| МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  **Кафедра програмних систем і технологій**  Дисципліна  **«Операційні системи»**  **Лабораторна робота №5**  **на тему:**  **«Створення об’єктів ФС у командному рядку»** | | | |
| --- | --- | --- | --- |
| **Виконав:** | Сторожук Богдан Миколайович | **Перевірила**: | Ткаченко Ольга Миколаївна |
| Група | ІПЗ-33 | Дата перевірки |  |
| Форма навчання | денна | Оцінка |  |
| Спеціальність | 121 |
| 2024 | | | |

**Мета роботи** — одержання практичних навичок роботи з файловими системами (ФС) FAT32 та NTFS за допомогою утиліт ОС Windows.

1. На розділі із файловою системою NTFS створити дерево каталогів source\_OS з відповідним вмістом, де 1 – номер варіанту:

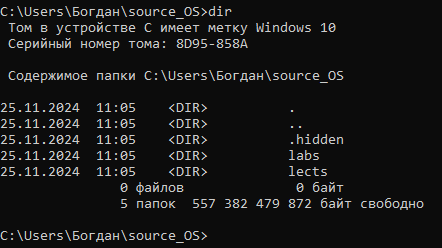
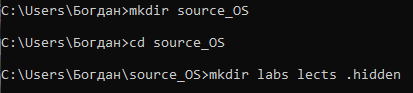
− каталог .hidden – прихований;

− файли lab1.txt, lab(1+1).txt наповнити рядками lab1, lab(1+1);

− user1.txt наповнити іменем поточного користувача,

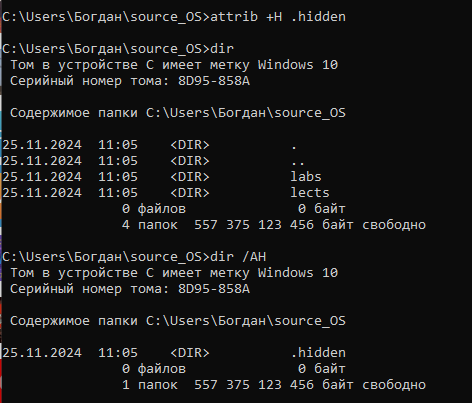
− tree1.txt – деревом source\_OS.

* 1. Створюємо дерево каталогів, скориставшись командою mkdir.

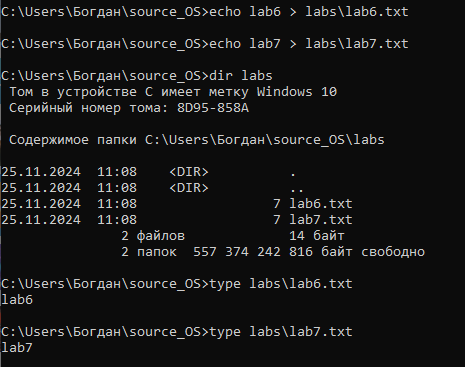


1.2 Надамо .hidden властивості прихованого каталогу, скориставшись командою attrib.

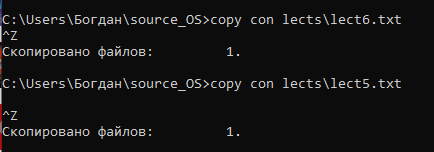
Для відображення прихованих об’єктів ФС командою dir необхідно використовувати ключ /AH.

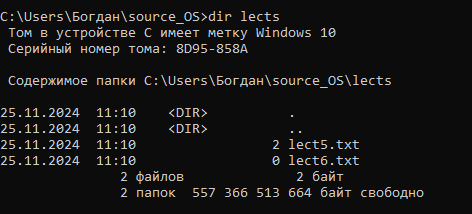


1.3 Створимо файли в labs, наповнивши їх відповідним вмістом:

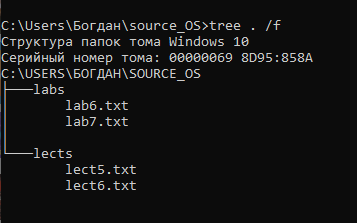


1.4 Створимо пусті файли в lects:



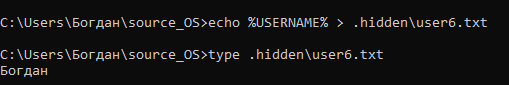
t

1.5 Переглянемо дерево каталогу OS, виконавши команду tree:

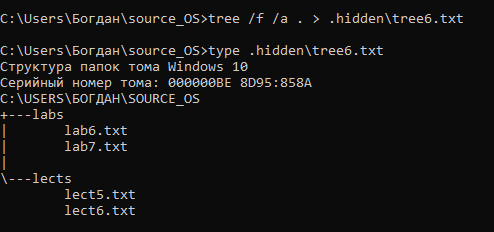


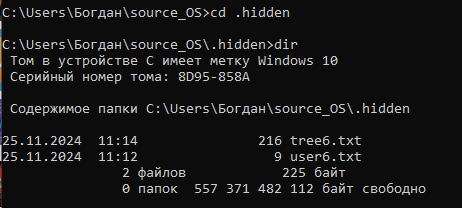
1.6 Створимо файл .hidden\user6.txt, наповнивши його іменем поточного

користувача:



1.7 Створимо файл .hidden\tree6.txt, наповнивши його деревом каталогу OS:





2. В домашній директорії користувача створити директорію OS з відповідним

вмістом:

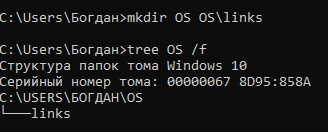
− labs, lects – посилання (junction та symlinkd) на відповідні каталоги;

− tree1\_h.txt – жорстке посилання на файл source\_OS/.hidden/tree1.txt,

− tree\_OS.txt містить дерево каталогів OS,

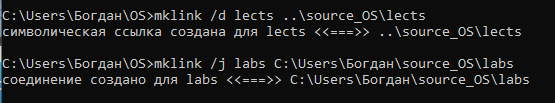
− result1.txt містить результати виконання лабораторної роботи.

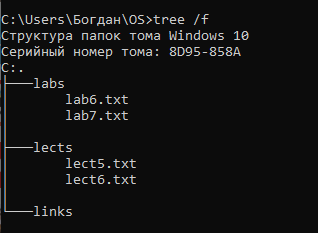
2.1. Створюємо дерево каталогів OS, скориставшись командою mkdir.



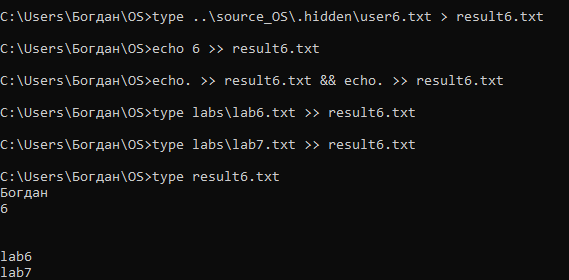
2.2. Створюємо символічний зв'язок на каталог lects та junction на каталог labs,

скориставшись командою mkdir.





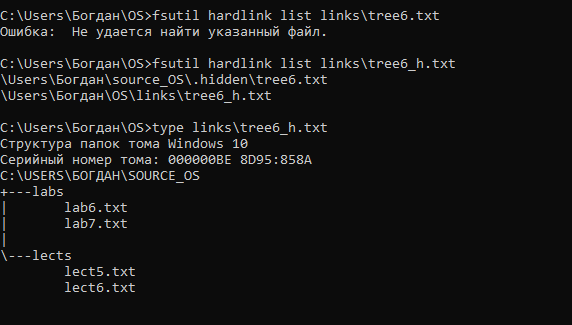
2.3. Створюємо файл результату OS\result6.txt та заповнюємо його наступним:



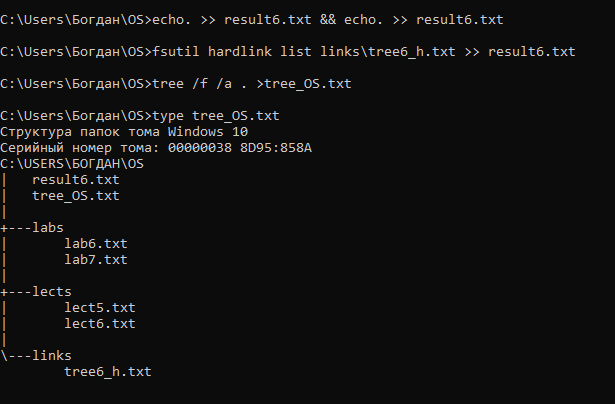
2.4. Створюємо файл links\tree6\_h.txt – жорстке посилання на

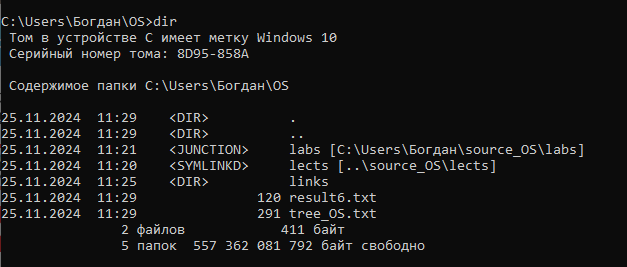
source\_OS\.hidden\tree1.txt





2.5. Оформлюємо файл результату result6.txt:

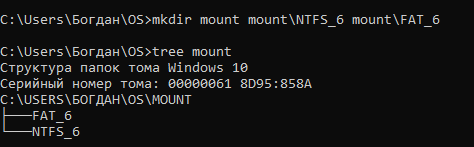




3. Створити точки майбутнього монтування розділів – каталоги mount\NTFS\_6,

mount\FAT\_6.

3.1. Створюємо каталоги, скориставшись командою mkdir:

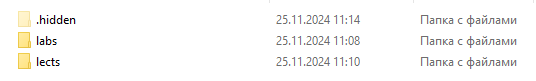


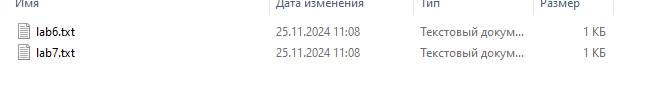
Результат:

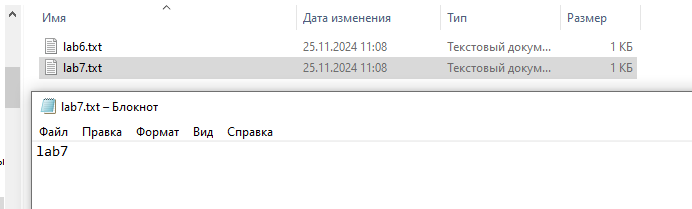


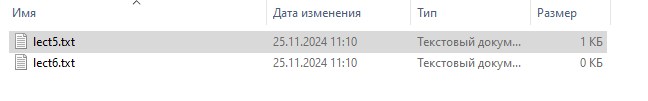


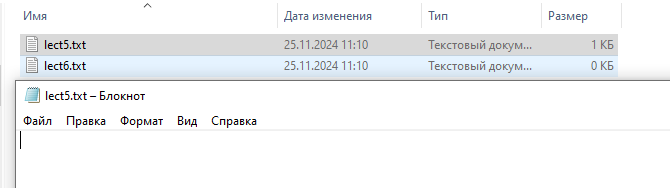
Source\_OS:



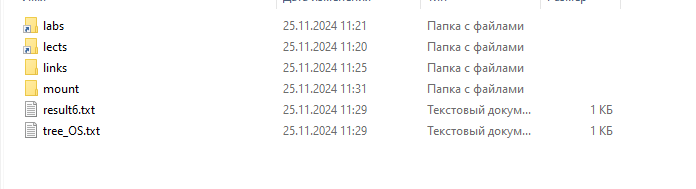


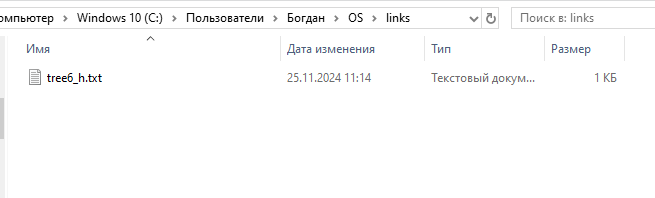


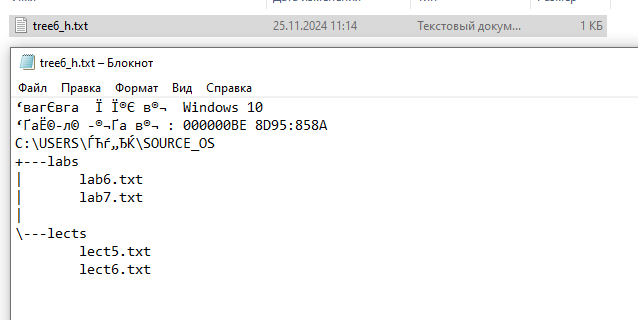


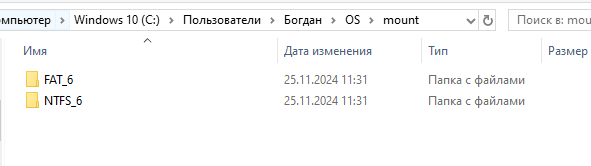


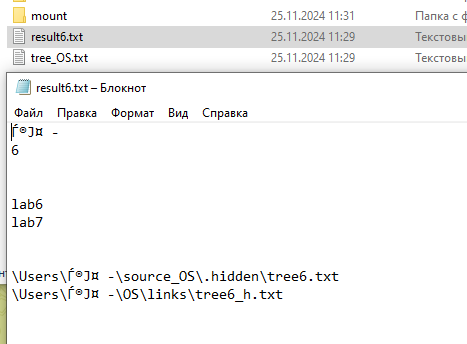
OS:

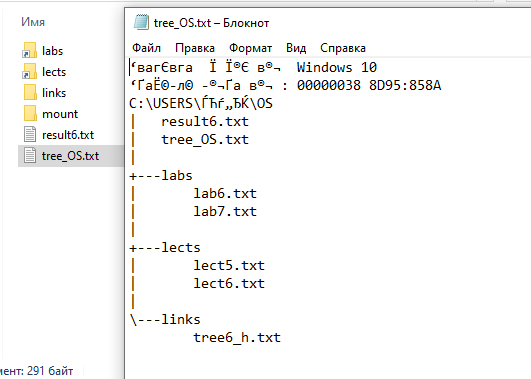












Питання для самоперевірки:

1. Базові та динамічні диски:

- Базові диски є простими та зручними для зберігання даних. Вони можуть мати первинні та розширені розділи (логічні диски).

- Динамічні диски надають більше можливостей для управління розділами (які називаються томами у динамічних дисках), такими як створення, видалення та розміщення томів без перезавантаження системи.

2. Розділ та том:

- Розділ — це частина фізичного диска, яка виступає як окремий логічний диск у системі. Базові диски використовують розділи.

- Том — це сегмент диска, який створений на динамічному диску. Том може складатися з простору одного або декількох дисків.

3. Файлова система для монтування томів у папки:

- NTFS (New Technology File System) дозволяє монтувати томи у папки.

4. Монтування томів командами `mountvol` та `mklink`:

- `mountvol` дозволяє прямо змонтувати том, вказуючи точку монтування.

- `mklink` створює символічні або жорсткі посилання на файл або каталог, але не використовується для монтування томів у традиційному розумінні.

5. Переваги використання UID тому:

- Використання UID (унікальний ідентифікатор) для ідентифікації томів робить систему більш гнучкою та надійною, оскільки UID завжди унікальний, у той час як літерні позначення можуть змінюватися.

6. Жорсткі та символічні зв’язки, junctions:

- Жорсткі зв’язки (`Hard Links`) можна створювати тільки для файлів, вони посилаються на те ж місце зберігання даних, що і оригінальний файл.

- Символічні зв’язки (`Symbolic Links`) та `Junctions` можна створювати для файлів та каталогів. Вони фактично є посиланнями на інші файли чи каталоги.

- Символічні зв’язки та Junctions можна використовувати в NTFS.