

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
Центр заочної, дистанційної та вечірньої форм навчання  
Кафедра комп'ютерних наук

Кваліфікаційна робота магістра

**«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ КЕРУВАННЯ ПЛАТНИМИ  
ПІДПИСКАМИ ІНТЕРНЕТ-СЕРВІСІВ»**

Здобувач освіти гр. ІН.мз-13с

Артем МАРЧУК

Науковий керівник,  
доцент, к.т.н.

Віктор ОБОДЯК

В.о. завідувача кафедри,  
доцент, к.т.н.

Ігор ШЕЛЕХОВ

Суми 2023

Сумський державний університет

(назва вузу)

Факультет ЕЛІТ Кафедра Комп'ютерних наук

Спеціальність «122 – Комп'ютерні науки»

Затверджую \_\_\_\_\_

В.о. зав. кафедри Шелехов І.В.

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

## ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТУ

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Інформаційна технологія керування платними підписками інтернет-сервісів

затверджую наказом по університету від « \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

№ \_\_\_\_\_

2. Термін здачі студентом закінченого проєкту (роботи) \_\_\_\_\_

3. Вхідні дані до проєкту (роботи) \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Дослідження актуальності проблеми та аналіз існуючих інструментів. 2) Формування мети та постановки задачі, визначення етапів реалізації проєкту. 3) Вибір технологій та інструментів, що необхідні для розробки. 4) Проєктування структури інформаційної системи. 5) Розроблення інформаційної системи. 6) Тестування готового програмного продукту та аналіз результатів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання \_\_\_\_\_

Керівник

\_\_\_\_\_

(підпис)

Завдання прийняв до виконання

\_\_\_\_\_

(підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1.	Дослідження актуальності проблеми та аналіз існуючих інструментів	01.11–07.11	
2.	Формування мети та постановки задачі, визначення етапів реалізації проекту	08.11–18.11	
3.	Вибір технологій та інструментів, що необхідні для розробки	19.11–29.11	
4.	Проектування структури інформаційної системи	18.11–24.11	
5.	Розроблення інформаційної системи	25.11–02.12	
6.	Тестування готового програмного продукту та аналіз результатів	03.12-17.12	
7.	Оформлення пояснювальної записки до кваліфікаційної магістерської роботи	08.12-12.12	

Студент–дипломник

\_\_\_\_\_

(підпис)

Керівник проекту

\_\_\_\_\_

(підпис)

## РЕФЕРАТ

Записка: 74 стор., 52 рис., 1 додаток, 23 джерела.

**Об'єкт дослідження** – процес проєктування та створення інформаційної технології керування платними підписками інтернет-сервісів.

**Мета роботи** – розроблення інформаційної технології керування платними підписками інтернет-сервісів.

**Методи дослідження** – під час розроблення інформаційної системи використані: Windows 10, RDP Wrapper, мова програмування Python, Скриптовий язык програмування PowerShell.

**Результати** – було проведено аналіз предметної області застосування та описана актуальність створення інформаційної технології; був зроблений вибір середовища розробки; розроблена інформаційна технологія керування платними підписками інтернет-сервісів; після реалізації проєкту проведено тестування роботи інформаційної системи.

RDP, ОПТИМІЗАЦІЯ, WINDOWS, PYTHON, POWERSHELL,  
SCRIPT, CHROME, КОМАНДА, ПІДПИСКИ, БРАУЗЕР

# Зміст

ВСТУП .....	6
1 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	7
1.1 Дослідження актуальності проблеми .....	7
1.2 Аналіз існуючих інструментів .....	11
1.3 Постановка задачі.....	21
2 МЕТОДИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ КЕРУВАННЯ ПЛАТНИМИ ПІДПИСКАМИ ІНТЕРНЕТ-СЕРВІСІВ.....	22
2.1 Вибір середовища розробки.....	22
2.2 Проектування структури скрипту та вибір модулів .....	47
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	52
3.1 Розробка інформаційної технології керування платними підписками інтернет-сервісів.....	52
3.2 Тестування інформаційної технології.....	62
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	65
ДОДАТОК А .....	68

## ВСТУП

Для багатьох компаній потрібні різні інструменти для роботи, аналізу та розвитку. Інструменти бувають різні, це можуть бути веб-сервіси або прикладне програмне забезпечення, яке встановлюється на локальний комп'ютер. Всі ці інструменти коштують гроші, та у різних інструментах різні вимоги для користування.

Якщо цими інструментами користується одна людина в компанії, то це просто: зробили підписку або оплатили ліцензію і співробітник користується інструментами.

Якщо команда збільшується, збільшується не тільки команда, а також і географія використання. Зазвичай ліцензія дається на один локальний комп'ютер або одна підписка на користувача без можливості одночасного використання.

Для багатьох компаній це стає проблемою, так як ціна на такі інструменти може бути непомірною, і компанія не може розширюватись або втрачає гроші.

Таким чином, актуальною є проблема використання платних інструментів не одним співробітником, при цьому не порушувати правила використання та економити на цьому бюджет компанії.

Метою кваліфікаційної магістерської роботи є розробка інформаційної технології керування платними підписками інтернет-сервісів.

# 1 ІНФОРМАЦІЙНИЙ ОГЛЯД

## 1.1 Дослідження актуальності проблеми

Дослідження актуальності проблеми будемо розглядати на двох прикладах.

Перший - це велика ІТ компанія, де працює більше семисот осіб. Серед них більше трьохсот осіб працює віддалено у відділі по оптимізації пошукових систем. Для спрощеності дослідження взято один відділ віддаленого аутрічу та один інструмент, яким вони користуються. Оцінкою актуальності буде виступати економія бюджету, яку можна буде спрямувати на покращення відділу або на залучення додаткових осіб до компанії та створення нових робочих місць.

Другий приклад являє собою невеликий бізнес - звичайна фізична особа підприємець, у якої є три магазини з побутовою технікою. Буде розглянуто, чи можна в такому випадку щось оптимізувати та оптимізувати бюджет.

Отже розглянемо перший приклад.

Велика ІТ компанія, є відділ віддаленого аутрічу, всі співробітники працюють з різних куточків планети. Кожному співробітнику потрібно надати доступ до інструменту для аналізу сайтів, таких інструментів багато, одним з них є Ахрефс (рис. 1.1) [1].

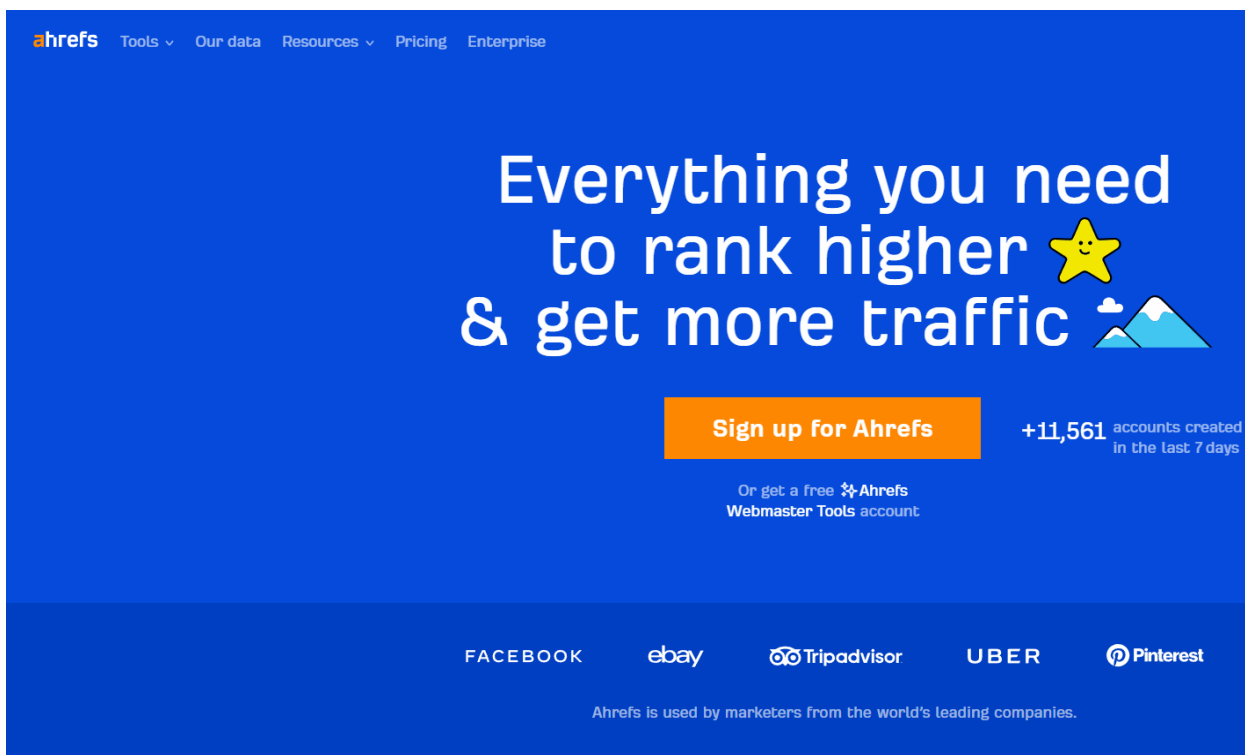



Рисунок 1.1 - Головна сторінка інструменту «Аhrefс» [1]

Аhrefс – це якісний інструмент для аналізу веб-ресурсів на предмет трафіку, ключових слів, позицій видачі в пошуковій системі. Також завдяки цьому інструменту можна побачити, як розвивається веб-ресурс кампанії, а також веб-ресурси конкурентів. Отже для ефективної роботи цей інструмент потрібно надати всім особам у відділі. Тому розглянемо умови використання та ціни на цей інструмент (рис. 1.2 та рис. 1.3) [1].



# Plans & pricing

We've been crawling the entire web 24/7 since 2010, indexing and structuring petabytes of information. Get access to these insights with an Ahrefs subscription and use them to improve your business.

[Pay monthly](#) / Pay annually, get 2 months free 

Monthly price	Lite	Standard	Advanced	Enterprise
Additional taxes may apply depending on your country.	Starts at <sup>ⓘ</sup>	Starts at <sup>ⓘ</sup>	Starts at <sup>ⓘ</sup>	Starts at <sup>ⓘ</sup>
	<b>\$99</b>	<b>\$199</b>	<b>\$399</b>	<b>\$999</b>
	<a href="#">Subscribe</a>	<a href="#">Subscribe</a>	<a href="#">Subscribe</a>	<a href="#">Talk to us</a>
	Essential data for small businesses and hobby projects.	Perfect for SEO professionals and in-house marketers.	Indispensable tools and more data allowance as your business scales.	Ideal for big agencies and enterprises.
Feature comparison	<ul style="list-style-type: none"> <li>✓ SEO Dashboard <sup>ⓘ</sup></li> <li>✓ Site Explorer <sup>ⓘ</sup></li> <li>✓ Keywords Explorer <sup>ⓘ</sup></li> <li>✓ Site Audit <sup>ⓘ</sup></li> <li>✓ Rank Tracker <sup>ⓘ</sup></li> <li>✓ Alerts <sup>ⓘ</sup></li> </ul>	<ul style="list-style-type: none"> <li>✓ All in <b>Lite</b></li> <li>+ 6 months of history <sup>ⓘ</sup></li> <li>+ Position history chart <sup>ⓘ</sup></li> <li>+ SERP updates <sup>ⓘ</sup></li> <li>+ Site Explorer <sup>ⓘ</sup></li> <li>Content gap <sup>ⓘ</sup></li> <li>Broken backlinks <sup>ⓘ</sup></li> <li>Broken links <sup>ⓘ</sup></li> </ul>	<ul style="list-style-type: none"> <li>✓ All in <b>Standard</b></li> <li>+ 2 years of history <sup>ⓘ</sup></li> <li>+ Site Explorer <sup>ⓘ</sup></li> <li>Site Structure <sup>ⓘ</sup></li> <li>HTML source <sup>ⓘ</sup> <span>Soon</span></li> <li>+ Site Audit <sup>ⓘ</sup></li> <li>HTTP authentication <sup>ⓘ</sup></li> <li>+ Google Data Studio <sup>ⓘ</sup></li> </ul>	<ul style="list-style-type: none"> <li>✓ All in <b>Advanced</b></li> <li>+ Unlimited history <sup>ⓘ</sup></li> <li>+ Access management <sup>ⓘ</sup></li> <li>+ Pay by Invoice <sup>ⓘ</sup></li> <li>+ Directory listing <sup>ⓘ</sup></li> <li>+ Audit log <sup>ⓘ</sup></li> <li>+ API <sup>ⓘ</sup></li> </ul>

Рисунок 1.2 - Тарифні плани інструменти «Аhrefс» [1]

	Lite \$99 /mo	Standard \$199 /mo	Advanced \$399 /mo	Enterprise \$999 /mo
	<a href="#">Subscribe</a>	<a href="#">Subscribe</a>	<a href="#">Subscribe</a>	<a href="#">Talk to us</a>
What's included	Each plan comes with <b>1 Power user</b> <sup>ⓘ</sup> who can consume up to <b>500 credits</b> <sup>ⓘ</sup> per month. Credits are consumed when using all Ahrefs tools except Rank Tracker and Site Audit. <a href="#">Learn more</a>			
Projects	<ul style="list-style-type: none"> <li>Verified ownership <sup>ⓘ</sup>: Unlimited</li> <li>Unverified ownership <sup>ⓘ</sup>: 5</li> </ul>	<ul style="list-style-type: none"> <li>Unlimited</li> <li>20</li> </ul>	<ul style="list-style-type: none"> <li>Unlimited</li> <li>50</li> </ul>	<ul style="list-style-type: none"> <li>Unlimited</li> <li>100</li> </ul>
Data	<ul style="list-style-type: none"> <li>Max rows per report <sup>ⓘ</sup>: 2,500</li> <li>Export rows /mo <sup>ⓘ</sup>: 500,000</li> </ul>	<ul style="list-style-type: none"> <li>30,000</li> <li>1.5M</li> </ul>	<ul style="list-style-type: none"> <li>75,000</li> <li>4M</li> </ul>	<ul style="list-style-type: none"> <li>150,000</li> <li>10M</li> </ul>
Rank Tracker <sup>ⓘ</sup>	<ul style="list-style-type: none"> <li>Tracked keywords <sup>ⓘ</sup>: 750</li> <li>Update frequency <sup>ⓘ</sup>: Weekly</li> <li>Daily updates <sup>ⓘ</sup>: \$100 /mo</li> </ul>	<ul style="list-style-type: none"> <li>2,000</li> <li>Weekly</li> <li>\$150 /mo</li> </ul>	<ul style="list-style-type: none"> <li>5,000</li> <li>Weekly</li> <li>\$200 /mo</li> </ul>	<ul style="list-style-type: none"> <li>10,000</li> <li>Weekly</li> <li>\$250 /mo</li> </ul>
Site Audit <sup>ⓘ</sup>	<ul style="list-style-type: none"> <li>Crawl credits /mo <sup>ⓘ</sup>: 100,000</li> <li>Max pages per project <sup>ⓘ</sup>: 25,000</li> </ul>	<ul style="list-style-type: none"> <li>500,000</li> <li>50,000</li> </ul>	<ul style="list-style-type: none"> <li>1.5M</li> <li>250,000</li> </ul>	<ul style="list-style-type: none"> <li>5M</li> <li>5M</li> </ul>
Keywords Explorer <sup>ⓘ</sup>	<ul style="list-style-type: none"> <li>Keywords lists <sup>ⓘ</sup>: 5</li> </ul>	<ul style="list-style-type: none"> <li>50</li> </ul>	<ul style="list-style-type: none"> <li>100</li> </ul>	<ul style="list-style-type: none"> <li>250</li> </ul>

Рисунок 1.3 - Умови використання в тарифних планах «Аhrefс» [1]

Аналіз рисунків показує, що цей інструмент не дешевий, для великої ІТ компанії потрібно максимум лімітів та максимальний функціонал, а це дев'ятсот дев'яносто дев'ять доларів США за один обліковий запис. До цього облікового запису можливо доєднати ще двох осіб по п'ятдесят доларів США за особу. Сумарно обліковий запис для трьох осіб виходить тисячу дев'яносто дев'ять доларів США. Також для того, щоб забезпечити триста осіб цим інструментом, компанія витратить більше ста тисяч доларів США щомісячно. Це неймовірна сума навіть для великих компаній, тому оптимізація платних підписок для таких компаній дуже актуальна проблема, бо тільки при економії на одному інструменті компанія вже може створити п'ятдесят робочих місць з досить непоганою заробітною платою. При спробах використати інструменти, які були проаналізовані у пункті 1.2, обліковий запис блокувався, унеможливаючи роботу з ним.

Розглянемо другий приклад.


Є невеликий бізнес - три магазини побутової техніки. Для цих магазинів використовується програмне забезпечення для обліку товарного залишку, керування постачальниками, зберігання списків клієнтів, дисконтних карт, ведення звітності та інше. Оскільки магазинів три, потрібно три копії програми для касирів, а також для бухгалтерії та спеціаліста по закупівлі товару. Також потрібні три копії ПРРО (Програмний реєстратор розрахункових операцій).

Проблема не тільки в початковій ціні за ліцензії програмного продукту і ПРРО, а також в щомісячному обслуговуванні. Ціна за програмний продукт з USB ключем показана нижче (рис. 1.4) [2].

ГОЛОВНА ПРОДУКТИ ТА ЦІНИ ЗАХОДИ ПРИКЛАДИ ВПРОВАДЖЕНЬ ПАРТНЕРИ КОНТАКТИ

Головна > Продукти та ціни > Спеціальний ключ

### СПЕЦІАЛЬНИЙ КЛЮЧ



**Microinvest йде Вам назустріч - ціни знижені більш ніж на 25%**


Вказана ціна дійсна тільки на території України.


**Ціна: ~~18450~~ грн **12678** грн**


Введіть номер телефону


[ЧЕКАЮ ДЗВІНКА](#)


**НАШІ ПЕРЕВАГИ**


  
 Постійний розвиток

  
 Великий список обладнання

  
 Доступні ціни

  
 Зростаюча база відео-уроків і документації

  
 Програма лояльності

  
 Безкоштовне тестування

#### Спеціальний ключ ПЗ Microinvest

З цим ключем можна користуватися всіма модулями ПЗ Microinvest на одному комп'ютері



Рисунок 1.4 - Ціна ключа продукту з офіційного сайту [2]

Таким чином, щоб повноцінно користуватися програмним забезпеченням, потрібно придбати п'ять копій. Потрібно також щомісячно витрачати гроші на обслуговування.

Якщо оптимізувати використання програмного забезпечення, щоб однією копією могли користуватися одночасно всі кому це потрібно, то це зменшить одночасні витрати на придбання цього програмного забезпечення, а також щомісячні витрати на обслуговування. Отже і для невеликого бізнесу ця проблема актуальна.

Можна запропонувати використовувати інформаційну систему на базі віддаленого підключення до робочої станції або групування сеансів.

## 1.2 Аналіз існуючих інструментів

Інструменти на основі віддаленого підключення до комп'ютера:

1. Apple Remote Desktop (ARD ) - це додаток Macintosh, розроблений Apple Inc., вперше випущений 14 березня 2002, який замінив аналогічний продукт під назвою Apple Network Assistant. Apple Remote Desktop, призначений для адміністраторів комп'ютерів, що відповідають за велику кількість комп'ютерів, а також вчителів, яким необхідно допомагати окремим особам або проводити групові демонстрації. Цей додаток дозволяє користувачам віддалено керувати або контролювати інші комп'ютери через мережу [3,4].

2. HP ZCentral Remote Boost, раніше відомий як HP Remote Graphics Software або HP RGS, — це клієнт-серверне програмне забезпечення для віддаленого робочого столу, розроблене HP Inc. і вперше запущене в 2003 році. HP RGS забезпечує віддалений доступ до високопродуктивних робочих станцій (або віртуальних робочих станцій) з багатьох різних пристроїв, включаючи інші робочі станції, ПК, планшети Windows, MacBook. Це програмне забезпечення призначене для віддаленого доступу до інтенсивних графічних програм, редагування відео та складних 3D-моделей. Також підтримується співпраця або спільне використання екрана між кількома користувачами, віддалений доступ до USB та звуку. У 2020 році HP оновила та змінила бренд RGS як частину рішення HP ZCentral. ZCentral Remote Boost отримав нагороду Engineering Emmy Award у 2020 році за те, що телепродукція по всьому світу вийшла в ефір, незважаючи на те, що команди не мали доступу до своїх звичайних студій [7]. За допомогою HP RGS уся обробка, включаючи апаратно-прискорену графіку з найновішим рендерингом OpenGL або DirectX, виконується на робочій станції, а на клієнтський пристрій надсилаються лише стиснуті растрові зображення (екран). HP RGS вирізняється власними алгоритмами стиснення, які дозволяють передавати складні 3D-зображення та відео в реальному часі, з чим важко справляються традиційні протоколи віддаленого робочого столу. Це відкриває можливість використання віддалених настільних комп'ютерів

для галузей із інтенсивним використанням графіки, таких як САПР, розвідка нафти й газу, анімація, архітектурна інженерія та будівництво, для яких раніше були потрібні локальні робочі станції. Підвищення продуктивності в RGS 7.1 відкрило ще більш вимогливі варіанти використання, включаючи віддалене відображення 4К-дисплеїв, передачу 60 кадрів на секунду або сеанси віддаленого робочого столу програм 3D CAD із використанням максимальної пропускної здатності до 3 Мбіт/с. Існує два компоненти програмного забезпечення: відправник (для робочої станції або сервера) і приймач (для клієнтського пристрою). Програмне забезпечення підтримує OpenGL і Microsoft DirectX. Програмне забезпечення продається окремо для серверів, віртуальних машин і робочих станцій не HP. HP почала включати RGS у всі свої настільні робочі станції бренду Z. RGS можна завантажити з HP і безкоштовно запускати на робочих станціях HP Z і мобільних робочих станціях ZBook. Для іншого апаратного забезпечення потрібна ліцензія для відправника HP RGS. Приймач можна безкоштовно завантажити для клієнтів Windows, Linux і macOS [3,5].

3. Незалежна обчислювальна архітектура (ICA) власний протокол для системи серверів додатків, розроблений Citrix Systems. Протокол встановлює специфікацію для передачі даних між сервером і клієнтами, але не прив'язаний до якої-небудь однієї платформи. Citrix ICA є альтернативою протоколу віддаленого робочого столу Microsoft (RDP). Практичними продуктами, відповідними ICA, є продукти Citrix WinFrame, Citrix XenApp (раніше називалися MetaFrame / Presentation Server) і продукти Citrix XenDesktop. Вони дозволяють запускати звичайні програми Windows на відповідному сервері Windows, а будь-який підтримуваний клієнт може отримати доступ до цих програм. Окрім Windows, ICA також підтримується на ряді серверних платформ Unix і може використовуватись для надання доступу до додатків, що працюють на цих платформах. На клієнтських платформах не потрібно запускати Windows; є клієнти під дуже велику

кількість пристроїв на різних платформах. Клієнтське ПО ICA також вбудовано в різні платформи планшет-клієнтів. ICA в цілому аналогічний кінцевим серверам, таким як X Window System. Він також забезпечує зворотний зв'язок користувацького вводу від клієнта до сервера та різні засоби для сервера для відправки графічного виводу, а також інших носіїв, таких як аудіо, від запущених програм до клієнта. Ключовими проблемами такої архітектури є підтримка мережі та продуктивність - графічно інтенсивний додаток (як більшість із них представлено з використанням графічного інтерфейсу), обслуговуване через повільне або мережеве з'єднання з обмеженою пропускною здатністю вимагає значного стиснення та оптимізації, щоб зробити додаток придатним для клієнта. Клієнтська машина може бути іншою платформою і не мати такого ж програмного графічного інтерфейсу, доступного локально – у цьому випадку серверу може знадобитися відправити фактичні дані бітової карти через підключення. Залежно від можливостей клієнта, сервери можуть також передавати клієнту частину графічної обробки, наприклад, для рендеринга мультимедійного контенту. ICA за замовчуванням працює через TCP-порт 1494 або може бути інкапсульований в (CGP) на TCP 2598. ICA підтримує концепцію каналів на рівні сеансу для інкапсуляції пере направлення мультимедіа або розширення USB в ICA [3,6].

#### 4. Віддалений робочий стіл Microsoft (Microsoft Remote Desktop)

Підключення до віддаленого робочого столу добре тим, що для віддаленого доступу до комп'ютера з ним, не потрібно встановлювати будь-яке додаткове програмне забезпечення, за допомогою протоколу RDP, який використовується при доступі, в достатній мірі захищений і добре працює. Але є і недоліки. Для того щоб підключитися до віддаленого робочого столу, ви можете, не встановлюючи додаткові програми для всіх версій Windows 7, 8 і Windows 10 (а також для інших операційних систем, у тому числі Android і iOS, завантажуючи безкоштовний клієнт Microsoft Remote Desktop ), в

якості комп'ютера, до якого підключаються (сервера), може бути використаний тільки комп'ютер або ноутбук з Windows Pro і вище. Ще одне обмеження — без додаткових налаштувань, підключення до віддаленого робочого столу Microsoft працює тільки, якщо комп'ютери та мобільні пристрої знаходяться в одній локальній мережі (наприклад, підключені до одного маршрутизатора у випадку домашнього використання) або ж мають статичний IP-адресу в Інтернеті (при цьому знаходяться не за маршрутизаторами).

Розглянуті приклади мають схожий принцип дії, це віддалене підключення до сервера або комп'ютера, вони можуть вирішити такі питання, як використання програмного забезпечення чи підписок командою, але мають суттєву ваду, для великої команди це дуже незручно, тому що підключатись віддалено до комп'ютера зможуть тільки по черзі, і тому ефективність праці буде дуже низькою.

В сучасних компаніях використовують не тільки локальні додатки, а також різні веб інструменти, для таких інструментів також є рішення – це антидетект сервіси або браузерери. Антидетект браузер - це спеціально налаштований анонімний браузер, з технологією підміни або приховування параметрів [3,8].

Розглянемо деякі з них:

1. Octo Browser - це маркетинговий інструмент, який може допомогти у просуванні продуктів або послуг. Це також веб-переглядач із кількома обліковими записами та базується на Chromium. Це один із найкращих інструментів, коли справа доходить до керування необмеженою кількістю онлайн-акаунтів на кількох платформах. Що відрізняє Octo Browser від багатьох інших конкурентів, так це те, що він може використовуватися без будь-яких збоїв різноманітними користувачами, включно з маркетплейсами, веб-майстрами, криптовалютними трейдерами та онлайн-букмекерами.

Це правда, що існує багато інших чудових маркетингових інструментів або браузерів, але більшість із них можна легко виявити або їм бракує маркетингових стратегій, які пропонує Octo Browser. Не тільки це, з Octo Browser, можна використовувати кілька облікових записів, не боячись бути заблокованим. Він підтримує проксі, які є дуже популярними. Деякі хороші маркетингові інструменти з такими ж якісними функціями виявляються надзвичайно дорогими, на відміну від Octo Browser. А іншим таких якостей просто не вистачає.

Цей браузер має такі функції:

А) Кілька облікових записів - необмеженою кількістю облікових записів можна керувати за допомогою лише одного пристрою, і кожен профіль ідентифікується за відбитком пальця браузера.

Б) Підробка відбитків пальців - одна з найкращих особливостей Octo Browser полягає в тому, що використовується глибокий внутрішній код Chromium, щоб уникнути трекерів, принтерів пальців і шашок.

В) Арбітраж - сотні або тисячі облікових записів належать фахівцям з трафіку в Traffic arbitrage. За допомогою різних методів оплати для різних цілей арбітр отримує гроші щоразу, коли виконується бажана дія. Однак оферентом є веб-сторінка або щось загалом, що може знадобитися для вашого трафіку. Є два типи пропозицій: одні білі, а інші сірі, тобто дозволені та заборонені.

Г) Партнерський маркетинг - Партнерський маркетинг схожий на арбітраж трафіку, але з іншим механізмом отримання доходу. Інструменти для роботи однакові – Facebook, Instagram, Google тощо, відповідно ризики однакові – блокування акаунтів одного користувача. Водночас у афілійованому маркетингу цінність рекламних акаунтів вища, а отже, і ступінь безпеки має бути вищою. Octo Browser дозволяє мінімізувати ризик блокування та захищає бюджети та бренди, з якими працює партнер. Інструменти для афілійованого маркетингу такі ж, як і для арбітражу, як-от



Instagram, Facebook і Google тощо. Однак рекламні облікові записи мають більше значення, коли йдеться про афілійований маркетинг. Octo Browser мінімізує всі ризики блокування, захищаючи бренди, які співпрацюють з ним.

Д) Командна робота - з Octo Browser ви можете насолоджуватися використанням кількох облікових записів без будь-яких заборон або навіть контрольних точок на будь-якому пристрої. Безпека ваших відбитків пальців зберігається, навіть якщо всі ваші профілі веб-переглядача спільні для членів команди.

Е) Файли cookie - Octo Browser постачається з вбудованим роботом cookie, який збирає файли cookie автоматично, коли йдеться про паралельний безголовий режим. Він дозволяє імпортувати файли cookie у форматах JSON і Netscape.

Є) Автоматизація API - З Octo Browser ви можете використовувати Selenium, Puppeteer та деякі інші бібліотеки шляхом автоматичної взаємодії профілів за допомогою API.

Ж) Швидкісні характеристики - браузер Octo дуже схожий на Google Chrome, тому він зручний для користувачів.

З) Універсальність - Octo Browser можуть використовувати кілька людей. Його можуть використовувати веб-майстри для створення профілів на багатьох платформах і торгових майданчиках [13].

2. GoLogin — це програма для захисту від виявлення, яка в основному використовується для кількох облікових записів. Він багато в чому схожий з віртуальними приватними мережами та проксі-серверами щодо маскуванню вашої цифрової особистості в Інтернеті. Однак цей інструмент більше підходить для масових завдань. У двох словах, це дозволяє створювати кілька окремих облікових записів для різних видів сайтів. Різні види цифрових маркетологів можуть знайти цінність у GoLogin. Якщо керувати партнерськими програмами або жонглювати рекламними кампаніями,

блокування може злити зусилля на вітер. Необхідність змінити параметри браузера не менш важлива під час серфінгу в Інтернеті. GoLogin може допомогти витратити ваші фізичні та віртуальні ресурси більш ефективно, щоб ви могли досягати більшого з меншими витратами.

#### Функції GoLogin:

База даних відбитків пальців - використовуючи технологію GoLogin, можна налаштувати параметри свого цифрового відбитка пальця. Він являє собою колекцію даних монітора сайтів, щоб ідентифікувати вас через ваш браузер. Завдяки можливості підробити 50+ характеристик з'єднання можна створити абсолютно унікальний профіль браузера та приховати свою особу.

За допомогою GoLogin можна налаштувати такі параметри:

Операційні системи.

Типи підключення.

Мережа Tor (Німеччина, ЄС, Франція, Великобританія та США).

HTTP проксі.

Проксі SOCKS 4.

Проксі SOCKS 5.

Часовий пояс.

На основі зовнішнього IP.

На основі особистих переваг.

WebRTC.

Геолокація.

Навігатор.

Шрифти.

Маска мультимедійних пристроїв.

Обладнання.

Варіанти зберігання.

Плагіни для браузера.

Cookie.

Імпорт файлів cookie.

Безкоштовні проксі.

Як згадувалося, можна отримати доступ до безкоштовних проксі-серверів GoLogin, щоб отримати IP-адреси (протокол Інтернету) з п'яти країн. Отже, використання стороннього постачальника проксі-сервера преміум-класу необов'язково. Під час створення нового профілю можна наказати GoLogin створити унікальний відбиток пальця одним натисканням кнопки. Кожен профіль зберігає свій цифровий відбиток пальця та файли cookie, щоб уникнути необхідності повторної авторизації. Крім зручності, використання цього інструменту для виконання важкої роботи допомагає гарантувати надійність створеної цифрової ідентифікації. Змішування параметрів без міри чи причини може призвести до дивного профілю браузера, який може викликати підозру. Крім того, можна замінити відбиток пальця наявного профілю на новий одним клацанням миші. Якщо піти в цьому напрямку, можна негайно перетворити скомпрометований відбиток пальця на нову особу. Таким чином, теоретично можна генерувати необмежену кількість ідентичностей.

Профілі браузера - існує обмеження на кількість профілів браузера, які ви можете зберігати в GoLogin, від 1 до 8000. Ці профілі, які безпечно зберігаються в хмарі, не збігаються локально, оскільки жоден із них не має однакового відбитка пальця. У результаті сайти навряд чи заблокують ваші облікові записи.

Наразі GoLogin доступний для Windows, macOS, Linux та Android. Мобільний додаток знаходиться на стадії бета-тестування, але до нього доступний ранній доступ у Google Play.

Загалом GoLogin сумісний із такими ОС:

Windows 7 або новішої версії (тільки для 64-розрядних систем) і Server 2012 і 2016.

macOS 10.11 (El Capitan) до 10.14 (Mojave).

Linux Ubuntu 18.04.

Більшість сімейства Linux.

Так само GoLogin має хмарну версію. Це означає, що можна увійти у свій обліковий запис за допомогою будь-якого звичайного браузера на будь-якому пристрої, не завантажуючи відповідну програму.

Одна з визначальних особливостей цього браузера для захисту від виявлення — технологія контролю відбитків пальців WebGL (Web Graphics Library). Він може відтворювати 2D- і 3D-графіку в браузері без використання плагінів [14,15].

3. FraudFox — це віртуальна машина на базі Windows 7, яка також дозволяє підробити вашу цифрову ідентичність. це універсальний інструмент для підробки агентів користувачів і пристроїв, та повністю деталізована віртуальна машина у своєму класі.

Параметри захисту від агітації також включені, FraudFox пропонує параметри налаштування елемента canvas HTML 5, щоб ви мали повний контроль над ним. FraudFox також запобігає витоку IP-адрес і DNS. Запити WebRTC і DNS змушені виконувати протокол VPN або SOCKS. Витік незашифрованих даних добре контролюється, що означає справді анонімний перегляд, більше ніж просте редагування за допомогою агента користувача, FraudFox пропонує набагато більше функцій, таких як маніпуляція деталями пристрою через консоль віртуальної машини, зміна мережевого сліду за допомогою кнопки обфускації та багато іншого. Він також відкритий для подальших інсталяцій програмного забезпечення, наприклад VPN-клієнта у віртуальній машині, щоб мати можливість підвищити рівень анонімного перегляду.

Але головна проблема полягає в тому, що FraudFox є набагато менш безпечним, оскільки Windows 7 — це продукт, у якого закінчився термін підтримки, і він більше не оновлюється, а це означає, що його безпека, можливо, вже давно застаріла. На відміну від браузерів із засобом виявлення,

FraudFox підробляє лише кілька основних параметрів, таких як шрифти, агент користувача, мова, часовий пояс і роздільна здатність екрана. Для багатьох веб-сайтів такого примітивного маскування буде недостатньо, і вас можуть легко виявити.

Отже було розглянуто багато інструментів для оптимізації командної роботи, але всі вони досить не універсальні, на одних незручно працювати великими командами, на інших неможливо використовувати локальні додатки, також антидетект браузері більше підходять для команд, які не працюють з одними і тими ж інструментами, а більше використовують один комп'ютер в багатьох облікових записах. Також всі ці інструменти з помісячною оплатою, і дивлячись наскільки велика команда, вони можуть коштувати не дешево [14,15].

### **1.3 Постановка задачі**

Виходячи з аналізу проблеми та існуючих інструментів, треба розробити інформаційну технологію керування платними підписками інтернет-сервісів великою командою або більш ніж одним користувачем одночасно.

Для досягнення поставленої мети було проведено аналіз існуючих інструментів та описана актуальність задачі.

За результатами аналізу визначено наявність проблем, сформульована мета та поставлено такі задачі:

- вибрати середовище розробки;
- визначити порядок розробки;
- розробити інформаційну технологію керування платними підписками інтернет-сервісів;
- після реалізації системи провести тестування роботи.

Практична значимість роботи полягає у підвищенні ефективності діяльності компаній за рахунок економії бюджету.

## 2 МЕТОДИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ КЕРУВАННЯ ПЛАТНИМИ ПІДПИСКАМИ ІНТЕРНЕТ- СЕРВІСІВ

### 2.1 Вибір середовища розробки

Для досягнення мети було обрано «WINDOWS 10» (рис. 2.1) на базі цієї операційної системи будуть створюватись умови для користування платними інструментами великою командою.



Рисунок 2.1 - Загальний вигляд робочого столу «WINDOWS 10»

Вибір «WINDOWS 10» для виконання поставленої задачі обґрунтований тим, що це найпопулярніша операційна система [11] (рис. 2.2 та рис. 2.3) доволі легко встановлюється та обслуговується. До того ж, майже всі вмiють користуватися цією операційною системою.

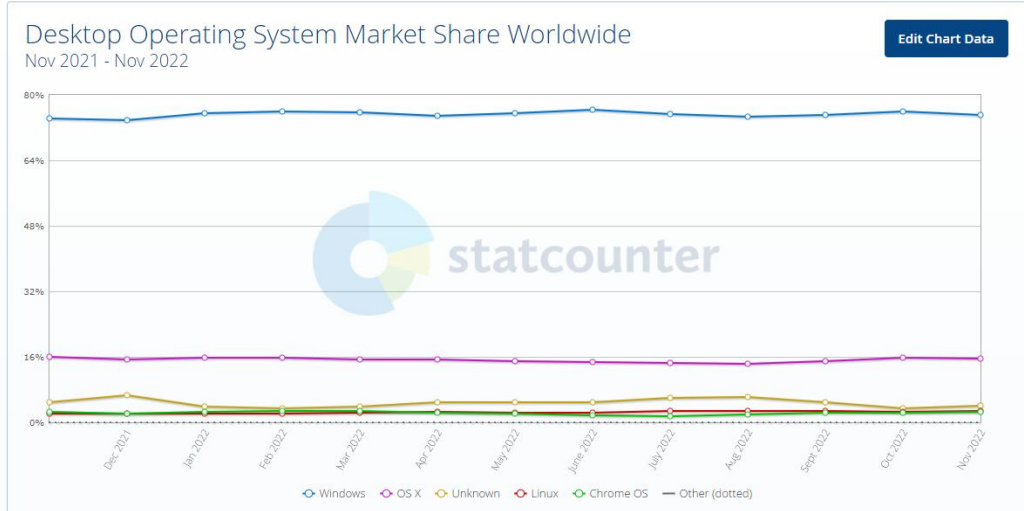


Рисунок 2.2 - Дані сайту statcounter.com по всім операційним системам [11]

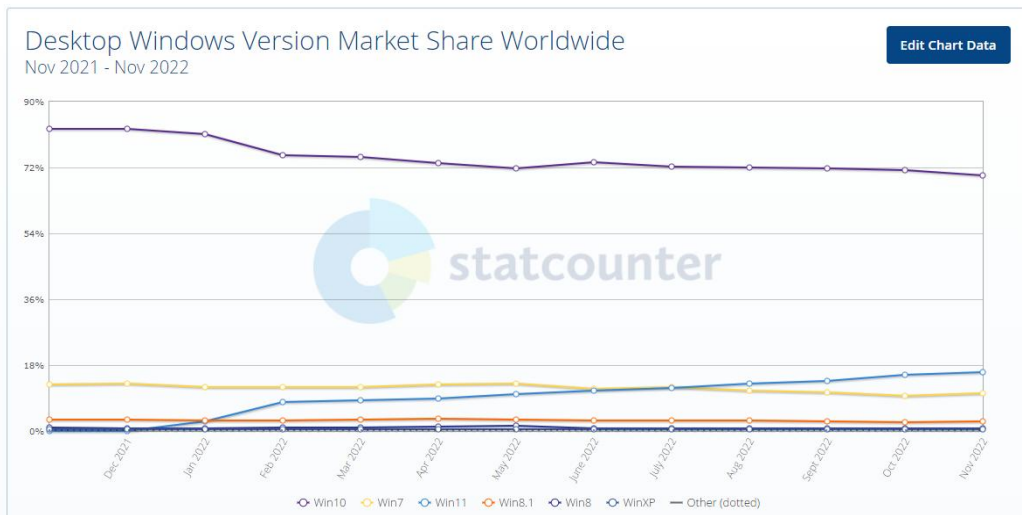


Рисунок 2.3 - Дані сайту statcounter.com по операційним системам сімейства

«WINDOWS» [11]

Це також позитивно впливає на результат, не потрібно навчати користуватися командою.

Для вирішення поставленої мети ми будемо використовувати метод віддаленого підключення до комп'ютера. Тож це ще один привід використання цієї операційної системи, тому що в систему вже вмонтований такий інструмент як Remote Desktop Protocol (RDP) (рис. 2.4 та рис. 2.5) [8,9,16-18].

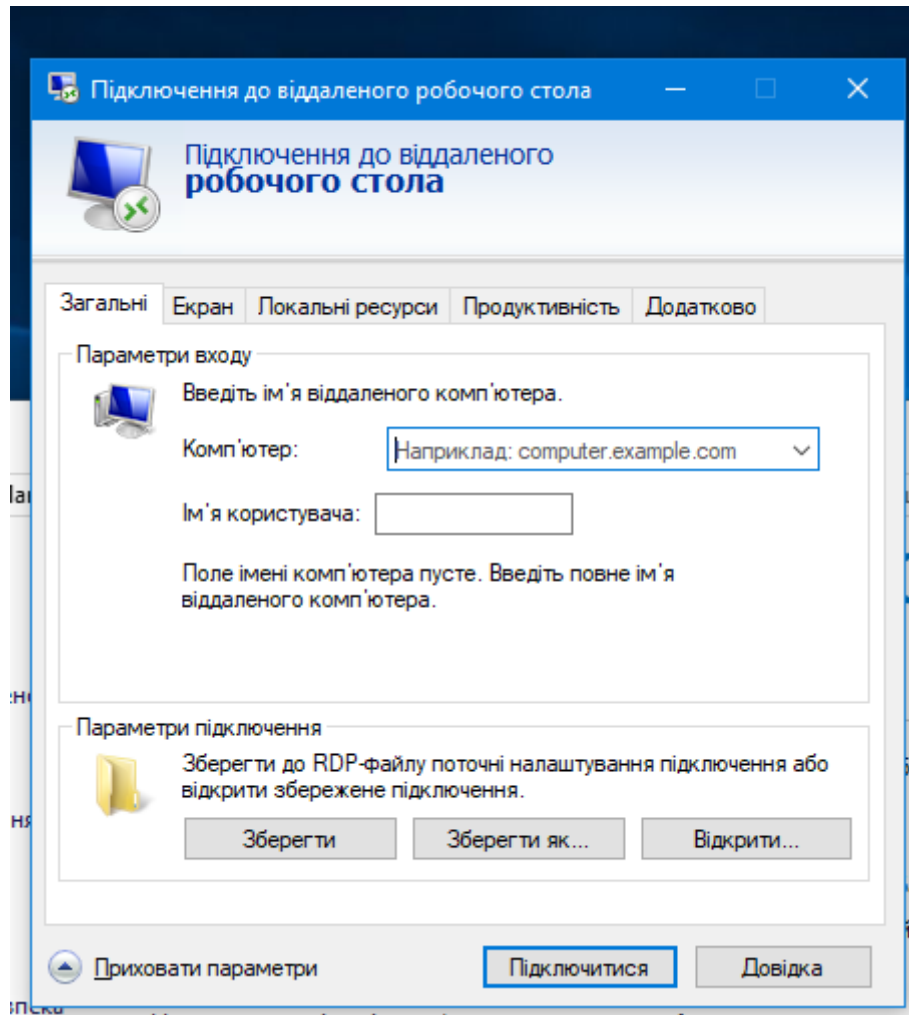


Рисунок 2.4 - Діалогове вікно підключення до віддаленого робочого столу за допомогою RDP

В першому розділі вже описувався цей протокол, але в чистому вигляді він нам не підійде, тому що в нього є недолік для використання більш ніж одним користувачем, а саме при підключенні RDP до комп'ютера



користувач заходить до системи повністю, і любий хто був підключений до комп'ютера або працював на ньому буде відключений.

Це звісно основна проблема яку потрібно вирішити для досягнення мети, бо переваг у такого способу дуже багато.

При використанні RDP користувач фізично входить на комп'ютер, тому може використовувати все встановлене на комп'ютері програмне забезпечення, навіть те, що ліцензується за допомогою USB ключа, якщо ключ фізично присутній в роз'ємі того комп'ютера до якого підключився користувач.

До того ж якщо ліцензія діє тільки на одного користувача, то можна підключатись під цим користувачем не порушуючи умов використання, так як фактично, комп'ютер один і на ньому встановлено програмне забезпечення.

Також до переваг можна віднести захищеність підключення по цьому протоколу тож розглянемо цей протокол більш детально.



Рисунок 2.5 - Загальне відображення RDP

Зв'язок у RDP базується на кількох каналах, а протокол теоретично підтримує до 64 000 унікальних каналів.

Основна функція RDP полягає в передачі відображення монітора (пристрою виведення) від віддаленого сервера до клієнта та клавіатури та/або миші (пристроїв введення) від клієнта до віддаленого сервера. Зв'язок під час RDP-з'єднання буде надзвичайно асиметричним (рис. 2.6), тоді як більшість даних надходитиме від сервера до клієнта. Зв'язок RDP за замовчуванням шифрується за допомогою блокового шифру RSA RC4 [9].



Рисунок 2.6 - Асиметрична комунікація протоколу RDP

Перш ніж ми розберемося, як насправді працює з'єднання RDP, давайте розглянемо протоколи/стандарти, на яких базується RDP. (рис. 2.7)

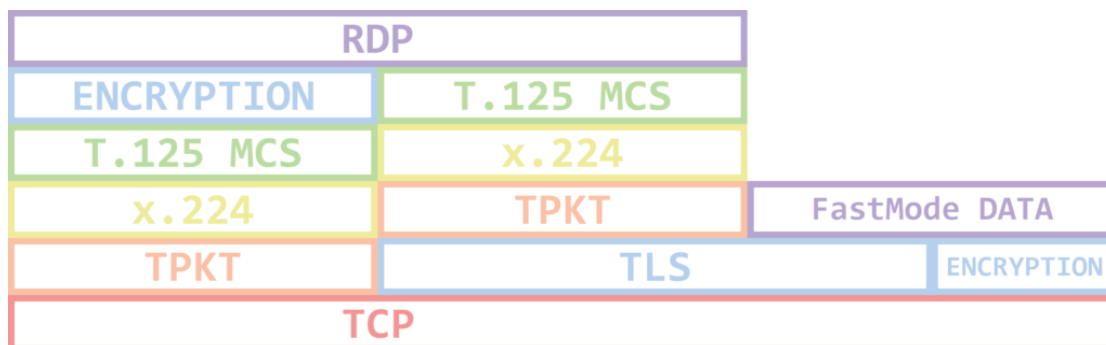


Рисунок 2.7 - Стэк протоколів RDP

TPKT відомий як транспортна служба ISO на основі TCP . TPKT дозволяє одноранговим вузлам обмінюватися одиницями інформації, відомими як одиниці даних транспортного протоколу ( TPDU або PDU ).

X.224 — це транспортний протокол , орієнтований на з'єднання, він надає транспортну службу в режимі з'єднання. RDP використовує його в початковому запиті на підключення та відповіді.

T.125 MCS — це служба багатоточкового зв'язку . Це дозволяє RDP спілкуватися через кілька каналів і керувати ними [9,19].

Надсилання та отримання даних через стек RDP по суті те саме, що й 7-рівнева модель OSI для зв'язку. Передані дані розділяються, спрямовуються на канал, шифруються, обгортаються, обрамляються та упаковуються перед тим, як перейти по дроту іншій стороні, а потім проходять той самий процес у зворотному порядку.

Реалізація MS RDP абстрагувала всю складність стека протоколів, і це дозволяє розробникам легко писати розширення для протоколу [9].

Етапи підключення RDP (рис. 2.8).

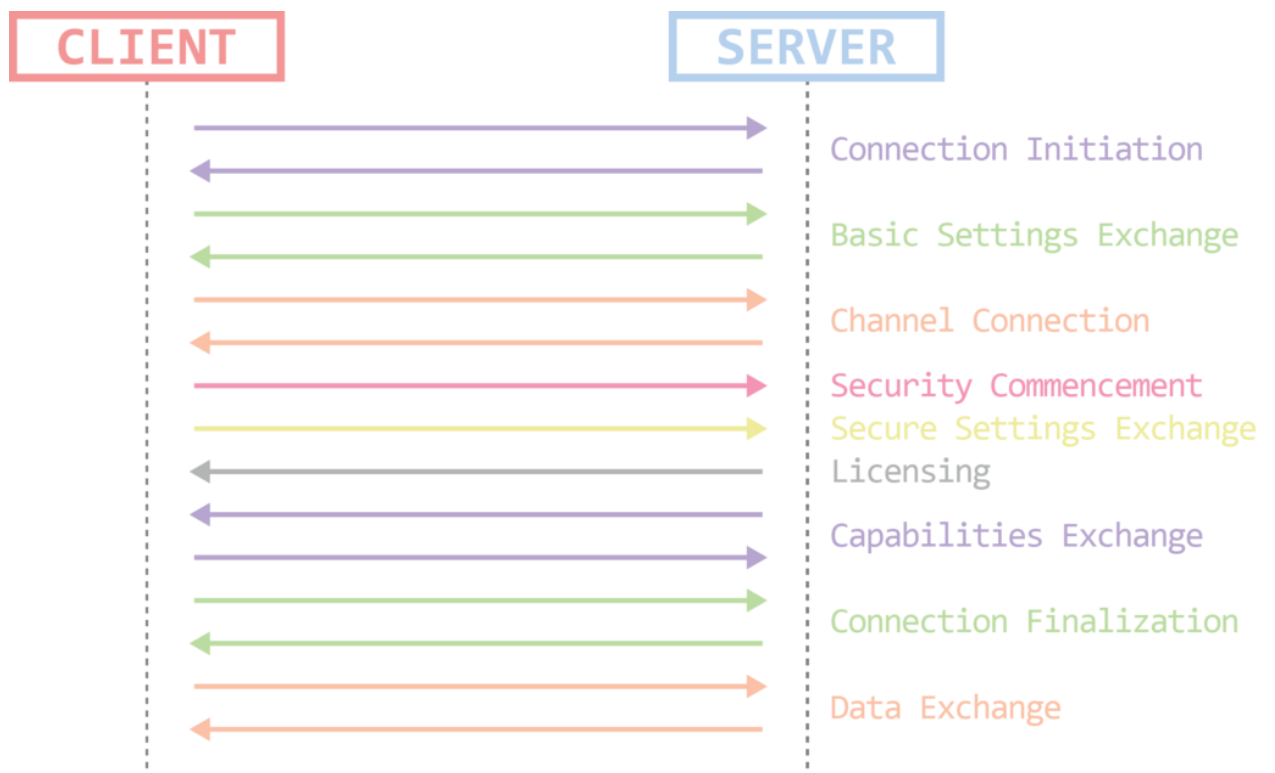


Рисунок 2.8 - Етапи підключення RDP

Підключення RDP можна розбити на кілька етапів:

- Ініціалізація підключення.
- Обмін основними налаштуваннями.
- Підключення каналу.
- Початок блоку безпеки.
- Захищений обмін налаштуваннями.

- Ліцензування.
- Обмін можливостями.
- Завершення підключення.
- Обмін даними.

Розглянемо етапи більш детально:

**Ініціалізація підключення** (рис. 2.9) - підключення RDP ініціюється клієнтом за допомогою PDU запиту на підключення X.224. Цей пакет містить запит на узгодження RDP, який містить кілька позначок підключення та протоколи безпеки, які підтримуються клієнтом. Ці протоколи безпеки можуть належати до однієї з двох категорій:

1. Стандартна безпека RDP:

За замовчуванням шифрування RSA RC4.

2. Покращена безпека RDP:

TLS.

CredSSP (TLS + NTLM/ Kerberos ).

RDSTLS – розширений RDP з TLS.

Підключення підтверджується сервером за допомогою X.224 Connection Confirm PDU. Цей пакет містить відповідь RDP Negotiation Response, яка використовується для інформування клієнта про вибраний протокол безпеки (вибраний із підтримуваних клієнтом протоколів), який використовуватиметься протягом усього терміну служби з'єднання [9,19].

З цього моменту подальші дані будуть загорнуті в X.224 Data PDU [9].

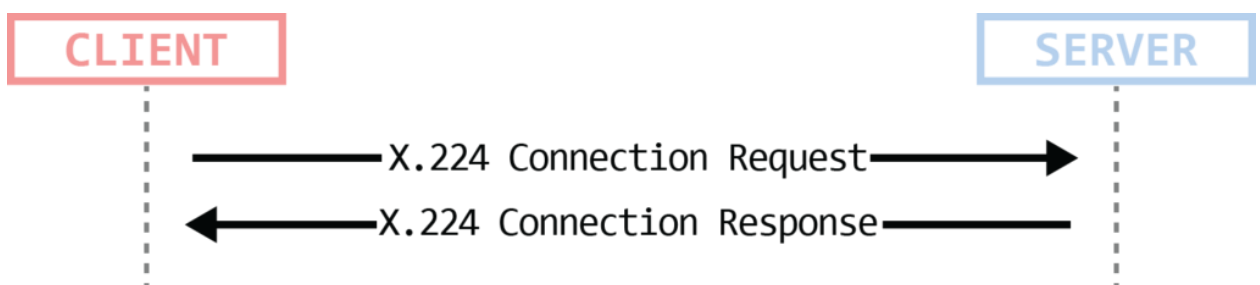


Рисунок 2.9 - Ініціалізація підключення RDP

**Обмін основними налаштуваннями** (рис. 2.10) - на цьому етапі відбувається обмін основними налаштуваннями між клієнтом і сервером за допомогою MCS Connect Initial PDU і MCS Connect Response PDU (відповідно). Ці налаштування (як від клієнта, так і від сервера) включають:

Основні дані – версія RDP, роздільна здатність робочого столу, глибина кольору, інформація про клавіатуру, ім'я хоста, інформація про програмне забезпечення клієнта (ідентифікатор продукту, номер збірки) тощо.

Дані безпеки – методи шифрування, розмір ключів сеансу, випадковий сервер (пізніше використовується для створення ключів сеансу) і сертифікат сервера (деяке з цього актуально лише під час використання стандартної безпеки RDP).

Дані мережі – інформація про запитані та виділені віртуальні канали. Він містить кількість каналів і масив певних віртуальних каналів. Клієнт запитує точний тип каналів у запиті, а сервер надає фактичні ідентифікатори каналів у відповіді [9].

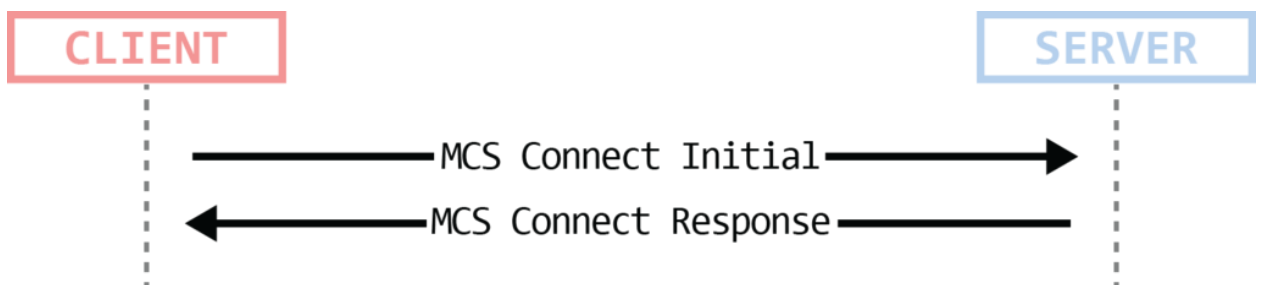


Рисунок 2.10 - Обмін основними налаштуваннями

**Підключення каналу** (рис. 2.11) - після створення списку віртуальних каналів, які використовуватимуться в сеансі RDP, настає етап, на якому здійснюється підключення кожного окремого каналу. Це має декілька під етапів :

Запит MCS Erect Domain – висота в домені MCS. Оскільки RDP не використовує переваги передових топологій MCS, він буде 0.

MCS Attach User Request – запит на ідентифікатор каналу користувача

### MCS Attach User Confirm – ID каналу користувача

Запити та підтвердження приєднання до каналу MCS – клієнт почне надсилати запити на приєднання до віртуальних каналів, використовуючи свої ідентифікатори. Починаючи з каналу користувача, каналу вводу/виводу та продовжуючи віртуальними каналами, узгодженими під час обміну основними налаштуваннями. Сервер, у свою чергу, підтверджуватиме кожне успішне приєднання до каналу [9].

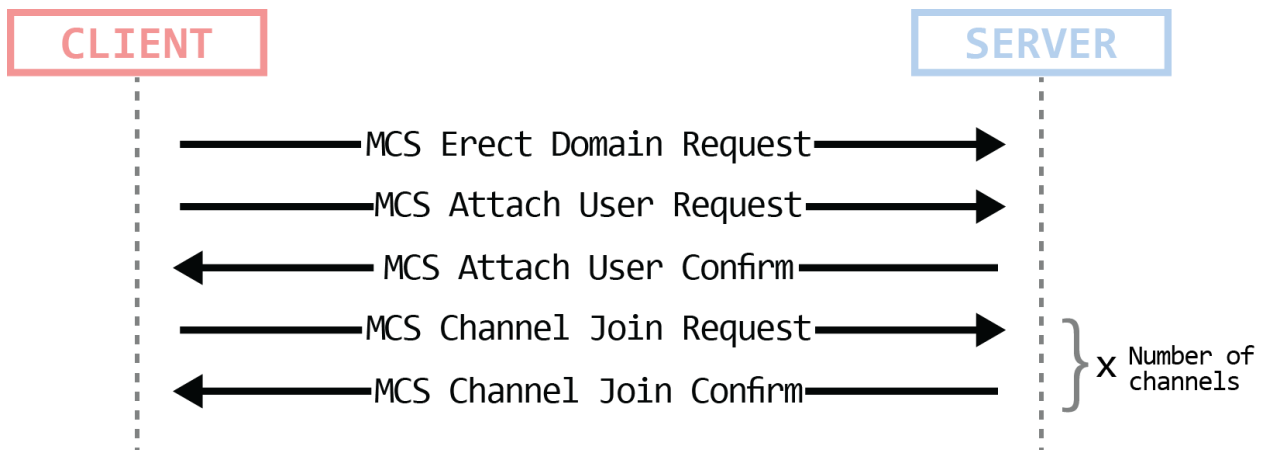


Рисунок 2.11 - Підключення каналу

**Початок блоку безпеки** (рис. 2.12) - клієнт надсилає PDU Security Exchange, що містить клієнт, випадково зашифрований відкритим ключем сервера. Потім клієнт і сервер використовують випадкові числа (як із даних безпеки Basic Settings Exchange, так і з Security Exchange PDU), щоб створити ключі шифрування сеансу. З цього моменту наступний трафік RDP може бути зашифрований [9].

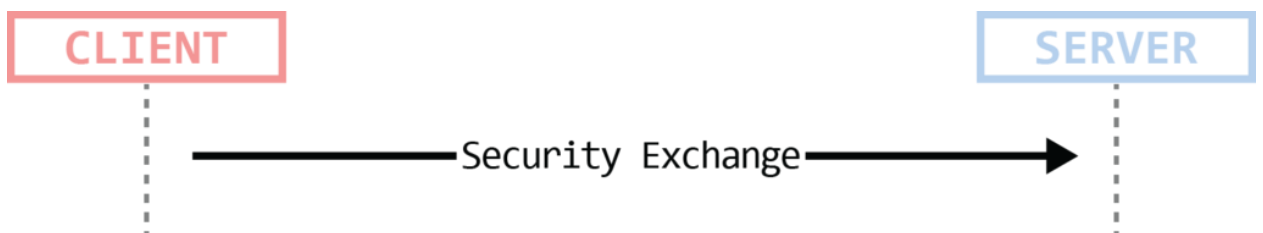


Рисунок 2.12 - Початок блоку безпеки

**Захищений обмін налаштуваннями** (рис. 2.13) - на цьому етапі клієнт надсилає зашифрований PDU клієнтської інформації, що містить інформацію про підтримувані типи стиснення, домен користувача, ім'я користувача, пароль, робочий каталог тощо [9].

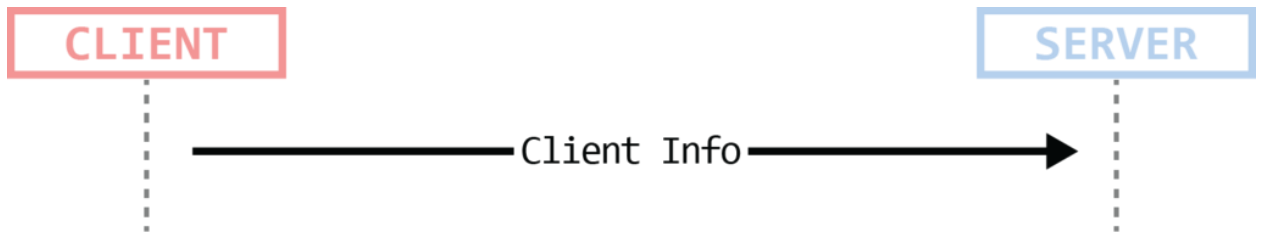


Рисунок 2.13 - Захищений обмін налаштуваннями

**Ліцензування** (рис. 2.14) - цей етап призначений для того, щоб дозволити авторизованим користувачам підключатися до термінального сервера. Це означає підтримку більш ніж 2 одночасних підключень (що є типовим для «сервера RDP Windows») до сервера. Для цього потрібно придбати ліцензію від Microsoft.

У багатьох випадках для сервера RDP не налаштовано сервер ліцензування, у цьому випадку сервер RDP просто надішле PDU клієнту, який «затверджує» його ліцензію (лише до 2 сеансів) [9].

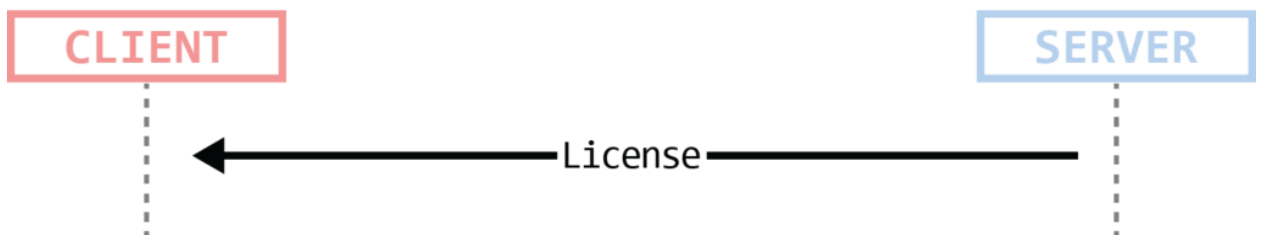


Рисунок 2.14 - Ліцензування

**Обмін можливостями** (рис. 2.15) - сервер надсилає підтримувані можливості в Demand Active PDU. Цей PDU містить структуру, яка має багато можливостей різних типів. За даними Microsoft, маємо 28 типів

наборів можливостей. Основні типи: загальні (версія ОС, загальне стиснення), введення (тип і функції клавіатури, підтримка швидкого доступу тощо), шрифти, віртуальні канали, растрові кодеки та багато іншого. Тоді сервер може надіслати або не надіслати PDU макета монітора для опису моніторів на сервері. Потім клієнт відповідає Confirm Active PDU, що містить власний набір можливостей [9].

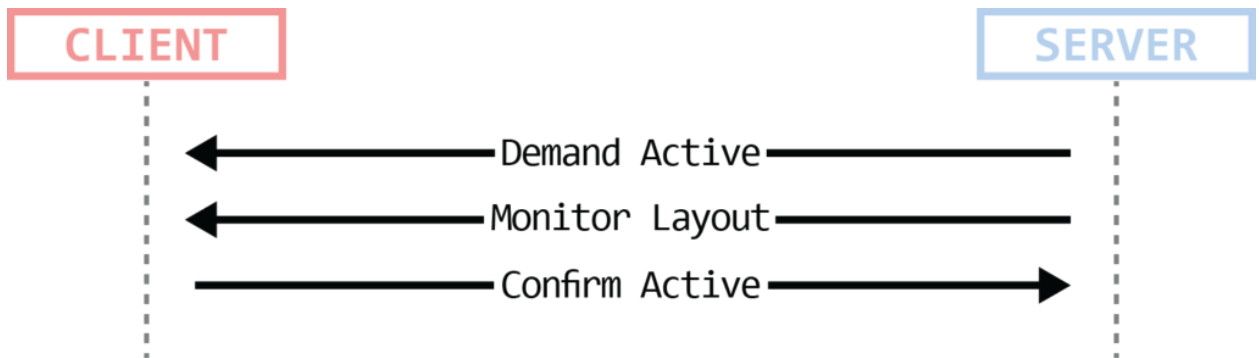


Рисунок 2.15 - Обмін можливостями

**Завершення підключення** (рис. 2.16) - клієнт і сервер обмінюються кількома типами PDU, щоб завершити з'єднання. Усі ці PDU надходять від клієнта (PDU можна надсилати один за одним, не чекаючи відповіді). PDU включає :

Client/Server Synchronize PDU – використовується для синхронізації ідентифікаторів користувачів між клієнтом і сервером.

PDU керування клієнтом/сервером (співпраця) – і клієнт, і сервер надсилають цей PDU, щоб вказати спільний контроль над сеансом.

Client Control PDU (Request/Grant Control) – клієнт надсилає запит на керування, сервер надає його.

PDU/PDU з постійним списком ключів (необов'язково) – клієнт надсилає серверу список ключів, кожен ключ ідентифікує кешовану растрову карту. Це дозволяє кешу растрового зображення бути постійним (на відміну від обмеження терміну служби з'єднання). Кешування растрового зображення — це механізм, який використовується для зменшення



мережевого трафіку, необхідного для передачі графічного виводу від сервера до клієнта.

PDU списку шрифтів/карти – ці PDU зберігають інформацію про шрифти для сеансу RDP (назва шрифту, середня ширина, підпис тощо) [9].

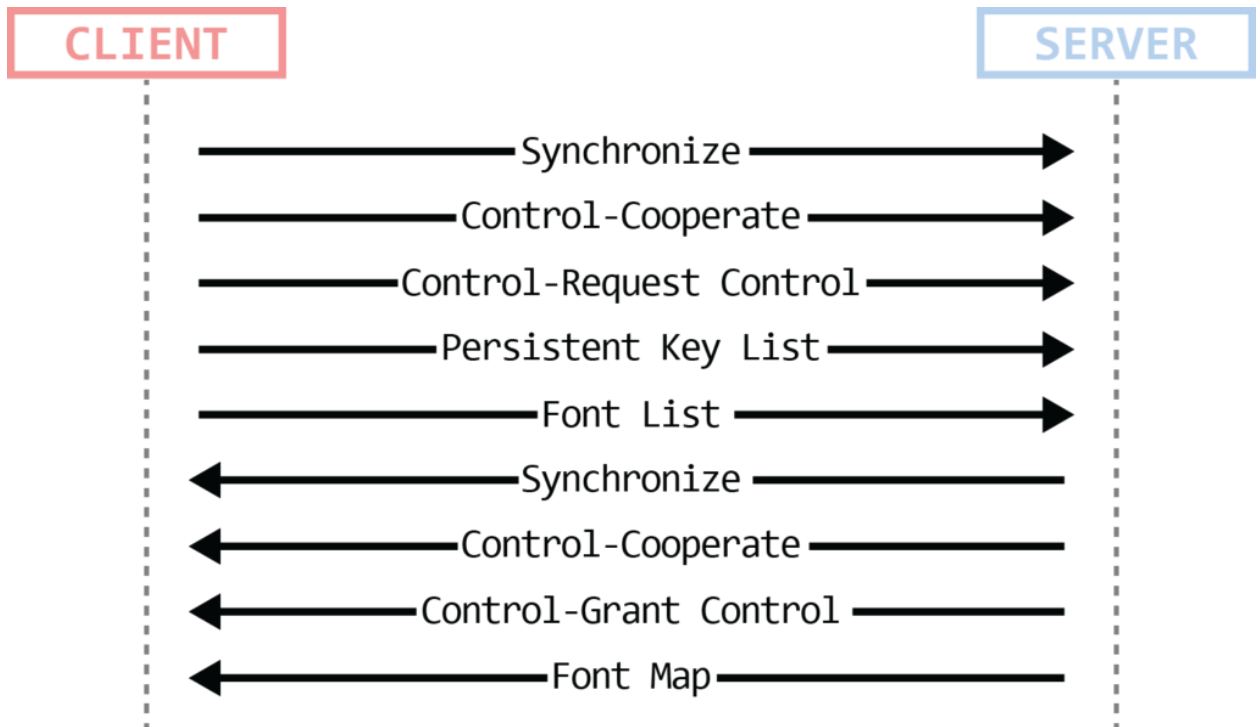


Рисунок 2.16 - Завершення підключення

**Обмін даними** (рис. 2.17) - після завершення підключення основна частина даних, що надсилаються між клієнтом і сервером, буде входом (клієнт->сервер) і графічними даними (сервер->клієнт). Додаткові дані, які можна передати, включають інформацію про керування з'єднанням і повідомлення віртуального каналу:

Базовий вхід і вихід - протягом життя з'єднання клієнт і сервер обмінюються основними вхідними/вихідними даними. Клієнт надсилає вхідні дані, а сервер надсилає вихідні дані.

Вхідні дані – містять інформацію про мишу та клавіатуру, а також періодичну синхронізацію (наприклад, стан клавіш `NAM_LOCK` / `CAPS_LOCK`)

Вихідні дані – основні вихідні дані містять растрові зображення сеансу користувача на сервері. Крім того, сервер може надсилати звукову інформацію (лише у формі дуже простого «гудка» – частота + тривалість).

Ці основні дані введення/виведення можуть передаватися одним із двох способів: повільним або швидким шляхом .

Slow-Path – звичайний PDU з усіма заголовками стеку протоколу RDP

Fast-Path – як випливає з назви, його було створено, щоб зменшити як обсяг переданих даних, так і обсяг обробки, необхідний для їх обробки. Це робиться шляхом зменшення/видалення заголовків PDU з певних типів PDU (наприклад, введення з клавіатури/миші) [9].

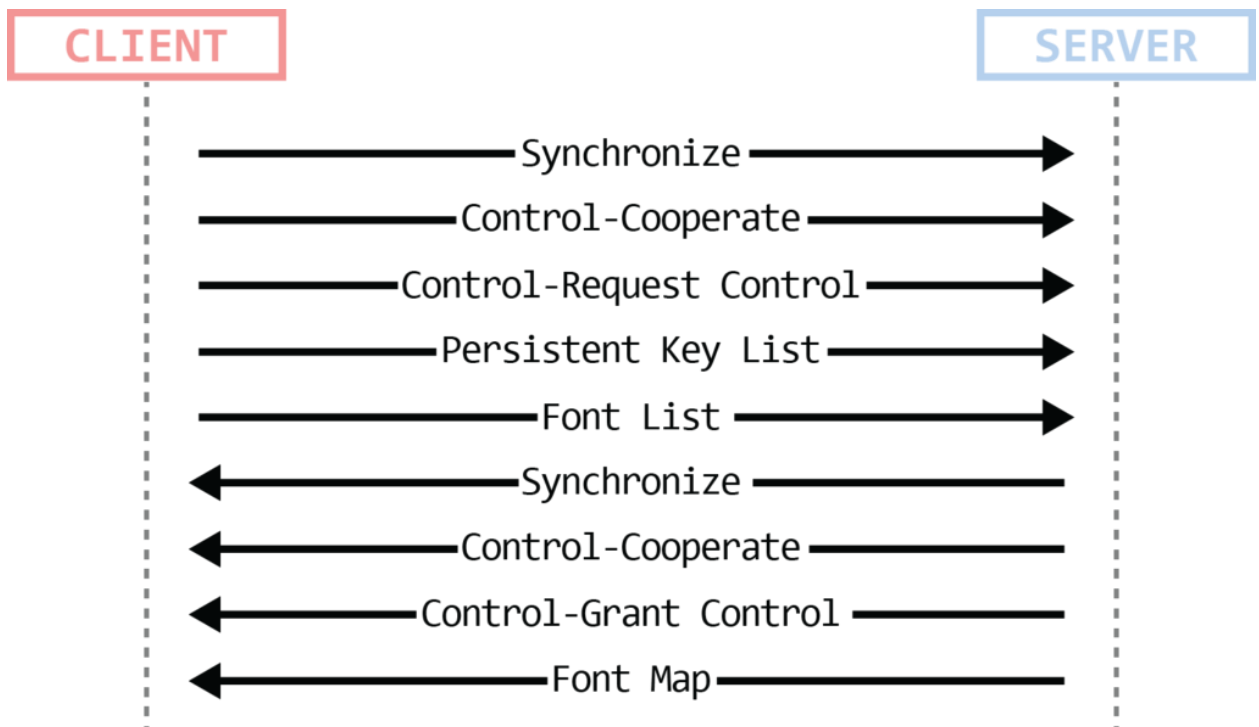


Рисунок 2.17 – Обмін даними

З більш детального розгляду протоколу RDP ми можемо зробити висновок, що цей протокол безпечний, та може підтримувати не одне з'єднання, а використання актуальної операційної система надасть гарантію на своєчасне оновлення компонентів для захисту від витоку інформації.

Для того щоб вирішити питання одночасного підключення до RDP достатньої кількості користувачів нам знадобиться звернутися до проекту з відкритим вихідним кодом RDP Wrapper (рис. 2.18 та рис. 2.19) [12].

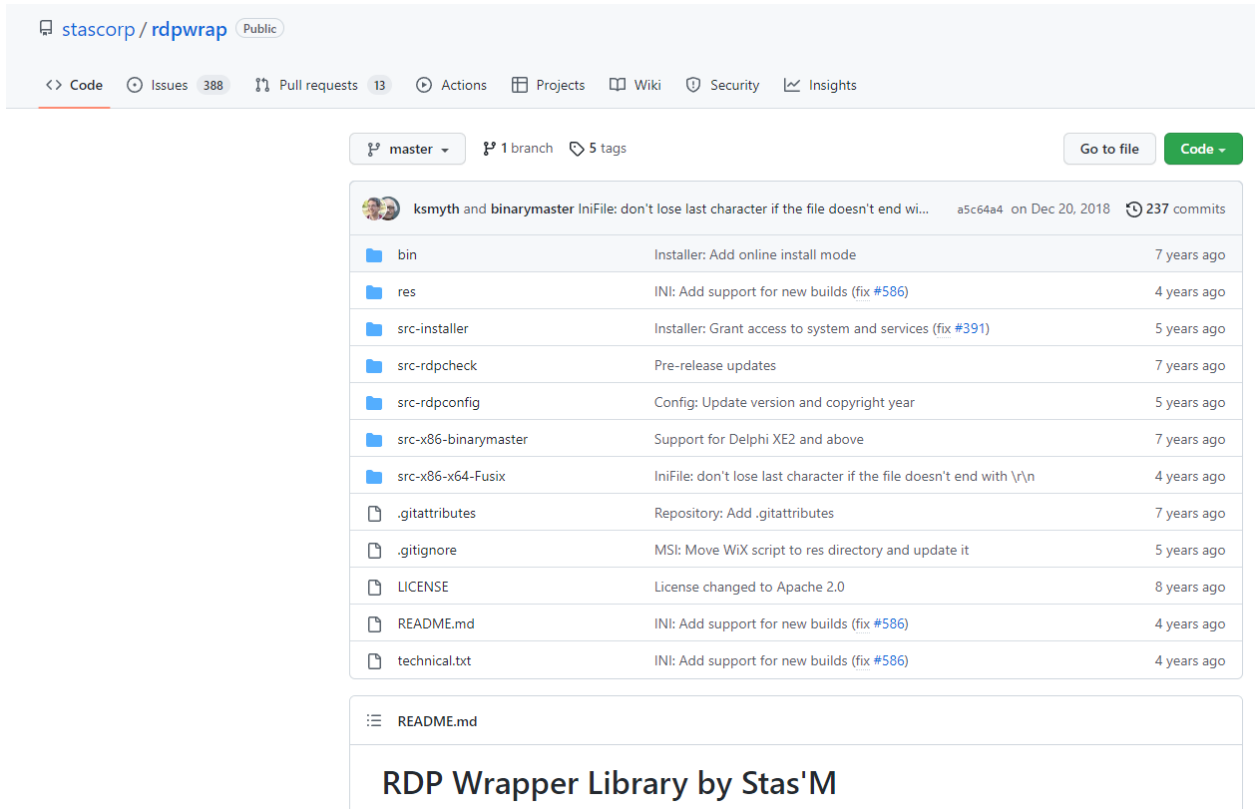


Рисунок 2.18 - Сторінка проекту на ресурсі GitHub.com [12]

RDP Wrapper діє як шар між службою віддалених робочих столів і диспетчером служб управління (SCM). Цей додаток дозволяє користувачам увімкнути підтримку кількох одночасних сеансів RDP.

Цей додаток не змінює код та данні файлу Termsrv.dll, а тільки завантажує бібліотеку «termsrv» з іншими параметрами, необхідними для одночасного підключення. Це означає, що додаток працюватиме навіть коли оновиться операційна система, а це в свою чергу означає, що користувачам не потрібно боятися оновлення Windows та порушення умов ліцензії [12].

Ім'я	Дата змінення	Тип	Розмір
install	23.07.2016 20:19	Пакетний файл ...	1 КБ
RDPCheck	12.08.2015 20:23	Застосунок	1 021 КБ
RDPConf	27.12.2017 21:29	Застосунок	1 071 КБ
RDPWInst	27.12.2017 20:20	Застосунок	1 426 КБ
uninstall	06.03.2016 5:40	Пакетний файл ...	1 КБ
update	06.03.2016 5:40	Пакетний файл ...	1 КБ

Рисунок 2.19 - Інсталяційні файли додатка «RDP Wrapper»

Також дуже корисною функцією цього додатка є можливість підключатися по RDP всім кому це потрібно під одним користувачем, тобто якщо потрібний нам додаток ліцензується тільки під одного користувача, то все одно можна користуватися цим додатком, також можна змінити порт підключення до віддаленого робочого столу, це може бути корисно коли у провайдера Інтернету закриті порти, всі налаштування можна побачити на рис. 2.20.

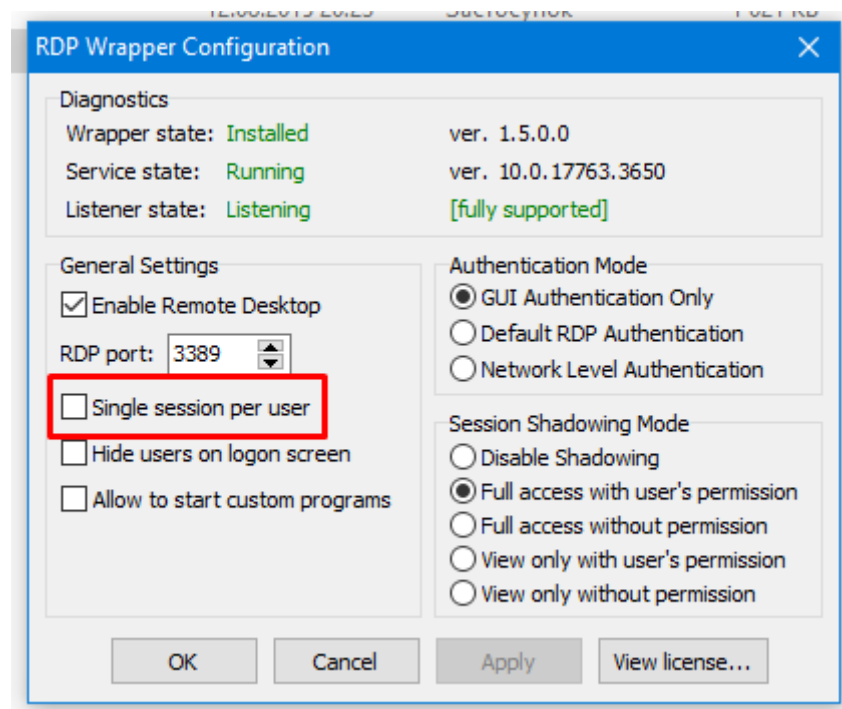


Рисунок 2.20 - Діалогове вікно налаштувань «RDP Wrapper»

По суті єдине, що нам потрібно від цього додатка і є можливість підключення під одним користувачем багато разів, тоді як кількість з'єднань можна налаштувати в групових політиках самої операційної системи. Для цього нам потрібно зайти в командну строку та вписати «gpedit.msc». Далі зайти в «Конфігурація комп'ютера» потім вибрати вкладку «Адміністративні шаблони» далі «Компоненти Windows» далі «Служби віддалених робочих столів» потім «Вузол сеансів віддалених робочих столів» та перейти на вкладку «Підключення» в цій папці є політика «Обмежити кількість підключень» (рис. 2.21) потрібно зайти в нього та включити його, при цьому вибрати кількість підключень яка потрібна, або вписати велике число (рис. 2.22).

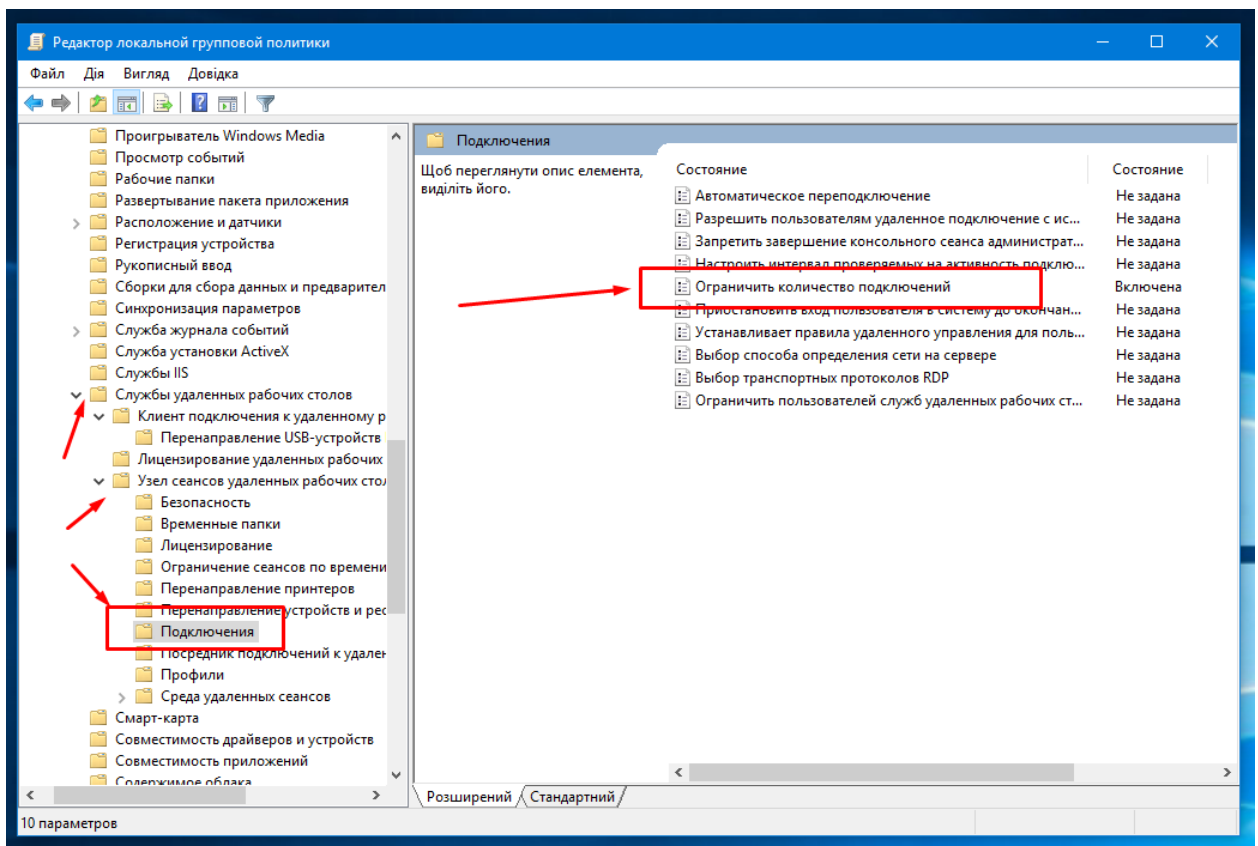


Рисунок 2.21 - Діалогове вікно редактора групових політик Windows

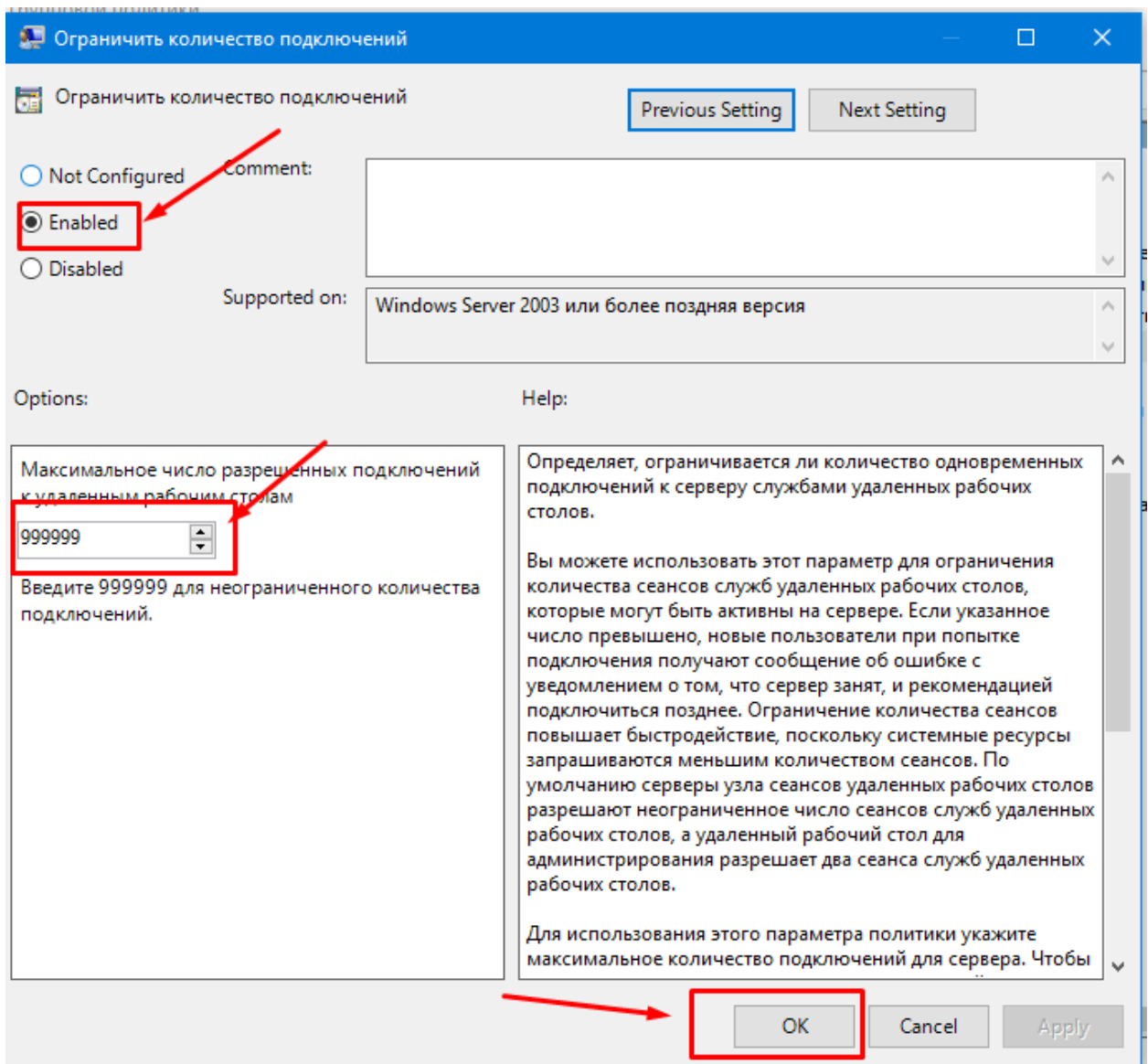


Рисунок 2.22 - Діалогове вікно налаштувань груповою політики по обмеженню кількості підключень до віддаленого робочого столу

Після встановлення додатку RDP Wrapper та налаштувань групових політик це вже вирішує проблему використання багатою кількістю користувачів локальних програм, які ліцензуються на одному комп'ютері, за допомогою кодів, USB ключів, та навіть прив'язують ліцензію до користувача. Якщо взяти приклад з магазинами побутової техніки, то це дуже оптимізує використання програмного додатка для ведення обліку, продажів так як під одним користувачем можна запусити декілька версій цього додатка, а вже в самому додатку зробити різні авторизації для касирів,

бухгалтера, та людини яка буде ставити прихід товару, а також власник зможе все це контролювати з дому не доплачуючи за кількість копій. Це й є перший тип оптимізації для локальних додатків.

Проте для платних підписок на веб-сервісах цього не достатньо, тому що браузер «Google Chrome», «Opera», «Mozilla» не буде запускатись більше одного разу, тобто браузер зможе запустити тільки той користувач, який запустив його першим, для всіх інших він не буде працювати. Єдиний браузер який може запускатися в такому режимі це «Internet Explorer», але цей браузер вже не підтримується майже всіма веб-сервісами, наприклад тим що ми розглядали у першому прикладі, сервісом Ахревс.

Для прикладу можна побачити як виглядає сторінка авторизації Ахрефс в «Google Chrome» (рис. 2.23) та «Internet Explorer» (рис. 2.24) [1].

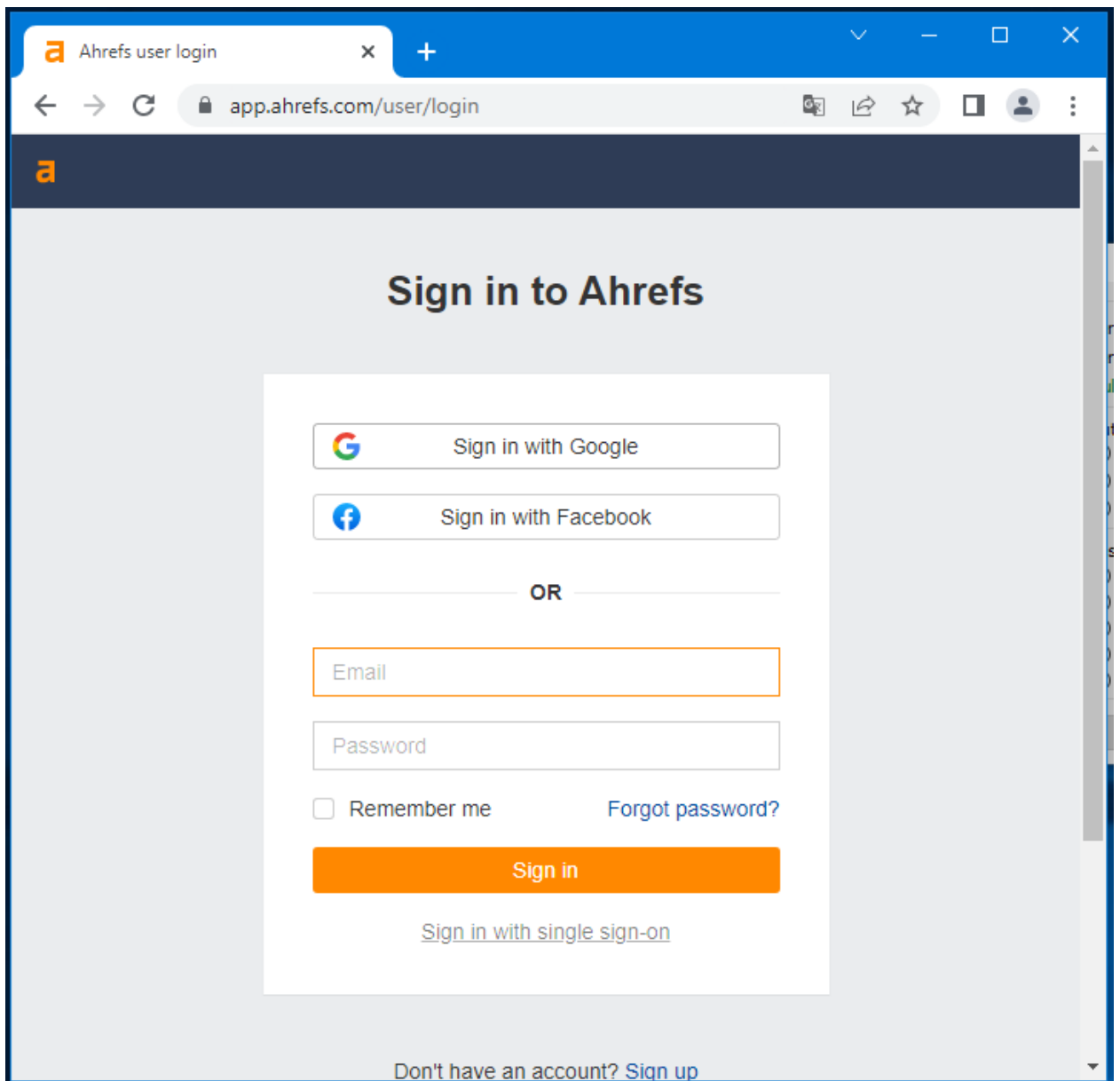


Рисунок 2.23 - Сторінка авторизації Ахрефс у «Google Chrome» [1]



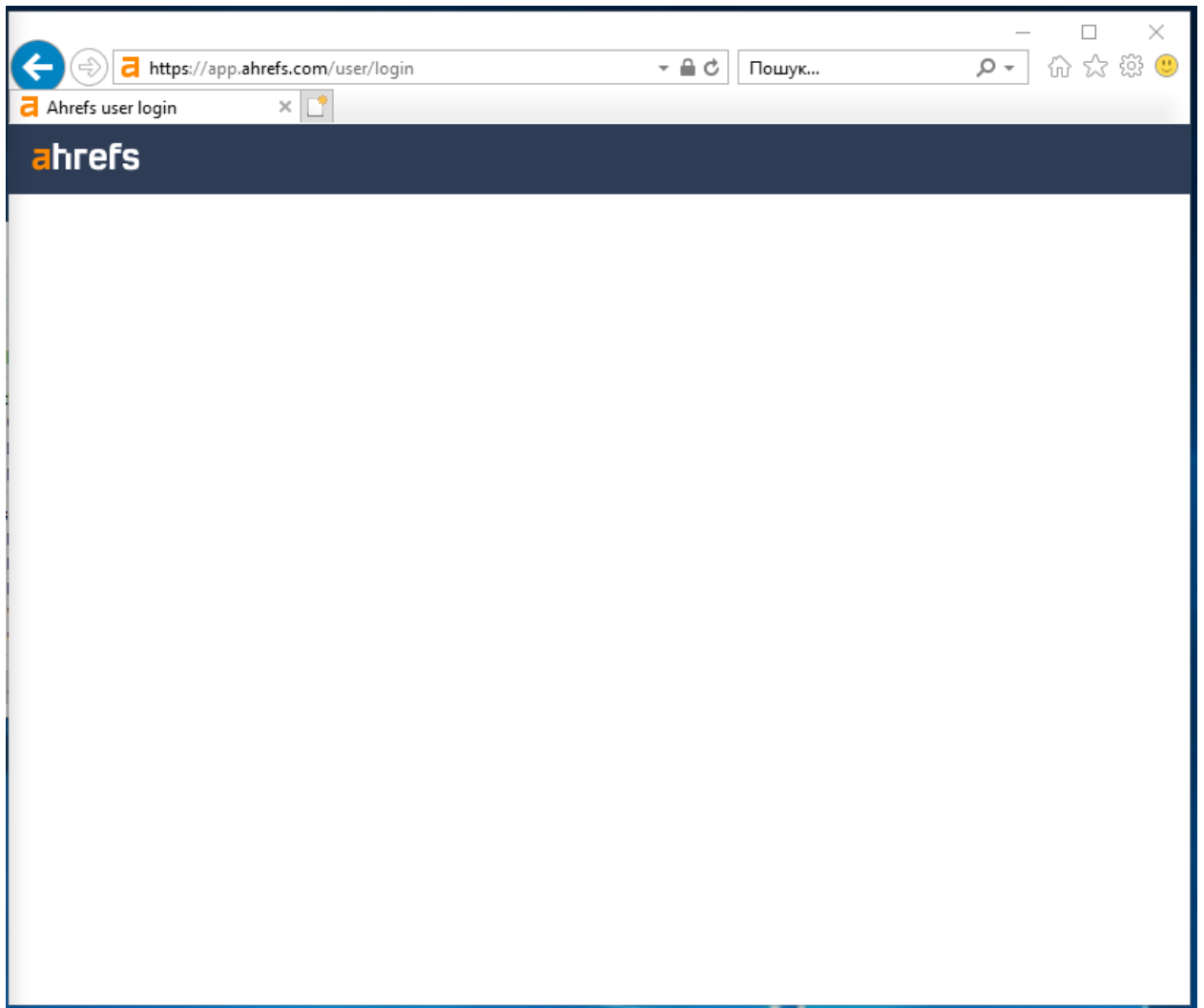


Рисунок 2.24 - Сторінка авторизації Ахрефс у «Internet Explorer» [1]

Як бачимо браузер «Internet Explorer» не підтримує роботу Java Script тому його використання не можливе.

Для командної роботи підійде браузер «Google Chrome», але для його запуску нам знадобиться міняти параметри запуску.

Так саме параметри запуску, їх можна міняти при запуску з командної строки (рис. 2.25) або дописуючи команду у ярлик (рис. 2.26).

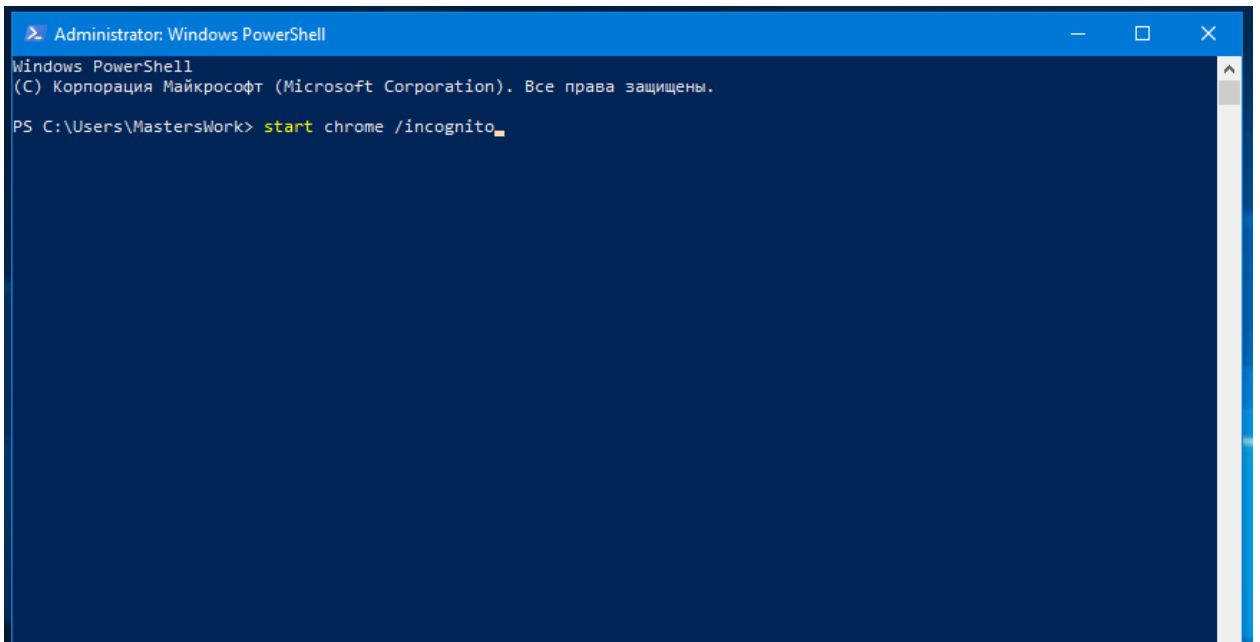


Рисунок 2.25 - Запуск «Google Chrome» в режиме інкогніто за допомогою команди в «PowerShell»

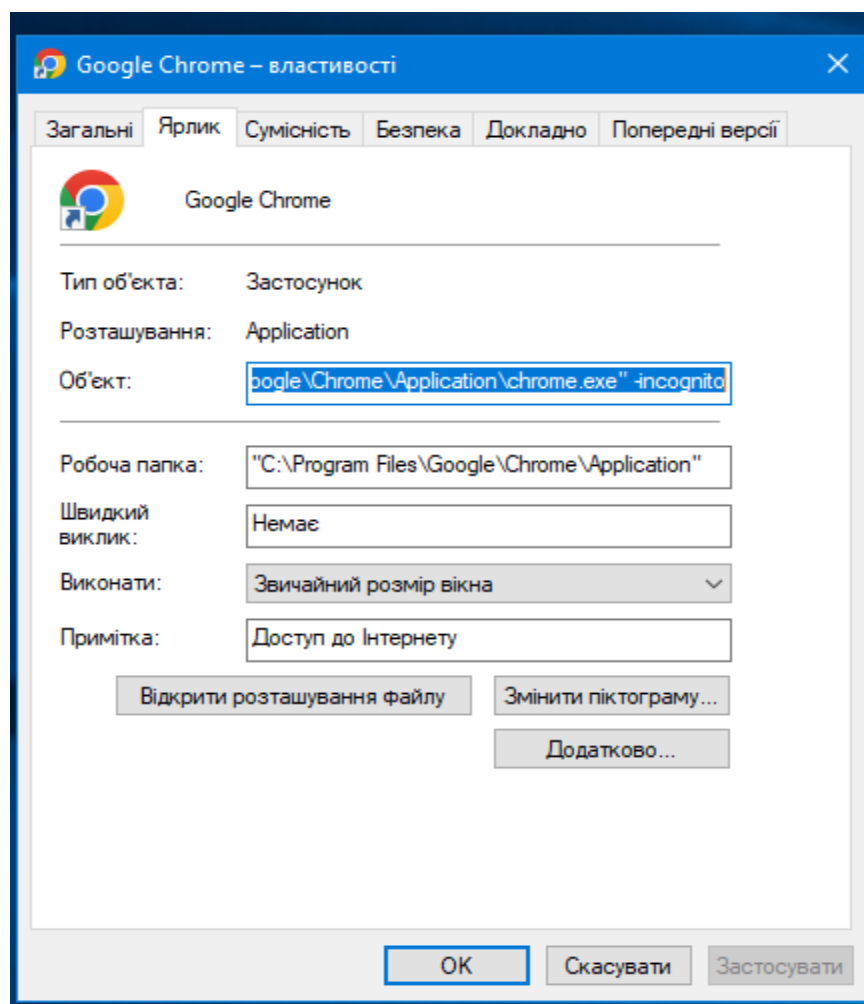


Рисунок 2.26 - Запуск «Google Chrome» в режимі інкогніто за допомогою доданого параметра в ярлик

Отже браузер не може запуснитися декілька разів тому що використовується одна і та сама папка з даними користувача, яка і зберігає всі дані користувача такі як cookies, сесії, тощо. Але для того щоб велика команда могла одночасно працювати нам потрібні ці дані, наприклад якщо один раз авторизувався в особистому кабінеті, то на всіх запуснених копіях ця авторизація повинна бути. Таким чином ніякі тригери які повинні бачити що сервісом користується не одна людина не будуть спрацьовувати, тому що, для сервісу це один комп'ютер зі своїм ідентифікатором, одна IP адреса, одна збірка браузеру і ідентифікатор, це як би всі одночасно просто сиділи за одним комп'ютером і робили.

Тож так як користувач комп'ютеру один і той самий то й папка з даними знаходиться за однією адресою це:

«%LOCALAPPDATA%\Google\Chrome\User Data»

Тому для запуску браузеру потрібно міняти цю папку за допомогою параметру запуску, а саме:

«--user-data-dir="user\_data"»

Де user\_data – це шлях до іншої папки користувача

Наприклад для першого користувача який запускає браузер запуск буде с параметром:

"C:\Program Files\Google\Chrome\Application\chrome.exe" --user-data-dir="C:\chromefolders\chromedata1"

Для другого:

"C:\Program Files\Google\Chrome\Application\chrome.exe" --user-data-dir="C:\chromefolders\chromedata2"

І так в подальшому, для кожного користувача, своя пака.

Але цю папку потрібно кожного разу копіювати з основної папки в якій як раз і зберігаються усі данні що користувач вже авторизований, також сесії та файли cookies щоб не потрібно було кожному користувачу окремо

авторизуватись, тим самим кожного разу міняючи активну сесію на сервері сервісу.

Для прикладу запусимо браузер та авторизуємось на порталі МІХ СумДУ (рис. 2.27) [20] том що цей портал зберігає сесію і після закриття браузеру не потрібно авторизуватись знову, а перед цим скопіюємо файли даних з користувацької папки в іншу, для того щоб в цій папці не було збережено активної сесії, та запусимо браузер з іншою папкою (рис. 2.28).

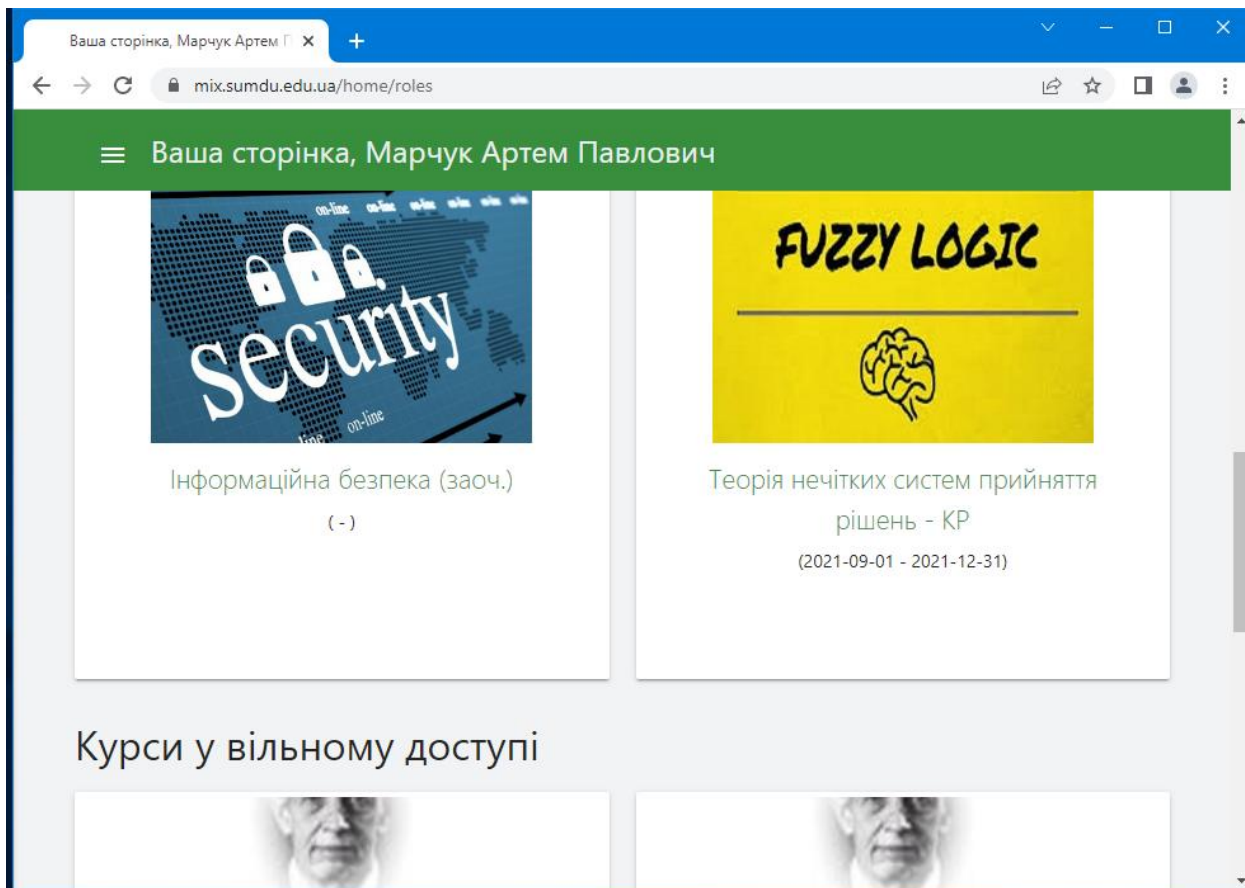


Рисунок 2.27 - головна сторінка порталу МІХ СумДУ при авторизації [20]

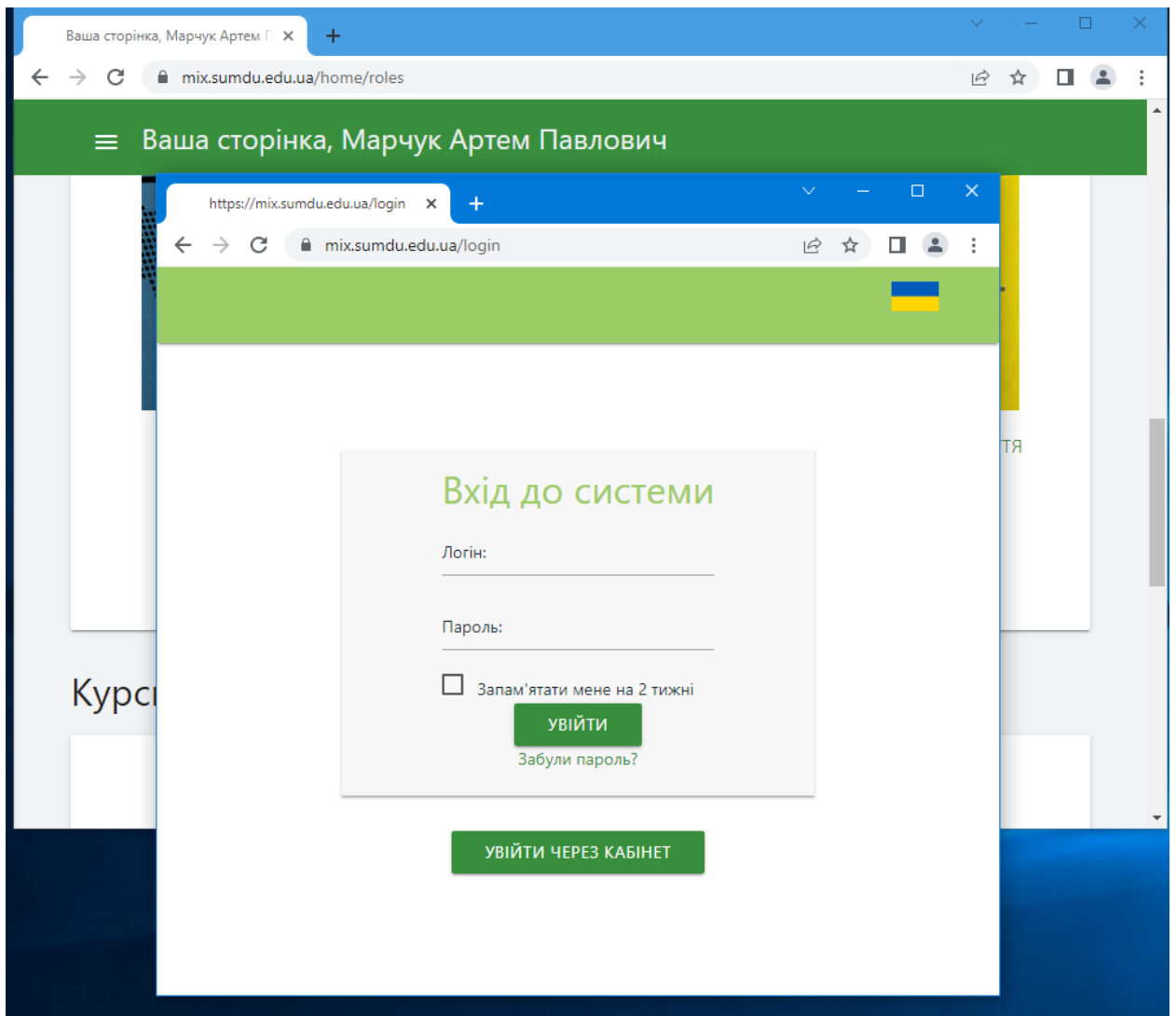


Рисунок 2.28 - Одночасний запуск іншої копії браузера зі зміненої користувацькою папкою [20]

Після запуску іншої копії зі зміненої користувацькою папкою, та скопіювавши і перейшовши за посиланням з авторизованого вже порталу у іншій копії браузера ми не можемо відразу туди зайти, бо в цій паці з даними не було активних сесій, і тому авторизуватись буде потрібно заново.

Але якщо закрити всі браузери та скопіювати після цього всі файли з папки даних користувача першого браузера, до папки другого, то при відкритті обох браузерів авторизація буде в обох (рис. 2.29), навіть розмір вікна який був під час закриття першого браузера буде у другого.

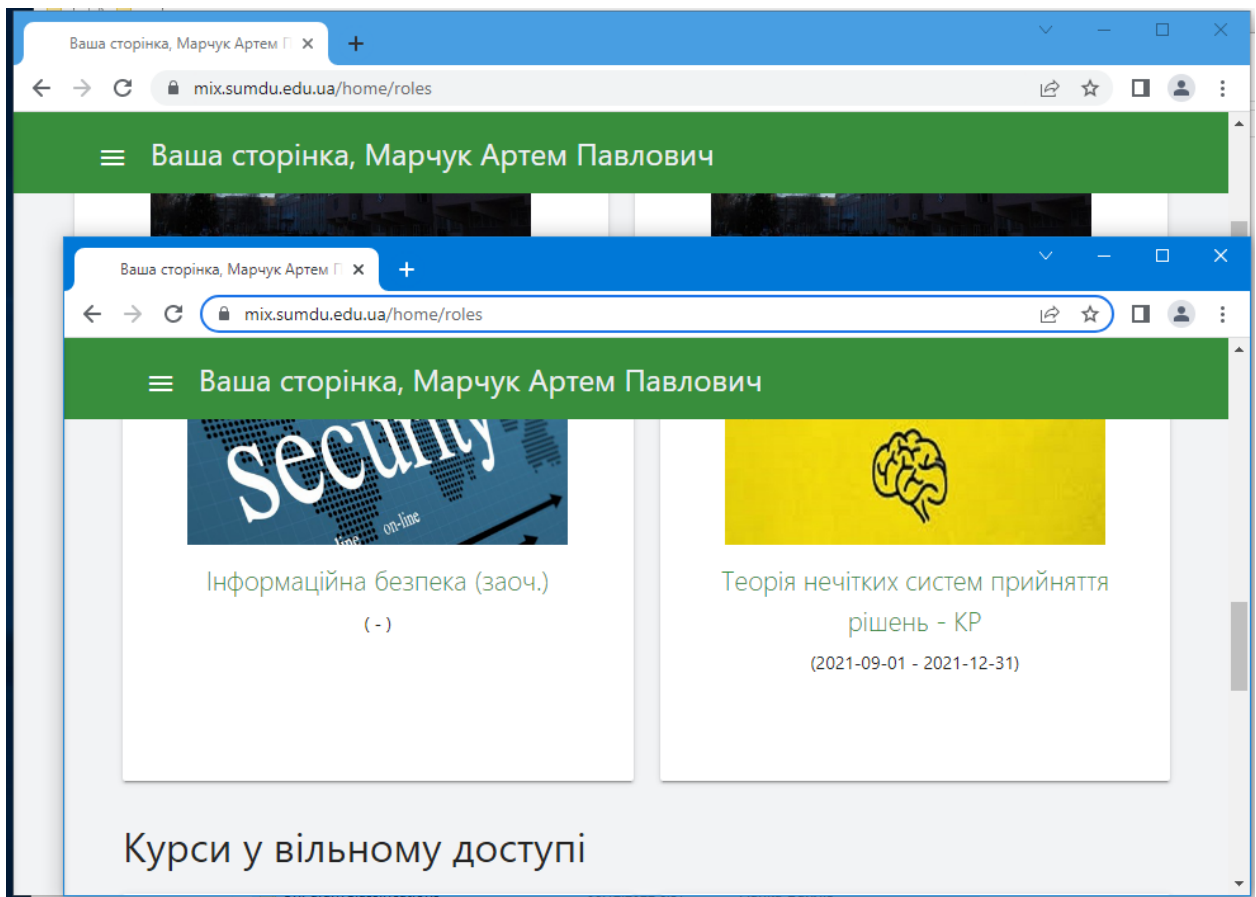


Рисунок 2.29 - Відкриття обох браузерів з різними папками користувацьких даних [20]

Таким чином буде працювати і на віддаленому робочому столі відкриття браузера з однаковою авторизацією с одним користувачем комп'ютера. Але для цього потрібно буде кожен раз копіювати основну папку з даними основного користувача до інших, до того ж змінювати параметри запуску для кожного. Тому потрібно автоматизувати цей процес.

Для автоматизації процесу на мою думку краще всього підійде написання скрипту на мові Python, бо для його використання достатньо буде встановити середу IDE безкоштовно (рис. 2.30) [21] та виконувати його через скрипт PowerShell, для якого можна зробити зручний ярлик.

The Python Programming Language

python™

Download Python 3.11.0

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [macOS](#), [Other](#)

Want to help test development versions of Python? [Prereleases](#), [Docker images](#)

Support Python in 2022! End of year fundraiser and membership drive are live now! [Join in today!](#)

Active Python Releases

For more information visit the [Python Developer's Guide](#).

Python version	Maintenance status	First released	End of support	Release schedule
3.11	bugfix	2022-10-24	2027-10	PEP 664
3.10	bugfix	2021-10-04	2026-10	PEP 619
3.9	security	2020-10-05	2025-10	PEP 596
3.8	security	2019-10-14	2024-10	PEP 569

Рисунок 2.30 - Сторінка завантаження Python IDE з офіційного сайту [21]

## 2.2 Проектування структури скрипту та вибір модулів

Для автоматизації процесу та зручності користування потрібно розробити скрипт на мові Python який буде виконувати такі дії:

- Вести запис в лог файл.
- Перевіряти папки з яким номером вже є.
- Копіювати данні з основної папки в нову.
- Запускати браузер з параметрами нової папки даних.
- Перевіряти активність користувача.
- Перевіряти чи є процес браузеру.
- Видаляти папку після закриття браузеру.
- Відключати сесію якщо користувач не активний.
- Написати скрипт в PowerShell для запуску Python скрипта.

Розберемо докладніше кожний пункт:

**Вести запис в лог файл** – це потрібно робити для подальшої зручності для виявлення проблем, якщо на якомусь етапі все піде не так як планувалось то можна буде це відстежити за допомогою перегляду логів.

**Перевіряти папки з яким номером вже є** – як було написано вище для запуску декількох копій браузеру потрібно змінювати папку з даними користувача, логічно ці папки називати якимось типу chrome1, chrome2, chrome3 і так далі, тому потрібно перед тим як створювати нову папку, сканувати її, перевіряти останній номер папки і створювати наступний за ним номер.

**Копіювати данні з основної папки в нову** – сесії мають термін дії, тобто періодично, потрібно буде заходити в браузер з основною папкою даних та авторизуватись, тому всі другорядні папки повинні брати дані з неї.

**Запускати браузер з параметрами нової папки даних** – коли папка з даними скопійована в неї буде свій шлях, потрібно цей шлях зберігати в змінній та при запуску браузеру прописувати актуальній параметр.

**Перевіряти активність користувача** – Користувач може просто згорнути вікно віддаленого комп'ютера, та займатись своїми справами, проте він все ще буде використовувати ресурси віддаленого комп'ютера такі як місце в оперативній пам'яті та навантажувати процесор, тому потрібно відслідковувати неактивних користувачів.

**Перевіряти чи є процес браузеру** – в кожного процесу є свій PID (Process identifier) тож потрібно при відкритті браузеру записувати в змінну PID та перевіряти його на активність.

**Видаляти папку після закриття браузеру** – після того як закривається браузер і перевірка бачить що такого процесу вже немає, потрібно видалити ту папку, яка була створена для нього, все ж таки ці папки та дані досить великі і якщо їх не видаляти то пам'ять на жорсткому диску врешті закінчиться і буде неможливо створювати нові папки та відкривати нові копії браузеру.



**Відключати сесію якщо користувач не активний** – як вже було описано, неактивні користувачі займають ресурси, тому їх потрібно відключати від віддаленого робочого столу для визволення ресурсів для інших користувачів.

**Написати скрипт в PowerShell для запуску Python скрипта** – найпростіше запустити скрипт написаний на Python через командну строку PowerShell, до того ж можна створити ярлик запуску цього скрипта.

Це всі необхідні нам функції скрипта для автоматизації та спрощення роботи користувачів.

Виходячи з необхідних функцій потрібно використати такі модулі:

- Pyautogui;
- sleep з бібліотеки time;
- argparse;
- os;
- shutil;
- signal;
- subprocess;
- logging;
- datetime.

Розглянемо докладніше кожен модуль:

**Pyautogui** – це модуль, яка дозволяє емулювати дії користувача як з клавіатурою (натискання клавіш, введення тексту, гарячі клавіші), так і мишкою (права/ліва кнопка, натискання, перенесення). А також за допомогою цієї бібліотеки можна перевіряти активність миші, та відслідковувати її переміщення, і якщо мишка не ворухиться та не змінює позицій якийсь час то це буде означати що користувач неактивний.

**Sleep з бібліотеки time** – сам модуль працює з часом, а функція потрібна для того щоб робити паузи у виконанні програми або функції

**Argparse** – цей модуль дозволяє розбирати аргументи, що передаються скрипту під час його запуску з командного рядка, і дає можливість користуватися цими аргументами у скрипті. Тобто йдеться про те, що цей модуль дозволяє надавати скрипту деякі дані в момент його запуску, а цими даними скрипт зможе скористатися під час виконання коду. Простими словами це засіб, за допомогою якого можна налагодити спілкування між автором програми та тим, хто їй користується. Використання `argparse` означає, що при необхідності змінити поведінку скрипта або при необхідності передачі йому деяких даних, якщо це передбачено, користувачеві не потрібно редагувати програмний код. В результаті в скрипті з'являється певний рівень гнучкості.

**os** – це бібліотека функцій для роботи з операційною системою. Методи, включені до неї дозволяють визначати тип операційної системи, отримувати доступ до змінних оточення, керувати директоріями та файлами.

**shutil** – цей модуль містить набір функцій високого рівня для обробки файлів, груп файлів, і папок. Зокрема, доступні тут функції дозволяють копіювати, переміщати та видаляти файли та папки.

**signal** – цей модуль надає механізми для використання обробників сигналів у Python. Сигнали – це функція операційної системи, яка дозволяє повідомляти програму про подію та обробляти її асинхронно. Вони можуть бути створені системою або відправлені з одного процесу в інший. Оскільки сигнали переривають регулярний потік програми, можливо, деякі операції, особливо введення/виведення, можуть призвести до помилок, якщо сигнал приймається посередині. Як і в інших формах програмування, заснованого на подіях, сигнали приймаються шляхом створення функції зворотного виклику, визваної обробником сигналу, що викликається у разі виникнення сигналу. Аргументами обробника сигналу є номер сигналу та кадр стека від точки у програмі, яка була перервана сигналом.

**subprocess** – цей модуль дає можливість запускати процеси програм із Python. Іншими словами, ви можете запускати програми та передавати їм аргументи за допомогою модуля subprocess.

**logging** – це модуль Python, який дозволяє настроїти логування зі скрипту. У модуля logging дуже багато можливостей і безліч варіантів налаштування.

**datetime** – цей модуль надає класи для обробки часу та дати різними способами. Підтримується і стандартний спосіб представлення часу, проте більший наголос зроблено на простоту маніпулювання датою, часом та його частинами.

Тож знаючи все необхідне перейдемо до практичної реалізації.

### 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

#### 3.1 Розробка інформаційної технології керування платними підписками інтернет-сервісів.

Почнемо створення інформаційної технології керування платними підписками інтернет-сервісів з налаштування віртуальної машини на базі Oracle VM VirtualBox (рис. 3.1) [22].

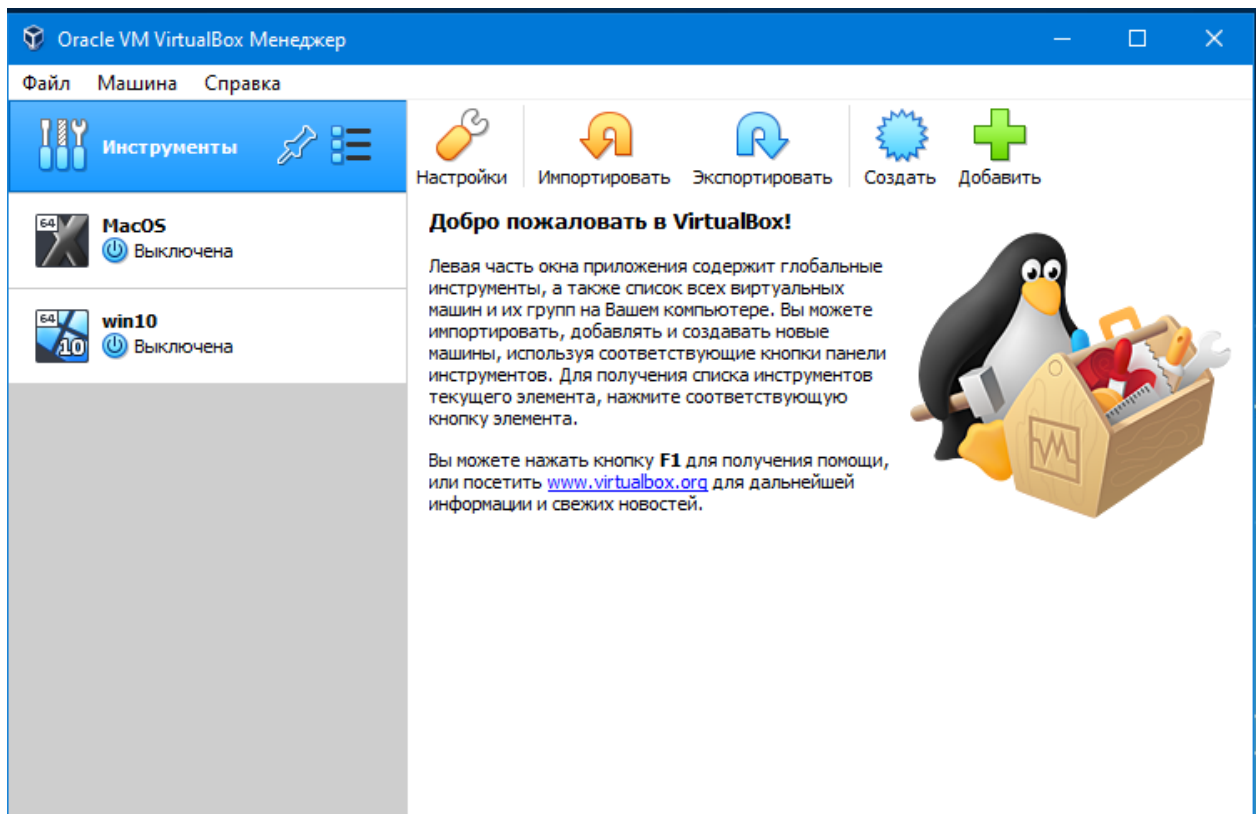
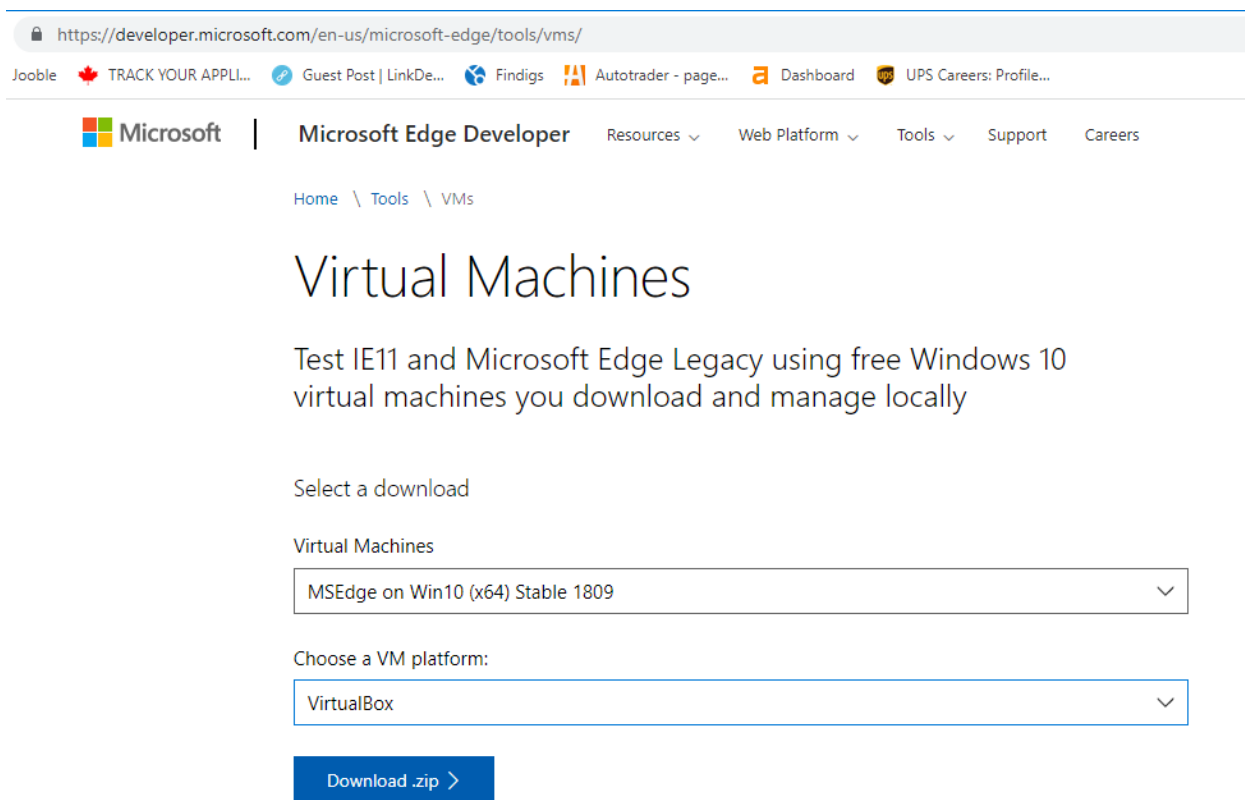


Рисунок 3.1 - Загальний вигляд додатка «Oracle VM VirtualBox» [22]

Після встановлення віртуальної машини потрібно завантажити з офіційного сайту Microsoft архів з готовою віртуальною машиною для VirtualBox на базі Windows 10 Enterprise, вона надається безкоштовно на 90 днів (рис. 3.2) [23].



The screenshot shows the Microsoft Edge Developer website. The browser address bar displays the URL: <https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/>. The page header includes the Microsoft logo and navigation links: Microsoft Edge Developer, Resources, Web Platform, Tools, Support, and Careers. Below the header, there are breadcrumb links: Home > Tools > VMs. The main heading is "Virtual Machines". The sub-heading reads: "Test IE11 and Microsoft Edge Legacy using free Windows 10 virtual machines you download and manage locally". Under the heading, there is a section titled "Select a download" with a dropdown menu labeled "Virtual Machines" showing the selected option "MSEdge on Win10 (x64) Stable 1809". Below this is a section titled "Choose a VM platform:" with a dropdown menu showing the selected option "VirtualBox". At the bottom of the form is a blue button labeled "Download .zip >".

Рисунок 3.2 - Вигляд сторінки завантаження архіву з готовою віртуальною машиною на базі Windows 10 [23]

Після завантаження, архів з віртуальною машиною розпаковуємо та додаємо до VirtualBox, після чого запускаємо віртуальну машину, встановлюємо Google Chrome. Також завантажуюмо з GitHub RDP Wrapper і також встановлюємо його (рис. 3.3).

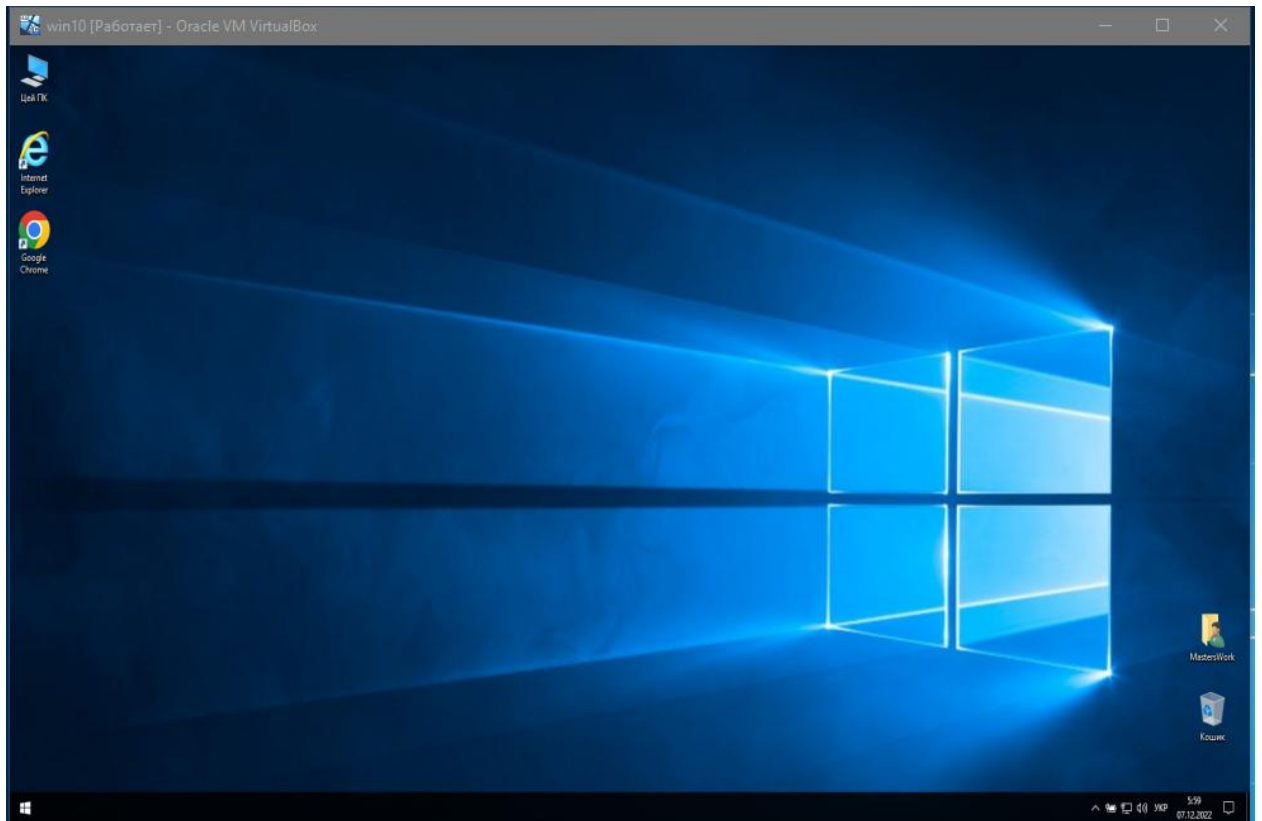


Рисунок 3.3 - Загальний вигляд запущеної віртуальної машини на базі Windows 10

Налаштовуємо RDP Wrapper таким чином щоб до віддаленого робочого столу можна було підключатись від одним і тим же користувачем, для цього запускаємо RDPConf та знімаємо галочку з пункту Single session per user (рис. 3.4) всі інші налаштування не змінюємо.

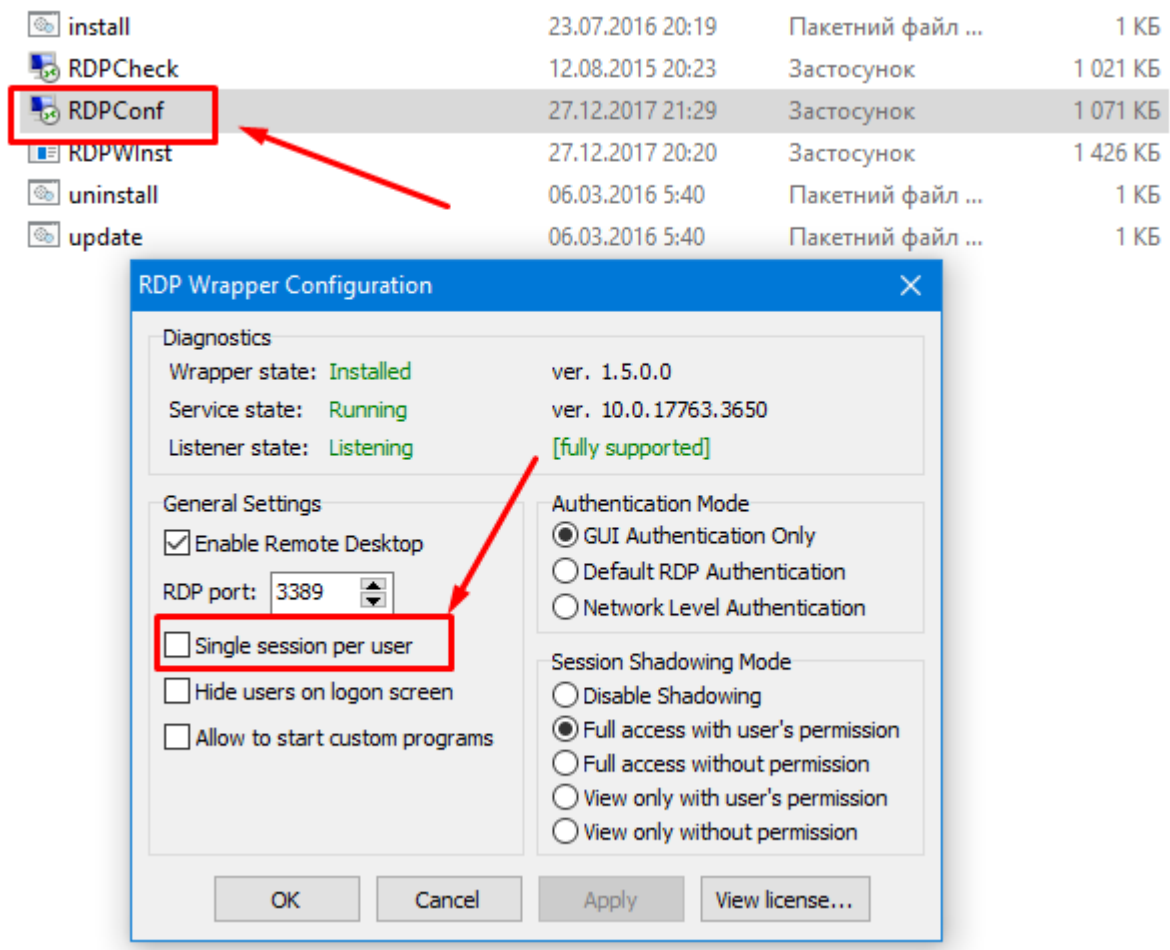


Рисунок 3.4 - Діалогове вікно налаштувань RDP Wrapper

Тепер перевіримо чи працює підключення до віддаленого робочого столу для декількох підключень і чи можемо ми зайти під одним користувачем, для цього можна використати додаток RDPCheck, який знаходиться в папці з завантаженими файлами RDP Wrapper (рис. 3.5).

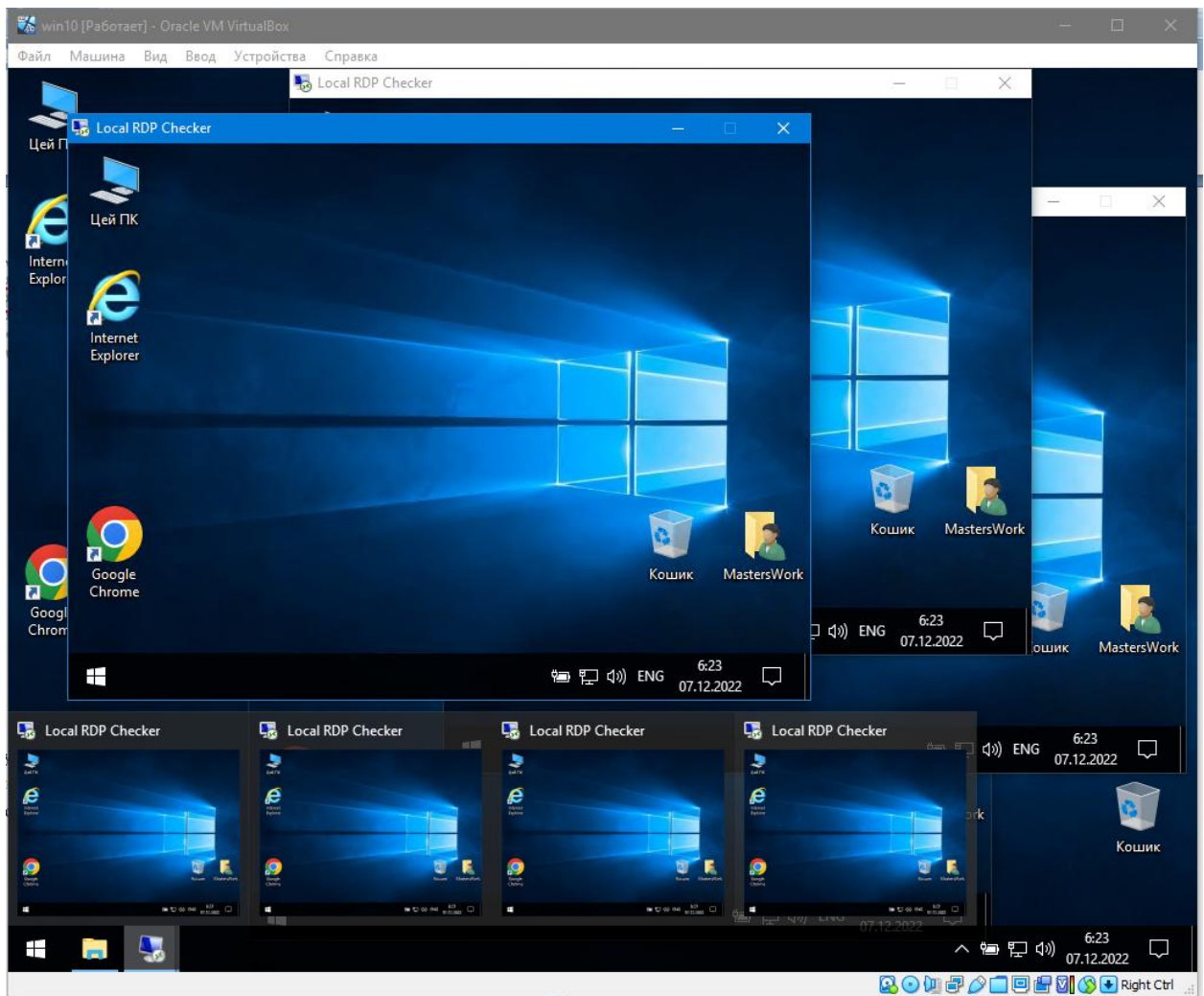


Рисунок 3.5 - Запуск чотирьох сеансів віддаленого робочого столу під одним користувачем

Також для перевірки чи дійсно всі віддалені сеанси працюють під одним користувачем потрібно запустити диспетчер задач, та перейти на вкладку «Користувачі» (рис. 3.6), де ми побачимо всіх користувачів які на даний момент авторизовані у системі.



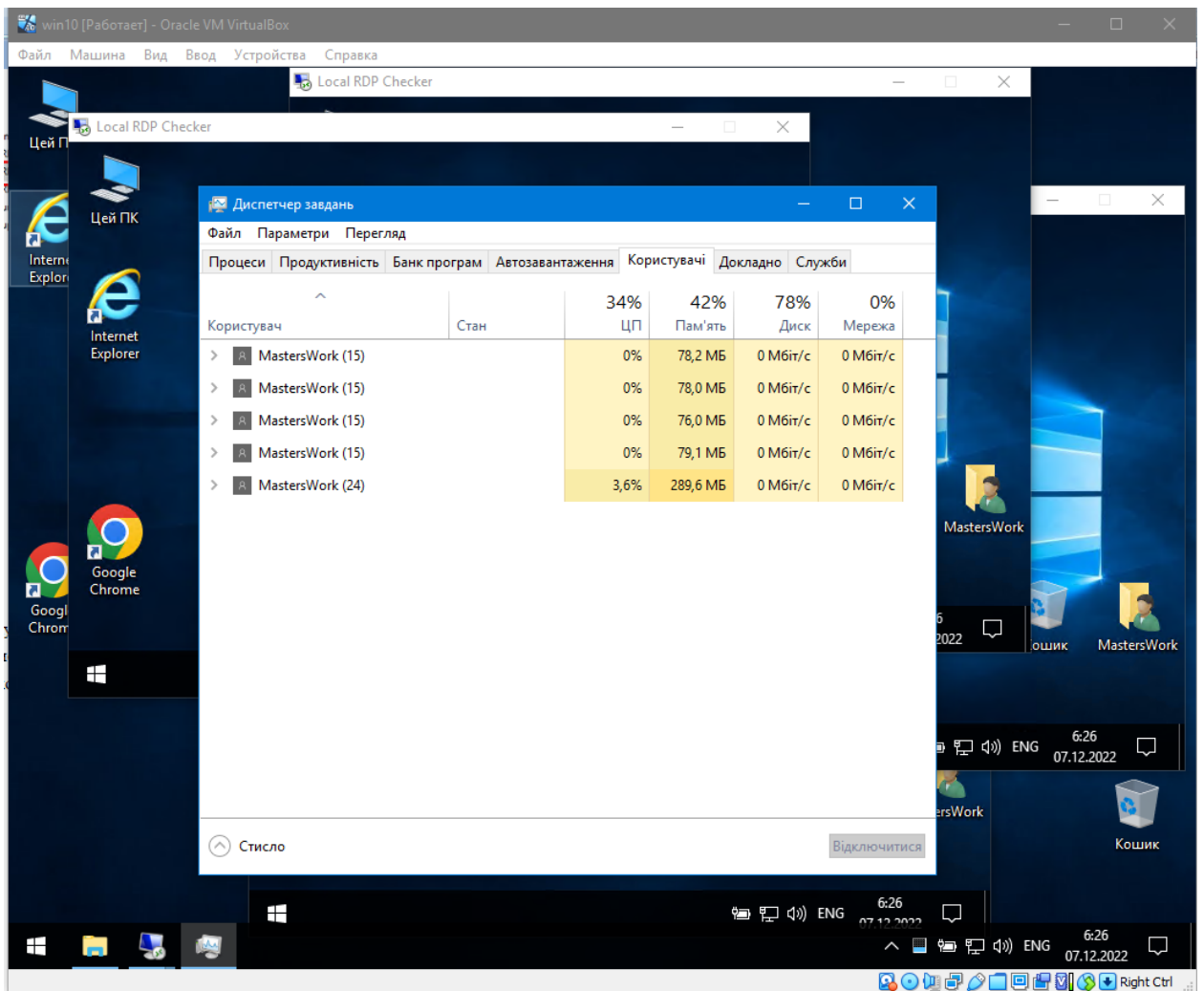
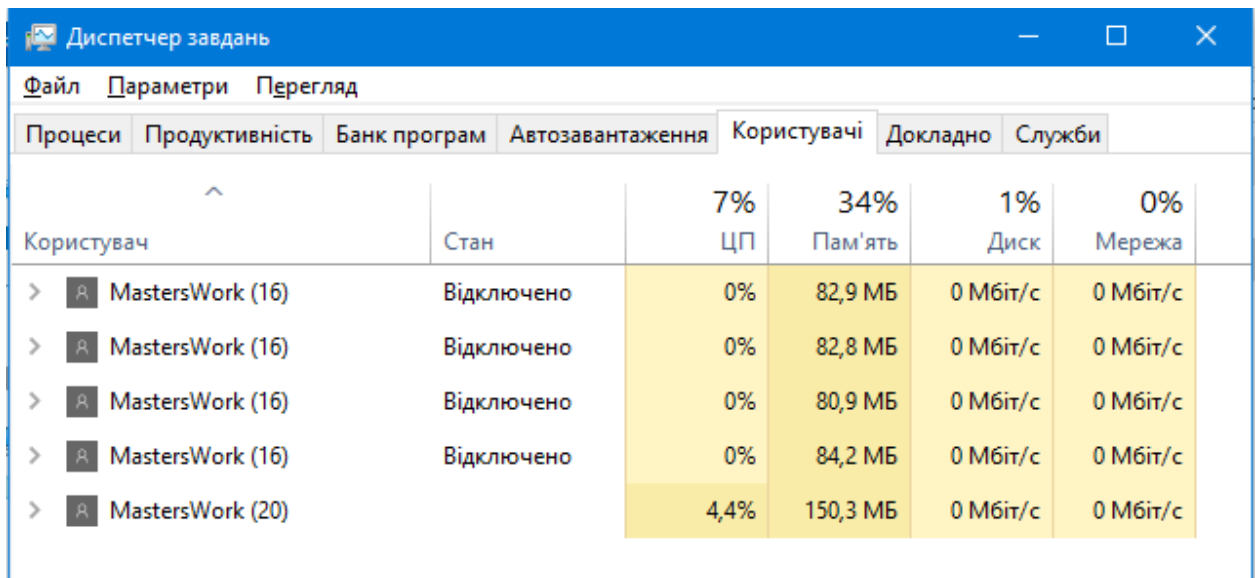


Рисунок 3.6 - Запущений диспетчер задач на віртуальній машині

На цій вкладці ми бачимо, що одночасно авторизований один і той же користувач, але п'ять разів, також можна побачити скільки програм та служб запущено кожним з користувачів і скільки вони використовують оперативної пам'яті, також якщо просто закрити вікна віддаленого підключення та не вийти з системи ми все одно будемо бачити ці авторизації і вони будуть використовувати деякі ресурси, але будуть з позначкою «відключено» (рис. 3.7).



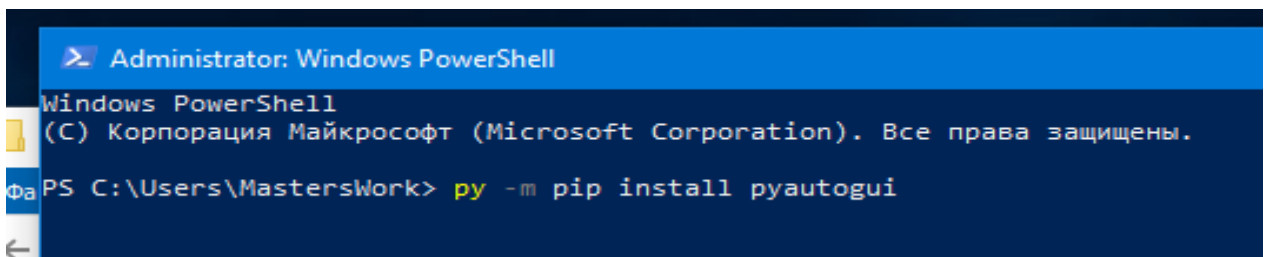
Користувач	Стан	7% ЦП	34% Пам'ять	1% Диск	0% Мережа
> MastersWork (16)	Відключено	0%	82,9 МБ	0 Мбіт/с	0 Мбіт/с
> MastersWork (16)	Відключено	0%	82,8 МБ	0 Мбіт/с	0 Мбіт/с
> MastersWork (16)	Відключено	0%	80,9 МБ	0 Мбіт/с	0 Мбіт/с
> MastersWork (16)	Відключено	0%	84,2 МБ	0 Мбіт/с	0 Мбіт/с
> MastersWork (20)		4,4%	150,3 МБ	0 Мбіт/с	0 Мбіт/с

Рисунок 3.7 - Запущений диспетчер задач на віртуальній машині

Тож перша частина виконана, і цієї системи вже достатньо для користування великою командою локальних програм.

Для подальшого виконання поставленої мети, встановлюємо середовище Python, яке можна скачати з офіційного сайту. Під час встановлення обов'язково потрібно вибрати допис у директорію PATH, це потрібно для того щоб в PowerShell знав коли звертаються саме до Python, для цього під час встановлення потрібно вибрати пункт «додати Python до змінних середовища» після цього потрібно перезавантажити віртуальну машину.

Також після завантаження віртуальної машини нам потрібно встановити модуль `ruautogui`. Для його встановлення потрібно запустити PowerShell та запустити його встановлення командою як відображено на рис. 3.8.



```

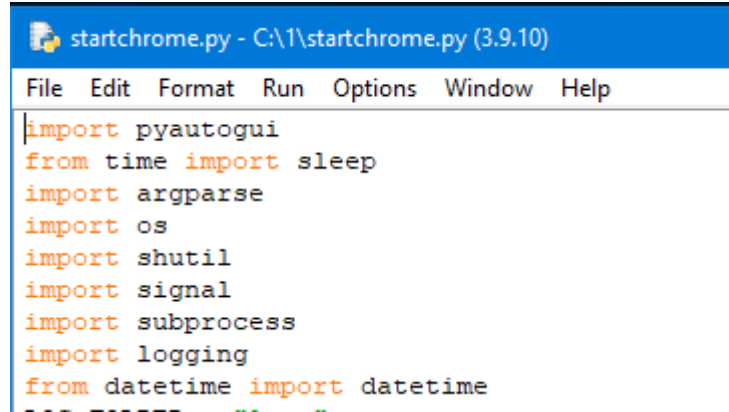
Administrator: Windows PowerShell
Windows PowerShell
(C) Корпорація Майкрософт (Microsoft Corporation). Все права захищені.
PS C:\Users\MastersWork> py -m pip install ruautogui

```

Рисунок 3.8 - Встановлення модуля для Python за допомогою PowerShell

Тепер за допомогою Python IDLE розробляємо скрипт згідно вимог, які були поставлені на етапі проектування.

Підключаємо необхідні модулі (рис. 3.9).



```
startchrome.py - C:\1\startchrome.py (3.9.10)
File Edit Format Run Options Window Help
import pyautogui
from time import sleep
import argparse
import os
import shutil
import signal
import subprocess
import logging
from datetime import datetime
-----
```

Рисунок 3.9 - Додавання необхідних модулів

Пишемо основні функції (рис. 3.10, рис. 3.11, рис. 3.12, рис. 3.13).

```
LOG_FOLDER = "logs"
if LOG_FOLDER not in os.listdir(os.path.dirname(__file__)):
    os.mkdir(os.path.join(os.path.dirname(__file__), LOG_FOLDER))

logging.basicConfig(filename=os.path.join(os.path.dirname(__file__), LOG_FOLDER,
                                         f'{datetime.today().date()}.txt'),
                    level=logging.INFO,
                    format='[% (asctime)s] %(levelname)s\t: %(message)s',
                    datefmt='%d.%m.%Y-%H:%M:%S')
```

Рисунок 3.10 - Зміст коду для створення логів

```

def copy_directory():
    source_dir = r"C:\Users\MastersWork\AppData\Local\Google\Chrome\User Data"
    destination_dir = r"C:\chromefolders\"

    print("Copy directory...")
    logging.info("Copy directory...")

    try:
        needed_directory_list = os.listdir(destination_dir)
        logging.info(needed_directory_list)
        needed_directory_list = list(map(lambda dir_name: dir_name.replace('chrome', ''), needed_directory_list))
        logging.info(needed_directory_list)
        last_dir_index = max(list(map(int, needed_directory_list))) |
        logging.info(last_dir_index)
    except ValueError as VE:
        logging.error(f'value error {VE}')
        last_dir_index = 0

    print("Last directory number ", last_dir_index)
    print("New directory number ", last_dir_index+1)

    current_dir_index = destination_dir + 'chrome' + str(last_dir_index + 1)
    logging.info(f'current dir_index {current_dir_index}')

    print('Copy...')
    shutil.copytree(source_dir, current_dir_index)
    logging.info(f'copy dir done {source_dir} {current_dir_index}')

    return current_dir_index

```

Рисунок 3.11 - Зміст функції копіювання папки з даними, та перевірка тих папок що вже є для визначення індексу та назви нової папки

```

def run_browser(path_to_browser, user_data):
    browser_run = f"{path_to_browser} --user-data-dir={user_data}"
    logging.info(browser_run)
    browser = subprocess.Popen(browser_run)
    logging.info('open browser chrome completed')
    return browser

```

Рисунок 3.12 - Зміст функції запуску браузера з додаванням параметрів запуску

```

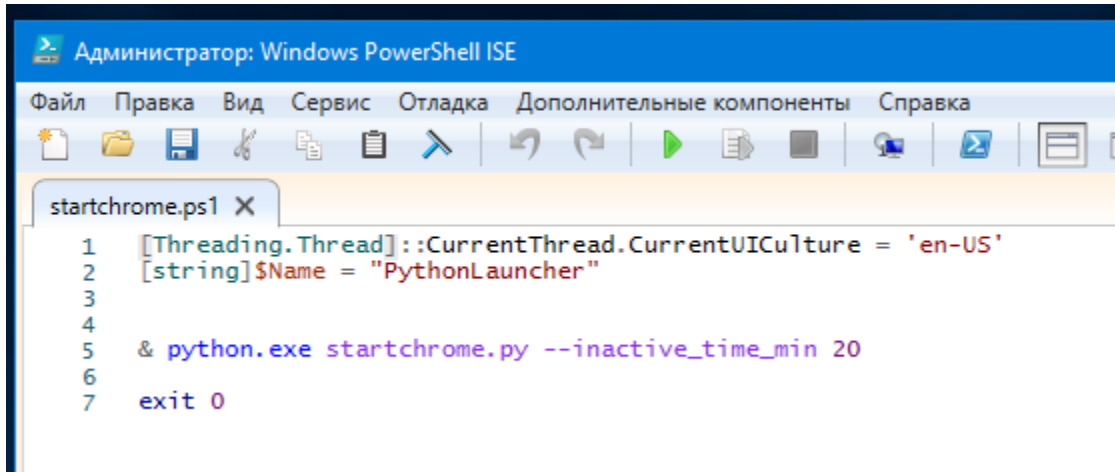
def check_pid(pid):
    try:
        os.kill(pid, signal.SIGTERM)
        print('browser close')
        logging.info(f'os.kill {pid} {signal.SIGTERM}')
    except OSError:
        return False
    except Exception as e:
        logging.error(e)
        print(e)
        print('Browser PID not found')
    else:
        return True

```

Рисунок 3.13 - Зміст функції закриття браузеру

Повний код скрипту додається у Додатку А.

Після написання скрипту для автоматизації роботи нам потрібно зробити невеликий скрипт для його запуску у PowerShell, для цього запускаємо PowerShell ISE та пишемо скрипт (рис. 3.14).



```
1 [Threading.Thread]::CurrentThread.CurrentUICulture = 'en-US'
2 [string]$Name = "PythonLauncher"
3
4
5 & python.exe startchrome.py --inactive_time_min 20
6
7 exit 0
```

Рисунок 3.14 - Код скрипту для запуску скрипта на мові Python

Після цього створюємо ярлик для запуску PowerShell з нашим скриптом, який буде запускати основний скрипт на мові Python, також для зручності міняємо іконку ярлику на іконку Google Chrome (рис. 3.15).

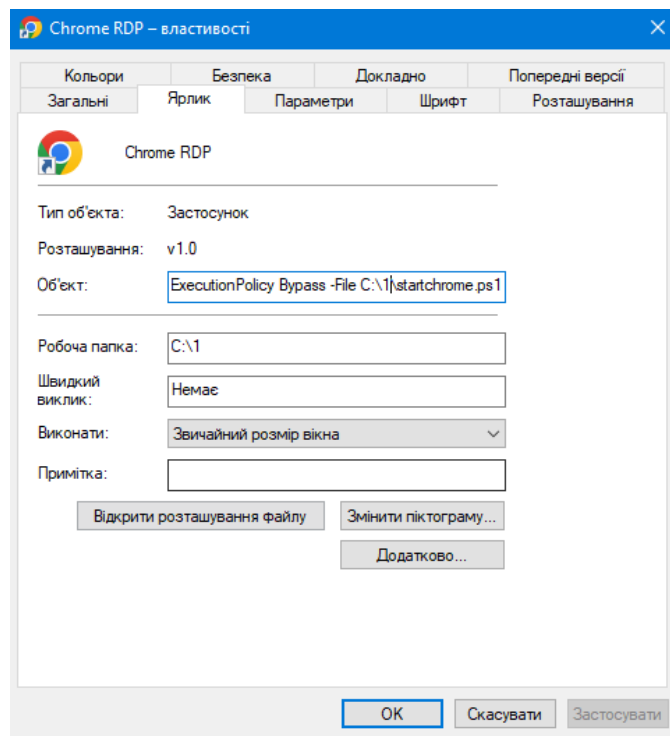


Рисунок 3.15 - Ярлик запуску скрипту PowerShell

Це був заключний етап розробки, тепер можна переходити до тестування.

### 3.2 Тестування інформаційної системи

Для тестування інформаційної системи заїдемо до аканту на МІХ СумДУ [20] з основного браузеру, потім запусимо п'ять сесій віддаленого робочого столу та випробуємо скрипт (рис. 3.16).

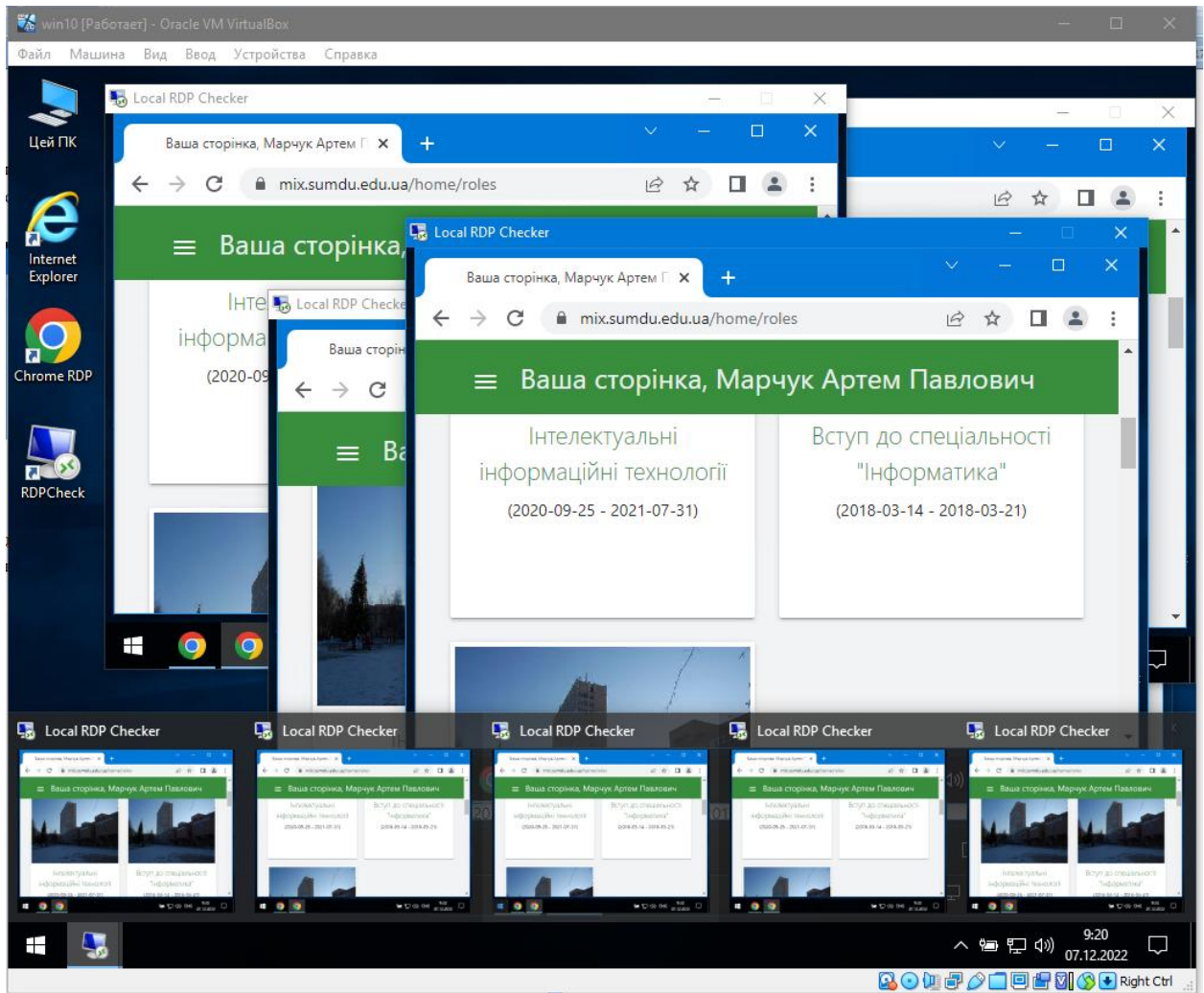


Рисунок 3.16 - Тестування роботи скрипта на п'яти сесіях віддаленого робочого столу [20]

Як бачимо інформаційна система ідеально працює, у кожного користувача відкривається один і той самий браузер, до того ж всі користувацькі данні копіюються, cookies та відкриті сесії. Кожен користувач може робити свою роботу нікому не заважаючи, кількість користувачів обмежена тільки тим на скільки вистачить ресурсів комп'ютера, сервера або

віртуальної машини. Папки з користувацькими даними створюються у правильному порядку (рис. 3.17).

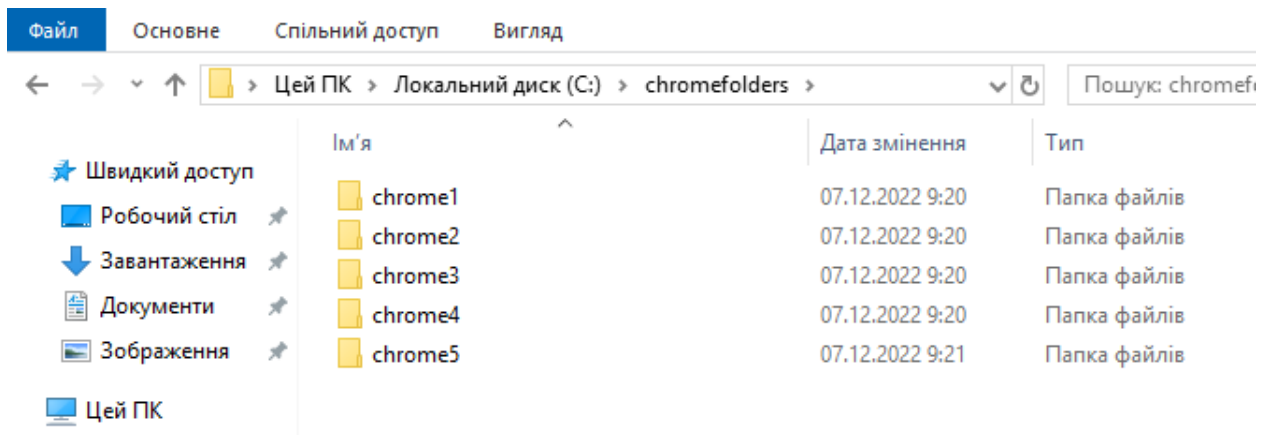


Рисунок 3.17 - Структура створення папок

Також створюються логи роботи скрипту (рис. 3.18).

```

-09:18:50] INFO      : C:\Program Files\Google\Chrome\Application\chrome.exe --user-data-dir=C:\chromefolders\
-09:18:50] INFO      : open browser chrome completed
-09:19:04] INFO      : Copy directory...
-09:19:04] INFO      : ['chrome1', 'chrome2']
-09:19:04] INFO      : ['1', '2']
-09:19:04] INFO      : 2
-09:19:04] INFO      : current dir_index C:\chromefolders\chrome3
-09:19:06] INFO      : copy dir done C:\Users\MastersWork\AppData\Local\Google\Chrome\User Data C:\chromefolder
-09:19:06] INFO      : C:\Program Files\Google\Chrome\Application\chrome.exe --user-data-dir=C:\chromefolders\
-09:19:06] INFO      : open browser chrome completed
-09:19:21] INFO      : Copy directory...
-09:19:21] INFO      : ['chrome1', 'chrome2', 'chrome3']
-09:19:21] INFO      : ['1', '2', '3']
-09:19:21] INFO      : 3
-09:19:21] INFO      : current dir_index C:\chromefolders\chrome4
-09:19:22] INFO      : copy dir done C:\Users\MastersWork\AppData\Local\Google\Chrome\User Data C:\chromefolder
-09:19:22] INFO      : C:\Program Files\Google\Chrome\Application\chrome.exe --user-data-dir=C:\chromefolders\
-09:19:22] INFO      : open browser chrome completed
-09:19:36] INFO      : Copy directory...
-09:19:36] INFO      : ['chrome1', 'chrome2', 'chrome3', 'chrome4']
-09:19:36] INFO      : ['1', '2', '3', '4']
-09:19:36] INFO      : 4
-09:19:36] INFO      : current dir_index C:\chromefolders\chrome5
-09:19:39] INFO      : copy dir done C:\Users\MastersWork\AppData\Local\Google\Chrome\User Data C:\chromefolder
-09:19:39] INFO      : C:\Program Files\Google\Chrome\Application\chrome.exe --user-data-dir=C:\chromefolders\
-09:19:39] INFO      : open browser chrome completed

```

Рисунок 3.18 - Приклад логу скрипта який створюється

Помилки при використанні системи не виникало, всі браузери в усіх сесіях працюють стабільно, програми запуснені в цих сесіях відкриваються та також працюють стабільно. Тож рахую що інформаційна система повністю протестована та готова до роботи з великою командою.

## ВИСНОВКИ

Для успішної роботи компанії або бізнесу існує дуже багато корисних програм або сервісів які покращують якість роботи чи аналізу. Але при розширенні компанії росте й кількість співробітників, і далеко не у всіх в компаній є можливість забезпечити їх необхідною кількістю цих інструментів, тарифи на ліцензії дуже високі а умови використання в більшості випадків не дають можливості використання цих інструментів одночасно всіма співробітниками.

У кваліфікаційній магістерській роботі проведена така робота:

- Досліджена актуальність проблеми та проаналізовані існуючі інструменти.
- Сформульована мета та поставлені задачі, визначені етапи реалізації проєкту.
- Було вибрано технології та інструменти необхідні для розробки.
- Була розроблена інформаційна технологія керування платними підписками інтернет-сервісів.
- Після реалізації проєкту було проведено тестування роботи інформаційної системи.

Інформаційна система реалізована в повному обсязі та відповідно до усіх заданих норм і вимог. Усі поставлені завдання виконані.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ahrefs - SEO Tools & Resources To Grow Your Search Traffic [Електронний ресурс] // ahrefs.com. – 2022. – Режим доступу до ресурсу: <https://ahrefs.com/>.
2. Програмне забезпечення для автоматизації бізнесу Microinvest [Електронний ресурс] // microinvest.ua. – 2022. – Режим доступу до ресурсу: <https://microinvest.ua/ua/index.html>.
3. Remotedesktopsoftware [Електронний ресурс] // wikibrief.org. – 2021. – Режим доступу до ресурсу: [https://de.wikibrief.org/wiki/Remote\\_desktop\\_software](https://de.wikibrief.org/wiki/Remote_desktop_software).
4. Apple Remote Desktop [Електронний ресурс] // apple.com. – 2022. – Режим доступу до ресурсу: <https://apps.apple.com/app/id409907375>.
5. ZCentral Remote Boost | HP® Official Site [Електронний ресурс] // hp.com. – 2022. – Режим доступу до ресурсу: <https://www.hp.com/us-en/workstations/zcentral-remote-boost.html>.
6. What is Independent Computing Architecture (ICA)? [Електронний ресурс] // Dominance Software. – 2005. – Режим доступу до ресурсу: <http://www.domsoft.com/citrix/ica.htm>.
7. The Show Must Go On - How HP ZCentral Remote Boost Made Remote Post-Production a Reality During a Global Pandemic [Електронний ресурс] // hp.com. – 2020. – Режим доступу до ресурсу: <https://press.hp.com/us/en/blogs/2020/show-must-go-on-hp-zcentral.html>.
8. Understanding the Remote Desktop Protocol (RDP) [Електронний ресурс] // microsoft.com. – 2021. – Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/troubleshoot/windows-server/remote/understanding-remote-desktop-protocol>.
9. [MS-RDPBCGR]: Remote Desktop Protocol: Basic Connectivity and Graphics Remoting [Електронний ресурс] // microsoft.com. – 2022. – Режим доступу до ресурсу: <https://learn.microsoft.com/en->

- [us/openspecs/windows\\_protocols/ms-rdpbcgr/5073f4ed-1e93-45e1-b039-6e30c385867c](https://us.openspecs/windows_protocols/ms-rdpbcgr/5073f4ed-1e93-45e1-b039-6e30c385867c).
10. Fitzpatrick D. RDP Security Explained [Электронный ресурс] // microsoft.com. – 2019. – Режим доступа до ресурсу: <https://www.mcafee.com/blogs/other-blogs/mcafee-labs/rdp-security-explained/>.
  11. Desktop Operating System Market Share Worldwide [Электронный ресурс] // statcounter.com. – 2022. – Режим доступа до ресурсу: <https://gs.statcounter.com/os-market-share/desktop/worldwide>.
  12. Motylkov S. RDP Wrapper Library by Stas'M [Электронный ресурс] // github.com. – 2022. – Режим доступа до ресурсу: <https://github.com/stascorp/rdpwrap/>.
  13. Mushad U. Octo browser Review [Электронный ресурс] // www.theadreview.com. – 2022. – Режим доступа до ресурсу: <https://www.theadreview.com/octo-browser-review/>.
  14. Molokov K. Virtual Machine vs. Anti-Detect Browser [Электронный ресурс] // gologin.com. – 2021. – Режим доступа до ресурсу: <https://gologin.com/virtual-machine-vs-anti-detect-browser>.
  15. Best Antidetect Browser for Fingerprint Spoofing [Электронный ресурс] // proxysp.com. – 2022. – Режим доступа до ресурсу: <https://www.proxysp.com/best-antidetect-browser/>.
  16. Kim C. Design and Implementation Smart Office System Based on Remote Desktop Protocol (RDP) // The Journal of the Institute of Webcasting, Internet and Telecommunication. 2014. № 2 (14). С. 153–159.
  17. Lu J., Zhou W., Liu Y. Remote Access Technology of Hard Disk Based on Embedded System // 2021 International Conference on Electronic Information Engineering and Computer Science, EIECS 2021. 2021. С. 157–163.

18. Salian D. (Divyashree) [и др.]. Workplace Collaboration Using Remote Desktop // International Journal of Engineering and Applied Sciences. 2015. № 3 (2). С. 257973.
19. Yang H.-K. Security Analysis of a Secure Dynamic ID based Remote User Authentication Scheme for Multi-server Environment // The Journal of the Institute of Webcasting, Internet and Telecommunication. 2013. № 1 (13). С. 273–278.
20. eLearning@SumyStateUniversity [Электронный ресурс] // СумДУ. – 2022. – Режим доступа до ресурсу: <https://mix.sumdu.edu.ua/>.
21. Welcome to Python.org [Электронный ресурс] // Python Software Foundation. – 2022. – Режим доступа до ресурсу: <https://www.python.org/>.
22. Oracle VM VirtualBox [Электронный ресурс] // ORACLE. – 2022. – Режим доступа до ресурсу: <https://www.virtualbox.org/>.
23. Virtual Machines - Microsoft Edge Developer [Электронный ресурс] // microsoft.com. – 2022. – Режим доступа до ресурсу: <https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/>.

## Додаток А

### Лістинг ПРОГРАМНОГО КОДУ

```
import pyautogui

from time import sleep

import argparse

import os

import shutil

import signal

import subprocess

import logging

from datetime import datetime

LOG_FOLDER = "logs"

if LOG_FOLDER not in os.listdir(os.path.dirname(__file__)):

    os.mkdir(os.path.join(os.path.dirname(__file__), LOG_FOLDER))

logging.basicConfig(filename=os.path.join(os.path.dirname(__file__),

LOG_FOLDER,

                    f'{datetime.today().date()}.txt'),

                    level=logging.INFO,

                    format='[%(asctime)s] %(levelname)s\t: %(message)s',

                    datefmt='%d.%m.%Y-%H:%M:%S')
```

```
BROWSER_PATH = "C:\Program Files\Google\Chrome\Application\chrome.exe"

def delete_and_move_dir(path_to_move):

    del_dir = r"C:\chrome100"

    try:

        if os.path.exists(del_dir):

            try:

                os.remove(del_dir)

                logging.info(f"os.remove({del_dir})")

            except Exception as e:

                logging.error(f"os remove error {e}')

                shutil.rmtree(del_dir)

        else:

            print("The file|directory does not exist")

            logging.info(f"The file|directory does not exist {del_dir}")

    except Exception as e:

        logging.error(f"Dir not deleted {e}')

        print(e)

        print('Dir not deleted')

    try:
```

```
shutil.move(path_to_move, del_dir)

logging.info(f'move folder {path_to_move} {del_dir}')
```

```
except Exception as e:
```

```
    print(e)

    logging.error(f'Cant move directory {e}')

    print('Cant move directory')
```

```
def shutdown():
```

```
    logging.info('shutdown session')

    cmd = 'shutdown /l'

    os.system(cmd)
```

```
def run_browser(path_to_browser, user_data):
```

```
    browser_run = f'{path_to_browser} --user-data-dir={user_data}'

    logging.info(browser_run)

    browser = subprocess.Popen(browser_run)

    logging.info('open browser chrome completed')

    return browser
```

```
def copy_directory():
```

```
source_dir = r"C:\Users\MastersWork\AppData\Local\Google\Chrome\User
Data"

destination_dir = r"C:\chromefolders\\"

print("Copy directory...")

logging.info("Copy directory...")

try:

    needed_directory_list = os.listdir(destination_dir)

    logging.info(needed_directory_list)

    needed_directory_list = list(map(lambda dir_name:
dir_name.replace('chrome', ''), needed_directory_list))

    logging.info(needed_directory_list)

    last_dir_index = max(list(map(int, needed_directory_list))) # take last int in
filename: [chrome4, chrome5]

    logging.info(last_dir_index)

except ValueError as VE:

    logging.error(f'value error {VE}')

    last_dir_index = 0

print("Last directory number ", last_dir_index)

print("New directory number ", last_dir_index+1)
```

```
current_dir_index = destination_dir + 'chrome' + str(last_dir_index + 1)

logging.info(f'current dir_index {current_dir_index}')

print('Copy...')

shutil.copytree(source_dir, current_dir_index)

logging.info(f'copy dir done {source_dir} {current_dir_index}')

return current_dir_index

def check_pid(pid):

    try:

        os.kill(pid, signal.SIGTERM)

        print('browser close')

        logging.info(f'os.kill {pid} {signal.SIGTERM}')

    except OSError:

        return False

    except Exception as e:

        logging.error(e)

        print(e)

        print('Browser PID not found')
```



```
else:  
    return True
```

```
def run(inactive_time):  
    log_off = False  
    mouse_activity = []  
  
    user_data = copy_directory()  
  
    print('Running browser...')  
    browser = run_browser(BROWSER_PATH, user_data)  
    while not log_off:  
        mouse_activity.append(tuple(pyautogui.position()))  
        sleep(60)  
        if len(mouse_activity) >= inactive_time:  
            if len(set(mouse_activity)) == 1:  
                logging.info('inactive session, break')  
                break  
            else:  
                mouse_activity = []  
    print(mouse_activity)
```

```
while True:

    if check_pid(browser.pid):

        delete_and_move_dir(user_data)

        break

    else:

        pass

    sleep(10)

shutdown()

if __name__ == '__main__':

    parser = argparse.ArgumentParser()

    parser.add_argument(

        "--inactive_time_min", type=int, help="Inactive time in minutes for logoff",
required=True

    )

    args = parser.parse_args()

    run(inactive_time=args.inactive_time_min)
```