

# Лабораторна робота №1

## Основи роботи з UNIX-подібними системами

**Мета роботи:** вивчення основних прийомів і отримання практичних навичок роботи з *UNIX*-подібними системами.

### Зміст:

Короткі теоретичні відомості.....	2
UNIX та UNIX-подібні системи.....	2
Доступ до UNIX-серверів .....	2
Клієнт доступу до серверів PuTTY .....	4
Користувач, привілейований користувач.....	11
Файлова система .....	12
Програми.....	13
Командна мова ОС Unix (shell) .....	13
Структура команд .....	15
Довідка по деяких командах.....	15
man .....	15
apropos .....	16
ls .....	16
mkdir.....	21
cp .....	21
mv .....	22
rm.....	23
Завдання для виконання .....	25
Контрольні питання .....	27

## Короткі теоретичні відомості

### *UNIX та UNIX-подібні системи*

**UNIX** – сімейство операційних систем, які є багатозадачними, ними можуть користуватися декілька користувачів та їх легко переносити.

Перша система **UNIX** була розроблена в 1969 році в підрозділі *Bell Labs* компанії *AT&T*. З тих пір була створена велика кількість різних **UNIX**-систем. Юридично лише деякі з них мають повне право називатися «**UNIX**»; інші ж, хоча і використовують подібні концепції і технології, поєднуються терміном «**UNIX**-подібні» (англ. *Unix-like*).

На теперішній час **UNIX**-системи широко використовуються на серверах, а також як вбудовані системи для різного устаткування. Також є варіанти і для особистого використання.

Серед прикладів відомих **UNIX**-подібних операційних систем є **BSD**, **Solaris**, **Linux**, **Android**, **MeeGo**, **NeXTSTEP**, **Mac OS X**, **Apple iOS**.

### *Доступ до UNIX-серверів*

Традиційно для доступу (роботи чи адміністрування) до роботи із сервером під керуванням **UNIX**-подібних системам можна використовувати клієнти термінального доступу.

Такі клієнти можуть використовувати різні протоколи доступу до сервера, наприклад **RLOGIN**, **TELNET** чи **SSH**. В даний час для віддаленого доступу до системи найчастіше застосовується мережевий протокол **SSH** (особливо його версія 2), оскільки він є більш захищеним. На відміну від **RLOGIN**, **TELNET**, протокол **SSH** шифрує весь трафік, включаючи і паролі, що передаються.

**SSH** дозволяє безпечно передавати в незахищеному середовищі практично будь-який інший мережевий протокол. **SSH** може використовувати стиснення даних, що передаються, що є зручним, наприклад, при роботі через повільні канали зв'язку.

Підтримка **SSH** реалізована у всіх **UNIX**-подібних системах, і на більшості з них в числі стандартних утиліт присутні клієнт і сервер **ssh**. Існує безліч реалізацій **SSH**-клієнтів і для не-**UNIX** ОС. Велику популярність протокол отримав після широкого розвитку аналізаторів трафіку і способів порушення роботи локальних мереж, як альтернатива незахищеному протоколу **Telnet** для управління важливими вузлами.

Для роботи по **SSH** потрібен **SSH**-сервер та **SSH**-клієнт. Сервер прослуховує з'єднання від клієнтських машин та при встановленні зв'язку виконує аутентифікацію, після чого починає обслуговування клієнта. Клієнт використовується для входу на віддалену машину та виконання команд.

Існує велика кількість **SSH**-клієнт для різних платформ, наприклад:

- GNU/Linux, \*BSD: kdessh, lsh-client, openssh-client, putty, ssh, Vinagre, Tectia SSH (SSH Communications Security) Client,
- Windows: PuTTY\KiTTY, SecureCRT, ShellGuard, Axessh, ZOC, SSHWindows, ProSSHD, XShell, Tectia SSH (SSH Communications Security) Client,
- MS Windows Mobile: PocketPuTTY, mToken, sshCE, PocketTTY, OpenSSH, PocketConsole, Tectia SSH (SSH Communications Security) Client,
- Mac OS: iTerm2, NiftyTelnet SSH, vSSH, ZOC,
- Symbian OS: PuTTY,
- iPhone: i-SSH, ssh (в комплекті із Terminal), Termius, OpenTerm, LibTerm, Shelly,
- Android: connectBot, Admin Hands, Server Auditor, JuiceSSH,
- Blackberry: BBSSH,
- MAEMO 5: OpenSSH,
- MeeGo 1.2 Harmattan: OpenSSH.

## ***Клієнт доступу до серверів PuTTY***

***PuTTY*** – це клієнт, який вільно розповсюджується, і його можна використовувати для різних протоколів віддаленого доступу до сервера, включаючи ***SSH, Telnet, rlogin***.

У ***PuTTY*** реалізована тільки клієнтська сторона з'єднання – сторона на якій щось відображається, в той час як сама робота виконується на стороні сервера. Клієнт ***PuTTY*** спочатку розроблявся для ***Microsoft Windows***, проте пізніше був перенесений і на ***Unix***.

***PuTTY*** входить в репозиторії практично всіх основних систем ***Linux*** (в т.ч. ***Ubuntu, Debian, ALT Linux***).

Код самої програми ***PuTTY*** повністю розроблений на ***C***. ***PuTTY*** не залежить від динамічних бібліотек, інших додатків, пакетів оновлень ОС. Пакет складається тільки з файлів що виконуються, які можуть бути встановлені в будь-якому місці. Програма є додатком, що вільно поширюється, із відкритим вихідним кодом і випускається під ***Open Source*** ліцензією ***MIT***.

Скачати останню версію ***PuTTY*** клієнта або його інсталера можна на офіційній сторінці:

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Потрібні для роботи із ***Microsoft Windows*** чи ***Linux*** інсталери наведені червоним на рисунку нижче:

## Package files

You probably want one of these. They include versions of all the PuTTY utilities (except the new ones). (Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

We also publish the latest PuTTY installers for all Windows architectures as a free-of-charge download.

### MSI ('Windows Installer')

64-bit x86: [putty-64bit-0.81-installer.msi](#) ([signature](#))

64-bit Arm: [putty-arm64-0.81-installer.msi](#) ([signature](#))

32-bit x86: [putty-0.81-installer.msi](#) ([signature](#))

### Unix source archive

.tar.gz: [putty-0.81.tar.gz](#) ([signature](#))

Поточну лабораторну роботу можна виконувати на сервері, до якого можна підключитись за допомогою клієнта *PuTTY*.

Для кожного студента створений персональний обліковий запис виду:

***dmXXYY (чи dpXXYY)***

де *XX* – номер (шифр) групи, *YY* - порядковий номер за списком (див. журнал успішності). Тобто номер *XXYY* повинен співпадати із номером залікової книжки.

Наприклад, перший студент по списку в групі ДМ-11 буде мати обліковий запис (логін) *dm1101*.

За бажанням, після першого запуску програми, можна змінити пароль, набравши в робочому рядку команду *passwd*.

Якщо на комп'ютері, на якому планується виконання лабораторної роботи, використовується ОС *Linux*, *MacOS* чи щось інше, що має вбудований термінал та *ssh* клієнт, то достатньо набрати в терміналі наступну команду:

```
ssh sandbox.ee.kpi.ua -p 6022 -l $LOGIN
```

де \$LOGIN – це ваш логін на сервері як це було описано вище (тобто перший студент в списку в групі ДМ-11 повинен набрати в командному рядку наступну команду: “ssh sandbox.ee.kpi.ua -p 6022 -l \$dm1101”).

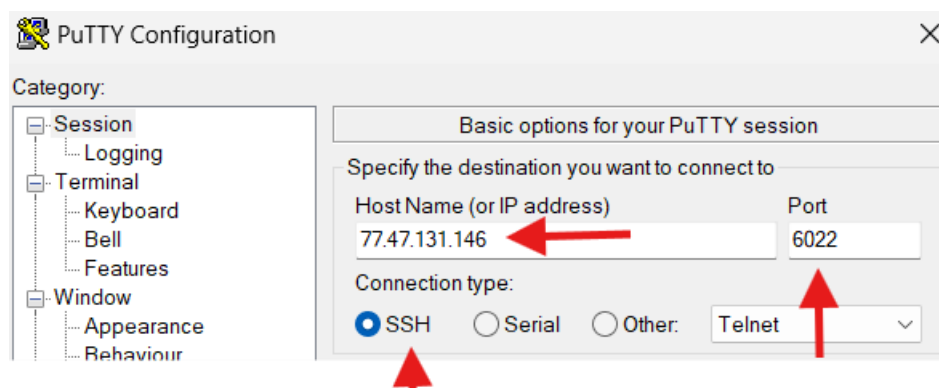
ПРИМІТКА 1: зверніть увагу, що після номеру порту (6022) записується ключ із латинської букви Ель, а не одиниця.

ПРИМІТКА 2: інколи, коли не вдається підключитися із вищевказаною інструкцією, спробуйте скористатися наступною командою: “ssh 77.47.131.146 -p 6022 -l \$dm1101”.

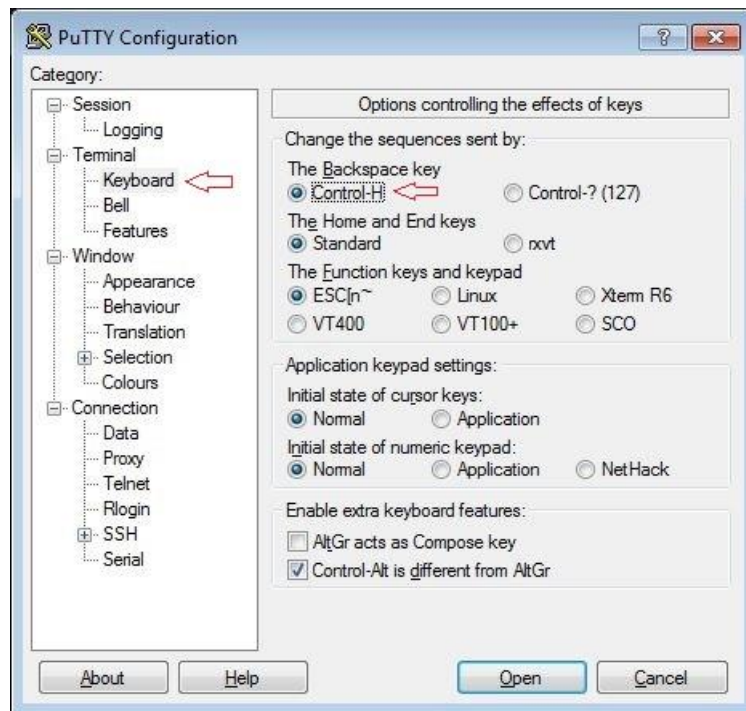
Якщо планується використовувати клієнт **PuTTY**, то після установки та запуску клієнта, для доступу до сервера, необхідно виконати наступні налаштування:

- a. Вказати ім'я сервера, протокол *SSH* та порт, відповідно до вказівок адміністратора. В нашому випадку ми повинні використовувати наступні налаштування:

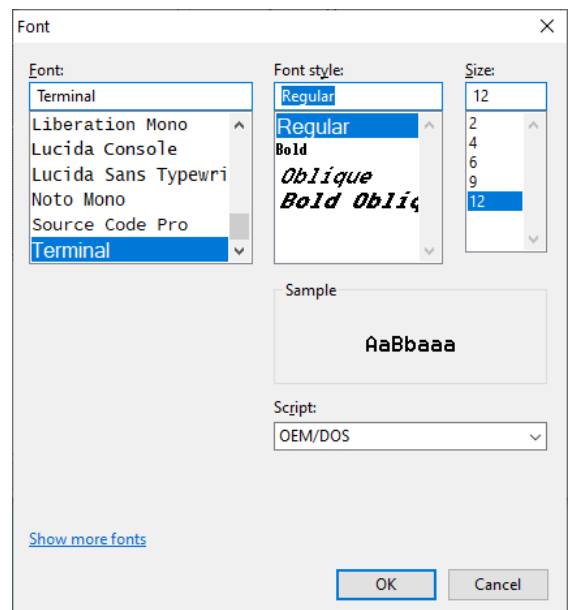
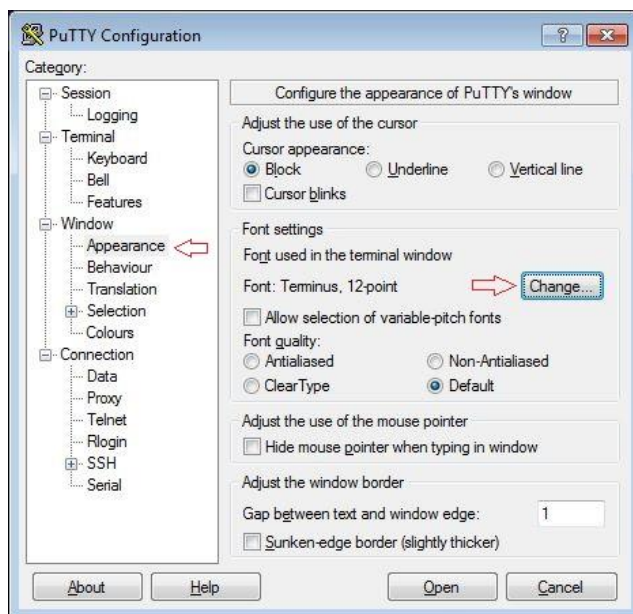
Налаштування	Ми працюємо	
	З Інтернету	В мережі КПІ
<i>Host Name:</i>	77.47.131.146	<i>sandbox.ee.kpi.ua</i>
<i>Port:</i>	6022	22
<i>Connection type:</i>	<i>SSH</i>	<i>SSH</i>



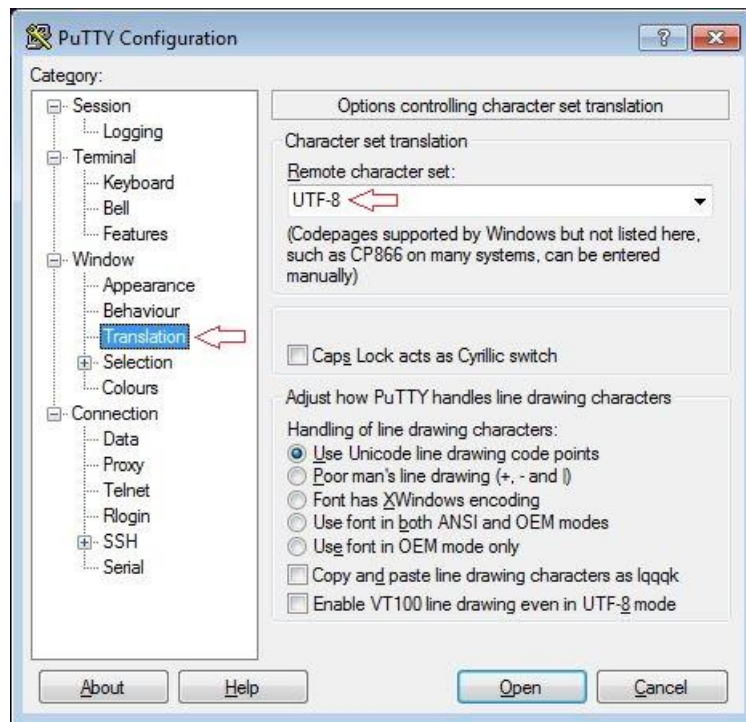
- b. Встановити трансляцію кодів клавіатури:



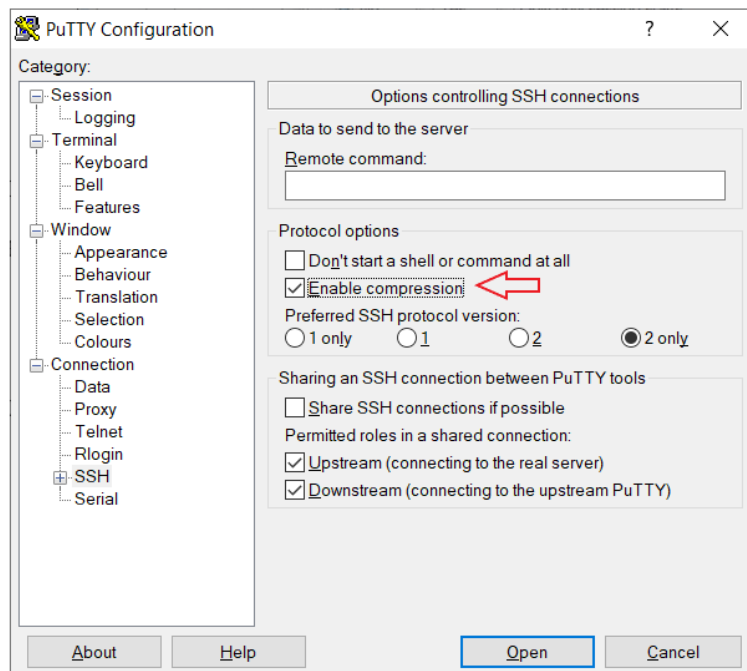
с. Встановити бажаний тип та розмір шрифту для консолі:



д. Встановити трансляцію кодування:



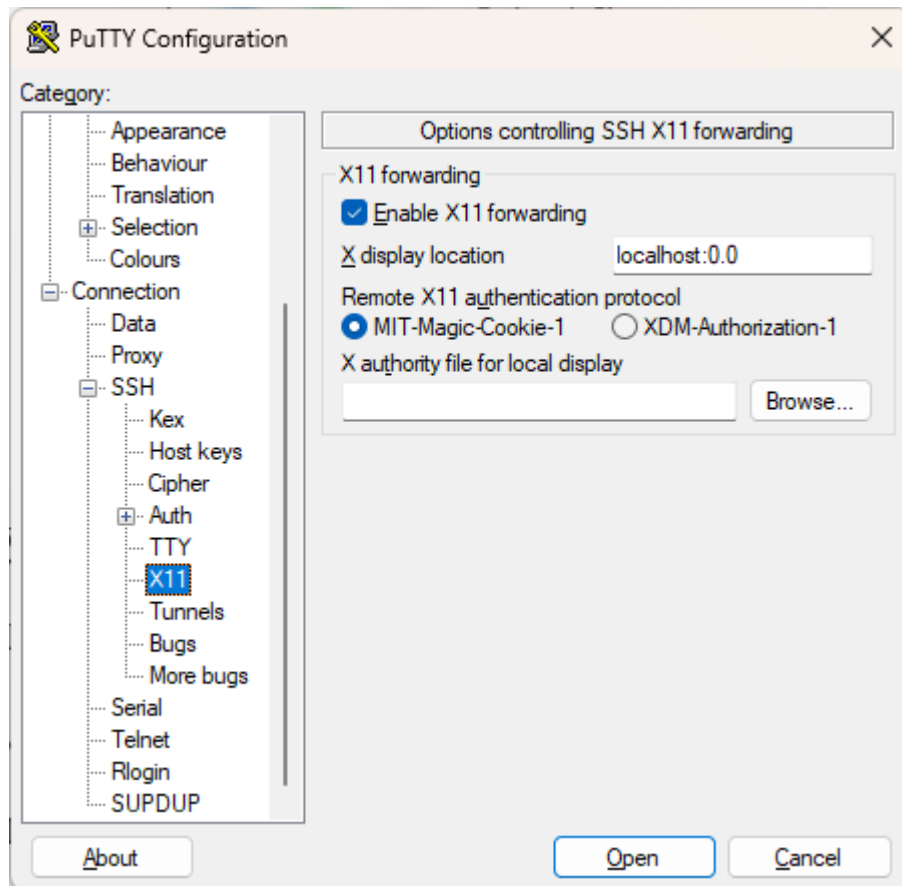
- е. При повільному інтернет з'єднанні комфорт роботи можна поліпшити вмиканням стиснення даних:



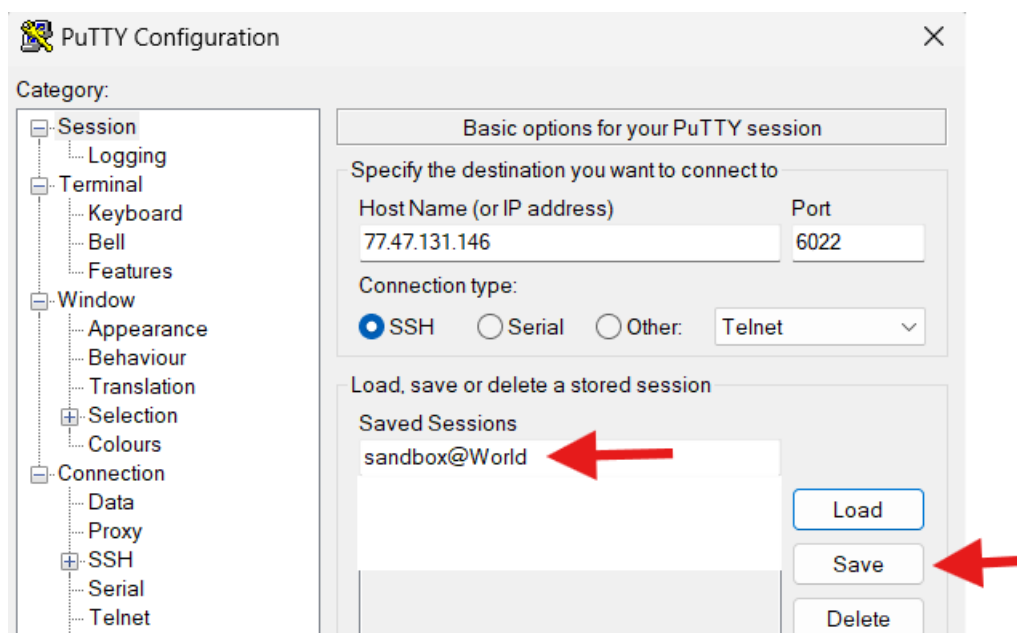
- ф. Налаштувати пересилку графічного зображення виконуваних програм. Для потреб поточної лабораторної роботи це не обов'язково, але оскільки ми збережемо ці налаштування, є сенс подбати про них. Для того, щоб графіка виконуваної програми відображалась на віддаленій



машині, слід виконати додаткові налаштування термінального клієнта PuTTY, а саме у підменю Connection/SSH/X11 дозволити пересилку (Enable X11 forwarding) та вказати розташування програми отримувача (localhost:0.0).

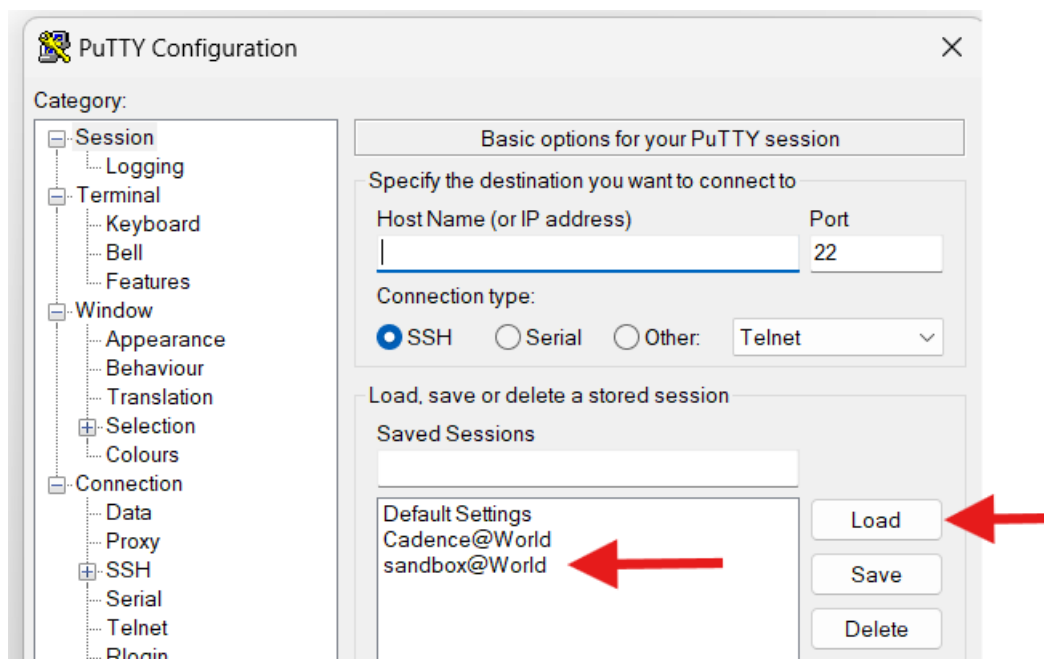


г. І нарешті зберегти профіль з'єднання для подальшого використання:

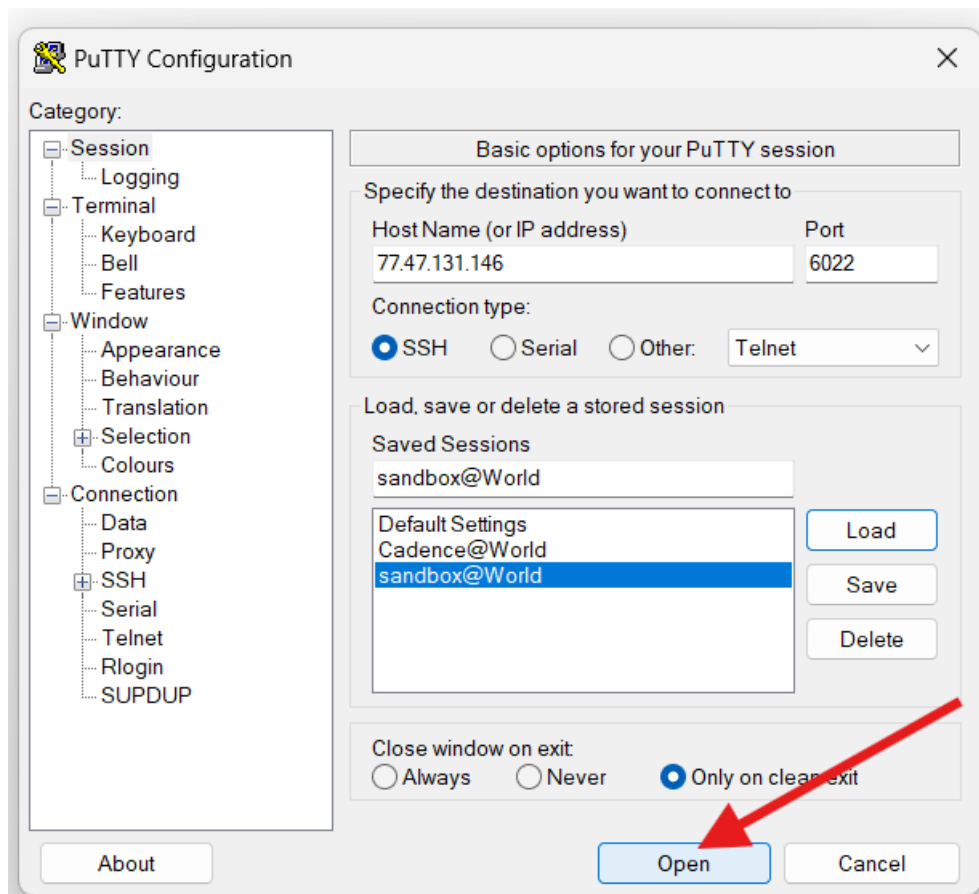


Після підключення до сервера, у консолі **PuTTY** можна запускати програми.

При наступних підключеннях до сервера не потрібно заново виконувати усі ці налаштування – достатньо завантажити збережені налаштування за допомогою кнопки *Load*, вибравши потрібний профіль зі списку.



Після цього вже можна підключитися натиснувши кнопку **Open**:



### ***Користувач, привілейований користувач***

Ядро ОС *UNIX* визначає права та можливості кожного користувача за його ідентифікатором (*UID - User Identifier*), який є унікальним цілим значенням, що присвоюється користувачеві при реєстрації в системі. Крім того, кожен користувач відноситься до певної групи користувачів, яка також ідентифікується деяким цілим значенням (*GID - Group Identifier*). Значення *UID* і *GID* для кожного зареєстрованого користувача зберігаються в облікових файлах системи та приписуються процесу, в якому виконується командний інтерпретатор, запущений при вході користувача в систему. Ці значення успадковуються кожним новим процесом, запущеним від імені даного користувача, і використовуються ядром системи для контролю правомочності доступу до файлів, виконання програм і т.д.

Адміністратор системи, який теж є зареєстрованим користувачем, повинен мати більші можливості, ніж звичайні користувачі. В ОС *UNIX* це

завдання вирішується шляхом виділення одного значення *UID* (нульового). Користувач з таким *UID* називається суперкористувачем (*superuser*) або *root*. Він має необмежені права на доступ до будь-якого файлу і на виконання будь-якої програми.

### **Файлова система**

Будь який файл ОС *UNIX* у відповідності з його типом може бути віднесений до однієї з наступних чотирьох груп: звичайні файли, каталоги, спеціальні файли, канали.

Файлова система ОС *UNIX* має ієрархічну (деревоподібну) структуру. У вершинах дерева знаходяться каталоги (директорії), що містять списки файлів. Ці файли в свою чергу можуть бути або знову каталогами, або звичайними файлами, або спеціальними файлами, що представляють різні пристрої введення-виведення.

На відміну від інших ОС, в *UNIX*-подібних системах не використовуються логічні диски. Усі наявні носії належать до єдиного дерева, вершиною якого є кореневий каталог. Кореневий каталог має ім'я *"/*". Він, зазвичай, містить каталоги:

**bin** для програм, які найбільше використовуються;

**dev** для спеціальних файлів, що представляють пристрої (дисплеї, диски, ...);

**etc** для зберігання налаштувань системи;

**lib** найважливіші бібліотеки;

**mnt** для підключення (монтування) нових файлових систем;

**sys** засоби для зміни конфігурації системи;

**tmp** для зберігання тимчасових файлів;

**usr** каталоги і звичайні файли, які залучаються при розв'язанні задач користувача.

Зверніть увагу, що на відміну від деяких інших ОС, для відокремлення імен каталогів використовується символ «прямий слеш» "/", а не "\".

Кожному зареєстрованому користувачеві відповідає деякий каталог файлової системи, який називається «домашнім» (*home*) каталогом користувача. При вході в систему користувач зазвичай одержує необмежений доступ до свого домашнього каталогу і всіх каталогів і файлів, що містяться в ньому. Потенційно можливий доступ і до всіх інших файлів, однак він може бути обмежений, якщо користувач не має достатніх привілеїв.

Доступ до домашнього каталогу можна отримати за допомогою його псевдоніма `~`, наприклад, команда

```
cd ~/work
```

здійснює перехід в підкаталог *work* домашнього каталогу поточного користувача.

### *Програми*

Програма, що виконується може бути запущена в інтерактивному режимі як команда **shell** або виконана в окремому процесі, утвореному вже запущеною програмою.

### *Командна мова ОС Unix (shell)*

Традиційний спосіб взаємодії користувача з *UNIX*-подібною системою ґрунтується на використанні командних мов.

Зазвичай в системі підтримується кілька командних інтерпретаторів зі схожими командними мовами, які можуть мати різні можливості. Загальна

назва для будь-якого командного інтерпретатора ОС *UNIX* – це **shell** (оболонка), оскільки будь-який інтерпретатор є зовнішнім оточенням ядра системи.

Працюючи з командною мовою, користувач може вводити змінні, присвоювати їм значення, виконувати прості команди, управляти потоком виконання команд, об'єднувати послідовність команд в процедури (командні файли). На рівні командної мови доступні такі властивості системи як поєднання процесів через програмний канал, напрямок стандартного вводу/виводу в конкретні файли, синхронне та асинхронне виконання команд.

Командна мова **shell** фактично є мовою програмування дуже високого рівня. На цій мові користувач здійснює управління комп'ютером. Зазвичай, після входу в систему ви починаєте взаємодіяти з командною оболонкою. Ознакою того, що оболонка (**shell**) готова до прийому команд є запрошення (промптер), яке з'являється на екрані. У найпростішому випадку це символ долара ("**\$**").

**Shell** не є необхідною і єдиною командною мовою. Наприклад, чималою популярністю користується мова **csh**, є також **ksh**, **bash** та інші.

**ЗВЕРНІТЬ УВАГУ.** **shell** - це одна з багатьох команд *UNIX*. Тобто в набір команд оболонки (інтерпретатора) "**shell**" входить команда "**sh**" - виклик інтерпретатора "**shell**". Перший "**shell**" викликається автоматично при вашому вході в систему і видає на екран промптер. Після цього ви можете викликати на виконання будь-які команди, в тому числі і знову сам "**shell**", який вам створить нову оболонку всередині колишньої.

Командні мови, що використовуються в ОС *UNIX*, достатньо прості, щоб нові користувачі могли швидко почати працювати, і досить потужні, щоб можна було використовувати їх для написання складних програм. Остання можливість спирається на механізм командних файлів (**shell scripts**), які можуть складатися з довільної послідовності командних рядків. Якщо

викликати командний файл замість чергової команди інтерпретатор читає файл рядок за рядком і послідовно інтерпретує команди.

### **Структура команд**

Будь яка командна мова сімейства **shell** фактично складається з трьох частин:

- ✓ службових конструкцій, що дозволяють маніпулювати з текстовими рядками і будувати складні команди на основі простих команд;
- ✓ вбудованих команд, які виконуються безпосередньо інтерпретатором командної мови;
- ✓ команд.

Команди в **shell** зазвичай мають наступний формат:

*<ім'я команди> <ключ(i)> <аргумент(u)>*

наприклад:

*ls -s /usr/bin*

*ls*            ім'я команди видачі вмісту директорія,  
*-s*            ключ ("-") - ознака ключів, *s* - обсяг файлів в блоках).  
*/usr/bin*     директорія, для якої виконується команда.

### **Довідка по деяких командах**

#### **man**

У більшості **UNIX**-подібних систем є вбудована детальна довідкова інформація по більшості команд та програм. Для отримання довідки по команді використовуйте команду *man*, наприклад:

*man ls*

дозволяє отримати довідку на команду *ls*.

Для виходу із режиму довідки потрібно натиснути клавішу “q”.

## **apropos**

Для пошуку довідки за ключовими словами, використовуйте команду *apropos*.

Наприклад, команда

*apropos “working directory”*

дозволяє знайти команди, які працюють з поточним каталогом, наприклад знайти команду, яка служить для виведення поточного каталогу.

## **ls**

*ls* – видача інформації про файли або каталоги.

### **СИНТАКСИС**

*ls [-R][-a][-d][-C][-x][-m][-l][-n][-o][-g][-r][-t][-u][-c][-p][-F][-b][-q][-i][-s][-f] [ім'я ...]*

### **ОПИС**

Команда *ls* для кожного імені каталогу роздруковує список файлів які містяться у поточному каталозі; для файлів – повторюється ім'я файлу і виводиться додаткова інформація відповідно до зазначених прапорів. За замовчуванням імена файлів виводяться в алфавітному порядку. Якщо імена не задані, видається вміст поточного каталогу. Якщо задані кілька аргументів, то вони сортуються за алфавітом, проте спочатку завжди йдуть файли, а потім каталоги з їх вмістом.

За замовчуванням результат видається по одному файлу в рядку; ключі **-C** та **-x** дозволяють видавати інформацію в декілька колонок, а ключ **-m** задає вільний формат.

Команда *ls* може використовуватися із такими ключами:

- R** Рекурсивно обійти підкаталоги, які зустрічаються.
- a** Вивести список всіх файлів (зазвичай не виводяться файли, імена яких починаються з крапки).



- d** Якщо аргумент є каталогом, то виводити тільки його ім'я, а не вміст. Часто використовується з ключем **-l** для отримання відомостей про стан каталогу.
- C** Вивід у кілька колонок з сортуванням по колонках.
- x** Вивід у кілька колонок з сортуванням по рядках.
- m** Вивід у вільному форматі, імена файлів розділяються комами.
- l** Вивід в довгому форматі: перед іменами файлів видається режим доступу, кількість посилань на файл, імена власника та групи, розмір у байтах і час останньої модифікації (див. нижче). Якщо файл є спеціальним, то в поле розміру виводиться старший і молодший номери пристрою.
- n** Те ж, що і **-l**, але ідентифікатори власника та групи виводяться у вигляді чисел, а не у вигляді імен.
- o** Те ж, що і **-l**, але ідентифікатор групи не виводиться.
- g** Те ж, що і **-l**, але ідентифікатор власника не виводиться.
- r** Змінити порядок сортування на зворотний алфавітний або, при наявності ключа **-t**, спочатку виводити старіші файли.
- t** Імена файлів сортуються не за алфавітом, а за часом (спочатку йдуть найсвіжіші файли). За замовчуванням використовується час останньої зміни.
- u** Замість часу останньої зміни використовувати час останнього доступу для сортування (з ключем **-t**) або для виводу (з ключем **-l**).
- c** Замість часу останньої зміни використовувати час останньої модифікації описувача файлу (тобто час створення файлу, зміни режиму доступу до нього і т.п.) для сортування (з ключем **-t**) або для виводу (з ключем **-l**).
- p** Якщо файл є каталогом, то видавати після його імені символ **/**.
- F** Якщо файл є каталогом, то видавати після його імені символ **/**; якщо файл може виконуватися, то видавати після його імені символ **\***.

- b** Видавати недруковані символи, що входять до імені файлу, у вісімковому вигляді (`\ddd`).
- q** Видавати недруковані символи, що входять до імені файлу, у вигляді символа ?.
- i** Видавати в першій колонці номери дескрипторів файлів.
- s** Видавати розмір файлів в блоках.
- f** Розглядати кожен аргумент як каталог і виводити його вміст. Цей ключ скасовує ключі **-l**, **-t**, **-s**, **-r** і включає ключ **-a**. Сортування імен файлів не проводиться; імена видаються в такому порядку, в якому вони перераховані в каталозі.

Режим доступу до файлу з ключем **-l** виводиться у вигляді 10 символів.

Перший символ означає:

- d** Файл є каталогом.
- b** Файл є спеціальним блочним файлом.
- c** Файл є спеціальним символьним файлом.
- p** Файл є іменованим каналом.
- Звичайний файл.

Решта 9 символів поділяються на три групи по три символи:

- ✓ 3 символи з правами доступу власника,
- ✓ 3 символи з правами доступу інших користувачів з його групи,
- ✓ 3 символи з правами доступу всіх інших користувачів.

У середині кожної групи використовуються три символи, що позначають права на читання, запис і виконання файлу відповідно. Для каталогу під правом на виконання маєтись на увазі право на перегляд у пошуках потрібного файлу.

При використанні команди

`ls -l /util/by`

результат може виглядати наступним чином:

```
-rwxr-xr-x 1 root sys 50 Jun 22 10:42 /util/by
```

Читаючи справа наліво можна побачити, що вміст файлу */util/by* останній раз змінювався о 10 годині 42 хвилини 22 січня. Розмір файлу 50 байт. Власником цього файлу належить є суперкористувачем (вхідне ім'я – *root*), також цей файл асоційовано з групою *sys*. Наступне число, в даному випадку 1, позначає кількість посилань на файл */util/by*. Нарешті, послідовність мінусів і букв вказує, що власник, члени групи та інші користувачі можуть читати і виконувати файл, а власник (і тільки він) має право писати у файл.

Права доступу позначаються наступним чином:

*r* Право на читання.

*w* Право на запис.

*x* Право на виконання (для каталогів – право на читання вмісту каталогу).

- Дане право доступу відсутнє.

*l* Враховується блокування доступу (біт переустановки ідентифікатора групи дорівнює 1, біт права на виконання членами групи дорівнює 0). Розташовується на місці права на виконання для членів групи.

*s* Право встановлювати заново ідентифікатор групи або ідентифікатор власника та право виконання файлу для членів групи або власника.

*S* Невизначена комбінація біт: право встановлювати заново ідентифікатор власника є, а право виконання файлу для власника відсутнє.

*t* Встановлено «липкий» біт [см. команду *chmod* для більш детальної інформації] у файла, який можуть виконувати інші користувачі. Розташовується на місці права на виконання для інших користувачів.

*T* «Липкий» біт встановлений, а права на виконання у інших користувачів немає. Розташовується на місці права на виконання для інших користувачів.

## ПРИКЛАДИ

1. Якщо файл має режим

*-rwxr--r--*

він доступний власнику для читання, запису і виконання, а членам групи і іншим користувачам – тільки для читання.

2. Режим

*-rwsr-xr-x*

свідчить про те, що файл доступний власнику для читання, запису і виконання, а членам групи і іншим користувачам тільки для читання та виконання. Дозволена переустановка при виконанні ідентифікатора користувача на ідентифікатор власника файлу.

3. По команді

*ls -a*

будуть виведені імена всіх файлів в поточному каталозі, включаючи і ті, які починаються з точки («приховані»), і звичайно не видаються.

4. По команді

*ls -aisn*

буде видана різноманітна інформація: список всіх файлів, включаючи ті, які зазвичай не виводяться (*-a*); номери описувачів файлів будуть виведені в лівій колонці (*-i*); розміри файлів (в блоках) виводяться в другій колонці (*-s*); нарешті, будуть видані числові ідентифікатори власників та груп (*-n*).

## **mkdir**

*mkdir* – створення каталогу.

### **СИНТАКСИС**

*mkdir [-m режим\_доступу] [-p] каталог ...*

### **ОПИС**

За командою *mkdir* створюється один або кілька каталогів з режимом доступу «*rwX*» (читання, запис і право на пошук доступні всім). Стандартні файли створюються автоматично; їх не можна створити по імені. Для створення каталогу необхідно мати право запису в поточному каталозі.

Ідентифікатори власника і групи нових директорій встановлюються відповідно рівними реальним ідентифікаторам власника та групи процесу.

Командою *mkdir* обробляються дві опції:

**m** Режим доступу. Явне завдання режиму доступу для створюваних каталогів [см. команду *chmod* для більш детальної інформації].

**p** При використанні цієї опції перед створенням нового каталогу попередньо створюються всі неіснуючі проміжні каталоги.

### **ПРИКЛАДИ**

Щоб створити піддерево каталогів *tmpdir/temp/dir*, потрібно виконати команду

*mkdir -p tmpdir/temp/dir*

## **cp**

*cp* – копіювання файлів

### **СИНТАКСИС**

*cp файл1 [файл2 ...] цільовий\_файл*

### **ОПИС**

Команда *cp* копіює *файл1* в *цільовий\_файл*. *Файл1* не повинен співпадати з *цільовим\_файлом* (будьте уважні при використанні метасимволів **shell'a**).

Якщо *цільовий\_файл* є каталогом, то *файл1*, *файл2*,..., копіюються в нього під

своїми іменами. Тільки в цьому випадку можна вказувати кілька вихідних файлів.

Якщо *цільовий\_файл* існує і не є каталогом, його попередній вміст втрачається.

Режим доступу власника та групи *цільового\_файла* при цьому не змінюється.

Якщо *цільовий\_файл* не існує або є каталогом, нові файли створюються з тими ж режимами, що і вихідні (крім «липкого» біта, якщо ви не суперкористувач).

Час останньої модифікації *цільового\_файла* (і останнього доступу, якщо він не існував), а також час останнього доступу до вихідних файлів встановлюється рівним часу, коли виконується копіювання. Якщо *цільовий\_файл* був посиланням на інший файл, всі посилання зберігаються, а вміст файлу змінюється.

## **mv**

*mv* – переміщення (перейменування) файлів

### **СИНТАКСИС**

*mv [-f] файл1 [файл2 ...] цільовий\_файл*

### **ОПИС**

Команда *mv* переміщує (перейменовує) *файл1* в *цільовий\_файл*. *Файл1* не повинен співпадати з *цільовим\_файлом*. Якщо *цільовий\_файл* є каталогом, то *файл1*, *файл2*, ..., переміщуються в нього під своїми іменами. Тільки в цьому випадку можна вказувати кілька вихідних файлів

Якщо *цільовий\_файл* існує і не є каталогом, його старий вміст втрачається.

Якщо *файл1* є каталогом, то він перейменовується в *цільовий\_файл*, тільки якщо у цих двох каталогів загальний надкаталог; при цьому всі файли, що знаходилися в *файл1*, переміщуються під своїми іменами в *цільовий\_файл*.

## **rm**

*rm* - видалення файлів або каталогів

### **СИНТАКСИС**

*rm [-f] [-i] файл ...*

*rm -r [-f] [-i] каталог ... [файл ...]*

### **ОПИС**

Команда *rm* служить для видалення зазначених файлів з каталогу. Якщо задане ім'я є посиланням на файл, то файл знищується. Для видалення користувач повинен мати **право запису в каталог**; мати **право на читання або запис файлу не обов'язково**.

Якщо немає права на запис у файл і стандартний вивід є термінал, то видається режим доступу до файлу і запитується підтвердження; якщо воно починається з літери *y*, то файл видаляється, інакше – ні. Якщо стандартний ввід не є термінал, команда *rm* поводитьсь так само, як при наявності опції *-f*.

Допускаються такі опції:

- f** Команда не видає повідомлень, коли видаляється файл який не існує, не запитує підтвердження при видаленні файлів, на запис в які немає прав. Якщо немає права і на запис у каталог, файли залишаються. Повідомлення про помилку видається лише при спробі видалити каталог, на запис в який немає прав (див. опцію *-r*).
- r** Відбувається рекурсивне видалення всіх каталогів і підкаталогів, перелічених у списку аргументів. Спочатку каталоги спустошуються, потім видаляються. Підтвердження при видаленні файлів, на запис в які немає прав, не запитується, якщо задана опція *-f* або стандартний ввід не є термінал і не задана опція *-i*.
- i** Перед видаленням кожного файлу запитується підтвердження. Опція *-i* усуває дію опції *-f*; вона діє навіть тоді, коли стандартний ввід не є термінал.

**ПРИКЛАДИ** Опція *-i* часто використовується разом з *-r*. По команді:

*rm -ir dirname*

запитуються підтвердження:

*directory dirname: ?*

При позитивній відповіді запитуються підтвердження на видалення всіх розміщених в каталозі файлів (для підкаталогів виконуються ті ж дії), а потім підтвердження на видалення самого каталогу.

Прикладом частого **руйнівного жарту** над новачками є порада набрати команду:

*rm -rf* або *rm -rf/*

яка рекурсивно видаляє всі файли та каталоги від поточного, чи від кореневого каталога, що, відповідно, призводить до втрати даних і, можливо, пошкодження операційної системи.

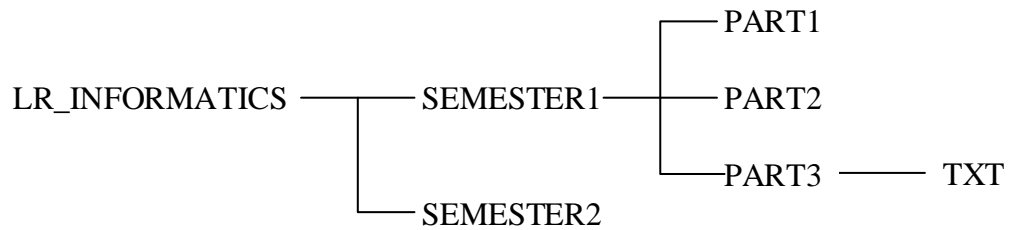


## Завдання для виконання

В цій лабораторній роботі потрібно використовувати тільки засоби командного рядка.

1. Встановити клієнт *PuTTY* та виконати усі його налаштування, як показано в теоретичних відомостях до цієї лабораторної роботи (якщо на комп'ютері є будь-який інший SSH-клієнт, то *PuTTY* можна не встановлювати, а користуватися зручним для вас клієнтом).
2. Увійдіть у систему, використовуючи як свій логін та пароль за замовчуванням. Змінити пароль за замовчуванням на більш зручний та запам'ятати його.
3. Продемонструвати вміння користуватися довідкою в UNIX-подібній системі (команда *man*), вивівши на екран інформацію про команду *uname*. Визначте тип і версію операційної системи нашого сервера.
4. За допомогою команди *apropos 'working directory'* вивести на екран основні команди роботи з робочим каталогом.
5. Вивчіть довідку по команді *ls*, звертаючи увагу на вплив ключів на роботу команди.
6. Перегляньте вміст домашнього каталогу в декількох форматах (скориставшись командою *ls* із відповідними ключами) та поясніть отриманий результат:
  - ✓ здійснити перегляд каталогу з виводом файлів в кілька колонок без детальної інформації про файли;
  - ✓ здійснити перегляд каталогу у довгому форматі (із зазначенням режиму доступу, власника, групи та датою останньої модифікації) з переглядом розміру файлів в блоках;
  - ✓ здійснити перегляд у вільному форматі – імена файлів повинні виводитися через кому.
7. Перегляньте вміст будь-якого каталогу у системі, який знаходиться вище вашого домашнього каталогу.

8. Створіть у домашньому каталозі структуру каталогів як показано на малюнку:



9. До каталогу *PART1* скопіюйте декілька будь-яких файлів (можна створити свої файли або скопіювати із будь-якого каталогу, до якого є доступ, наприклад, можна скопіювати кілька текстових файлів налаштувань з каталогу */etc*).
10. Зробити копії файлу із каталога *PART1* в каталоги *PART2* та *TXT*.
11. Виведіть на екран вміст цих каталогів.
12. Видаліть рекурсивно каталог *PART3*.
13. Видаліть рекурсивно каталог *LR\_INFORMATICS*.
14. Виведіть на екран і перегляньте один з скопійованих текстових файлів.

## Контрольні питання

1. У чому принципові відмінності між звичайним користувачем і суперкористувачем в *UNIX*-подібних системах?
2. Як можна подивитися права доступу до конкретного файлу (директорії) певного користувача?
3. Які команди роботи з каталогами в *UNIX*-подібних системах вам відомі?
4. Які команди роботи з файлами в *UNIX*-подібних системах вам відомі?
5. Чи змінюють ключі роботу команд в *UNIX*-подібних системах?