

Лабораторна робота №2

Тема: «*Основи сучасної термінальної міжкомп'ютерної взаємодії*»

Мета роботи: придбання навичок із сучасної термінальної взаємодії між різними операційними системами на прикладі ОС *Windows* та ОС *Linux*.

План.

1. Теоретичні відомості
2. Завдання для виконання
3. Вимоги до оформлення протоколу виконання лабораторної роботи
4. Контрольні питання.

1 Теоретичні відомості

1.1 Термінальні пристрої

Сьогодні у всіх студентів є електронний комп'ютер, оформлений як великий настільний комп'ютер, середнього розміру ноутбук, розміром з книгу планшетний комп'ютер або смартфон. Ми господарі цих комп'ютерів і використовуємо їх одноосібно.

Але чи завжди ми їх використовуємо на 100% їх можливостей? У період часу, коли ми не використовуємо комп'ютер, чи можемо ми його дати в оренду іншим людям? Але не віддати його в тимчасове користування фізично, щоб після завершення терміну оренди нервувати, намагаючись знайти нечесного орендаря, який не бажає повертати комп'ютер. Чи можна віддати в оренду можливості комп'ютера, орендувати самі послуги, що надаються комп'ютером, дистанційно орендувавши сам комп'ютер? Споживачі комп'ютера про це стали замислюватися ще понад 60 років тому, коли для доступу до одного дорогого комп'ютера стояла черга з тисяч споживачів.

Ще в другій половині 19 століття з'явився Телетайп (англ. *Teletype*, скорочено - *TTY*) - електромеханічна друкарська машина, яка використовувалася для передачі між двома абонентами текстових повідомлень через найпростіший електричний канал (зазвичай по парі проводів). Тому для вирішення завдання оренди комп'ютера були придумані термінальні пристрої, що дозволяють множині користувачів одночасно працювати з одним комп'ютером, який називається хост (*host* - господар).

Термінальний пристрій складався з мінімум трьох компонент: пристрій введення, наприклад, клавіатура, пристрій виведення, наприклад, дисплей, і комунікаційної пристрій для зв'язку через комп'ютерну мережу з комп'ютером-хостом.

Багатокористувацьке термінальне пристрій дозволяє організувати на базі одного комп'ютера кілька незалежних місць - терміналів - з можливістю одночасної роботи користувачів.

Спершу у кожного типу комп'ютера-хоста були свої типи термінальних пристроїв - «дурні» термінали (англ. *Dumb terminal*). У вони були замінені на інтелектуальні термінали (англ. *Intelligent terminal*), які можна було програмувати за допомогою завантажуються з стрічки емуляторів терміналів, налаштовуючи на роботу із заданим комп'ютером-хостом.

Коли з'явилася ОС *Unix* багато програм, що використовують текстовий інтерфейс командного рядка, були спочатку розроблені для роботи через термінал. У більшості сучасних комп'ютерів, які не використовують спеціалізовані термінали, робота терміналу емулюється засобами ОС для сумісності зі старими програмами.

Для емуляції використовується псевдотермінал (англ. *Pseudo terminal*, або *PTY*) у вигляді пари псевдопристроїв, один з яких, другорядний, емулює справжній текстовий термінал, а другий, головний, надає засоби, за допомогою яких емулятор терміналу контролює процеси.

Багато років розробники і споживачі термінальних систем не турбувалися про безпеку даних і процесів. Тільки для зручності адміністрування користувачі повинні були проходити процеси ідентифікації, вводячи свої логіни, і підтверджувати їх процесом аутентифікації, вводячи, наприклад, секретну фразу - пароль. Для цих цілей багато років використовувався

мережевий протокол *Telnet* (англ. *Teletype network*) як алгоритм взаємодії між терміналами і комп'ютером-хостом. Тому у протоколі не передбачено використання ні шифрування, ні перевірки достовірності даних, і при бажанні стороння людина може легко перехопити конфіденційні дані, основним з яких є пароль користувача.

Коли споживачів стали хвилювати питання мережевої безпеки, *telnet* був замінений на безпечні протоколи, наприклад, на мережевий протокол *SSH* (англ. *Secure SHell* — «безпечна оболонка»).

1.2 Структура файлової системи UNIX

В класичній ОС *UNIX* та її нащадках, наприклад, ОС *Linux*, файлові системи виглядають як каталоги єдиного дерева. Дерево починається з кореневого каталогу, який починається з символу */*, як показано на рисунку 1. Всі *UNIX*-системи мають приблизно однакову структуру дерева каталогів:

- системні утиліти в каталозі */bin*
- системні бібліотеки – у каталозі */lib*,
- конфігураційні файли - у каталозі */etc*
- каталоги користувачів - у каталозі */home*

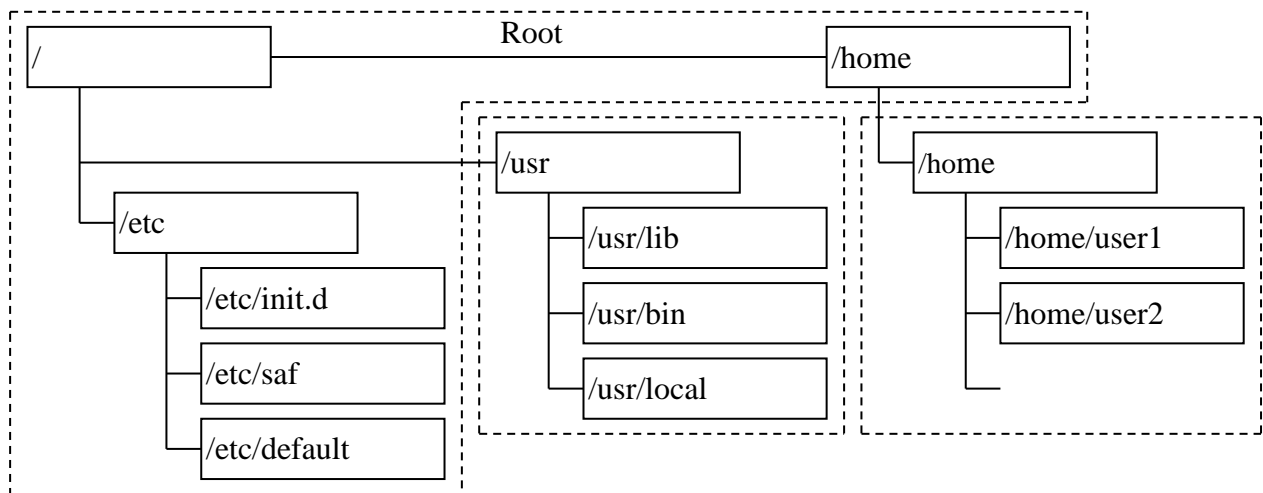


Рис. 1 - Фрагмент структури простору імен в *UNIX*

Базові команди огляду файлової системи:

ls - виводить зміст каталогу файлової системи;

pwd (англ. *print working directory* — надрукувати робочий каталог) - виводить повний шлях від кореневого каталогу до поточного робочого каталогу;

cd назва_каталогу - змінює поточний каталог на новий каталог за вказаною назвою (для переходу на батьківський каталог використовуються символи *..*, наприклад, *cd ..*);

pushd назва_каталогу - змінює поточний каталог на новий каталог за вказаною назвою з можливістю повернення в зворотному порядку;

popd - повертає до попереднього каталогу, з якого виконано переміщення після команди *pushd*;

less назва_файлу — перегляд вмісту файлу за вказаною назвою (для завершення перегляду файлу використовується клавіша *q*)

1.3 Основи багато користувальницької роботи з ОС

Як багато користувальницьке середовище ОС повинна забезпечувати зберігання інформації про її користувачів за рахунок ведення так званих профілів користувачів або облікових записів користувачів. У файловій системі створюється база даних користувачів, їх

паролів і профілів доступу до локальних ресурсів обчислювальної системи (наприклад, файл */etc/passwd* в Unix-подібних операційних системах).

Структура профілю користувача у файлі */etc/passwd* містить наступні поля:

- ім'я користувача;
- ідентифікатор користувача (*UID - user ID*) як число, з яким ОС легше працювати ніж зі строковими даними - імені користувача;
- Ідентифікатор групи (*GID - group ID*);
- реальне ім'я;
- початковий каталог користувача;
- командний інтерпретатор для обробки команд інтерфейсу командної строки.

Для керування профілями потрібен рівень адміністратора ОС.

Звичайний користувач може лише змінювати свій пароль командою *passwd*

Але при створенні нового паролю треба враховувати його складність, про що ОС повідомить.

Приклади команд огляду середовища та простої взаємодії з іншими користувачами:

whoami - виводить ім'я користувача, яке пов'язано з поточним ідентифікатором користувача;

who – виводить перелік користувачів, які працюють в ОС (ім'я користувача та мережева адреса комп'ютера, з якого він зайшов на сервер);

write ім'я користувача - дозволяє відправити повідомлення іншому користувачеві, який працює у цій же ОС (для завершення роботи можна виконати комбінацію клавіш *Ctrl+C*).

Пам'ятайте, що всі подробиці про роботу команд можна отримати через команду довідки *man назва команди*. Наприклад:

man pwd

1.4 Сучасна термінальна взаємодія

1.4.1 Завантаження програмного забезпечення

Для взаємодії з ОС *Linux*, яка розташована на віддаленому комп'ютері, з локального комп'ютера з ОС *Windows* часто використовується три програми:

PUTTY - програма віддаленого доступу до терміналу командного рядка, де в назві є слово *TTY* (від англ. *TTY* - телетайп);

PSCP - програма віддаленого копіювання файлів з інтерфейсом командного рядка;

WINSCP - програма віддаленого копіювання файлів з графічним інтерфейсом.

Посилання на отримання програми *putty.exe*

- 32-розрядна - <https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe>

- 64-розрядна - <https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe>

Посилання на отримання програми *pscp.exe* –

- 32-розрядна - <https://the.earth.li/~sgtatham/putty/latest/w32/pscp.exe>

- 64-розрядна - <https://the.earth.li/~sgtatham/putty/latest/w64/pscp.exe>

Посилання на отримання програми *winscp* –

<https://winscp.net/download/WinSCP-5.9.4-Setup.exe>

1.4.2 Віддалений доступ з ОС *Linux* до ОС *Linux*

Для перевірки доступності віддаленого *Linux*-сервера в мережі використовується команда *PING*:

ping мережева_адреса.

Команда відправляє запити (англ. *Echo-Request*) зазначеному серверу у мережі й фіксує відповіді (англ. *Echo-Reply*). Час між відправленням запиту й одержанням відповіді дозволяє визначати двосторонні затримки у маршруті й частоту втрати пакетів, тобто побічно визначати завантаженість каналів передачі даних і проміжних пристроїв.

Приклад використання команди: *ping 91.219.60.189*

Для припинення роботи команди можна виконати комбінацію клавіш *Ctrl+C*

Для віддаленого доступу з однієї ОС *Linux* до іншої ОС *Linux* використовується команда *SSH* - *Secure Shell* (англ. *Secure SHell* — «безпечна оболонка»):

```
ssh ім'я_користувача@мережева_адреса
```

Команда дозволяє передавати дані через шифроване з'єднання. Для шифрування використовується відкритий ключ *Linux*-сервера, на який виконується з'єднання.

Ключ можна отримати автоматично при першому з'єднанні. Найважливішими даними для шифрування є пароль користувача, який зможе розшифрувати лише сервер, використовуючи свій пароль.

Наприклад, для з'єднання користувачу з ім'ям користувача = *user1* до серверу з *IP=91.219.60.189*, необхідно виконати команду:

```
ssh user1@185.143.145.158
```

1.4.3 Віддалений доступ з ОС *Windows* до ОС *Linux*

Для віддаленого доступу з ОС *Windows* до терміналу командного рядка ОС *Linux* використовується програмна утиліта *PUTTY.EXE*

На рисунку 2 показано схему віддаленого доступу до сервера.

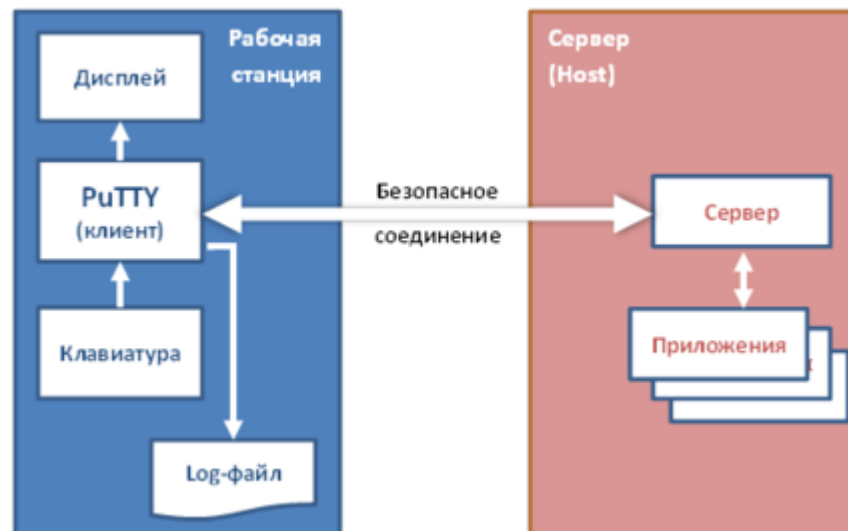


Рис. 2 - Схема віддаленого доступу до сервера

Приклад стартовою екранної форми програми *PuTTY* представлений на рисунку 3.

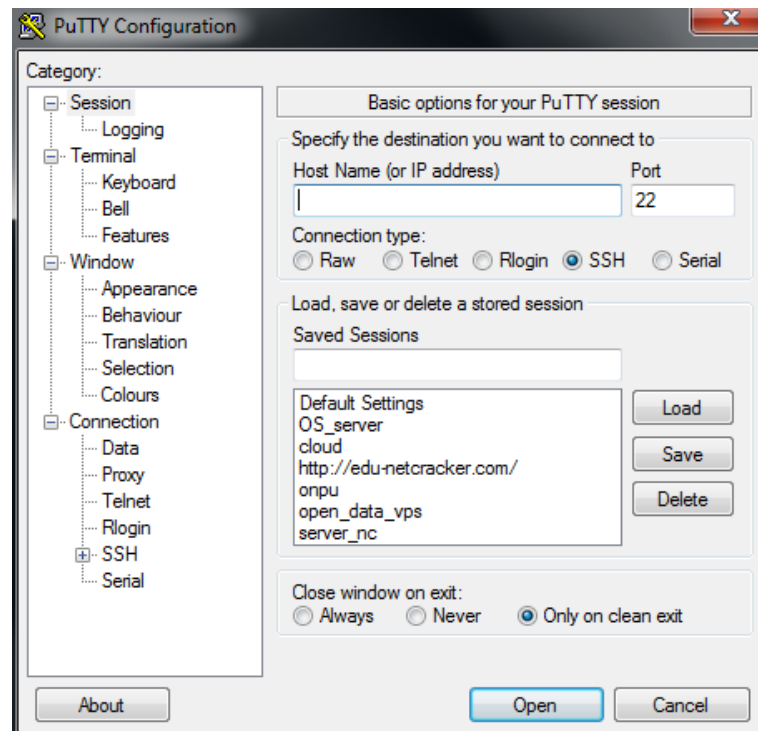


Рис. 3 – Приклад стартовою екранної форми програми *PuTTY.EXE*

Для встановлення з'єднання з віддаленою ОС необхідно:

- 1) у полі *Host Name* вказати IP-адресу комп'ютера з ОС *Linux*;
- 2) у полі *Connection Type* вибрати безпечне з'єднання (*SSH*), що автоматично встановить порт з'єднання = 22;
- 3) у полі *Saved Sessions* вказати назву з'єднання;
- 4) натиснути кнопку *Save* для збереження налаштувань з'єднання;
- 5) натиснути кнопку *Open* для встановлення з'єднання

Для налаштування колірної гами (білий фон, чорний тон) необхідно зайти в розділ *Window-Colours* і виставити кольори, як показано на рисунку 4.

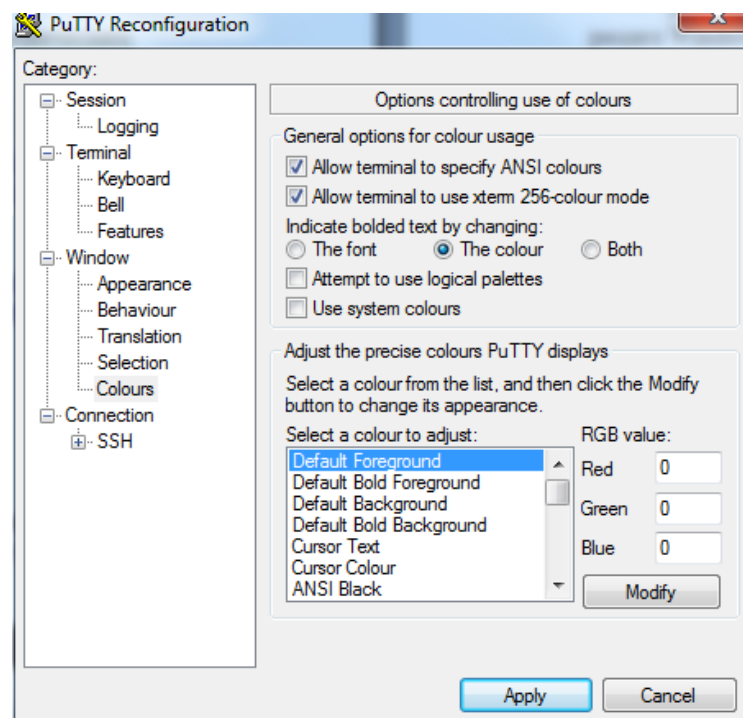


Рис. 4 – Приклад екранної форми налаштування колірної гами

Для налаштування розміру шрифту необхідно зайти в розділ Window-Appearance і виставити *Font setting*, натиснувши кнопку *Change* як показано на рисунку 5.

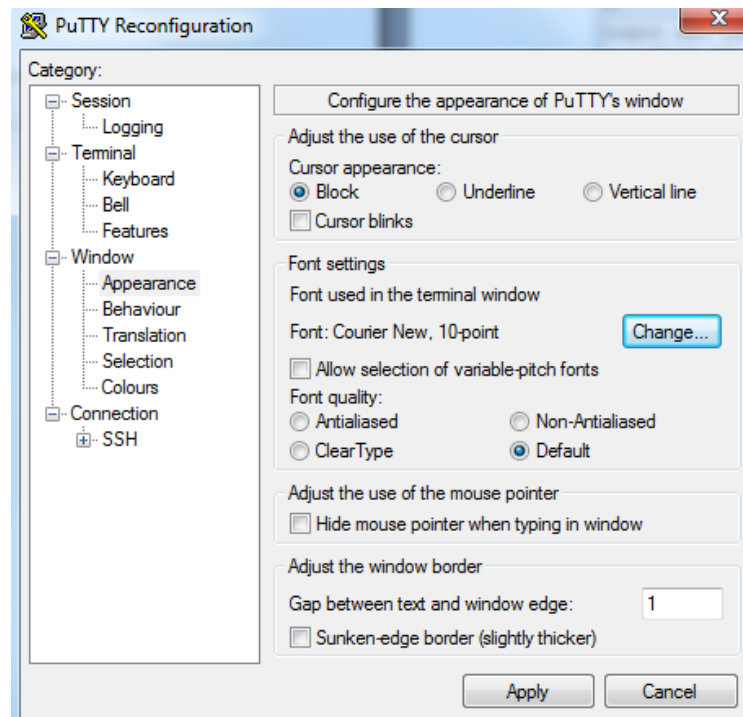


Рис. 5 – Приклад екранної форми налаштування розміру шрифту

В результаті виконаних дій з'явиться екран із запрошенням введення логіна і пароля, як показано на рисунку 6.

Для безпеки процес вводу символів паролю не відображається на екрані і курсор не змінює свої положення.

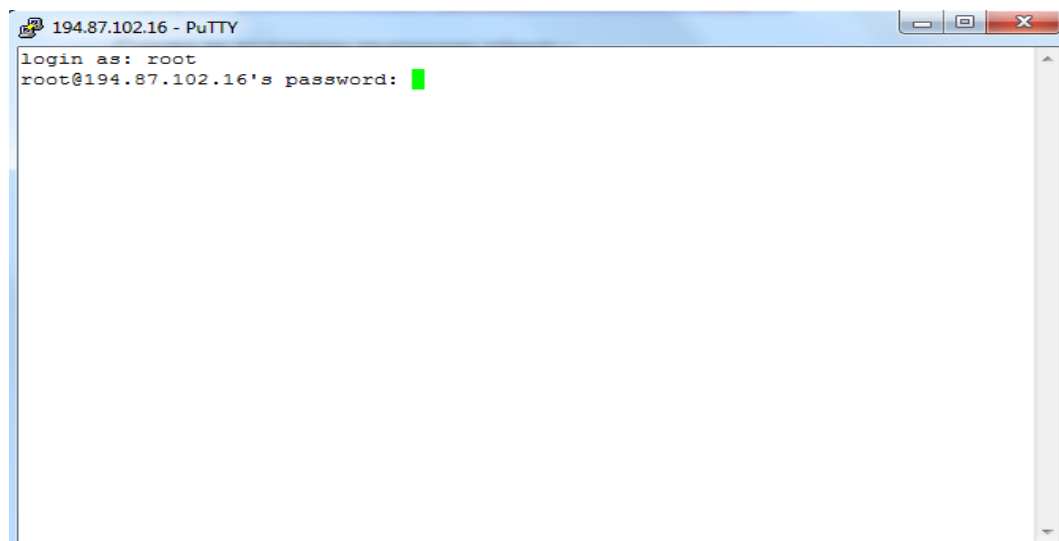


Рис. 6 – Приклад екранної форми вводу логіна і пароля

Для завершення роботи з терміналом командного рядка використовується команда `exit`

1.4.4 Віддалене копіювання файлів в командному рядку

В UNIX-подібних ОС для копіювання файлів використовується команда *CP*:
cp SourceFile TargetFile

Наприклад, для копіювання файлу File1, який розташовано у каталозі /dir1, у файл File2, який розташовано у каталозі /dir2, необхідно виконати:

```
cp /dir1/File1 /dir2/File2
```

В UNIX-подібних ОС існує утиліта *SCP* для безпечного віддаленого копіювання файлів (*Secure CP*).

Команда копіювання локального *SourceFile* на віддалену ОС:

```
scp SourceFile user@host:/directory/TargetFile
```

Команда копіювання *SourceFile* з віддаленої ОС:

```
scp user@host:/directory/SourceFile TargetFile
```

SourceFile має включати повний шлях до файлу, який копіюється.

Програмна утиліта *PSCP.EXE* є еквівалентом команди *SCP*.

Для роботи з утилітою необхідно використовувати термінал командного рядка в ОС Windows, який запускається через виконання команди *cmd* в рядку виконання команд.

Наприклад, для копіювання файлу 1.txt на віддалений комп'ютер з адресою 91.219.60.189 від імені користувача root в каталог /etc/ необхідно виконати:

```
pscp.exe 1.txt root@91.219.60.189:/etc/
```

Наприклад, для копіювання файлу /etc/1.txt з віддаленого комп'ютера з адресою 91.219.60.189 від імені користувача root в каталог d: локального комп'ютера необхідно виконати:

```
pscp.exe root@91.219.60.189:/etc/1.txt d:\
```

1.4.5 Віддалене копіювання файлів через графічний інтерфейс

Для копіювання файлів через графічний інтерфейс використовується програма *WinSCP*, приклад стартової екранної форми якої представлений на рисунку 7.

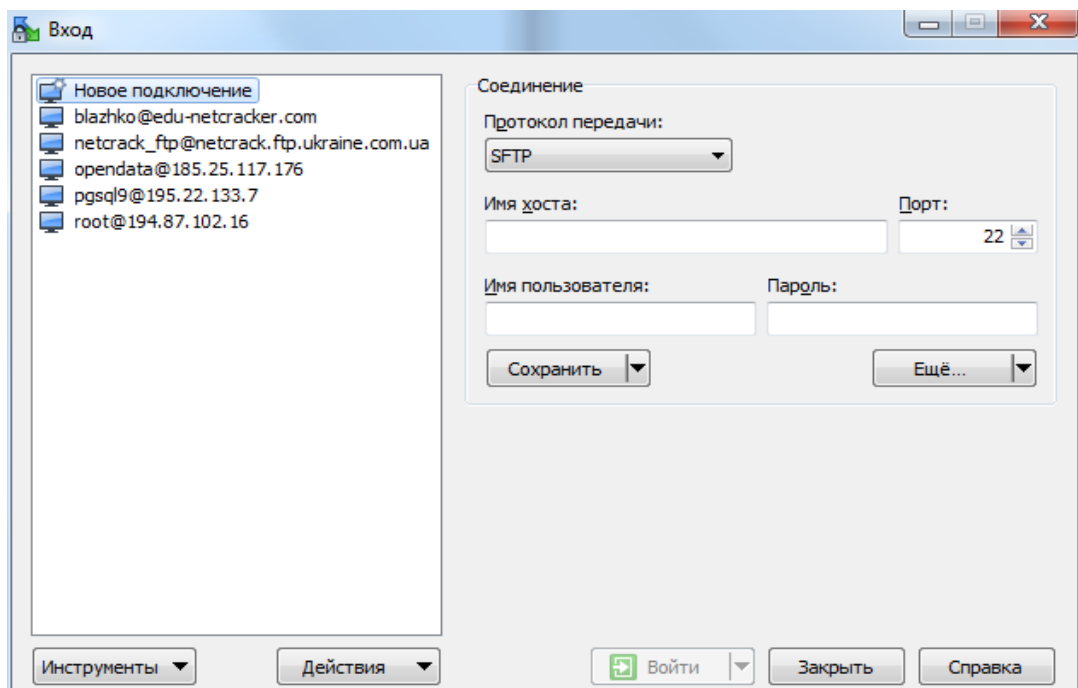


Рис. 7 – Приклад екранної форми налаштування з'єднання

Необхідно створити опис нового з'єднання, натиснувши кнопку «Нове підключення», і вказати параметри з'єднання з віддаленою ОС як показано:

Протокол передачі - *SFTP*

Ім'я хоста - 91.219.60.189

Ім'я користувача - ваше ім'я

Для подальшого підключення необхідно натиснути кнопку «Увійти», в результаті чого з'явиться екранна форма, в правій частині якої з'явиться вміст каталогу вашого користувача, а в лівій частині - вміст одного з каталогів файлової системи локальної ОС, як показано на рисунку 8.

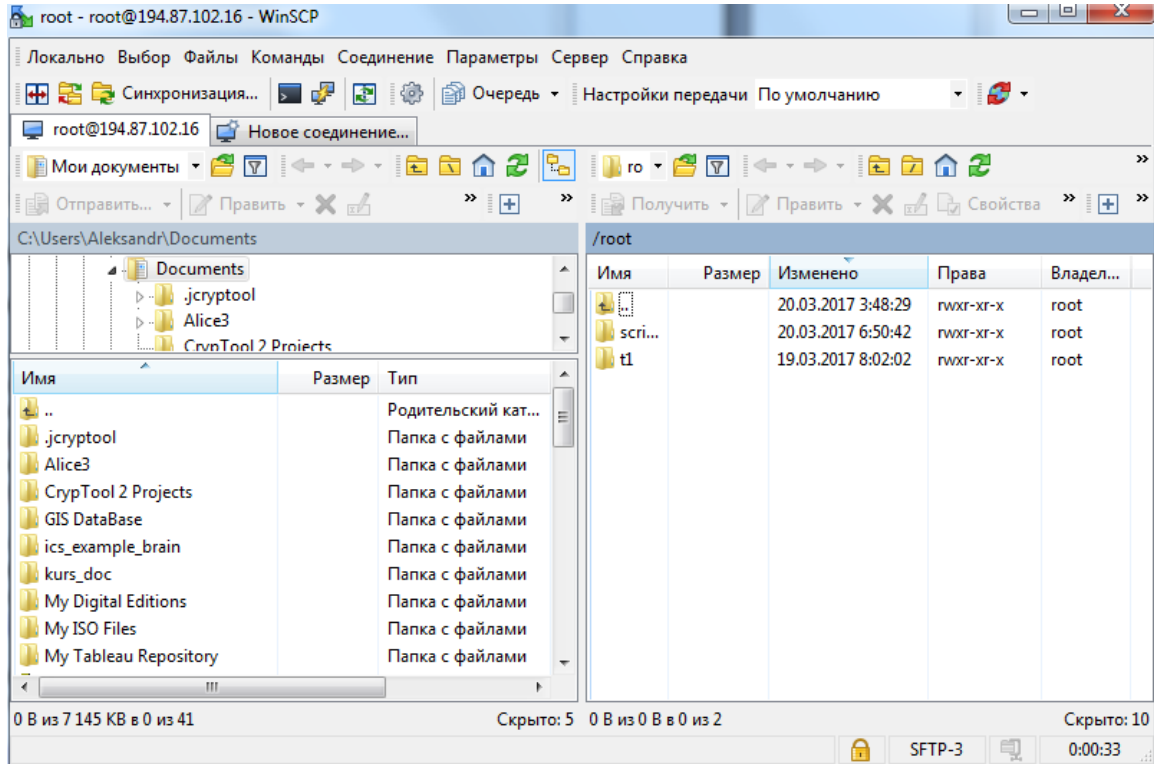


Рис. 8 – Приклад екранної форми із вмістом каталогів

З метою економії мережевого трафіку програма автоматично не оновлює зміст каталогів після початку роботи, тому для оновлення графічного змісту екранів рекомендується натискати графічний елемент «дві зелені стрілки».

2 Завдання для виконання

2.1 Робота з віддаленим *Linux*-сервером через термінальний режим роботи

1. Перевірте наявність мережевого з'єднання між вашим локальним комп'ютером та віддаленим сервером *Linux*, який знаходиться за мережевою адресою з *IP* = 91.219.60.189
2. Встановіть програму віддаленого доступу до ОС *Linux* *PUTTY.EXE*, отримавши її через інтернет за відповідним посиланням.
3. Запустіть програму *PUTTY.EXE* та налаштуйте з'єднання з віддаленим сервером: *IP* =91.219.60.189, тип з'єднання = *SSH*.
збережіть налаштування з урахуванням вашого прізвища та імені за назвою *OS_Linux_Прізвище_Ім'я* латиницею, наприклад, *OS_Linux_Blazhko_Oleksandr*
4. Встановіть з'єднання з віддаленим сервером *Linux*, використовуючи логін та пароль, наданий вам лектором.
5. Після успішного входу до ОС виконайте команду визначення поточного імені користувача.
6. Визначте мережеву адресу сервера, з якого ви увійшли до віддаленого серверу.
7. Визначте назву псевдотерміналу користувача.
8. Визначте назву поточного каталогу користувача.
9. Перейдіть до каталогу */etc*
10. Визначте вміст каталогу */etc*
11. Перегляньте вміст файлу з обліковими записами користувачів
12. Поверніться до каталогу свого користувача з використанням можливості швидкого повернення командами *pushd* та *popd*
13. Перейдіть до каталогу, який є на рівень вище (до батьківського каталогу)
14. Перегляньте вміст поточного каталогу
15. Змініть пароль вашого користувача.

2.2 Обмін повідомленнями в ОС *Linux*

1. Встановіть з'єднання з віддаленим *Linux*-сервером
2. Домовтеся з представником вашої команди про обмін повідомлень
3. Напишіть повідомлення представнику команди, в якому буде вказано ваше та його прізвище, а також дочекайтеся повідомлення від представника вашої команди.

2.3 Ініціалізація *Git*-репозиторію на віддаленому *Linux*-серверу

Знаходячись на віддаленому *Linux*-сервері в своєму домашньому каталозі, ініціалізуйте *Git*-репозиторій, виконавши команди:

1. Ініціалізуйте системний прихований каталог *Git*, використовуючи наступні команди:

```
git config --global user.name "Ваше Ім'я та Прізвище"  
git config --global user.email Ваш_E-Mail  
git config --global init.defaultBranch main  
git init
```
- 2 Клонуйте існуючий *GitHub*-репозиторій, створений в лабораторній роботі №1, використовуючи наступні команди:

```
git clone https://github.com/ПовнаНазваРепозиторію
```
- 3 Виконайте команди перегляду вмісту створеного каталогу *Git*-репозиторію

2.3 Обмін файлами між локальною ОС Windows, віддаленим Linux-сервером та GitHub-репозиторієм

1. Створіть в ОС *Windows* порожній текстовий файл з назвою, яка співпадає з вашим прізвищем та ім'ям в латиницю, наприклад, *Blazhko_Oleksandr.txt*
2. Запустіть в ОС *Windows* програму термінального рядка *CMD*
3. Встановіть програму віддаленого копіювання файлів *PSCP.EXE*, отримавши її через інтернет за відповідним посиланням.
4. Використовуючи програму *PSCP.EXE*, завантажте створений раніше файл у каталог з *Git*-репозиторієм, який раніше був створений в домашньому каталозі на віддаленому *Linux*-сервері.
5. Знаходячись в ОС *Linux*, в каталозі *Git*-репозиторію, зареєструйте завантажений раніше файл в репозиторії, виконавши команду:
git add Назва_файлу
6. Зафіксуйте зміни в репозиторії, вказавши коментар до цих змін, виконавши команду:
git commit -m 'будь-який коментар про нову версію вмісту репозиторію'
7. Перешліть зміни на віддалений *GitHub*-репозиторій, виконавши команду:
git push
8. Зайдіть на <https://github.com>, перегляньте вміст віддаленого *GitHub*-репозиторію та переконайтеся, що зміни успішно виконано.

3 Вимоги до оформлення протоколу виконання лабораторної роботи

Протокол виконання лабораторної роботи у *PDF*-форматі з назвою *ПрізвищеІм'я.pdf* повинен мати наступну структуру:

- 1) титульний аркуш з назвою дисципліни, теми лабораторної роботи, групи та ПІБ виконавця;
 - 2) аркуш із усіма пунктами завданням до лабораторної роботи;
 - 3) аркуші з результатами виконання пунктів завдань, які повинні містити:
 - пункт із завданням;
 - скріншот частини екрану з командами (з кольорами тексту білий фон/чорний тон);
 - скріншот частини екрану з результатом виконання команди;
 - 4) аркуш з висновками: перелік завдань, які були найскладнішими.
- Протокол необхідно розмістити на віддаленому *GitHub*-репозиторію.

4 Контрольні запитання

1. З якою метою створювались термінали?
2. Що таке ідентифікатор користувача ?
3. Для чого використовується програма *putty.exe* ?
4. Що таке безпечне з'єднання з сервером ?