```

та налаштування змінних середовищ, через виклик функції системної бібліотеки:

```
rundll32.exe sysdm.cpl,EditEnvironmentVariables
```

Такі пакетні файли запускаються через код додатка на C# як окремий процес відповідного класу простору імен (в дужках вказується шлях до відповідного пакетного файлу у каталозі проекту):

```
System.Diagnostics.Process.Start(@".\..\..\Bat-files\1-1.bat");
```

Тобто перша вкладка додатку (рис. 2.8) буде містити елементи запуску пакетних файлів, функцій, стороннього додатку і детальні коментарі до них.

Рисунок 2.8 - Вкладка «Installer» застосунка OSP.

WinSxS. Ще один системний каталог великого розміру розміщений за шляхом %windir%\WinSxS. В папці WinSxS зберігаються резервні копії системних файлів операційної системи до оновлень і не тільки. Тобто, кожного разу, після отримання і встановлення оновлення Windows, в цю папку зберігається інформація про змінювані файли та самі ці файли з тим, щоб була можливість видалити оновлення і відкотити зроблені зміни. Через якийсь час папка WinSxS може займати досить багато місця – кілька гігабайт, при цьому розмір цей весь час збільшується в міру установки нових оновлень Windows. Насправді, ця папка займає значно менше місця, ніж здається. Багато файлів з папки Windows проектується за допомогою жорстких посилань саме з папки WinSxS. Отже, звичним чином очищати таку папку не варто, адже вона зберігає не тільки файли, пов’язані з оновленнями, але і файли самої системи, що використовуються в процесі роботи, а також для того, щоб повернути ОС в початковий стан або виконати деякі операції, пов'язані з відновленням. [24]

Для безпечної роботи з цією папкою передбачено штатну систему обслуговування образів розгортання і управління ними – DISM (Deployment Image Servicing and Management). Це засіб командного рядка, який може використовуватися для обслуговування образу Windows або для підготовки образу середовища попереднього встановлення Windows (Windows PE). До його функцій входить: додавання, видалення і перерахування пакетів; додавання, видалення і перерахування драйверів; включення і відключення компонентів Windows; модернізація Windows до іншого випуску; обслуговування всіх платформ (32-розрядні, 64-розрядні і Itanium); використання старих сценаріїв диспетчера пакетів та інші [25–27].

За допомогою DISM можна детально проаналізувати сховище WinSxS:

```
Dism.exe /Online /Cleanup-Image /AnalyzeComponentStore
```

Приклад результату зображено на рис. 2.9, де:

```
Component Store (WinSxS) information:

Windows Explorer Reported Size of Component Store : 5.10 GB

Actual Size of Component Store : 5.07 GB

    Shared with Windows : 3.81 GB
    Backups and Disabled Features : 1.25 GB
    Cache and Temporary Data :  0 bytes

Date of Last Cleanup : 2021-09-16 20:59:33

Number of Reclaimable Packages : 0
Component Store Cleanup Recommended : No

The operation completed successfully.
```

Рисунок 2.9 - Результат аналізу сховища за допомогою DISM.

- Windows Explorer Reported Size… – об’єм сховища без врахування жорстких посилань;
- Actual Size of Componets Store – з врахуванням жорстких посилань (за винятком посилань на папку Windows);
- Shared with Windows – файли, спільні з Windows (є важливими і використовуються);
- Backups and Disable Features – резервні копії та відключені компоненти (умовно не важливі, не впливають на поточну роботу системи);
- Cache and Temporary Data – кеш і тимчасові файли (не є обов’язковими).

Також тут надається рекомендація про необхідність очищення – Component Store Cleanup Recommended.

За допомогою DISM можна здійснити очистку сховища компонентів:

```
Dism.exe /Online /Cleanup-Image /StartComponentCleanup
```

, видалити файли, необхідні для видалення пакетів оновлень, після чого неможливо буде видалити встановлені оновлення:

```
DISM.exe /online /Cleanup-Image /SPSuperseded
```

.

У параметра /StartComponentCleanup є додатковий ключ /ResetBase, за допомогою якого можна видалити всі попередні версії компонентів.

DISM дозволяє також здійснити процес сканування:

```
DISM /Online /Cleanup-Image /ScanHealth
```

та відновлення компонентів сховища у разі його пошкодження:

```
DISM /Online /Cleanup-Image /RestoreHealth
```

Оформивши вказані команди у відповідні пакетні файли, створено можливість їх запуску через елементи керування вкладки «DISM(winSxS)» застосунку на C# як окремих незалежних процесів. Додано також можливість очистки каталогу %windir%\WinSxS\Backup – сховища для розпакованих системних оновлень, потреба в якому відпадає після їх установки (рис. 2.10).

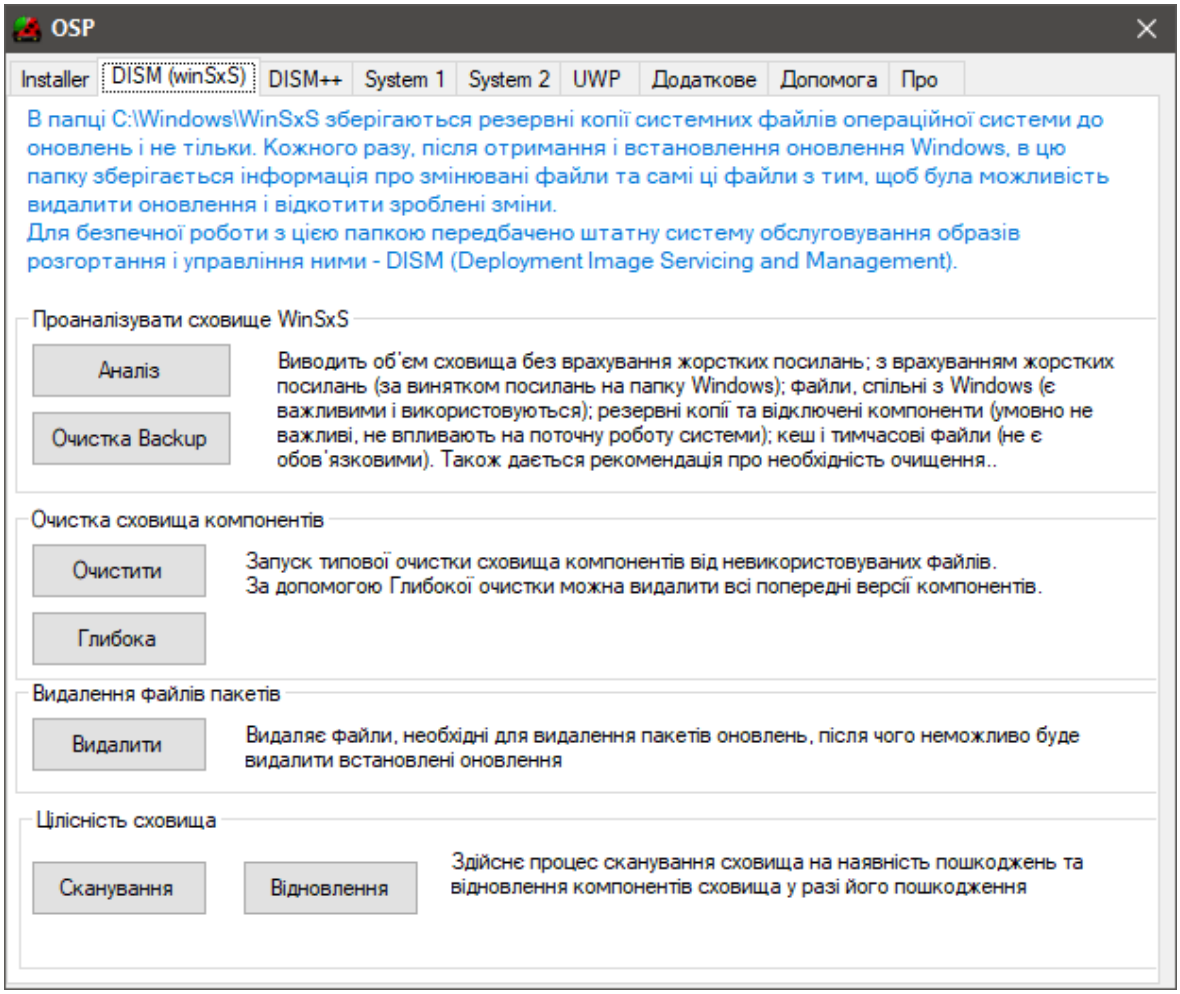

Рисунок 2.10 - Вкладка «DISM(winSxS)» застосунку OSP.

Наступна вкладка «DISM++» призначена для доступу до стороннього однойменного безкоштовного додатку (рис. 1.7, 2.11). Зокрема подано короткий коментар, що стосується його ролі в очистці системного розділу з зображенням відповідної частини його інтерфейсу, яка відповідає за необхідну функцію. В DISM++ поряд з іншим функціоналом реалізовано систему очистки каталогу WinSxS від сміття. Dism ++ спочатку був розроблений як графічна панель управління системою обслуговування образів розгортання і управління ними (DISM) для командного рядка. Однак, зараз додаток пропонує набагато ширші можливості, ніж створення і управління системними образами.

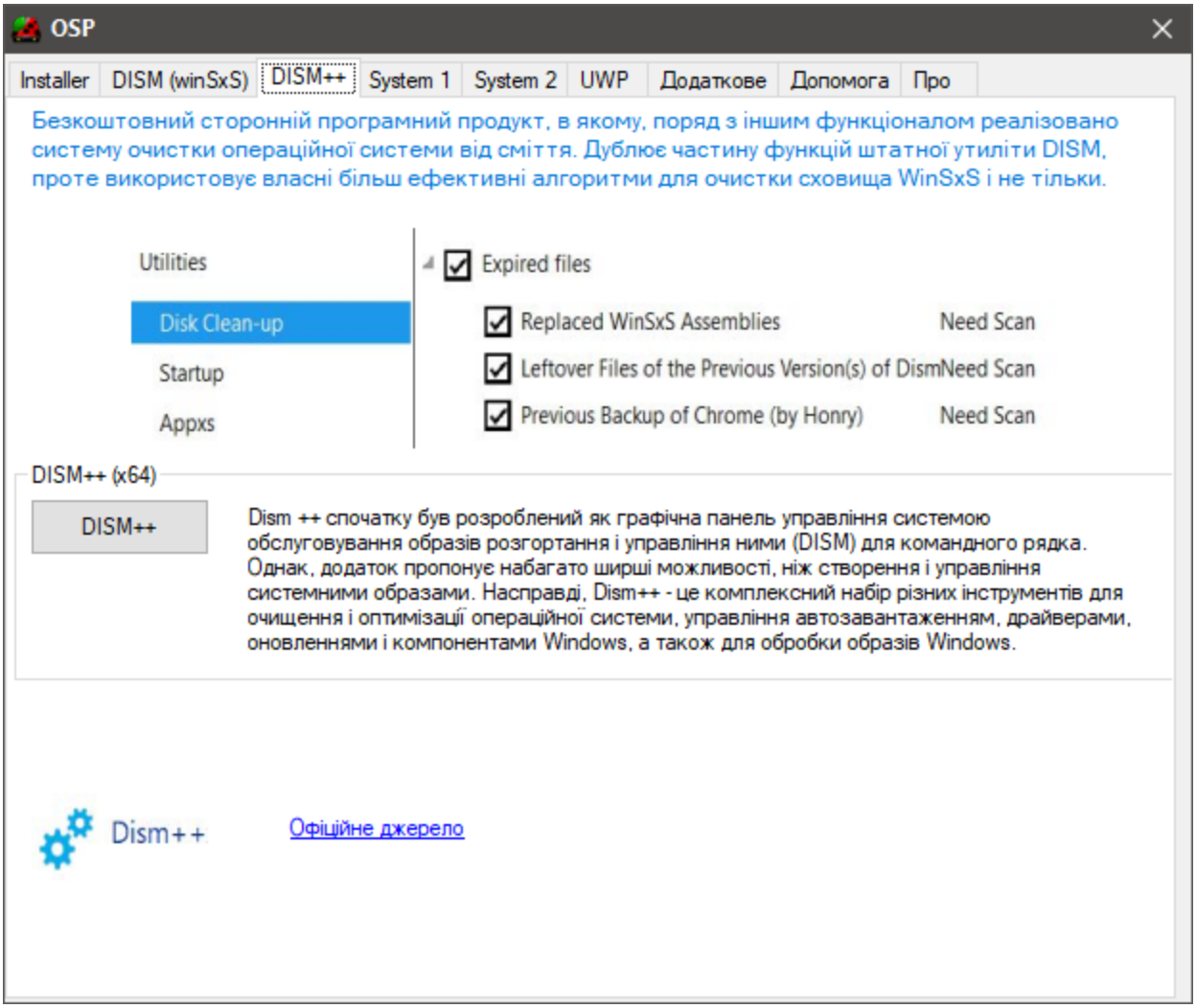

Рисунок 2.11 - Вкладка «DISM++» застосунку OSP.

Насправді, Dism++ – це комплексний набір різних інструментів для очищення і оптимізації операційної системи, управління автозавантаженням, драйверами, оновленнями і компонентами Windows, а також для обробки образів Windows. І хоча цей застосунок дублює частину функцій штатної утиліти DISM (від якого походить його назва), в останніх його версіях реалізовані власні, більш ефективні алгоритми обробки вище вказаного системного сховища і не тільки. Такий функціонал працює у два етапи: аналіз сховища з подальшими рекомендаціями та, власне, очистка такого сховища з можливістю вибору окремих опцій. Через цю вкладку OSP можна запустити такий додаток (який додано до складу OSP), або через подане посилання перейти на офіційний сайт виробника, звідки завантажити останню актуальну версію DISM++.

Наступні вкладки користувацького інтерфейсу OSP: «System 1» та «System 2» (Рис. 2.12, 2.13 відповідно) групують у собі елементи призначені для керування іншими механізмами вивільнення простору на системному розділі, пов’язані із особливостями функціоналу операційної системи, і призначені для зупинки/запуску деяких служб, відключення/включення деяких можливостей операційної системи, викликом сторонніх допоміжних інструментів для очистки тощо.

Детально розглянемо такі механізми.

Стиснення системних файлів (CompactOs). В останніх версіях ОС Windows з’явилась офіційна можливість стиснення системних файлів, на відміну від можливості стиснення цілого системного розділу в файловій системі NTFS, яка існувала в старших операційних системах Windows [28]. Для цього було введено штатну системну утиліту командного рядка: COPMACT.

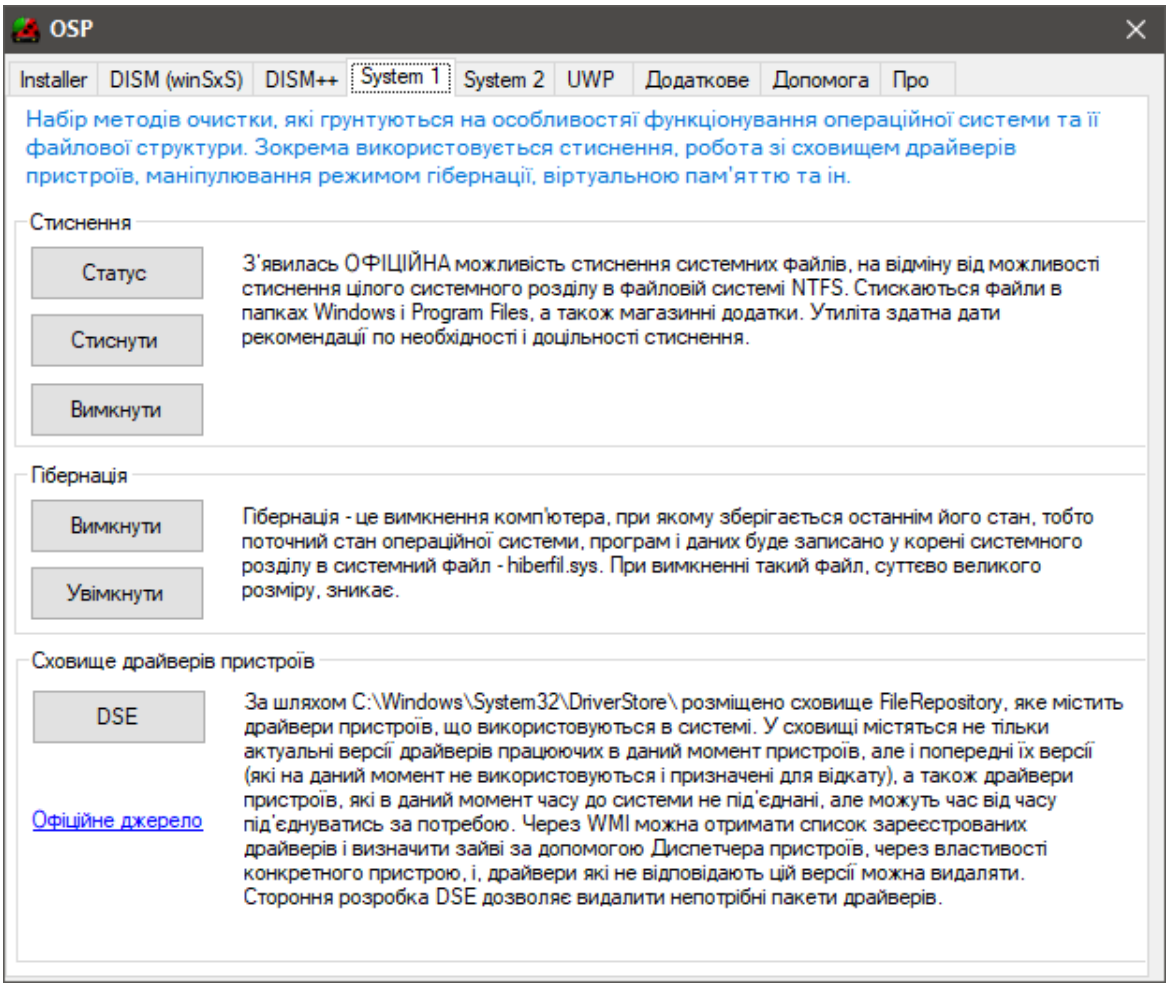

Рисунок 2.12 - Вкладка «System 1» застосунку OSP.

Стисненню підлягають файли в папках Windows і Program Files, а також магазинні додатки. Системні файли забезпечують значну частку зекономленого місця [28]. Це добре видно зі звіту DISM про сховище компонентів (рис. 2.9). І хоча, зрозуміло, «стиснена» система працюватиме повільніше, на практиці, враховуючи розвиток апаратних засобів (як то нові покоління процесорів, чіпсетів, твердотільних накопичувачів, тощо), таке сповільнення досить незначне, проте дозволяє зекономити декілька гігабайт вільного простору. І навіть вище вказана утиліта здатна дати рекомендації по необхідності і доцільності стиснення.

Зокрема для з’ясування статусу стиснення та отримання рекомендацій використовується ключ:

```
/CompactOS:query
```

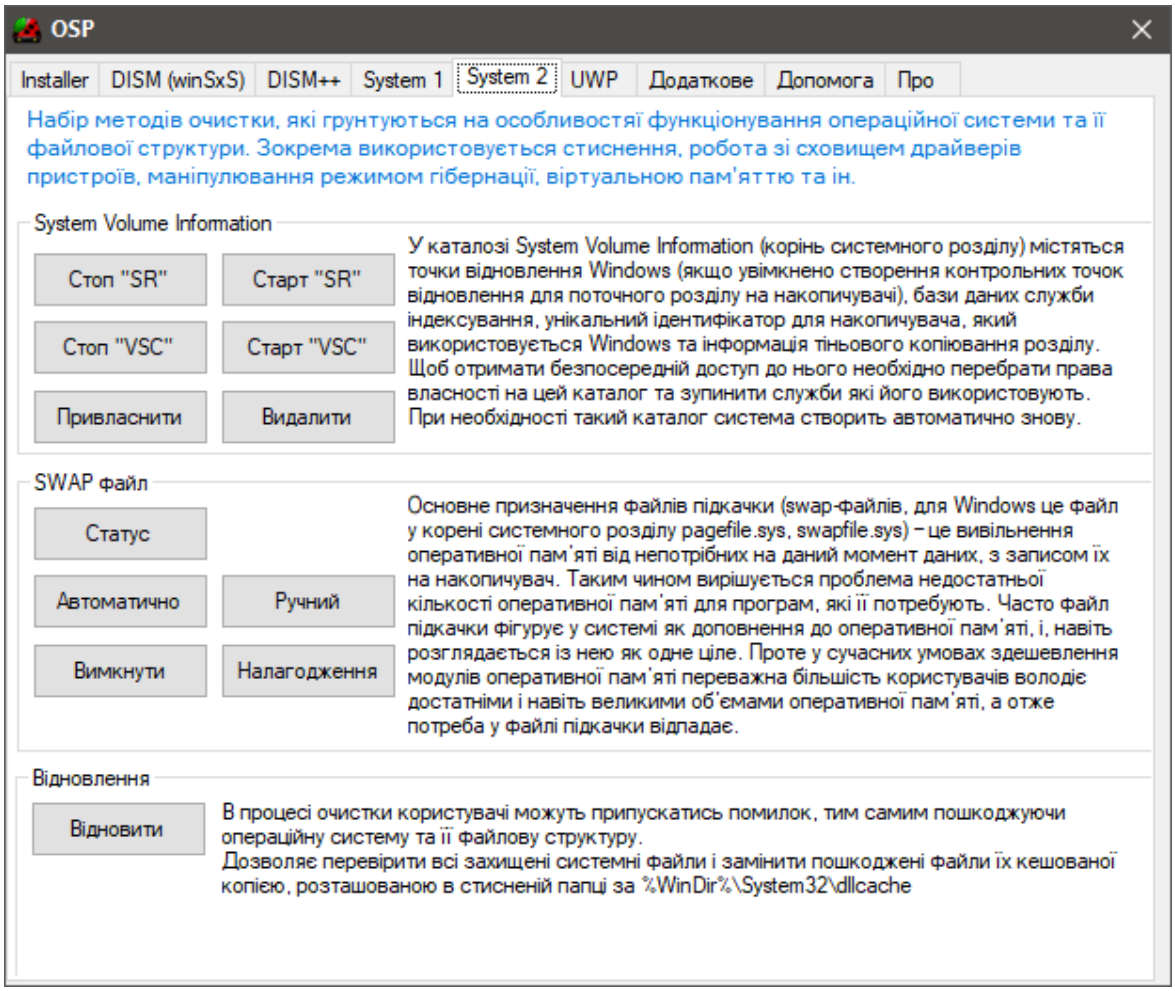

Рисунок 2.13 - Вкладка «System 2» застосунку OSP.

для здійснення стиснення:

```
/CompactOS:always
```

,

а відміни стиснення:

```
/CompactOS:never
```

.

Це реалізовано в окремих пакетних файлах, які можна запускати через додаток OSP.

Зрозуміло, що результати будуть різнитись між окремим конкретними системами, але орієнтовно можна зекономити біля 20% простору, який занятий системними фалами, що в абсолютних величинах може скласти 2–3 гігабайти.

Гібернація. Гібернація – це вимкнення комп’ютера, при якому зберігається останній його стан. Тобто поточний стан операційної системи, програм і даних буде записано у корені системного розділу в системний файл -

hiberfil.sys. При наступному увімкненні система відразу повертається в початковий стан, з усіма відкритими програмами, вікнами та даними на момент його вимкнення, і можна продовжувати роботу, не витрачаючи часу на повторний запуск системи та програм. Такий стан схожий із режимом сну, проте при гібернації живлення вимкнене повністю, адже вся поточна інформація записується на жорсткий диск, а в режимі сну споживається мінімальна потужність системою для підтримання інформації в оперативній пам’яті. Обидва режими корисні, коли тимчасово робота припиняється і суттєво або повністю потрібно зменшити споживання енергії. Режим гібернації дозволяє дещо швидше приводити систему у робочий стан, ніж її повторне повне завантаження, але повільніший ніж режим сну. Нас цікавить власне файл hiberfil.sys, розмір якого співрозмірний з об’ємом оперативної пам’яті. Відмовившись від режиму гібернації (затратити трохи більше часу на повне завантаження системи, або скористатись режимом сну), можна зекономити суттєвий простір на системному розділі в декілька, або навіть декілька десятків гігабайт, який займає hiberfil.sys. [29]

Це можна зробити, або через налаштування операційної системи, або через командний рядок:

```
powercfg -h off
```

Щоб увімкнути цей режим знову:

```
powercfg -h on
```

що реалізовано в окремих пакетних файлах, які запускаються через OSP.

Сховище драйверів. За шляхом %Systemroot%\System32\DriverStore\ розміщено сховище FileRepository, яке містить драйвери пристроїв, що використовуються в системі. Займати таке сховище може від кількох до десятка гігабайт, в залежності від кількості таких пристроїв та частоти оновлення їх драйверів. Причому у сховищі містяться не тільки актуальні версії драйверів працюючих в даний момент пристроїв, але і попередні їх версії (які на даний момент не використовуються і призначені для відкату), а також драйвери пристроїв, які в даний момент часу до системи не під’єднані, але можуть час від

часу під’єднуватись за потребою, наприклад: смартфони, камери, модеми, тощо. Просто очистити таку папку (DriverStore) не можна, оскільки це призведе до виходу з ладу усієї системи, або відключення окремих пристроїв. Проте очистка сховища FileRepository не призведе до краху системи, але при зміні складу устаткування може знадобитись повторне встановлення драйверів умовно нових пристроїв із зовнішніх джерел.

Тому для чистки користуються спеціальним прийомами за допомогою штатної утиліти командного рядка PNPUTIL. Ключ /e дозволяє повернути список усіх пакетів драйверів, що зберігаються у FileRepository (рис. 2.14). Проаналізувавши такий список (за датою, версією, назвою) можна видалити той пакет, в якому вже немає необхідності за допомогою ключа /d.

Наприклад:

`pnputil.exe /d oemXX.inf` , де oemXX.inf – файл, що ідентифікує пакет драйверів, а XX – номер драйвера.

```
Published name :            oem24.inf
Driver package provider :   Microsoft
Class :                     Принтери
Driver date and version :   06/21/2006 10.0.19041.1
Signer name :               Microsoft Windows

Published name :            oem26.inf
Driver package provider :   Samsung
Class :                     Принтери
Driver date and version :   07/31/2009 3.4.95.0
Signer name :               Microsoft Windows Hardware Compatibility Publisher

Published name :            oem33.inf
Driver package provider :   TP-Link
Class :                     Android Phone
Driver date and version :   08/28/2014 11.0.0.0
Signer name :               Microsoft Windows Hardware Compatibility Publisher
```

Рисунок 2.14 - Фрагмент списку поверненого командою *pnputil /e*.

Наступний крок – визначити драйвер пристрою, який на даний момент використовується. Це можна зробити за допомогою Диспетчера пристроїв, через властивості конкретного пристрою, і, драйвери які не відповідають цій версії можна видаляти. [6, 30] Більш зручно такі маніпуляції можна проводити за

допомогою сторонніх інструментів, наприклад Driver Store Explorer [31] (рис. 2.15).

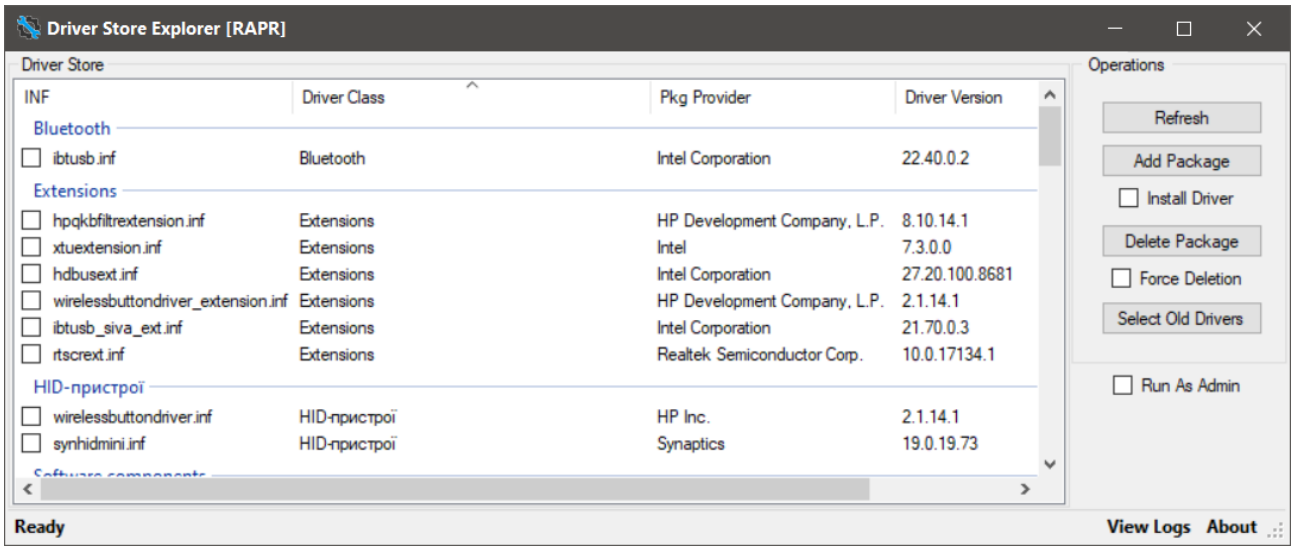

Рисунок 2.15 - Основне вікно застосунку Driver Store Explorer.

Такий інструмент було введено у склад OSP з можливістю його запуску та завантаження останньої версії з офіційної сторінки.

Окрім драйверів, що зберігаються в %Systemroot%\System32\DriverStore\, багато виробників під час запуску установника драйверів свого пристрою дублюють сховище у власному пакеті програмного забезпечення, яке встановлюється у систему, а також розміщують ці драйвері за нестандартними для операційної системи шляхами. Наприклад, при встановленні принтерів Xerox, або Samsung папки з драйверами крім штатного вищеописаного сховища розміщуються у %windir%\Drivers\Xerox\, %windir%\Drivers\Samsung\ та %systemdrive%\Program Files (x86)\Xerox\ і %systemdrive%\Program Files (x86)\Samsung\, що зроблено, очевидно, для сумісності власного ПЗ з старшими версіями операційних систем. Тому, варто здійснити перегляд вказаних каталогів на наявність аналогічних папок для подальшого їх видалення, що ніяк не вплине на подальше функціонування пристроїв.

System Volume Information. На розділах усіх накопичувачів (а, отже і на системному розділі) розміщені і автоматично створюються в корені системні

приховані каталоги System Volume Information, які у об’ємі можуть сягати кількох десятків гігабайт.

У цьому каталозі містяться точки відновлення Windows (якщо увімкнено створення контрольних точок відновлення для поточного розділу на накопичувачі), бази даних служби індексування, унікальний ідентифікатор для накопичувача, який використовується Windows та інформація тіньового копіювання тому (якщо працює історія файлів Windows [32]). Інакше кажучи, в папці System Volume Information зберігаються дані, необхідні для роботи служб з цим накопичувачем, а також дані для відновлення системи або файлів за допомогою засобів відновлення Windows.

Такий каталог на розділі з файловою системою NTFS має не тільки атрибут «системний» та «прихований», а й права доступу, що обмежують дії користувача з ним. Щоб отримати безпосередній доступ до нього необхідно змінити параметри безпеки у вкладці «Безпека» властивостей цього каталогу. Проте для файлових систем FAT, FAT32, exFAT будь-який користувач одразу має можливість працювати з таким каталогом.

Отримати повний доступ до такого каталогу на системному розділі можна, виконавши команду керування безпекою в командному рядку:

```
cacls %systemdrive%\"System Volume Information" /g <username>:f
```

де <username> – ім’я поточного користувача. Або можна перебрати права власності на цей каталог, автоматично отримавши повний доступ до неї (саме такий спосіб використано в OSP):

```
takeown /f "%SystemDrive%\System Volume Information" /r
```

Контрольні точки (які найбільше займають місця на розділі) можна видалити викликавши через командний рядок вікно налаштувань «Захист системи»:

```
SystemPropertiesProtection
```

Або за допомогою виклику відповідної функції системної бібліотеки

```
rundll32.exe shell32.dll,Control_RunDLL sysdm.cpl,,4
```

Проте цього може бути недостатньо, щоб одразу знищити весь каталог, оскільки служби, які його використовують, можуть працювати і використовувати в ньому файли та папки, а, отже, не дозволять їх знищити. Тому перед видаленням необхідно зупинити основні служби, а саме: «Захист системи» (SR), «Volume Shadow Copy» (VSC) та «Історія файлів». За замовчуванням «Історія файлів» відключена одразу, якщо її явно не увімкнув користувач при налагоджені (для цього необхідно задіяти окремий накопичувач) [32]. «Захист системи» можна зупинити за допомогою зміни параметра реєстру операційної системи (через редактор реєстру Regedit) DisableSR (0 – увімкнено «Захист системи» , 1 – вимкнено «Захист системи») за таким шляхом:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\SystemRestore],
```

або створити звичайний текстовий файл з таким вмістом:

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\SystemRestore]
"DisableSR"=dword:00000001
```

після чого змінити його розширення на .reg та запустити. Значення dword:00000000 – увімкне «Захист системи», а dword:00000001 – вимикає. Проте при імпорту у реєстр такого файлу через пакетний файл, який запускається через застосунок OSP, ми стикнулись з проблемою відсутності 64-бітного інтерпретатора командного рядка, який використовується в окремих командах C#. Вирішення проблеми може бути наступним: або «підстановка» заміною в системних каталогах 64-розрядного 32-розрядним інтерпретатором, копіюванням його з %Systemroot%\System32 у %Systemroot%\SysWoW64, або здійснення замість експорту у реєстр reg-файлу, модифікації реєстру командою REG (що і було зроблено):

```
reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows NT\SystemRestore" /v DisableSR /t Reg_Dword /d 1 /f
```

Для зупинки та запуску служби «Volume Shadow Copy» можна скористатись командою NET, прописавши у відповідні пакетні файли її параметри:

```
net stop vss
```
(для зупинки),

`net start vss` (для старту).

Після вказаних попередніх підготовчих дій, можна здійснювати очистку у каталозі System Volume Information через відповідний пакетний файл, запущений через OSP.

SWAP-файл. Основне призначення файлів підкачки (swap-файлів, для Windows 10 файли у корені системного розділу pagefile.sys, swapfile.sys) – це вивільнення оперативної пам'яті від непотрібних на даний момент даних, з записом їх на системний носій. Таким чином вирішується проблема недостатньої кількості оперативної пам'яті для програм, які її потребують. Часто файл підкачки фігурує у системі як доповнення до оперативної пам'яті, і, навіть розглядається із нею як одне ціле. Проте у сучасних умовах здешевлення модулів оперативної пам'яті переважна більшість користувачів володіє достатніми і навіть великими об'ємами оперативної пам'яті, а, отже, потреба у файлі підкачки відпадає. Хоча операційна система за замовчування його створює. Вона автоматично регулює його розмір, і навіть коли він не використовується, його мінімальний розмір не є нульовим, і може сягати кількох гігабайт.

В такому випадку можна відключити взагалі файл підкачки через налаштування у графічному інтерфейсі системи, або через командний рядок. Зокрема:

- `wmic pagefile list /format:list` – отримання статусу файлу-підкачки;
- `wmic computersystem set AutomaticManagedPagefile=True` – автоматичне регулювання розміру фала-підкачки операційно системою;
- `wmic computersystem set AutomaticManagedPagefile=False` – ручне регулювання розміру фала-підкачки користувачем;
- `wmic computersystem set AutomaticManagedPagefile=False`
  `wmic pagefileset delete` – вимкнення з подальшим видаленням файлу підкачки
- `SystemPropertiesPerformance.exe` – виклик системного вікна налагодження параметрів файлу підкачки.

Що і було реалізовано через пакетні файли з можливістю їх запуску через OSP.

В процесі очистки користувачі можуть припускатись помилок, тим самим пошкоджуючи операційну систему та її файлову структуру. В таких випадках варто скористатись штатною консольною утилітою SFC. Наприклад ввести у командному рядку (що теж додано до OSP):

```
sfc /scannow
```

Це дозволяє перевірити всі захищені системні файли і замінити пошкоджені файли їх кешованої копією, розташованою в стисненій папці за шляхом %WinDir%\System32\dllcache [33].

UWP-застосунки. Windows 10 поставляється з набором Modern, або UWP-застосунків (раніше вони називалися Metro Apps або APPX). Це Калькулятор, Календар, Пошта, Кортана, Карти, Новини, OneNote, Groove Music Камера і та інші. UWP програми Windows 10 автоматично в профіль користувача встановлюються при першому вході до системи. Більшість цих програм не потрібні користувачам, тому їх зазвичай бажають видалити. Розглянемо, як правильно видалити вбудовані UWP/APPX програми у Windows 10, що дозволить зберегти додаткове місце на системному розділі та вилучить непотрібні елементи у стартовому меню.

Є UWP-застосунки, які не можна видалити за допомогою панелі керування або налаштувань у Windows 10. Для видалення цих програм потрібно використовувати PowerShell або сторонні утиліти.

Для того, щоб видалити такі застосунки через Powershell спочатку необхідно отримати повний список встановлених таких застосунків, адже стандартні засоби операційної системи із панелі керування не дозволяють усіх їх показати, і, зрозуміло видалити. Такий список нам необхідний для визначення точного імені застосунку, яке надалі будемо використовувати у скриптах Powershell для видалення цих застосунків. Команда Get-AppxPackage | Select Name, PackageFullName повертає список встановлених застосунків (рис. 2.16).

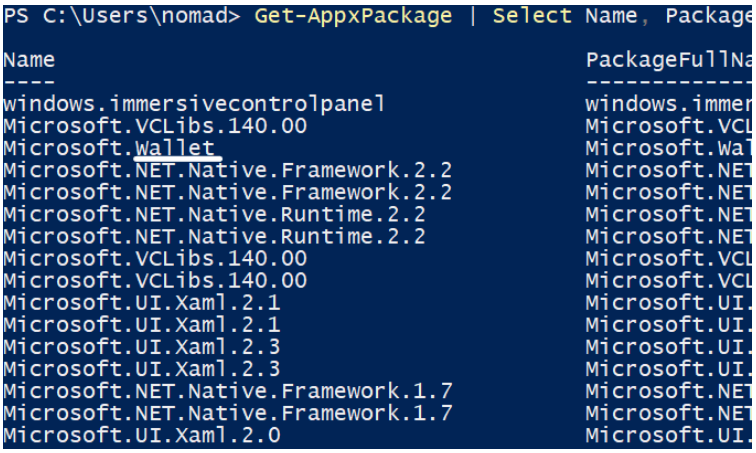

Рисунок 2.16 - Фрагмент списку встановлених UWP-застосунків, повернутий командою: *Get-AppxPackage | Select Name, PackageFullName*

Причому важливим є початок кожного із рядків, зокрема, спочатку йде назва виробника (в нашому випадку «Microsoft»), а після крапки йде точне ім’я програми, наприклад Wallet. (підкреслено на рис. 2.16).

Надалі видаляти такі застосунки можна або для конкретного користувача (що не зекономить місце на накопичувачі), або для усіх користувачів, використовуючи команду:

```
Get-AppxPackage -allusers *xxxxx* | Remove-AppxPackage",
```

де xxxxx – точне ім’я програми.

Проте запуск такого типу команд із застосунка C# в якості окремого процесу неможливий, доки не надано дозвіл на запуск скриптів Powershell з командного рядка, а отже і пакетного файлу, які ми використовуємо. За замовчуванням такий запуск з міркувань безпеки в операційній системі заборонено, хоча сама команда в складі скрипту дозволена на виконання з середовища Powershell. Повністю дозволяти запуск скриптів Powershell з командного рядка недоцільно, оскільки це підвищує рівень уразливості операційної системи, проте можна надати такий дозвіл окремо для кожної команди в межах окремого екземпляру середовища її виконання, адже такий екземпляр після виконання буде закритий, а, отже, і дозвіл буде відмінений. Для цього в пакетному файлі скористаємось:

```
-executionpolicy bypass -command
```

перед командою, яка виконуватиметься, і, відповідно типовий рядок пакетного файлу видалення UWP-застосунку набуде вигляду:

```
powershell -executionpolicy bypass -command "Get-AppxPackage -allusers *xxxxxx* | Remove-AppxPackage"
```

Проте дана команда фізично повністю не видаляє застосунки з розділу, а лише відміняє їх реєстрацію. Для їх видалення необхідно з каталогу %Systemdrive%\Program files\WindowsApps\ видалити папку(и) відповідного застосунку, що зручно також зробити за допомогою середовища Powershell через командний рядок, попередньо перебравши на себе права власності на такі об'єкти і встановивши відповідні права доступу. Це можна зробити за допомогою циклу в пакетному файлі, наприклад:

```
for /d %a in ("*xbox*") do (takeown /f "%a" /r /d y && icacls "%a" /grant administrators:F /t && rd /s /q "%a")
```

де takeown – перебирає на себе права (робить користувача власником) кожної папки, яка містить у назві «xbox», icacls надає користувачеві повний доступ до цієї папки з подальшим видаленням її вмісту (rd). До переліку застосунків на видалення буде включено такі, які найменш важливі для користувача, та які пересічні користувачі найчастіше видаляють: Bing Weather, Поради, Windows Photo, Groove Музика, 3D Builder, Maps Карти, Будильники і годинники, Skype, Телефон, Microsoft Solitaire Collection, Камера, Кіно і ТБ, Спорт, Новини, Звертання за допомогою, Paint 3D, Xbox, Xbox One SmartGlass. Видалення кожного із цих компонентів реалізуємо у вигляді окремих пакетних файлів, кожен з яких можна викликати через вкладку UWP із додатка OSP (рис. 2.17).

Додаткові інструменти та засоби. В процесі очистки часто виникають ситуації, коли не завжди можна повністю автоматизувати такий процес, або іноді необхідно зробити попередні маніпуляції, які вимагають додаткових знань, дій або засобів. У вкладці «Додаткове» (рис. 2.18) зібрано два такі засоби, для полегшення роботи користувача з операційною системою, системним розділом та самим застосунком.

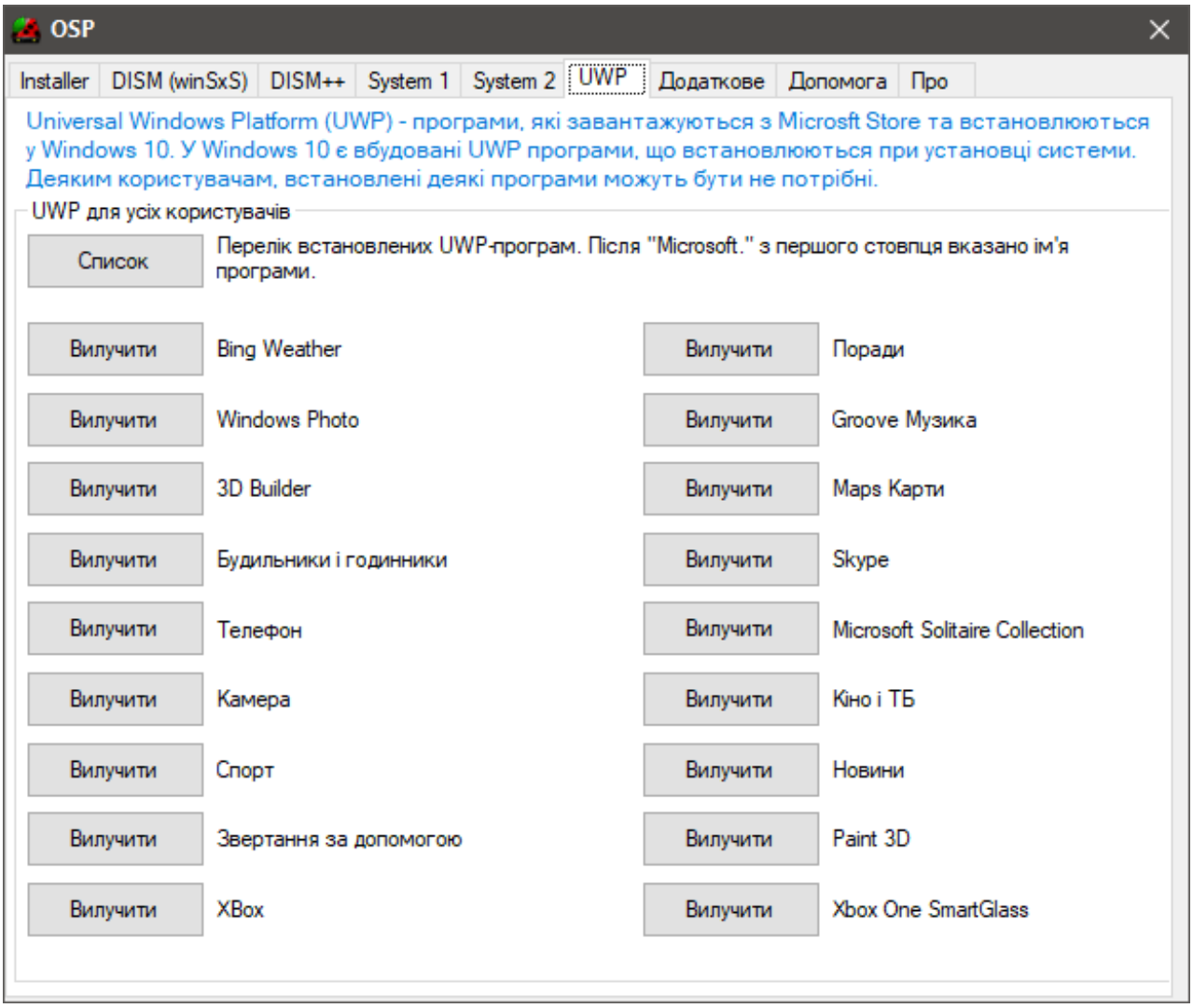

Рисунок 2.17 - Вкладка «UWP» застосунку OSP.

В розширене контекстне меню (викликається на файловому об’єкті з утримуванням клавіші Shift) операційної системи для роботи з файлами та папками можна через реєстр ввести додаткову команду, яка дозволяє швидко привласнити користувачеві будь-яку папку чи файл разом із повними правами доступу до неї (Get Owner) (рис. 2.19).

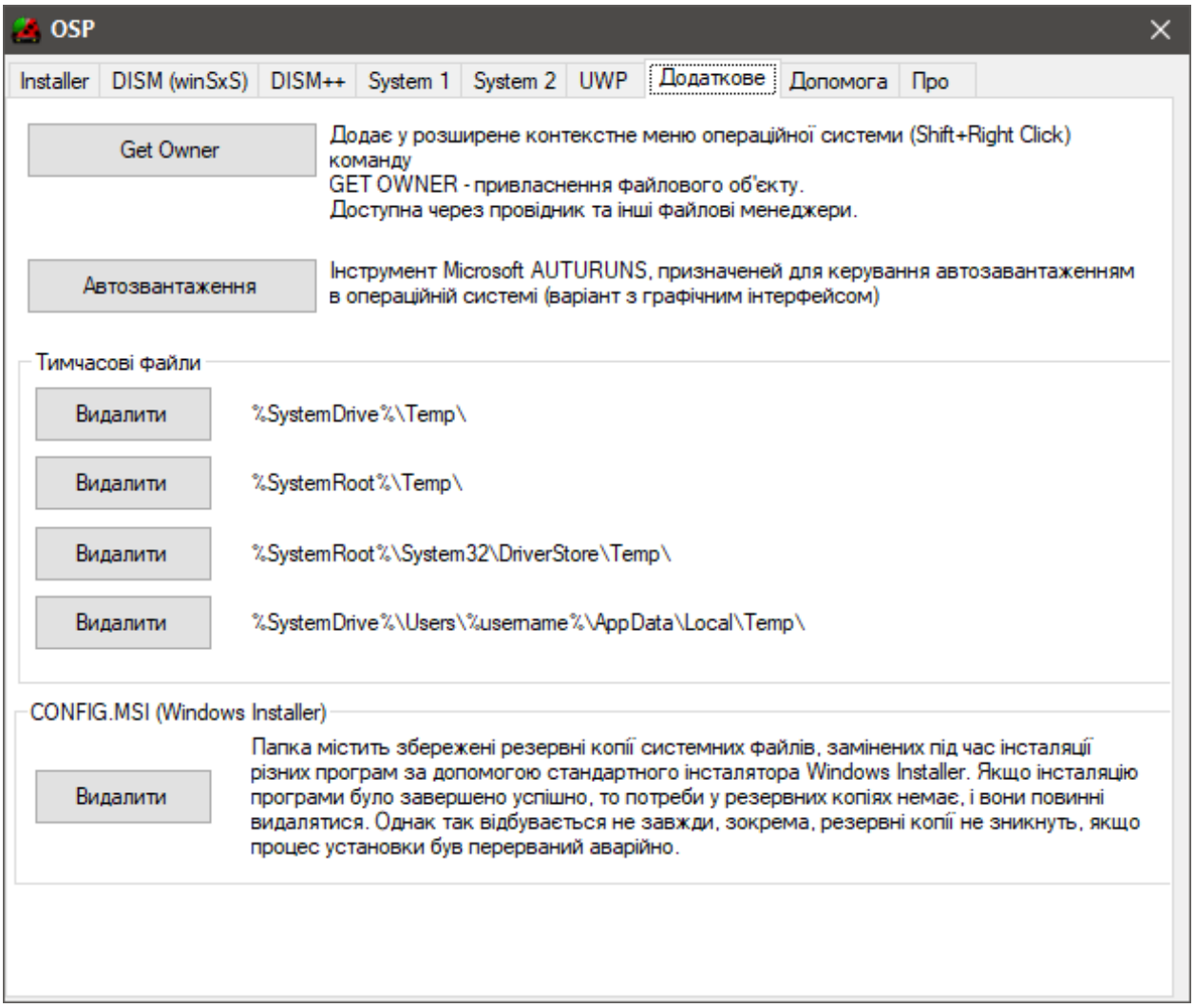

Рисунок 2.18 - Вкладка «Додаткове» застосунку OSP.

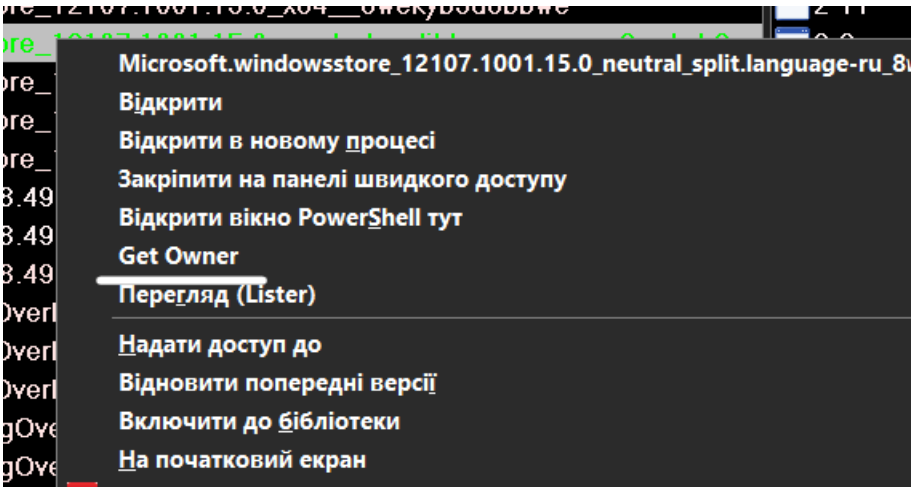

Рисунок 2.19 - Введена команда Get Owner в розширене контекстне меню

Для цього створимо пакетний файл, що вводить зміни до відповідних ключів реєстру:

```
rem Додає назву команди для файлового варіанту використання
reg add "HKCR\*\shell\runas" /d "Get Owner" /f
reg add "HKCR\*\shell\runas" /v Extended /f
reg add "HKCR\*\shell\runas" /v NoWorkingDirectory /f
rem Додає виконувані команди для файлового варіанту використання
reg add "HKCR\*\shell\runas\command" /d "cmd.exe /c takeown /f \"%1\" && icacls \"%1\" /grant
administrators:F" /f
reg add "HKCR\*\shell\runas\command" /v IsolatedCommand /d "cmd.exe /c takeown /f \"%1\" && icacls
\"%1\" /grant administrators:F" /f
rem Додає назву команди для варіанту використання каталогу
reg add "HKCR\Directory\shell\runas" /d "Get Owner" /f
reg add "HKCR\Directory\shell\runas" /v Extended /f
reg add "HKCR\Directory\shell\runas" /v NoWorkingDirectory /f
rem Додає виконувані команди для варіанту використання каталогу
reg add "HKCR\Directory\shell\runas\command" /d "cmd.exe /c takeown /f \"%1\" /r /d y && icacls \"
%1\" /grant administrators:F /t" /f
reg add "HKCR\Directory\shell\runas\command" /v IsolatedCommand /d "cmd.exe /c takeown /f \"%1\"
/r /d y && icacls \"%1\" /grant administrators:F /t" /f
reg add
"HKCR\AllFilesystemObjects\shellex\ContextMenuHandlers\{C2FBB630-2971-11D1-A18C-00C04FD75D13}" /f
```

Для оптимізації системи варто також бачити та керувати списком програм, які запускаються при завантаженні операційної системи. Для цього введемо в OSP виклик додаткової утиліти Autoruns (кнопка «Автозавантаження») [34].

В операційній системі є цілий список шляхів до каталогів із тимчасовими файлами, які теж періодично необхідно очищати. До них можна віднести:

%SystemDrive%\Temp\,

%SystemRoot%\Temp\,

%SystemRoot%\System32\DriverStore\Temp\,

%SystemDrive%\Users\%username%\AppData\Local\Temp\.

Тому до вкладки «Додаткове» було додано можливість очистки таких каталогів через запуск відповідного пакетного файлу. Для цього було необхідно передбачити видалення вмісту каталогу без видалення самого каталогу і сторонніх повідомлень про помилки:

```
SET THEDIR=%systemroot%\temp
Echo Deleting all files from %THEDIR%
DEL "%THEDIR%\*" /F /Q /A
Echo Deleting all folders from %THEDIR%
FOR /F "eol=| delims=" %%I in ('dir "%THEDIR%\*" /AD /B 2^>nul') do rd /Q /S "%THEDIR%\%%I"
EXIT
```

Аналогічно пакетний файл буде виглядати і для інших вказаних каталогів тимчасових файлів.

Папка CONFIG.MSI. На системному розділі Windows іноді з'являється прихована папка CONFIG.MSI, і при тривалій експлуатації системи її розмір може сягати кількох сотень мегабайт. Папка містить збережені резервні копії системних файлів, замінених під час інсталяції різних програм за допомогою стандартного інсталятора Windows Installer. Процес встановлення програмного забезпечення далеко не завжди протікає без ускладнень. Іноді трапляються збої або користувач перериває установку. У цьому випадку інсталятор здійснює відкат і повертає систему в початковий стані. Перш ніж приступити до виконання цієї операції, Windows Installer зберігає всю інформацію про замінені папки і файли, гілки і ключі реєстру Windows, і генерує скрипт для відкату установки. Ця інформація зберігається в спеціальному прихованому каталозі, який створюється в корені системного диска під час встановлення. Якщо інсталяцію програми було завершено успішно, то потреби у резервних копіях немає, і вони повинні видалятися. Однак так відбувається не завжди, зокрема, резервні копії не зникнуть, якщо процес установки був перерваний аварійно. Тому в папці %SystemDrive%\CONFIG.MSI з часом накопичуються непотрібні файли.

В OSP було введено у вкладку «Додаткове» ще одну панель призначену для знищення цього каталогу.

Останні вкладки «Допомога» і «Про» містять коротку інформацію про інші вкладки програми та про розробників відповідно.

В результаті розробки було отримано програмний продукт, який має файлову структуру зображену на рис. 2.20.

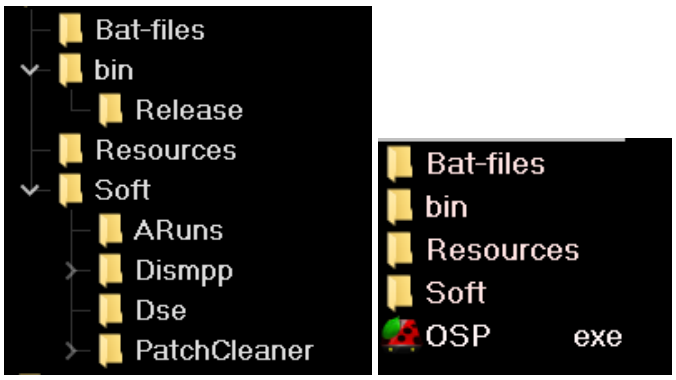

Рисунок 2.20 - Дерево каталогів OSP.

Bat-files – містить більше 60 пакетних файлів та інших скриптів, що запускаються через інтегроване середовище.

Bin\Release – містить скомпільований файл інтегрованого середовища-оболонки на C#.NET.

Resources – містить деякі нестандартні графічні об’єкти інтерфейсу інтегрованого середовища.

Soft – містить 4 пакети сторонніх утиліт, що викликаються з основної програми.

І нарешті у корені папки готового проекту розміщено стартовий виконуваний файл (OSP.exe) скомпільований із кореневого пакетного файлу, призначеного для запуску із будь-якого місця на диску, враховуючи файлову структуру програмного продукту.

2.6. Організація тестування та налагодження програмного засобу «OSP»

Процес тестування OSP передбачав наступні кроки:

1. Перевірка на сумісність з різними збірками та версіями операційної системи.
2. Перевірка на коректність роботи кожного пакетного файлу та інших зовнішніх модулів, що можуть викликатися з інтегрованого графічного середовища користувача.
3. Перевірка коректної роботи та відображення елементів графічного інтерфейсу інтегрованого середовища для різних варіантів роздільної здатності екрану та масштабування шрифтів (для ноутбуків).
4. Перевірка коректної роботи зовнішніх модулів саме під час їх виклику із інтегрованого середовища.
5. Перевірка операційної системи на працездатність та збереження функціоналу після роботи OSP.
6. Перевірка ефективності роботи (економія місця, зручність та швидкість взаємодії з користувачем).

Розроблений програмний засіб ефективно та стабільно працює в 32- та 64-розрядних операційних системах Windows 10 редакцій Home, Pro та Enterprise збірок від 1507 до 21H2 з системним розділом, що використовує файлову систему NTFS. Базовий функціонал OSP зберігається також в 32- та 64-розрядних операційних системах Windows Vista, 7, 8, 8.1 та в ОС Windows 11. Частково функціонал збережений при роботі в операційних системах Windows XP SP3. В цільовій операційній системі Windows 10 пакетні файли та інші скрипти викликались та спрацьовували коректно, що перевірялось явним чином за допомогою файлових менеджерів та системних засобів графічного інтерфейсу операційної системи та порівнянням з результатами їх роботи в ручному режимі. Робота графічного інтегрованого середовища була перевірена для роздільної здатності екрану від 800x600 для 16 та 32-бітної схеми передачі кольорів і масштабованості системних шрифтів до 125%. Після виклику кожного конкретного зовнішнього об’єкту перевірялась функціональність операційної системи, яка пов’язана із таким об’єктом на предмет її збереження та коректності роботи. Проводився аналіз стабільної та коректної роботи операційної системи в цілому в певний проміжок часу з виконанням типових задач після виконання серії скриптів та пакетних файлів з OSP. Ефективність очистки та оптимізації системного розділу оцінювалась за об’ємом зекономленого простору в момент після роботи OSP, та в динаміці його зміни в процесі типової експлуатації операційної системи.

2.7. Рекомендації по використанню та впровадженню програмного засобу «OSP».

Враховуючи досвід тестування та задачі, які ставились в процесі розробки OSP, можна сформулювати ряд рекомендацій для повнофункціонального використання даного програмного засобу.

1. Наявність апаратних засобів, що відповідають мінімальним системним вимогам, що ставляться до операційної системи Windows 10.[35]

2. Встановлена одна із операційних систем: Windows 10 x86/x64 Home/Pro/Enterprise, збірок від 1507 до 21H2.
3. Файлова система системного розділу NTFS версії не нижче 3.1 (v5.1).
4. Робота з-під облікового запису, які має локальні права адміністратора, або, при роботі у домені, з правами адміністратора домену.
5. Запуск OSP з правами адміністратора системи.
6. .NET Framework версії не нижче 3.5 SP1.
7. Powershell версії не нижче 2.0 SP1.

Даний програмний продукт може бути використаний як звичайними користувачами, які мають права адміністратора на локальних системах, так і адміністраторами систем, мережевими адміністраторами в якості основного інструменту для очищення робочих станцій для оптимізації їх роботи на предмет ефективного використання простору системного розділу в процесі експлуатації операційних систем Windows 10, а також частково Windows Vista, 7, 8, 8.1 та в ОС Windows 11.

ВИСНОВКИ

В роботі будо досліджено основні механізми та методи вивільнення простору системного розділу в процесі експлуатації операційної системи Windows 10 (x86/x64). Було розглянута структура розміщення системних файлів та папок операційної системи та їх призначення і роль у функціонуванні операційної системи. Визначено методи та засоби керування такими папками та їх розміром. Для цього було розглянуто як системний інструментарій, призначений для вирішення поставлених задач, доступний чи недоступний для типового користувача, так і безкоштовні засоби сторонніх розробників та можливість написання власних сценаріїв та пакетних файлів для очистки (прибирання) системного розділу та змін налаштувань операційної системи.

В роботі для розробки програмного засобу було використано можливості інтерпретатора командного рядка, розширений засіб роботи з інтерфейсом командного рядка PowerShell, засоби виклику системних бібліотек та інструменти для роботи з реєстром операційної системи. В якості середовища розробки кінцевого програмного продукту було використано Microsoft Visual C# 2010 Express Edition.

Як результат був розроблений програмний засіб OSP (Optimization of System Partition) здатен здійснювати поетапну, або вибіркову (за бажанням користувача) очистку системного розділу та змінювати окремі налаштування операційної системи Windows 10 вивільняючи місце на системному розділі, тим самим роблячи систему більш стабільною та ефективнішою у роботі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Булатецький В. В., Булатецька Л. В., Пруц Г. С. Методи та засоби вивільнення простору системного розділу ОС Microsoft Windows 10. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*, 2018. № 32. С. 85–89.
2. Windows and GPT FAQ. URL: https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/windows-and-gpt-faq (дата звернення: 13.11.2021)
3. What partitions does windows automatically create during instalation. URL: https://www.tenforums.com/installation-upgrade/34336-what-partitions-does-windows-automatically-create-during-instalation.html (дата звернення: 13.11.2021)
4. 4 Ways to Safely Delete Unused MSI and MSP Files from Windows Installer Folder. URL: https://www.raymond.cc/blog/safely-delete-unused-msi-and-mst-files-from-windows-installer-folder/ (дата звернення: 13.11.2021)
5. Как очистить папку FileRepository в DriverStore. URL: https://remontka.pro/driverstore-filerepository-folder-windows/ (дата звернення: 13.11.2021)
6. Как удалить папку Driverstore в Windows. URL: https://nastroyvse.ru/opersys/win/papka-driverstore-v-windows.html (дата звернення: 13.11.2021)
7. Очистка папки WinSxS в Windows 10, 8 и Windows 7. URL: https://remontka.pro/winsxs-windows/ (дата звернення: 13.11.2021)
8. How to reclaim space reducing size of WinSxS folder on Windows 10. URL: https://www.windowscentral.com/how-reclaim-space-reducing-size-winsxs-folder-windows-10 (дата звернення: 13.11.2021)
9. What is PowerShell? URL: https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7.1 (дата звернення: 13.11.2021)

10. Инструменты командной строки и автоматизации. URL: http://microsin.net/adminstuff/windows/command-line-and-automation-tools.html (дата звернення: 13.11.2021)
11. How to Create a GUI for PowerShell Scripts? URL: https://theitbros.com/powershell-gui-for-scripts (дата звернення: 13.11.2021)
12. Реестр Windows 10. URL: https://windowsprofi.ru/win10/reestr-windows-10.html (дата звернення: 13.11.2021)
13. Windows registry information for advanced users. URL: https://docs.microsoft.com/en-us/troubleshoot/windows-server/performance/windows-registry-advanced-users (дата звернення: 13.11.2021)
14. Reg commands. URL: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/reg (дата звернення: 13.11.2021)
15. Rundll32. URL: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/rundll32 (дата звернення: 13.11.2021)
16. Deployment Image Servicing and Management (DISM) Best Practices. URL: https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/deployment-image-servicing-and-management--dism--best-practices?view=windows-11 (дата звернення: 13.11.2021)
17. WiseCleaner Software - Stabilize, Secure and Speed Up Your Windows PC. URL: https://www.wisecleaner.com/ (дата звернення: 13.11.2021)
18. Dism++, probably the most powerful utility. URL: https://www.chuyu.me/uk/index.html (дата звернення: 28.11.2021)
19. Булатецький В. В., Булатецька Л. В. Технології проміжного коду в корпоративних інформаційних системах : Текст лекцій нормативної навчальної дисципліни “Платформи корпоративних інформаційних систем”. Луцьк: СНУ імені Лесі Українки, 2018. 48 с. URL: http://evnuir.vnu.edu.ua/handle/123456789/17724

20. Можно ли удалить папку Windows\Installer и файлы из неё. URL: https://remontka.pro/windows-installer-folder-delete/ (дата звернення: 13.11.2021)
21. Как очистить папку Installer в Windows. URL: https://windowstips.ru/kak-ochistit-papku-installer-v-windows (дата звернення: 13.11.2021)
22. Очистка места на RDS ферме (Installer, ServiceProfiles, WinSxS). URL: http://pyatilistnik.org/cleaning-rds-farm-space/ (дата звернення: 13.11.2021)
23. Downloaded installations что за папка. URL: https://kztarif.ru/kompjutery/downloaded-installations-chto-za-papka (дата звернення: 13.11.2021)
24. Очистка папки WinSxS в Windows 10, 8 и Windows 7. URL: https://remontka.pro/winsxs-windows/ (дата звернення: 13.11.2021)
25. DISM - Deployment Image Servicing and Management. URL: https://sites.google.com/site/raminaliyevit/veb-kasty/mswindows7/dism (дата звернення: 13.11.2021)
26. Как грамотно уменьшить размер папки WinSxS в Windows 10, 8.1 и 8. URL: https://www.outsidethebox.ms/15272/ (дата звернення: 13.11.2021)
27. Manage the Component Store. URL: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-8.1-and-8/dn251569(v=win.10)?redirectedfrom=MSDN (дата звернення: 13.11.2021)
28. Compact OS: как сжать системные файлы Windows 10 и сэкономить несколько гигабайт. URL: https://www.outsidethebox.ms/17965/ (дата звернення: 13.11.2021)
29. How to disable and re-enable hibernation on a computer that is running Windows. URL: https://docs.microsoft.com/en-US/troubleshoot/windows-client/deployment/disable-and-re-enable-hibernation (дата звернення: 13.11.2021)
30. Как очистить папку FileRepository в DriverStore. URL: https://remontka.pro/driverstore-filerepository-folder-windows/ (дата звернення: 13.11.2021)

31. DriverStoreExplorer URL:https://archive.codeplex.com/?p=driverstoreexplorer (дата звернення: 13.11.2021)
32. Backup and Restore in Windows. URL: https://support.microsoft.com/en-us/windows/backup-and-restore-in-windows-352091d2-bb9d-3ea3-ed18-52ef2b88cbef (дата звернення: 13.11.2021)
33. Sfc. URL: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/sfc (дата звернення: 13.11.2021)
34. Autoruns for Windows v14.06. URL: https://docs.microsoft.com/uk-ua/sysinternals/downloads/autoruns (дата звернення: 13.11.2021)
35. Windows 10 system requirements. URL: https://support.microsoft.com/en-us/windows/windows-10-system-requirements-6d4e9a79-66bf-7950-467c-795cf0386715 (дата звернення: 13.11.2021)

Додаток А

Технічне завдання на розробку програмного засобу «OSP»

1. ЗАГАЛЬНІ ВІДОМОСТІ

1.1. Назва системи

Повне найменування системи: система для оптимізації системного розділу Windows 10 (x86/x64) «OSP» (Optimization of System Partition).

1.2. Найменування організацій-учасників робіт

Виконавець: здобувач вищої освіти, навчально-наукового інституту неперервної освіти Волинського національного університету імені Лесі Українки – Булатецький Віталій Вікторович.

1.3. Передумови для проведення робіт

Останні версії операційних систем сімейства Windows вимагають все більше місця для збереження своєї файлової структури на жорсткому диску в системному розділі. Вони поповнюються новими компонентами, періодично отримують різноманітні оновлення та виправлення, зберігають в цьому розділі файли користувачів, заповнюють системний розділ різноманітними резервними копіями, тощо. В результаті розділ накопичувача, де міститься сама операційна система, з часом заповнюється інформацією, яка не завжди є критично важливою, проте викликає дефіцит вільного дискового простору, що впливає як на швидкодію операційної системи, так і на функціонал та навіть на можливість подальшого безперебійного отримання оновлень та виправлень. Основний шлях уникнути такої ситуації – вчасно здійснювати очистку системи від неважливої інформації. Хоча сама операційна система частково повинна позбуватись подібних даних в автоматичному режимі, проте, при неналежному налаштуванні та із-за можливих періодичних збоїв це не завжди відбувається, або відбувається не вчасно, чи некоректно. І хоча найпростішим способом є збільшення розміру системного розділу, але це не завжди є можливим, особливо для невеликих твердотільних накопичувачів, або при браку вільного місця на інших розділах. А

якщо і є можливим, то це призводить до незручності роботи з таким розділом, що пов’язано з тривалим його резервним копіюванням, відновленням та зміною його параметрів, адже він загромаджується великою кількістю несуттєвих файлів та каталогів.

1.4. Порядок оформлення передачі замовнику результатів робіт по розробці програмного забезпечення

Програмне забезпечення оформлене у вигляді інсталяційного пакету, який може бути записаний на оптичний, або інший носій, або розміщений на загальнодоступній платформі у мережі. Комплектується короткою вмонтованою інструкцією для користувача.

2. ПРИЗНАЧЕННЯ СИСТЕМИ

Програмна система призначена для оптимізації системного розділу Windows 10 (x86/x64) редакцій Home, Pro та Enterprise збірок від 1507 до 21H2 з системним розділом, що використовує файлову систему NTFS.

3. ВИМОГИ ДО СИСТЕМИ

3.1. Вимоги до функціональних характеристик

Програмний продукт повинен володіти наступними функціональними характеристиками:

- містити єдине інтегроване середовище з графічним інтерфейсом користувача для легкого доступу до засобів оптимізації звичайному користувачеві;
- враховувати особливості роботи програмного забезпечення в операційні системі (середовище виконання, організації процесів тощо);
- давати змогу взаємодіяти з системними засобами операційної системи доступними для звичайного користувача та прихованими від нього;

- допускати завантаження, встановлення та роботу сторонніх засобів оптимізації;
- містити в собі реалізацію додаткових інструментів, які полегшують здійснення процесу оптимізації;
- давати можливість вибіркового застосування методів оптимізації на вимогу користувача;
- містити у собі лаконічні та водночас вичерпні коментарі, зрозумілі користувачеві під час оптимізації.

3.2. Технічні вимоги

Програмна розробка повинна являти собою інтегроване графічне середовище у вигляді віконної форми, яка містить інструменти керування типу вкладок, кожна із яких представляє ту або іншу компоненту розробки, що стосується окремого об’єкту, або групи об’єктів, які підлягають оптимізації. Кожна із вкладок повинна містити елементи керування (і прив’язані до них коментарі) через які здійснюється запуск того або іншого інструмента для здійснення оптимізації. Кожен із інструментів запускається окремим процесом та може працювати при необхідності паралельно, незалежно один від одного. Робота програми повинна здійснюватися під обліковим записом користувача, що входить в групу адміністраторів від імені адміністратора.

3.3. Вимоги до експлуатаційних характеристик

Програмний засіб повинен здійснювати очистку (прибирання) ряду системних папок операційної системи Windows 10 такі як: Installer, WinSxS, тимчасові папки, DriverStore, System Volume Information, керувати стисненням системних файлів та папок, режимом гібернації, віртуальною пам’яттю, службами System Restore та Volume Shadow Copy та ін.

3.4. Вимоги до технічного забезпечення

Програмна розробка розрахована на функціонування при наступному наборі технічних засобів:

Мінімальна апаратна конфігурація комп’ютера:

- Процесор: 1 ГГц з підтримкою PAE, NX і SSE2;
- RAM: 1 Гбайт (32 бітна версія Windows 10) або 2 Гбайт (64 бітна версія Windows 10);
- HDD: 16 Гбайт (32 бітна версія Windows 10) або 20 Гбайт (64 бітна версія Windows 10);
- Відеокарта: підтримка Microsoft DirectX 9 з драйвером WDDM.

4. ОСНОВНІ РЕЖИМИ ФУНКЦІОНУВАННЯ ПРОГРАМНОГО ЗАСОБУ

Функціонування реалізованої програмної системи з точки зору користувача можна описати наступним чином:

1. Користувач встановлює (інсталює програмний продукт).
2. Користувач запускає програмний продукт в режимі (від імені) адміністратора.
3. Користувач вибирає необхідний інструмент оптимізації з вкладень основної форма графічного інтерфейсу.
4. Користувач обирає необхідні налаштування інструменту оптимізації (якщо такі передбачені).
5. Якщо необхідно, користувач переходить до наступного вкладення, яке відповідає за інший інструмент, та обирає потрібні налаштування.
6. Кожен із інструментів працює незалежно і може використовуватись паралельно з іншим.

5. ПОРЯДОК ПРИЙОМУ СИСТЕМИ В ЕКПЛУАТАЦІЮ

Робота приймається у випадку відсутності помилок у роботі та її ефективності. Система перевіряється на відповідність усім вимогам. При знаходженні недоліків у роботі програма доопрацьовується.

Додаток Б

Інструкція користувача програмного засобу «OSP»

УВАГА! Максимально ефективно програма працює від імені локального адміністратора!

Призначення основних вкладок:

Installer

Папка Installer - це системна папка, де зберігаються необхідні для видалення або оновлення програм, їх інсталятори, інсталятори різних програмних компонентів, патчі, файли не сталих системних оновлень та інші дані, які необхідні системі і встановленому програмному забезпеченню для певних процесів. Ця папка розміщується за шляхом C:\Windows\Installer, але потрапити в неї за допомогою системного провідника просто так не можна. Папка Installer прихована і захищена.

DISM (winSxS)

В папці C:\Windows\WinSxS зберігаються резервні копії системних файлів операційної системи до оновлень і не тільки. Кожного разу, після отримання і встановлення оновлення Windows, в цю папку зберігається інформація про змінювані файли та самі ці файли з тим, щоб була можливість видалити оновлення і відкотити зроблені зміни.

Для безпечної роботи з цією папкою передбачено штатну систему обслуговування образів розгортання і управління ними - DISM (Deployment Image Servicing and Management).

DSIM++

Безкоштовний сторонній програмний продукт, в якому, поряд з іншим функціоналом реалізовано систему очистки операційної системи від сміття. Дублює частину функцій штатної утиліти DISM, проте використовує власні більш ефективні алгоритми для очистки сховища WinSxS і не тільки.

System 1, System 2

Набір методів очистки, які ґрунтуються на особливостях функціонування операційної системи та її файлової структури. Зокрема використовується стиснення, робота зі сховищем драйверів пристроїв, маніпулювання режимом гібернації, віртуальною пам'яттю та ін.

UWP

Universal Windows Platform (UWP) - програми, які завантажуються з Microsft Store та встановлюються у Windows 10. У Windows 10 є вбудовані UWP програми, що встановлюються при установці системи. Деяким користувачам, встановлені деякі програми можуть бути не потрібні.

Додаткове

Призначена для додавання в розширене контекстне меню ОС команди перехоплення прав власності Get Owner. Завантаження інструменту маніпулюванням автозавантаженням програм при запуску ОС. Очистка тимчасових файлів ОС.

Анотація

Булатецький В.В. – Дослідження та реалізація засобів оптимізації системного розділу операційної системи Windows 10 – Рукопис.

Магістерська робота за спеціальністю 122 – Комп’ютерні науки – Волинський національний університет імені Лесі Українки, Луцьк. – 2021р.

В роботі досліджено механізми вивільнення простору на системному розділі операційної системи Windows 10. Розглянуто методи очистки та оптимізації системного розділу та розроблено програмний засіб, у якому реалізовано автоматизовану систему застосування таких методів.

Ключові слова: Операційна система Windows, очистка, оптимізація, системний розділ, командний рядок, пакетний файл.

Abstract

Bulatetskyi V.V. - Research and implementation of tools for optimizing the system partition of the operating system Windows 10 - Manuscript.

Master's thesis in specialty 122 - Computer Science - Lesya Ukrainka Volyn National University, Lutsk. - 2021p.

The mechanisms of freeing space on the system partition of the Windows 10 operating system are investigated in the work. Methods of cleaning and optimization of the system partition are considered and the software in which the automated system of application of such methods is implemented is developed.

Keywords: Windows operating system, cleanup, optimization, system partition, command line, batch file.