

## Лекція 2

### КОМАНДИ РОБОТИ З ФАЙЛАМИ

Завдання на повторення матеріалу Лекції 1. Вивести в файл D:\1.txt список файлів і підпапок директорії C:\Windows. В разі виникнення помилки вона повинна бути записана в файли D:\error.txt.

У інтерпретатора cmd є потужний інструментарій взаємодії з файловою системою. Нагадаємо, що файлова система має деревоподібну структуру (рис. 2.1.) і імена файлів задаються в форматі [диск:][шлях\]ім'я файлу, тобто обов'язковим параметром є тільки ім'я файлу. При цьому, якщо шлях починається з символу "\", то маршрут обчислюється від кореневого каталогу, інакше — від поточного каталогу.

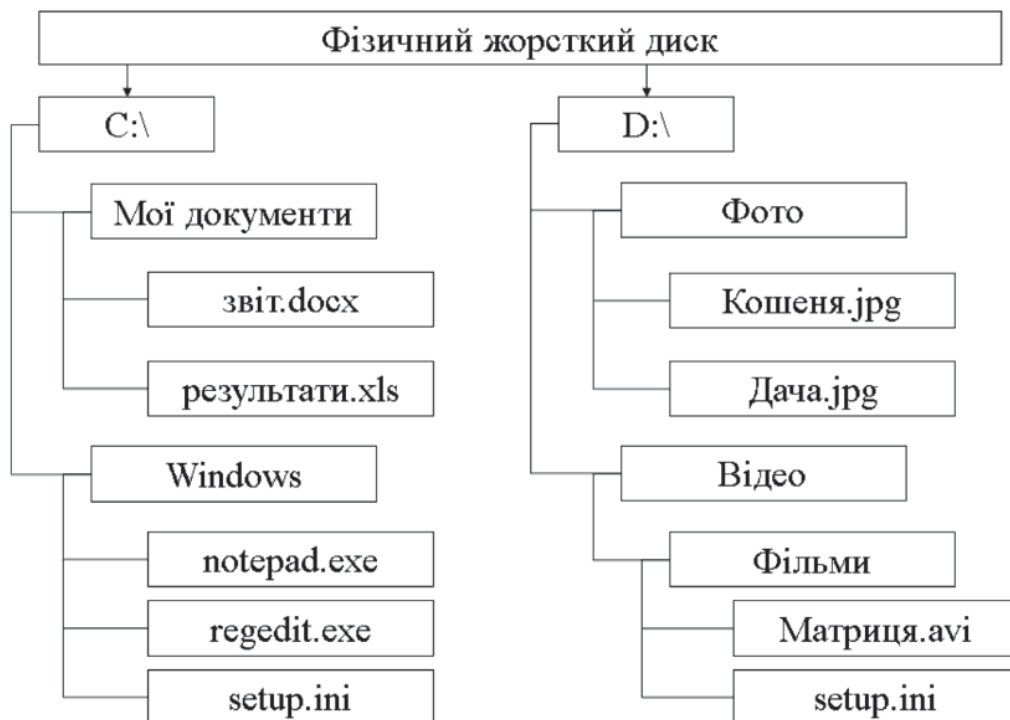


Рис. 2.1. — Деревоподібна структура файлової системи ОС Windows

Наприклад, ім'я C:123.txt задає файл 123.txt в поточному каталозі на диску C:, ім'я C:\123.txt — файл 123.txt в кореневому каталозі на диску C:, ім'я ABC\123.txt — файл 123.txt в підкаталозі ABC поточного каталогу.

Існують особливі позначення для поточного каталогу і батьківського каталогу. Поточний каталог позначається символом «.», його батьківський каталог — «..». Наприклад, якщо поточним каталогом є C:\WINDOWS, то шлях до файлу autoexec.bat в кореневому каталозі диска C: може бути записаний у вигляді ..\autoexec.bat.

В іменах файлів можна застосовувати так звані групові символи або шаблони:

? (знак питання);

\* (зірочка).

Символ «\*» в імені файлу означає будь-яку кількість будь-яких допустимих символів, символ «?» — один довільний символ або його відсутність. Скажімо, під шаблон text??1.txt підходять, наприклад, імена text121.txt і text911.txt, під шаблон text\*.txt — імена text.txt, textab12.txt, а під шаблон text.\* — всі файли з ім'ям text і довільним розширенням.

Однією з найважливіших команд роботи з файловою системою є команда COPY. Вона застосовується для копіювання одного або декількох файлів.

Формат запису команди COPY:

*COPY [/A|/B] джерело [/A|/B] [+ джерело [/A|/B] [+ ...]] [результат [/A|/B]] [/V]/[Y/-Y]*

Нехай на диску D у нас є 2 каталоги, з іменами 1 і 2. В каталозі 1 лежать текстові файли 1.txt, 2.txt і 3.txt (рис. 2.2.).

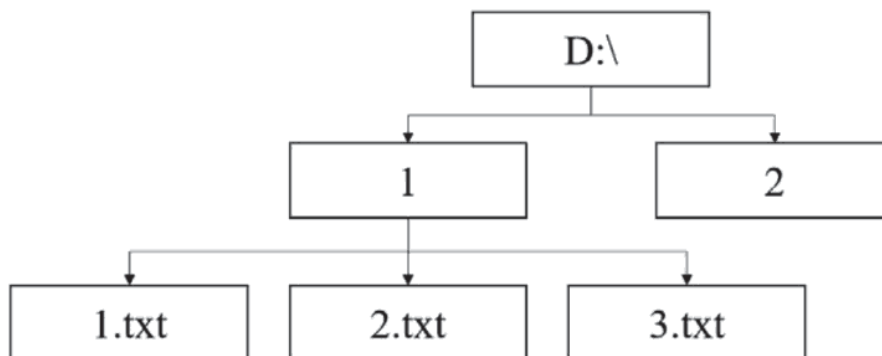


Рис. 2.2. — Приклад експериментального дерева каталогів для демонстрації роботи команди COPY

Скопіюємо кожен з файлів в папці 1 в папку 2 (лістинг 2.1.).

```

d:\>COPY d:\1\1.txt d:\2
The system cannot find the file specified.

d:\>COPY d:\1\1.txt d:\2
      1 file(s) copied.

d:\>COPY d:\1\2.txt d:\2
      1 file(s) copied.

d:\>COPY d:\1\3.txt d:\2
      1 file(s) copied.

d:\>DIR d:\2

Volume in drive D has no label.
Volume Serial Number is 2293-2D78
Directory of d:\2
05/30/2019  10:14 AM    <DIR>        .
05/30/2019  10:14 AM    <DIR>        ..
05/30/2019  10:13 AM                5 1.txt
05/30/2019  10:14 AM                4 2.txt
05/30/2019  10:14 AM                8 3.txt
              3 File(s)              17 bytes
              2 Dir(s)  788,302,704,640 bytes free
  
```

Лістинг 2.1. — Приклад роботи команди COPY

У нас було всього 3 файли. А якби їх було 3000? Як Ви думаєте, як можна застосувати команду COPY (і інші команди) відразу до серії файлів?

Для цього можна запросто використовувати групові символи ? і \*.

Для початку видалимо з директорії D:\2 всі файли, які ми скопіювали. Для цього зручно використати команду DEL:

*DEL [диск:][шлях]ім'я\_файлу [ключі]*

Ключі:

- /S — дозволяє видаляти зазначені файли з усіх підкаталогів;
- /F — примусово видаляти файли, доступні тільки для читання;
- /A[[:]атрибути] — вибирати файли для видалення по атрибутам.

Розглянемо варіанти застосування групових символів. Оскільки всі файли, які ми копіювали мають розширення txt і ім'я всього лише з однієї літери, вдалим буде використання наступного шаблону (лістинг 2.2.):

*? .txt*

```
d:\2>COPY D:\1\?.txt D:\2\  
D:\1\1.txt  
D:\1\2.txt  
D:\1\3.txt  
3 file(s) copied.
```

Лістинг 2.2. — Приклад застосування групового символу ?

Звичайно, можуть бути й інші шаблони. Поекспериментуємо ще. Відзначимо, що в папці D:\1 немає файлів з іншим розширенням, крім розширення txt. У цьому випадку ми можемо скопіювати всі файли з розширенням txt, використовуючи шаблон \*.txt (лістинг 2.3.).

```
d:\2>COPY D:\1\*.txt D:\2\
D:\1\1.txt
D:\1\2.txt
D:\1\3.txt
        3 file(s) copied.
```

Лістинг 2.3. — Приклад застосування групового символу \*

Ми також можемо скопіювати всі файли (лістинг 2.4.).

```
d:\2>COPY D:\1\* D:\2\
D:\1\1.txt
D:\1\2.txt
D:\1\3.txt
        3 file(s) copied.
```

Лістинг 2.4. — Приклад копіювання всіх файлів

У команди COPY є корисні ключі (табл. 2.1.).

Таблиця 2.1. — Ключі команди COPY

Ключ	Опис
/A	Файл є текстовим файлом ASCII, тобто кінець файлу позначається символом з кодом ASCII 26 (<Ctrl> + <Z>)
/B	Файл є двійковим. Цей ключ вказує на те, що інтерпретатор команд повинен при копіюванні зчитувати з джерела число байт, задане розміром в каталозі файлу, що буде копіюватися
/V	Перевірка правильності копіювання шляхом порівняння файлів після копіювання
/Y	Відключення режиму запиту підтвердження на заміну файлів
/-Y	Вмикання режиму запиту підтвердження на заміну файлів

За допомогою команди COPY також можливо створювати нові текстові файли і записувати в них інформацію без використання текстового редактора.

Для цього достатньо ввести команду COPY CON my.txt (лістинг 2.5.), яка буде копіювати те, що Ви набираєте на клавіатурі, в файл my.txt (якщо цей файл існував, то його буде перезаписано, інакше він буде створений).

```
d:\2>COPY CON 4.txt
Hello
people
!
^Z
1 file(s) copied.
```

Лістинг 2.5. — Приклад використання команди COPY для створення файлів

При цьому, для завершення введення необхідно ввести символ кінця файлу, тобто натиснути клавіші <Ctrl> + <Z>.

Команда COPY може також об'єднувати (склеювати) кілька файлів в один. Для цього необхідно вказати єдиний результуючий файл і кілька вихідних. Це досягається шляхом використання групових символів («?» і «\*») або формату *файл1* + *файл2* + *файл3* + ....

Наприклад, переглянемо вміст файлів, які ми раніше копіювали. Для цього зручно скористатися командою MORE (лістинг 2.6.).

```
d:\2>MORE 1.txt
Hello

d:\2>MORE 2.txt
dear

d:\2>MORE 3.txt
students
```

Лістинг 2.6. — Приклад перегляду вмісту файлів за допомогою команди MORE

Об'єднаємо їх в один файл (лістинг 2.7.).

```
d:\2>COPY /B 1.txt + 2.txt + 3.txt 5.txt
1.txt
2.txt
3.txt
1 file(s) copied.
```

Лістинг 2.7. — Приклад об'єднання вмісту текстових файлів за допомогою команди COPY

Ключ /B тут використовується для запобігання усічення файлів, що об'єднуються.

У новому файлі дійсно суперпозиція трьох об'єднаних нами файлів (лістинг 2.8.)

```
d:\2>MORE 5.txt
Hellodearstudents
```

Лістинг 2.8. — Приклад перегляду вмісту файлу 5.txt

Зауваження по команді COPY:

1. якщо ім'я цільового файлу відсутнє, то в його якості використовується перший файл зі списку. Наприклад, команда COPY 1.txt + 2.txt додасть до вмісту файлу 1.txt вміст файлу 2.txt;

2. якщо в команді COPY не вказати призначення копіювання, то файли будуть скопійовані в поточну директорію;

3. командою COPY можна скористатися і для присвоєння якому-небудь файлу поточної дати і часу без модифікації його вмісту. Для цього потрібно ввести команду типу:

*COPY /B 1.txt +,,*

Тут коми вказують на пропуск параметра приймача, що і приводить до необхідного результату.



Команда `COPY` має і свої недоліки. Наприклад, з її допомогою неможливо копіювати приховані і системні файли, файли нульової довжини, файли з підкаталогів. Крім того, якщо при копіюванні групи файлів `COPY` зустріне файл, який в даний момент не можна скопіювати (наприклад, він зайнятий іншим додатком), то процес копіювання повністю перерветься, і інші файли не будуть скопійовані.

Зазначені проблеми команди `COPY` можна вирішити за допомогою команди `XCOPY`, яка надає набагато більше можливостей при копіюванні:

*XCOPY джерело [результат] [ключі]*

Команда `XCOPY` має багато ключів, серед яких виділимо найважливіші, показані в табл. 2.2.

Таблиця 2.2. — Ключі команди `XCOPY`

Ключ	Опис
<code>/D[:[data]]</code>	Дозволяє копіювати тільки файли, змінені не раніше зазначеної дати. Якщо параметр <i>data</i> не вказано, то копіювання буде проводитися тільки якщо джерело новіше результату
<code>/S</code>	Дозволяє копіювати всі непусті підкаталоги в каталозі-джерелі
<code>/E</code>	Дозволяє копіювати взагалі всі підкаталоги, включаючи і порожні
<code>/H</code>	Можна копіювати приховані і системні файли
<code>/R</code>	Дозволяє замінювати файли з атрибутом «Тільки для читання»
<code>/Y</code> і <code>/-Y</code>	Визначають, чи потрібно запитувати підтвердження перед заміною файлів при копіюванні. <code>/Y</code> — придушення запиту підтвердження на перезапис існуючого цільового файлу, <code>/-Y</code> — запит підтвердження на перезапис існуючого цільового файлу

Для роботи з директоріями передбачені команди `MKDIR` і `RMDIR`. Для створення нового каталогу використовується команда:

*MKDIR [диск:]шлях*

У той час як для видалення каталогу використовується команда `RMDIR`:

*RMDIR [диск:]шлях [ключі]*



Замість цих команд допускається використання їх коротких аналогів MD і RD.

Команда MKDIR не може бути виконана, якщо каталог або файл із заданим ім'ям вже існує. Команда RMDIR не буде виконана, якщо каталог, що видаляється, не порожній.

Наприклад, створимо папку з ім'ям 11 і видалимо її (лістинг 2.9.).

```
d:\2>MD 11
d:\2>DIR /W
Volume in drive D has no label.
Volume Serial Number is 2293-2D78
Directory of d:\2
[.]      [...]    1.txt    [11]     2.txt    3.txt    4.txt    5.txt
                    5 File(s)                52 bytes
                    3 Dir(s)  788,302,675,968 bytes free
d:\2>RMDIR 11

d:\2>CD 11
The system cannot find the path specified.
```

Лістинг 2.9. — Приклад створення і видалення директорії

Перейменувати файли і каталоги можна за допомогою команди RENAME (REN). Синтаксис цієї команди має наступний вигляд:

*REN [диск:][шлях][каталог1|файл1] [каталог2|файл2]*

Тут параметр *каталог1|файл1* визначає назву каталогу/файлу, назву якого потрібно змінити, а *каталог2|файл2* задає нову назву каталогу/файлу. У будь-якому параметрі команди REN можна використовувати групові символи ? і \*. При цьому представлені шаблонами символи в параметрі *файл2* будуть ідентичні відповідним символам в параметрі *файл1*.

Якщо файл з ім'ям *файл2* вже існує, то команда REN припинить виконання, і станеться виведення повідомлення, що файл вже існує або зайнятий. Крім того, в команді REN можна вказати інший диск або каталог для створення результируючих каталогу і файлу.

Завдання: для всіх файлів з розширенням txt в поточному каталозі змінити розширення на doc.

Для переміщення файлів використовується команда MOVE:

*MOVE [/Y|/-Y] [диск:][шлях]ім'я\_файлу1[,...] результуючий\_файл*

Якщо вказано ключ /-Y, то при створенні каталогів і заміні файлів буде видаватися запит на підтвердження. Ключ /Y скасовує видачу такого запиту.

Наприклад, створимо на диску D:\ каталог 3 і перемістимо туди з каталогу 1 всі текстові файли (лістинг 2.10.).

```
d:\2>MKDIR D:\3
d:\2>MOVE d:\1\* d:\3
d:\1\1.txt
d:\1\2.txt
d:\1\3.txt
        3 file(s) moved.
```

Лістинг 2.10. — Приклад використання команди MOVE

**Команда ATTRIB** використовується для зміни атрибутів одного чи кількох файлів:

- **атрибут файла R «Тільки для читання»** забороняє вносити в нього зміни;
- **атрибут файла H «Прихований»** робить файли невидимими в провіднику;
- **атрибут файла S «Системний»** автоматично встановлюється операційною системою для всіх важливих системних файлів Windows. Змінити цей атрибут через властивості файлу неможливо, тому що такого поля немає (це можна зробити через командний рядок). Для системних файлів автоматично встановлюється атрибут «Прихований»;

- **архівний атрибут A.** Коли цей атрибут встановлений, це означає, що файл був змінений з часу проведення останнього резервного копіювання. ПЗ, за допомогою якого можна створювати резервні копії, також відповідає за зняття цього атрибута;

— **атрибут «індексації» I.** Означає, що відповідні підсистеми Windows не включатимуть вміст файлу в пошуковий індекс. Індексування необхідно для більш швидкого пошуку потрібних файлів на комп'ютері;

— **атрибут Compressed «Стискати вміст для економії простору на диску».** Встановлений атрибут означає, що Windows зберігає цей файл на диску в стисненому вигляді;

— **атрибут Encrypted «Шифрувати вміст для захисту даних»** дозволяє зашифрувати файл або папку. При встановленні даного атрибута файл або папка стануть доступні тільки користувачеві, що увійшов в систему з обліковим записом, з якого був встановлений атрибут.

Переглянути і змінити деякі атрибути встановлені для даного файлу можна і через пункт меню «Властивості файлу» (рис. 2.3.).

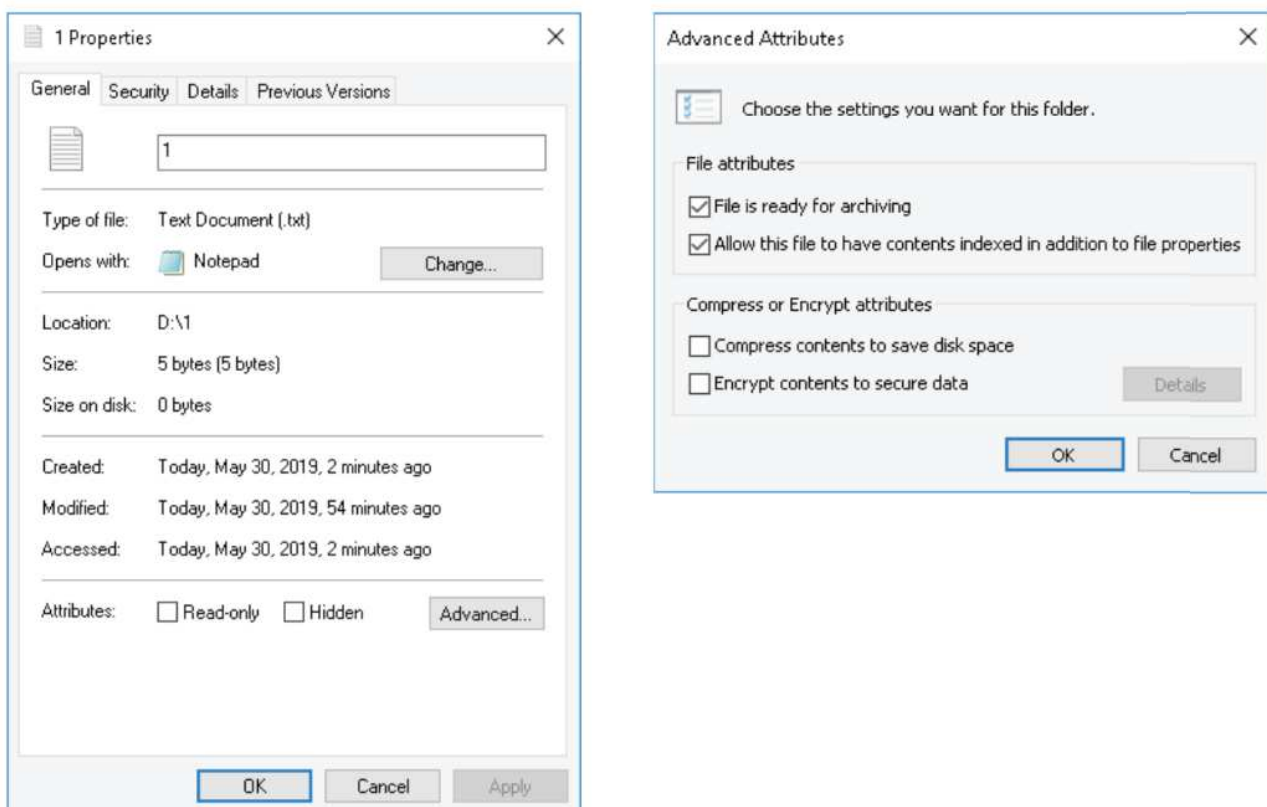


Рис. 2.3. — Перегляд і установка деяких атрибутів файлів через інтерфейс операційної системи Windows

Команда ATTRIB дає можливість змінювати атрибути файлу за допомогою командного рядка. Команда має синтаксис:

*ATTRIB [-A][+A][-S][+S][-R][+R][-H][+H][-I][+I][Диск:][Шлях]  
[Ім'я\_файла][/S][/D][/L]*

Букви A, S, R, H, I означають відповідні атрибути. Знак «+» і «-» означають відповідно установку і зняття атрибута.

За допомогою команди ATTRIB можна також переглянути атрибути файлів, що знаходяться в даному каталозі (лістинг 2.11.).

```
D:\3>ATTRIB
A                D:\3\1.txt
A                D:\3\2.txt
A                D:\3\3.txt
```

Лістинг 2.11. — Перегляд атрибутів файлів за допомогою команди ATTRIB

Наприклад, зробимо один із створених нами раніше файлів системним (лістинг 2.12.)

```
D:\3>ATTRIB +S 1.txt

D:\3>ATTRIB
A  S                D:\3\1.txt
A                D:\3\2.txt
A                D:\3\3.txt
```

Лістинг 2.12. — Приклад використання команди ATTRIB для присвоєння файлу атрибуту «системний»

І спробуємо видалити його (лістинг 2.13.).

```
D:\3>DEL 1.txt
Could Not Find D:\3\1.txt

D:\3>ATTRIB -S 1.txt

D:\3>DEL 1.txt
```

Лістинг 2.13. — Видалення файлу з атрибутом «системний»

У інтерпретаторі `cmd` є також так зване умовне виконання команд. Наприклад, в командному рядку Windows можна використовувати спеціальні символи, які дозволяють вводити кілька команд одночасно і керувати роботою команд в залежності від результатів їх виконання.

Використовуючи символ амперсанд `&`, можна розділити кілька утиліт в одній строчці командного рядка, при цьому вони будуть виконуватися одна за одною.

Наприклад, якщо набрати команду `DIR & PAUSE & COPY /?` і натиснути клавішу «Enter», то спочатку на екран буде виведено вміст поточного каталогу, а після натискання будь-якої клавіші — вбудована довідка команди `COPY`.

**Команда `PAUSE`** — припинення виконання пакетного файлу і виведення повідомлення:

*Press any key to continue . . .*

Команда **`ECHO`**, напевно, є однією з найбільш простих і найбільш використовуваних команд (лістинг 2.14.). Вона застосовується для виведення текстових повідомлень у стандартний потік виводу і для перемикання режиму відображення команд на екрані.

```
D:\3>echo Hello World!
Hello World!
```

Лістинг 2.14. — Приклад використання команди `ECHO`

Наприклад, якщо ввести в командному рядку:

*`ECHO Abc & COPY /?`*



і натиснути клавішу «Enter», то станеться виконання поспіль двох команд: ECHO Abc і виконання команди COPY /? .

А якщо, припустимо, ми хочемо, щоб команда ECHO просто вивела нам рядок «Abc & COPY /?»?

Символ ^ дозволяє використовувати командні символи як текст, тобто при цьому відбувається ігнорування значення спеціальних символів. Так, для цього необхідно ввести рядок:

*ECHO Abc ^& COPY /?*

У цьому випадку просто виконується одна команда ECHO з відповідними параметрами. На екран буде виведено повідомлення (лістинг 2.15.).

```
D:\3>ECHO Abc ^& COPY /?  
Abc & COPY /?
```

Лістинг 2.15. — Приклад використання команди ECHO для виведення повідомлень зі спеціальними символами

Умовна обробка команд в Windows здійснюється за допомогою символів && і || наступним чином. Подвійний амперсанд && запускає команду, яка стоїть за ним в командному рядку, тільки в тому випадку, якщо команда, що стоїть перед амперсандами, була виконана успішно (лістинг 2.16.).

```
D:\3>REN 10.txt 11.txt && ECHO File is renamed successfully  
File is renamed successfully  
D:\3>REN 3.txt 4.txt && ECHO File is renamed successfully  
The system cannot find the file specified.
```

Лістинг 2.16. — Приклад використання символу умовного виконання команд &&

Символи || здійснюють в командному рядку зворотну дію, тобто запускають команду, яка стоїть за цими символами, тільки в тому випадку, якщо команда, що



йде перед ними, не була успішно виконана. Тобто, якщо перша команда спрацює, то друга команда виконуватися не буде (лістинг 2.17.).

```
D:\3>REN 10.txt 11.txt || ECHO Error renaming file
The system cannot find the file specified.
Error renaming file
D:\3>REN 11.txt 12.txt || ECHO Error renaming file
```

Лістинг 2.17. — Приклад використання символу умовного виконання команд ||

Відзначимо, що умовна обробка діє тільки на найближчу команду. Кілька утиліт можна згрупувати в командному рядку за допомогою дужок (лістинг 2.18.).

```
D:\3>REN 4.txt 5.txt || (ECHO Error renaming file & ECHO '' > 4.txt)
The system cannot find the file specified.
Error renaming file
```

Лістинг 2.18. — Приклад групування команд

В даному випадку (лістинг 2.18.) алгоритм виконання буде такий:

1. якщо вдасться перейменувати файл 4.txt в файл 5.txt команди після оператора || виконуватися не будуть;
2. якщо не вдасться перейменувати файл 4.txt в файл 5.txt на екран буде виведено повідомлення «Помилка перейменування файлу» і файл 4.txt буде створений.

Завдання. Скопіювати файл 5.txt з кореневого каталогу диска D:\ в папку D:\1, при цьому ім'я файлу змінити на 4.txt. Якщо скопіювати файл не вийде, програма повинна створити файл 4.txt в каталозі D:\1 і вивести повідомлення, що файл не скопійований, а створений.

Домашнє завдання до Лекції 2.

Перемістити файл virus.exe з кореневого каталогу диска C:\ в каталог D:\1. При успішному переміщенні вивести відповідне повідомлення. При невдалому переміщенні вивести повідомлення про помилку в файл D:\1\error.txt і змінити розширення всіх файлів в директорії D:\1 на virus.