

Лабораторна робота №2

Робота з UNIX-подібними системами: маніпуляції із файлами

Мета роботи: вивчення основних прийомів роботи з файлами в *UNIX*-подібних системах.

Зміст:

Установка необхідного ПЗ	2
Графічний клієнт для копіювання файлів між сервером та комп'ютером для Windows	2
VcXsrv чи Xming	6
Короткі теоретичні відомості.....	9
Файлова система	9
Довідка по деяких командах.....	9
cat	9
Зміна напрямку вводу/виводу.....	11
Групування команд.....	12
Текстові редактори	13
nedit	13
joe	14
nano	15
vi	16
Завдання для виконання	23
Контрольні питання	25

Установка необхідного ПЗ

Графічний клієнт для копіювання файлів між сервером та комп'ютером для Windows

Для завантаження файлів зі свого комп'ютера під керівництвом *Windows* на сервер під керівництвом **UNIX**-подібної системи, і навпаки зручно користуватися програмою *WinSCP*.

WinSCP – це графічний клієнт протоколів *SFTP* та *SCP*, призначений для *Windows*. Має відкритий вихідний код та поширюється за ліцензією *GNU GPL*. Забезпечує захищене копіювання файлів між комп'ютером і серверами, що підтримують ці протоколи. Завдяки графічному інтерфейсу, звичному користувачам, спрощує освоєння початківцями.

Основні можливості WinSCP:

- Графічний інтерфейс в стилі провідника *Windows*.
- Всі основні файлові операції – копіювання, видалення і т. д.
- Автоматизація за допомогою скриптів і інтерфейсу командного рядка.
- Інтеграція з *Pageant (PuTTY Agent)* з підтримкою авторизації по відкритих ключах.
- Інтеграція з *Windows* (підтримка *Drag & Drop*, ярликів, підтримка схем *URL*).
- Робота з ключами і версіями протоколу *SSH*.
- Вбудований текстовий редактор.
- Підтримка різних типів авторизації: по паролю, аутентифікації з відкритим ключем, *Kerberos*.
- Можливість зберігати налаштування з'єднань.

- Локалізації інтерфейсу для кількох десятків мов.

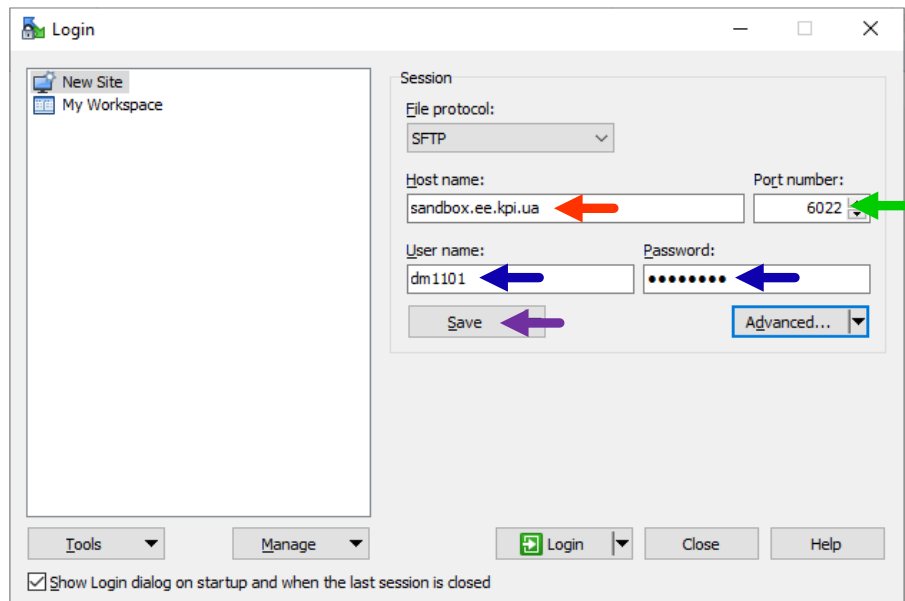
Для завантаження *WinSCP* можна скористатися посиланням:

<https://winscp.net/eng/download.php>

Після інсталяції ярлик *WinSCP* буде виглядати наступним чином:

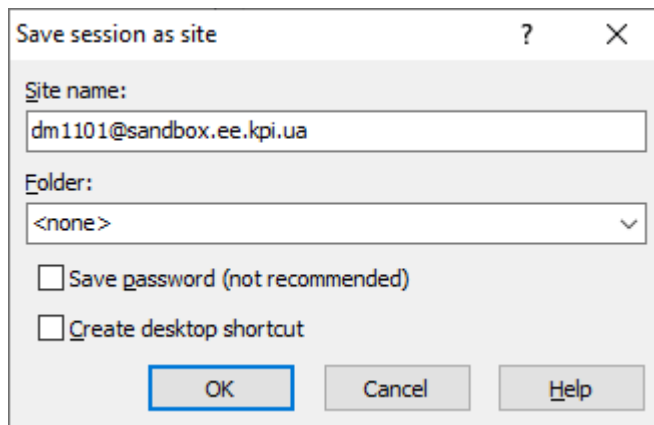


Після запуску *WinSCP* відкривається вікно налаштування з'єднання:



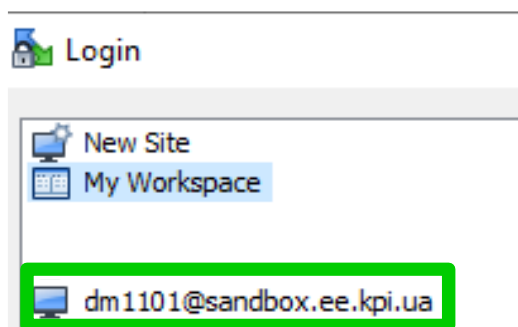
Для під'єднання до сервера потрібно вказати ім'я сервера (червона стрілка), номер порту (зелена стрілка), ім'я та пароль користувача (синя стрілка). Після цього можна натиснути кнопку *Login* і під'єднатися до сервера. Налаштування аналогічні до тих, які використовуються при підключенні до сервера за допомогою *PuTTY*. Якщо раніше було збережено налаштування підключень *PuTTY*, їх можна буде імпортувати.

Щоб кожного разу не вказувати усі ці параметри, можна зберегти налаштування, натиснувши кнопку *Save*, після чого з'явиться вікно збереження сесії:



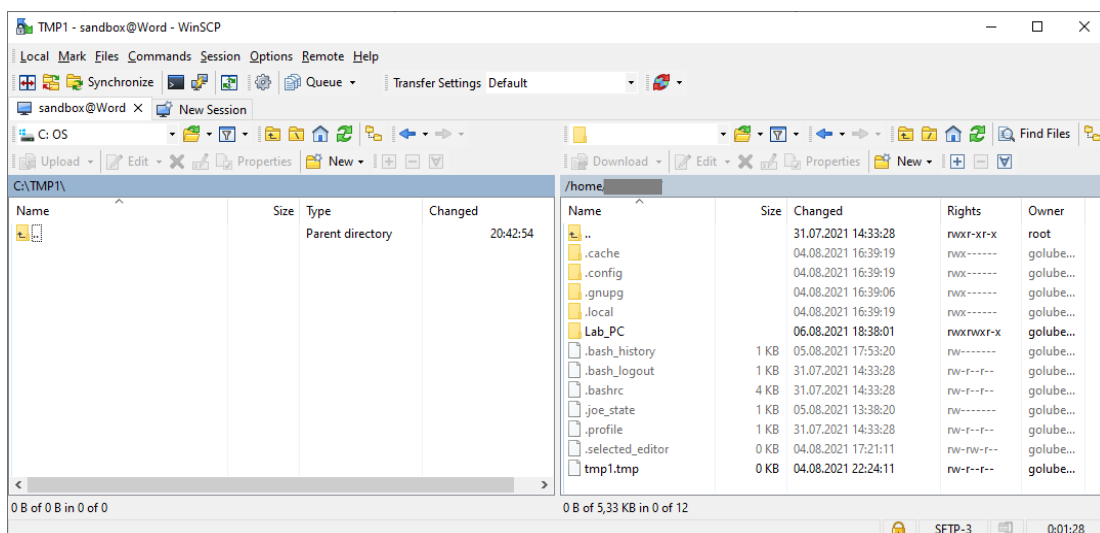
За замовчуванням, пароль не зберігається, але якщо поставити галочку напроти *Save password*, то при збереженні налаштувань в тому числі буде збережено і пароль.

Після збереження налаштувань, в панелі *Login* з'явиться запис, який відповідає збереженому профілю. На рисунку нижче він обведений зеленим:



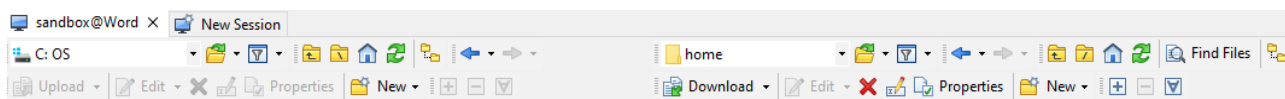
Налаштування підключення також можна імпортувати із раніше налаштованих підключень PuTTY.

Після з'єднання із сервером відкриється вікно, як показано на рисунку нижче:

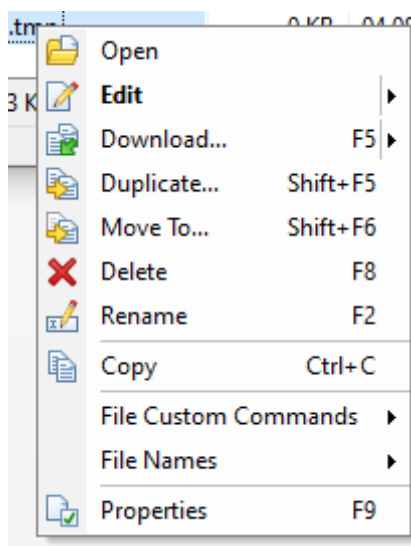



У цьому вікні *WinSCP* ліворуч знаходиться панель, яка показує файли, розташовані на локальному комп'ютері, а праворуч – панель, в якій показані файли на сервері, з яким виконано з'єднання. Перемикатися між цими панелями можна за допомогою клавіші *Tab*.

Виконувати маніпуляції із файлами можна або за допомогою гарячих клавіш, або за допомогою кнопок, які розташовані вище панелей:



Певні дії із конкретним файлом можна виконати також і за допомогою контекстного меню, яке відкривається натисканням правої кнопки миші:



Скопіювати файл із або на сервер можна обравши на ім'я файлу та натиснувши кнопку  **Download** відповідної панелі, або натиснувши гарячу клавішу *F5*.

ПРИМІТКА: Для правильного налаштування клавіатури в поточній сесії X-сервера (коли не вдається змінити мови на клавіатурі), потрібно виконати наступну команду:

```
setxkbmap -model pc105 -layout us,ua -variant qwerty -option grp:win_space_toggle
```

VcXsrv чи XMinG

SSH клієнт *PuTTY* працює у режимі командного рядка, але є можливість відобразити і графічну інформацію, яка передається із віддаленого сервера, для цього можна використовувати протокол для відображення та передачі зображень *X Windows System*. Цей протокол дозволяє відобразити графіку програмного забезпечення як на локальних дисплеях, так і як завгодно віддалених у мережевій доступності.

Для відображення графіки програмного забезпечення сервера на віддаленому робочому місці, необхідне налаштування спеціалізованого клієнта на робочому місці.

1. Найпоширенішими серверами *X11* для *Windows* є *VcXsrv* та *XMinG*.

Останні версії цих програм можна скачати по посиланням:

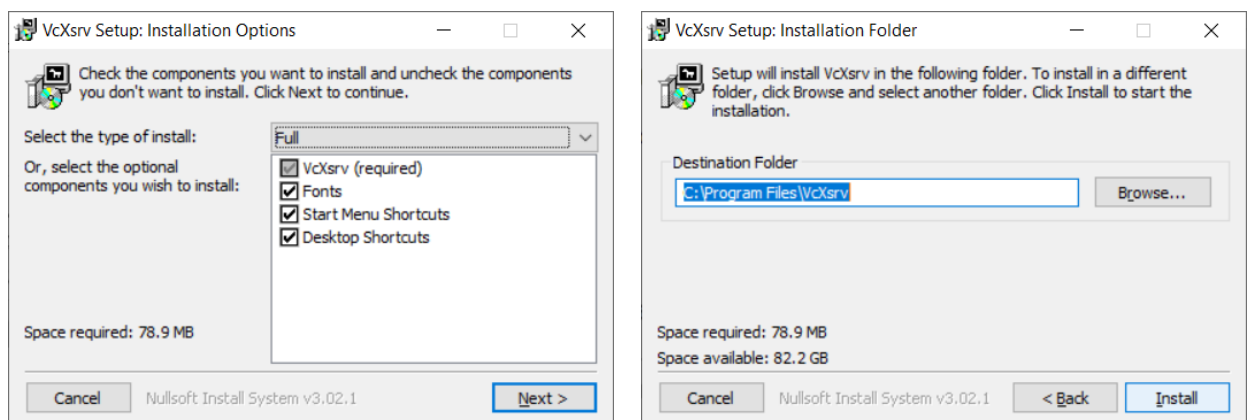
VcXsrv: <https://sourceforge.net/projects/vcxsrv/>

XMinG: <https://sourceforge.net/projects/xming/>

ПРИМІТКА: встановлювати потрібно тільки одну із них.

Нижче наведено приклад установки та налаштування *VcXsrv*.

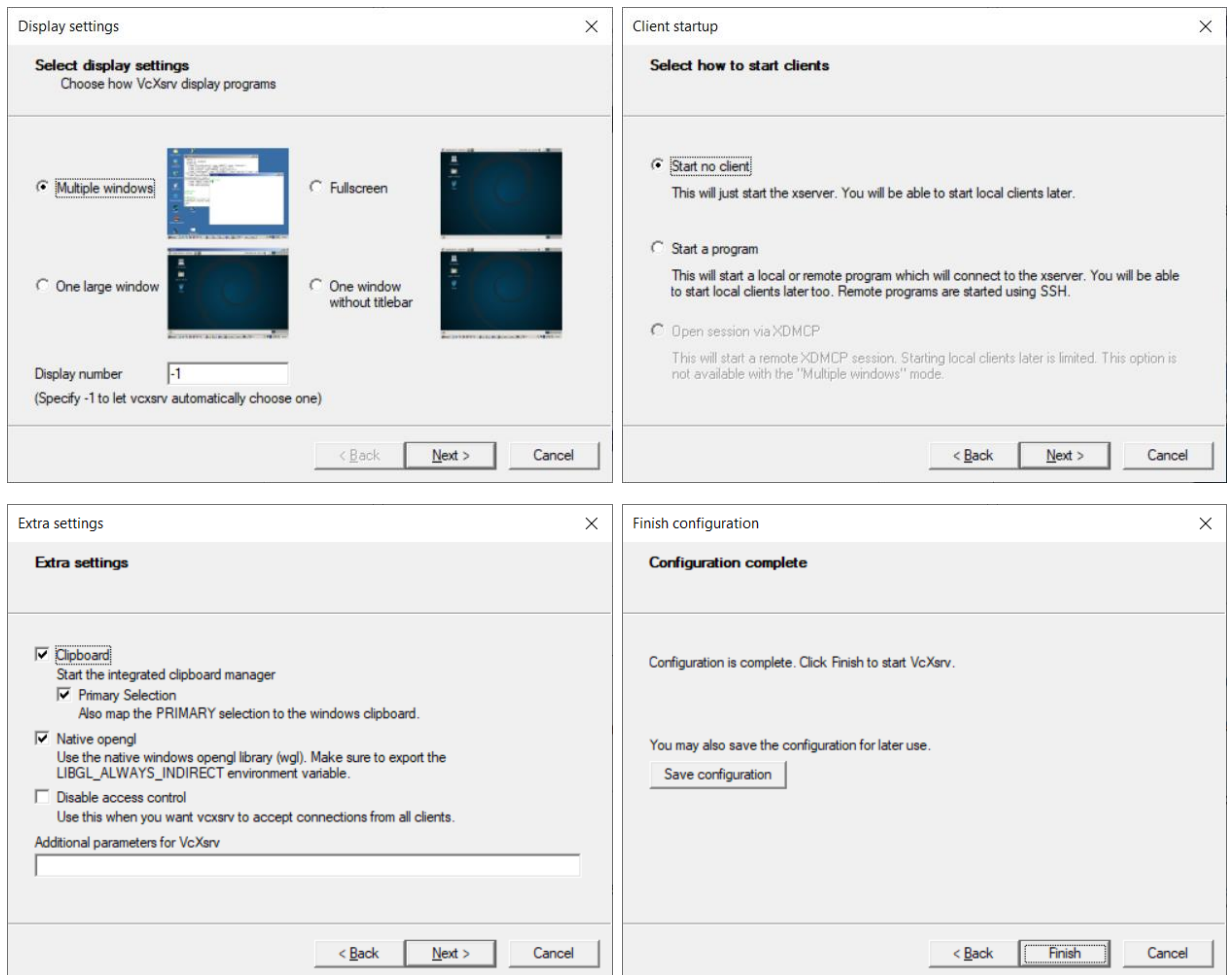
Натиснувши посилання, наведене вище, переходимо на сайт, де потрібно натиснути *Download* та завантажити файл інсталлятора. Далі встановити програму із наступними налаштуваннями:



Після встановлення слід запустити програму *XLaunch*:

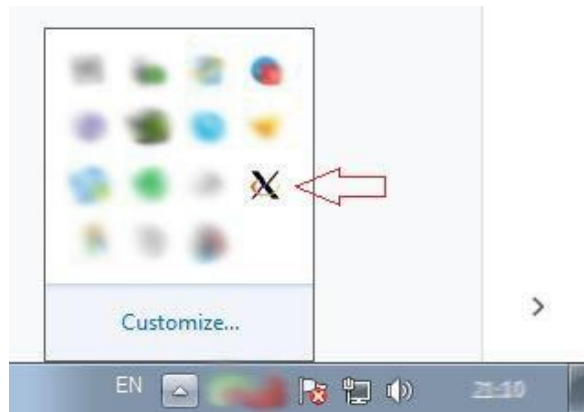


Та виконати налаштування, як показано на скріншотах нижче:



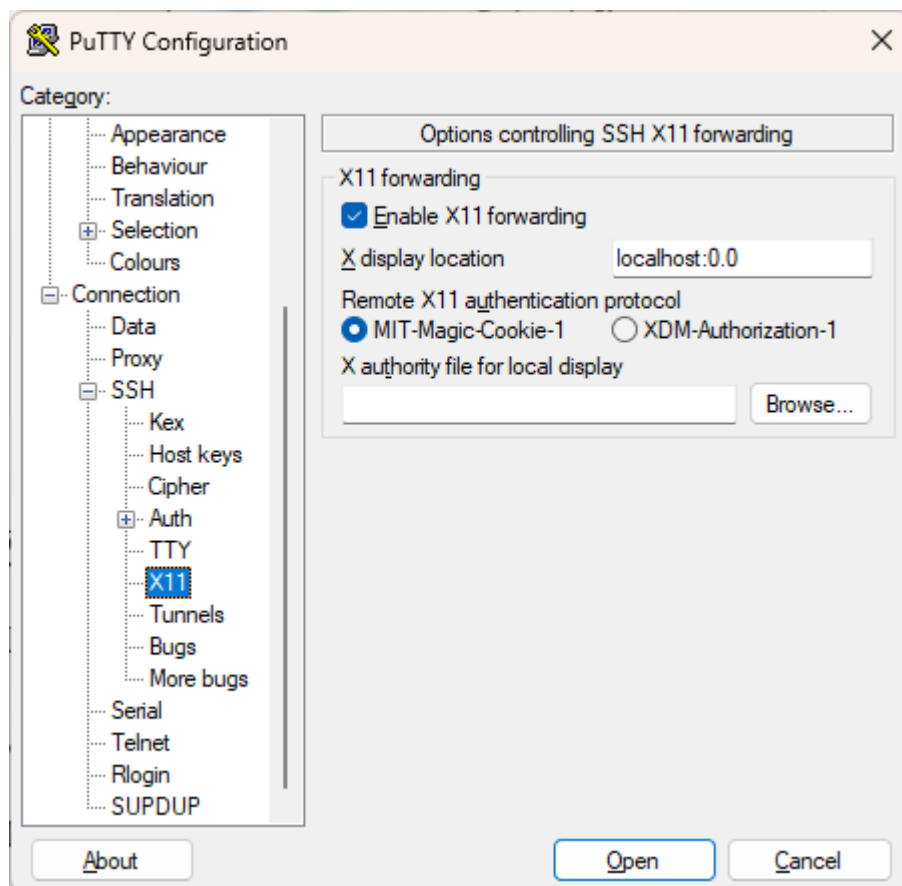
На останньому кроці можна зберегти конфігураційний файл у доступному місці, наприклад на робочому столі (після натискання кнопки *Save configuration*). Після цього натискаємо кнопку *Finish*. Це був початковий етап установки. Надалі достатньо запускати збережений конфігураційний файл перед початком кожного сеансу роботи із сервером, клікнувши на іконку на робочому столі.

Упевнитись, що клієнт працює, можна за іконкою у треї:



Після запуску *XMinG* або *VcXsrv* треба підключитися до сервера за допомогою термінального клієнта *PuTTY*.

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*).



Короткі теоретичні відомості

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.

Довідка по деяких командах

cat

cat – злиття та вивід файлів

СИНТАКСИС

cat [-u] [-s] [-v] [-t] [-e] файл ...

ОПИС

Команда *cat* по черзі читає вказані файли та видає їх вміст на стандартний вивід. Так, наприклад,

cat f

друкує вміст файлу *f*, а

cat f1 f2 > f3

зливає перші два файли (*f1* та *f2*) та поміщає результат в третій (*f3*).

Щоб додати файл *f1* до файлу *f2*, треба виконати команду

cat f1 >> f2

Якщо не вказано ні один файл або серед аргументів зустрівся -, команда *cat* читає дані зі стандартного вводу. Для завершення вводу слід натиснути *Ctrl-D* (символ кінця передачі, який згодом також стали використовувати як символ кінця файлу, *EoF*). Опції команди *cat* мають наступний зміст:

- u* Вивід не буферизується (за замовчуванням буферизується).
- s* Не повідомляється про неіснуючі файли.

-v Візуалізація недрукованих символів (крім табуляцій, переведення рядків і переходів до нової сторінки). Керуючі символи зображуються у вигляді $\wedge X$ (*CTRL* + *X*); символ *DEL* - у вигляді $\wedge ?$. Символи, що не входять в набір *ASCII* видаються у вигляді *M-x*, де *x* - визначається молодшими семи бітами символу.

З опцією **-v** можна використовувати наступні опції:

-t Візуалізація символів табуляції у вигляді $\wedge I$.

-e Візуалізація символів переведення рядка у вигляді $\$$

Якщо опція **-v** не вказана, то опції **-t** та **-e** ігноруються.

Це цікаво

Виконання команди `cat > file` дозволяє перенаправляти у файл символи, набрані з консолі. Цей прийом використовувався, зокрема, для створення файлів вихідного коду у прикладах на Фортрані, С та інших мовах, наприклад:

```
demo% cat greetings.f
PROGRAM GREETINGS
PRINT *, 'Real programmers write Fortran!'
END
```

Текст цієї простої програми перетворюється на файл виконуваної програми шляхом компіляції:

```
demo% f95 greetings.f
```

*За замовчуванням виконуваний файл мав назву `a.out`, що своєю чергою походить від скорочення “*assembler output*”, тобто в результаті компіляції створюється файл `a.out`.*

На основі цих традицій Інтернет-жартівники створили мем:

```
Real programmers write:
cat > a.out
```

який можна інтерпретувати так, що справжні програмісти безпосередньо з консолі набирають машинний код програми, виконуючи компіляцію подумки.

Зміна напрямку вводу/виводу

Механізм зміни напрямку вводу/виводу є одним з найбільш елегантних, потужних та водночас простих механізмів ОС *UNIX*. Мета, яка ставилася при розробці цього механізму, полягала в наступному. Оскільки *UNIX* – це інтерактивна система, то зазвичай програми вводять текстові рядки з терміналу і виводять текстові рядки на екран терміналу. Для того щоб забезпечити більш гнучке використання таких програм, бажано вміти забезпечити їм введення з файлу або з виходу інших програм і направити їх результат у файл або на вхід інших програм.

Реалізація механізму ґрунтується на наступних властивостях ОС *UNIX*.

По-перше, будь яке введення/виведення трактується як введення з деякого файлу та виведення в деякий файл. Клавіатура й екран терміналу теж інтерпретуються як файли (перший можна тільки читати, а в другий можна тільки писати).

По-друге, доступ до будь-якого файлу здійснюється через його дескриптор (додатне ціле число).

Фіксуються три значення дескрипторів файлів:

- ✓ Файл з дескриптором 1 називається файлом стандартного вводу (**stdin**),
- ✓ файл з дескриптором 2 – файлом стандартного виводу (**stdout**),
- ✓ файл з дескриптором 3 – файлом стандартного виводу діагностичних повідомлень (**stderr**).

По-третє, програма, запущена в деякому процесі, "успадковує" від батьківського процесу всі дескриптори відкритих файлів.

У головному процесі інтерпретатора командної мови файлом стандартного введення є клавіатура терміналу користувача, а файлами стандартного виводу і виводу діагностичних повідомлень – екран терміналу. Однак при запуску будь-якої команди можна повідомити інтерпретатору

(засобами відповідної командної мови), який файл повинен служити файлом стандартного вводу для програми яка запускається і який файл повинен служити файлом стандартного виводу чи виведення діагностичних повідомлень для програми. Тоді інтерпретатор перед виконанням системного виклику **exec** відкриває вказані файли, підмінюючи зміст дескрипторів 1, 2 та 3.

Звичайно, те ж саме може зробити і будь-яка інша програма, яка запускає третю програму в спеціально створеному процесі. Отже, все, що потрібно для нормального функціонування механізму зміни напрямку вводу/виводу – це дотримуватися при програмуванні угоди про використання дескрипторів **stdin**, **stdout** та **stderr**. Це не дуже важко, оскільки в найбільш поширених функціях бібліотеки вводу/виводу **printf**, **scanf** та **error** взагалі не потрібно вказувати дескриптор файлу. Функція **printf** неявно використовує **stdout**, функція **scanf** – **stdin**, а функція **error** – **stderr**.

Групування команд

Коротко перерахуємо засоби групування команд та зміни напрямку вводу/виводу:

- ✓ **cmd1 arg ...; cmd2 arg ...; ... cmdN arg ...** - послідовне виконання команд;
- ✓ **cmd1 arg ...& cmd2 arg ...& ... cmdN arg ...** - асинхронне виконання команд;
- ✓ **cmd1 arg ... && cmd2 arg ...** - залежність наступної команди від попередньої таким чином, що подальша команда виконується, якщо попередня видала нульове значення;
- ✓ **cmd1 arg ... || cmd2 arg ...** - залежність наступної команди від попередньої таким чином, що подальша команда виконується, якщо попередня видала ненульове значення;
- ✓ **cmd > file** - стандартна зміна напрямку виводу в файл **file**;
- ✓ **cmd >> file** - стандартна зміна напрямку виводу в кінець файлу **file** (дописування);

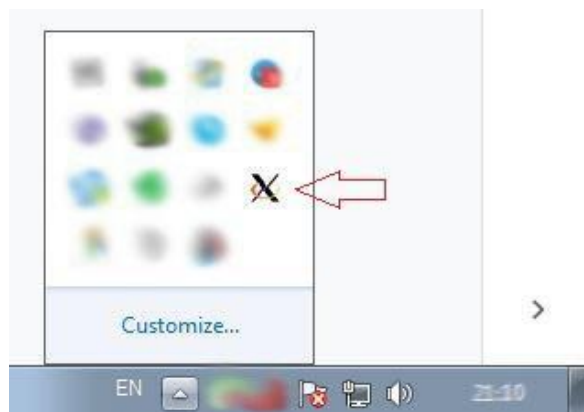
- ✓ **cmd < file** - стандартне введення інформації із файлу **file**;
- ✓ **cmd1|cmd2** - конвеєр команд, в якому стандартний вихід команди **cmd1** спрямований на стандартний вхід команди **cmd2**.

Текстові редактори

nedit

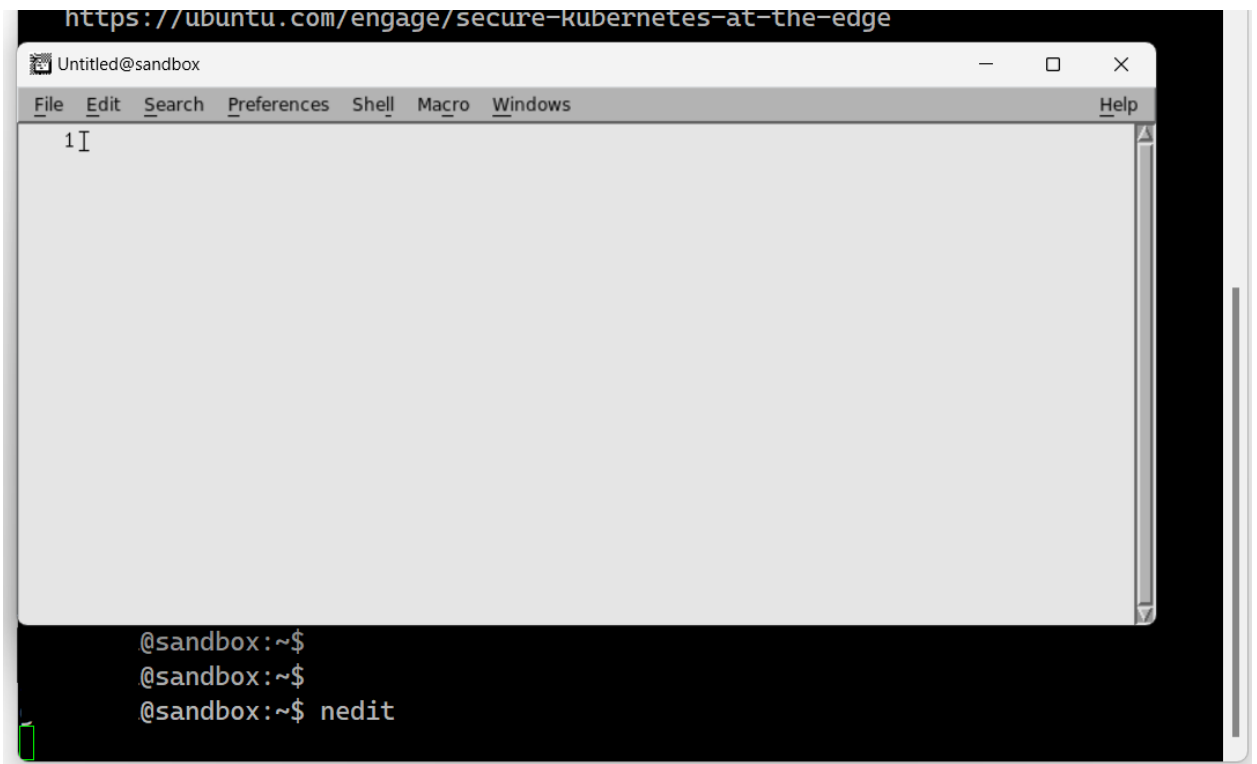
NEdit (Nirvana editor) – це текстовий редактор із редактором вихідного коду для системи *X Window*. Він має інтерфейс, подібний до текстових редакторів у *Microsoft Windows* та *Macintosh*. Спочатку він був розроблений Марком Еделем для *Fermilab* та випущений під дуже обмежувальною ліцензією, але сьогодні він поширюється під ліцензією із меншими обмеженнями GPL-2.0 або пізнішою версією і розроблений як незалежний відкритий код.

Перед використанням текстового редактору *NEdit* потрібно переконатися, що у вас запущено сервер *X11*. Упевнитись, що клієнт працює, можна за іконкою у треї:



Якщо ця іконка відсутня, то потрібно запуснути *VcXsrv* треба підключитися до сервера за допомогою термінального клієнта *PuTTY*.

Відкрити текстовий редактор із графічним інтерфейсом *nedit* можна за допомогою команди *nedit*, яка набирається в командному рядку вашого SSH клієнта, після чого відкривається графічне вікно редагування файлів із інтуїтивно зрозумілим інтерфейсом:



Запуск редактора призведе до того, що консоль буде зайнята його виконанням і ніякі інші команди запустити не можна, доки редактор не буде закритий. Аби запобігти цьому, редактор можна запустити асинхронно командою `nedit &`.

joe

joe – Joe's Own Editor, консольний екранний редактор.

Потужний текстовий редактор, який можна зустріти у багатьох системах. У найпростішому випадку запускається командою

```
joe ім'я.файла
```

У нижньому рядку відображається підказка

```
Type Ctrl-K Q to exit or Ctrl-K H for help
```

Новачкам варто скористатися порадою і увімкнути відображення підказок комбінацією «Ctrl-K H». Вікно редактора матиме вигляд:

```

ee.kpi.ua - PuTTY
REGION      GO TO      GO TO      DELETE     EXIT       SEARCH
^Arrow Select ^Z Prev. word ^U/^V PgUp/PgDn ^D Char. ^KX Save ^KF Find
^KB Begin ^X Next word MISC ^Y Line ^C Abort ^L Next
^KK End ^KU Top of file ^KJ Paragraph ^W >Word ^KQ All ^HELP
^KC Copy ^KV End of file ^KA Center line ^O Word< FILE Esc . Next
^KM Move ^A Beg. of line ^K Space Status ^J >Line ^KE Edit Esc , Prev
^KW File ^E End of line SPELL ^[O Line< ^KR Insert ^KH Off
^KY Delete ^KL To line no. Esc N Word ^_ Undo ^KD Save ^T Menu
^K/ Filter ^G Matching ( Esc L File ^^ Redo ^K` Revert

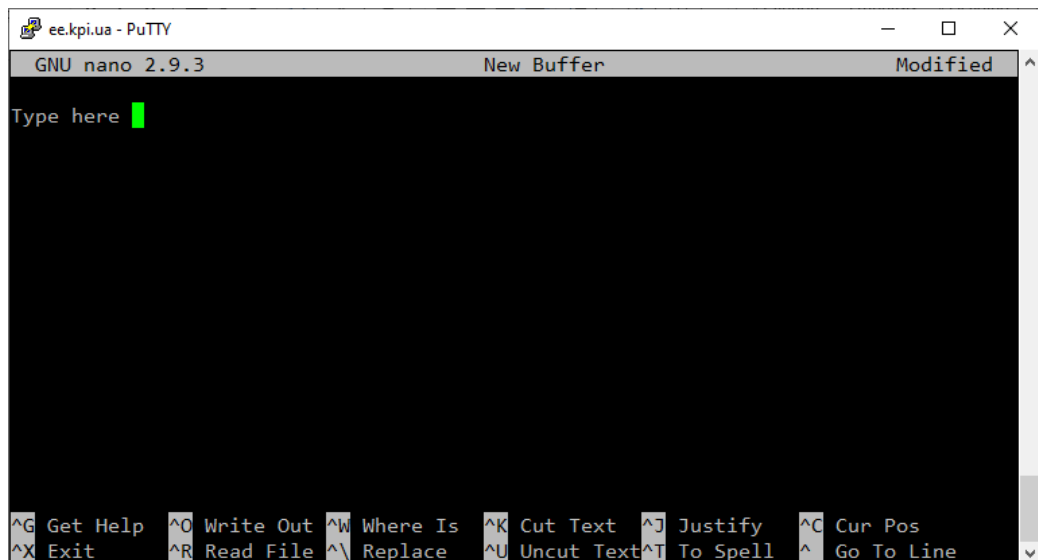
IW test.txt (Modified) Row 1 Col 26
Type here, see help above

```

Символ «^» означає клавішу Ctrl, а екран допомоги відображає підказки на найпоширеніші команди. Цей редактор не входить до стандартної конфігурації системи, тому на іншій UNIX-машині його може не бути, якщо адміністратор не подбав про це.

nano

nano – Nano's ANOther editor, компактний редактор із дружнім інтерфейсом. За замовчуванням, у нижніх двох рядках відображається підказка щодо найуживаніших команд:



Цей редактор не входить до стандартної конфігурації системи, тому на іншій UNIX-машині його може не бути, якщо адміністратор не подбав про це.

vi

vi, *view*, *vedit* – екранно-орієнтований редактор (**v**isual **e**ditor)

СИНТАКСИС

vi [*option...*] [*command...*] [*filename...*]

view [*option...*] [*command...*] [*filename...*]

vedit [*option...*] [*command...*] [*filename...*]

ОПИС

Зараз в це складно повірити, але у ранніх випусках UNIX-систем не було програми-редактора текстів у нашому теперішньому розумінні, тобто програми, яка б відображала результат редагування безпосередньо на екрані. Натомість, використовувався командний редактор **ed**. Користувач спочатку повинен був переглянути файл за допомогою програм відображення, визначити необхідні правки і дати команду редактору, наприклад, «додати символ А у третій рядок» і т.п. Робота з такими засобами вимагала уважності та вправності. Тому пізніше був створений редактор, де процес зміни тексту відображався на екрані по мірі редагування. Пізніше цей підхід розвинувся у концепцію WYSIWYG – What You See Is What You Get, тобто відтворення на

екрані зображення близького до того, що буде надруковано. Проте навіть попри відображення процесу редагування користування редактором **vi** вимагає певних навичок, з чим пов'язані численні жарти.

Цей редактор поставляється з переважною більшістю UNIX-подібних систем. Крім того, в аварійних режимах роботи системи дуже часто **vi** – це єдиний доступний редактор, тому базові навички роботи з ним знадобляться для відновлення працездатності пошкодженої системи.

Запуск і завершення VI

<code>vi filename</code>	Відкриття файлу <i>filename</i> для редагування
<code>vi -r filename</code>	Відкрити останню збережену версію <i>filename</i> після аварійного виходу
<code>vi filename file2 ...</code>	Відкрити <i>filename</i> , потім <i>file2</i> ... Перехід до наступного файлу - :n
<code>ZZ</code> або <code>:wq</code> або <code>:x</code>	Зберегти модифікації та завершити VI
<code>:w</code>	Зберегти модифікації в поточному файлі
<code>:w!</code>	Безумовно зберегти поточний файл
<code>:w file</code>	Зберегти поточний файл під ім'ям <i>file</i>
<code>:q!</code>	Завершити VI без зберігання модифікацій

Режими

vi працює у двох режимах – *тищати і все псувати*.

Інтернет-фольклор

VI може працювати у двох режимах: в **командному** режимі і режимі **вводу**. Відразу після запуску редактор знаходиться в командному режимі, в якому можна переміщатися по тексті, копіювати і вставляти текст з буфера. Спроби натискати символи на клавіатурі інтерпретуються як команди і якщо певний

символ чи комбінація не є чинною командою, видається звуковий сигнал, проте не відбувається зміна тексту на екрані («режим пицати»). Режим введення, тобто власне редагування тексту, включається командами вставки, доповнення, заміни. Натискання [ESC] повертає редактор у командний режим. Більшість команд виконується відразу після натискання клавіші. Виняток – команди рядкового редактора **ex**, що починаються з двокрапки, які виконуються після натискання [Return].

Зберігання файлу або завершення роботи програми після редагування вимагає переходу у командний режим (Esc). Новачки часто забувають про це, тому запит «how to exit vi» у пошуку Google повертає понад триста мільйонів результатів.

Порада: Перед будь-якими діями з документом введіть таку команду

:set showmode

Після цього поточний режим роботи редактора відображається в правому нижньому кутку екрана.

Вставка тексту

i	Вставка перед курсором
I	Вставка перед рядком
a	Додавання за курсором
A	Додавання після рядка
:r <i>file</i>	Вставити вміст <i>file</i> після поточного рядка

У режимі вводу:	
CTRL-i або TAB	Вставити зміщення вправо на shift width
CTRL-d	Зменшити зміщення вправо на shift width

CTRL-h або Backspace	Видалити попередній символ
CTRL-w	Видалити попереднє слово
CTRL-u	Видалити попередній рядок

Переміщення по тексту

h	Ліворуч
j	Вниз
k	Вгору
l	Праворуч
«стрілки»	Переміщення по тексту.
w	До наступного слова
b	До початку слова
^	До першого не пробільного символу рядка
+ або <cr>	До першого символу наступного рядка
-	До першого пробільного символу попереднього рядка
e	До кінця слова
(До початку речення
)	До кінця речення
{	До початку абзацу
}	До кінця абзацу
Примітка: абзаци розділяються порожнім рядком.	
0 або	До початку рядка
\$	У кінець рядка
1G	До першого рядка файлу
G	До останнього рядка файлу

H	До верхнього рядка екрана
L	До нижнього рядка екрана
CTRL-d	Прокрутка вниз на пів-екрана
CTRL-f	Прокрутка вниз на екран
CTRL-u	Прокрутка вверх на пів-екрана
CTRL-b	Прокрутка вверх на екран
%	К парної дужці: (), { }, []

Видалення тексту

Більшість команд видалення утворюються з символу **d** і команди переміщення.

Наприклад, **dw** видаляє слово. Інші команди видалення:

x	Видалити поточний символ
nx	Видалити n символів праворуч, починаючи з поточного
X	Видалити символ ліворуч від курсору
nX	Видалити n символів ліворуч від курсору
dd або :d	Видалити поточний рядок
"(a-z)dd	Вирізати поточний рядок в іменованій буфер

Копіювання в буфер

Подібно командам видалення, більшість команд копіювання в буфер складаються з символу **y** і команди переміщення.

yy або :y	Копіювати поточний рядок
"(a-z)yy	Копіювати поточний рядок в іменованій буфер (від a до z)

Модифікація тексту

Команди зміни діють як команди видалення з подальшим переведенням в режим введення. Більшість з них складається з символу **c** і команди переміщення. Наприклад **cw** замінює слово.

Вставка з буферу

p	Помістити текст з буфера після курсору
P	Помістити текст з буфера перед курсором
"(a-z)p або "(a-z)P	Помістити текст з буфера (з a до z) після/перед поточним рядком

Пошук/заміна підрядків

/str	Пошук <i>str</i> вперед від курсору
?str	Пошук <i>str</i> назад от курсору
n	Повторити пошук в тому же напрямку
:n,ms/str1/str2/opt	Шукати рядок <i>str1</i> з рядка n до m (n и m можуть бути пропущені); замінити <i>str1</i> на <i>str2</i> ; з опціями <i>opt</i> . Можливі опції: c - підтверджувати заміну, p - друк модифікованого рядка
&	Повторити останню команду :s
:g/str1/s/str2/str3/	Знайти рядок, яка містить <i>str1</i> , та замінити <i>str2</i> на <i>str3</i>
;	Повторити пошук
,	Повторити пошук в зворотному напрямку

Регулярні вирази

. (кома)	Будь-який символ, окрім «нового рядка»
*	0 або більше входжень кожного символу
[...]	Кожен із зазначених символів
[^...]	Будь-який з символів, крім зазначених
\<	Строго на початку слова
\>	Строго в кінці слова
^	На початку рядка

\$	В кінці рядка
\(...\)	Група умов
\n	Значення n-ої групи
\\	Скасування особливої дії символу \

Лічильники

Практично перед будь-якою командою можна вказати кількість її повторень. Наприклад, **5dw** призведе до видалення 5 слів, а **3fe** - до переміщення курсору вперед на 3-є входження символу «e». Подібним чином можна діяти і з командами вставки, наприклад, вставити 100 однакових рядків.

Режими VI

Зауважте: вводиться **:set option** для включення режиму та **:set nooption** для виключення. Щоб встановити настройки для кожного запуску VI, в домашньому каталозі створюється файл *.exrc*, що містить дані команди без двокрапки перед ними.

:set ai	Автоматичний відступ (по попередньому рядку)
:set all	Вивести на екран всі режими
:set eb	Звуковий сигнал при помилках
:set ic	Ігнорувати регістр при пошуку
:set list	Показувати символи табуляції (^ I) і кінця рядка (\$)
:set nu :set number	Показати номер рядка
:set showmode	Показати поточний режим в нижній частині екрану
:set wa	Виключити нормальну перевірку перед записом

Інше

~	Зміна регістра символу над курсором
---	-------------------------------------

J	Злиття наступного рядка з поточним
nJ	Злиття n рядків
.	Повторити останню команду, яка змінює текст
u	Скасувати останню зміну
U	Відмінити всі зміни в рядку
CTRL-G	Показати ім'я файлу, номер рядка, загальне число рядків і положення в файлі (%)

Редактор ві було розроблено під використання алфавітної клавіатури, без блоків стрілок та додаткової цифрової клавіатури. Якщо натискання клавіш стрілок приводить до пошкодження відображуваного тексту («режим все псувати»), то, найімовірніше, у програмі терміналу не налаштовано належну трансляцію кодів символів.

Завдання для виконання

1. Скопіюйте з консолі (набрати на клавіатурі) текстовий файл з розширенням *inf*. У нього запишіть номер групи, прізвище, ім'я. Збережіть файл у домашньому каталозі. (Підказка: можна використовувати команду *cat* та символи перенаправлення).
2. Скопіюйте цей файл під іншим ім'ям.
3. Переіменуйте копію так, щоб вона мала розширення *txt*.
4. За допомогою команд перенаправлення вводу/виводу кілька разів скопіюйте інформацію з будь якого файлу в новий файл так, щоб кількість рядків склало не менше 20.
5. Скористайтеся текстовим редактором *nedit* для редагування останнього файлу. В кінці файлу допишіть поточну дату та день тижня. Збережіть цей файл.
6. За допомогою клієнта *WinSCP* скопіюйте останній файл із сервера на свій робочий комп'ютер.

7. Продемонструйте скопійований файл на своєму комп'ютер. Допишіть в нього останнім рядком назву факультету та зберегти під новим ім'ям.
8. За допомогою клієнта *WinSCP* скопіюйте останній файл із свого робочого комп'ютера на сервер та продемонструйте розташування копії в домашньому каталозі та її вміст.

Контрольні питання

1. Які можливості надають команди перенаправлення вводу/виводу в *UNIX*-подібних системах?
2. Які програмні продукти вам відомі для скачування файлів із сервера та завантаження файлів на сервер?
3. Які текстові редактори для роботи із файлами в *UNIX*-подібних системах вам відомі?
4. Які переваги та недоліки відомих вам текстових редакторів для роботи із файлами в *UNIX*-подібних системах?