

Лабораторна робота №3

Основи роботи с 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.....	9
Файлова система	10
Програми.....	11
Командна мова ОС Unix (shell)	11
Структура команд	13
Зміна напрямку вводу/виводу.....	13
Групування команд.....	15
Довідка по деяких командах.....	16
man	16
apropos	16
ls	16
mkdir.....	21
cp	22
mv	22
cat	23
rm.....	24
vi.....	26
joe	33

nano	34
Midnight Commander	35
Завдання для виконання	42
Контрольні питання	44

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*).

В даний 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*-клієнтів для різних платформ:

- GNU/Linux, *BSD: *kdssh, 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,*
- Java: *MindTerm, AppGate Security Server,*
- J2ME: *MidpSSH,*

- 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 може використовуватися 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:

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date version of the code available out the [development snapshots](#), to see if the problem has already been fixed in those versions.

Package files

You probably want one of these. They include versions of all the PuTTY utilities.

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

MSI ('Windows Installer')			
32-bit:	putty-0.74-installer.msi	(or by FTP)	(signature)
64-bit:	putty-64bit-0.74-installer.msi	(or by FTP)	(signature)
Unix source archive			
.tar.gz:	putty-0.74.tar.gz	(or by FTP)	(signature)

Alternative binary files

The installer packages above will provide versions of all of these (except PuTTYtel), but you can download standalone binaries one by one if you prefer.

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

putty.exe (the SSH and Telnet client itself)			
32-bit:	putty.exe	(or by FTP)	(signature)
64-bit:	putty.exe	(or by FTP)	(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* та пароль *q1w2e3r4t5*.

При першому входженні усі облікові записи мають однаковий пароль:

q1w2e3r4t5

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

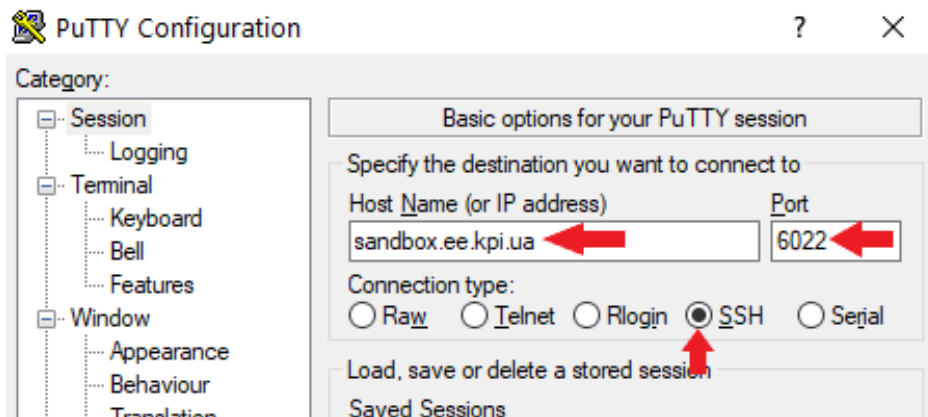
термінал та *ssh* клієнт, то достатньо набрати в терміналі наступну команду:

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

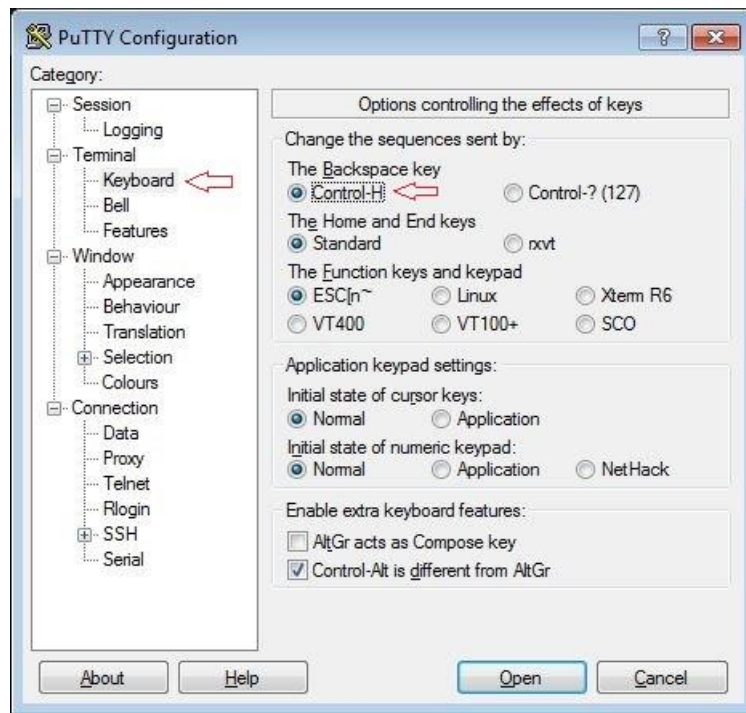
Якщо планується використовувати клієнт *PuTTY*, то після установки та запуску клієнта, для доступу до сервера, необхідно виконати наступні налаштування:

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

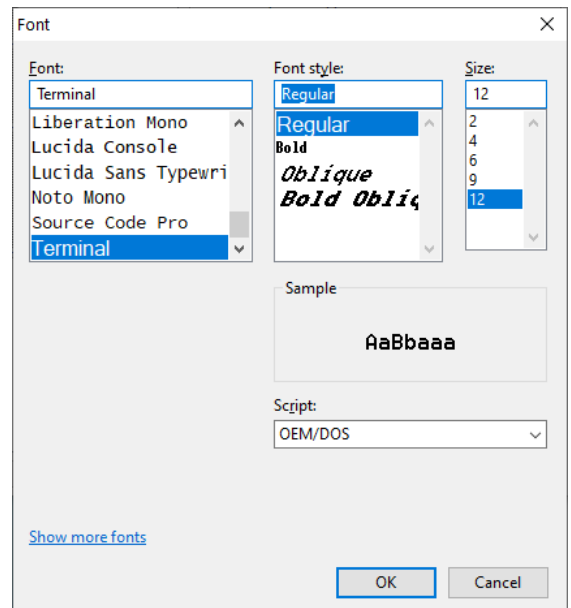
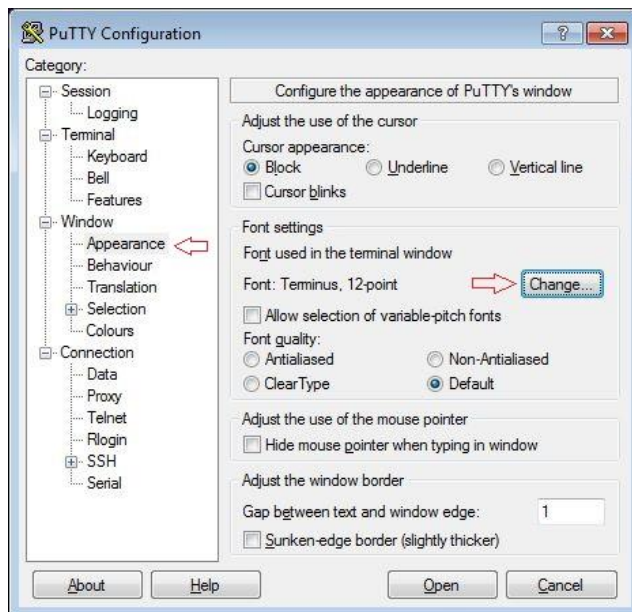
Налаштування	Ми працюємо	
	З Інтернету	В мережі КПІ
<i>Host Name:</i>	<i>sandbox.ee.kpi.ua</i>	<i>sandbox.ee.kpi.ua</i>
<i>Port:</i>	<i>6022</i>	<i>22</i>
<i>Connection type:</i>	<i>SSH</i>	<i>SSH</i>



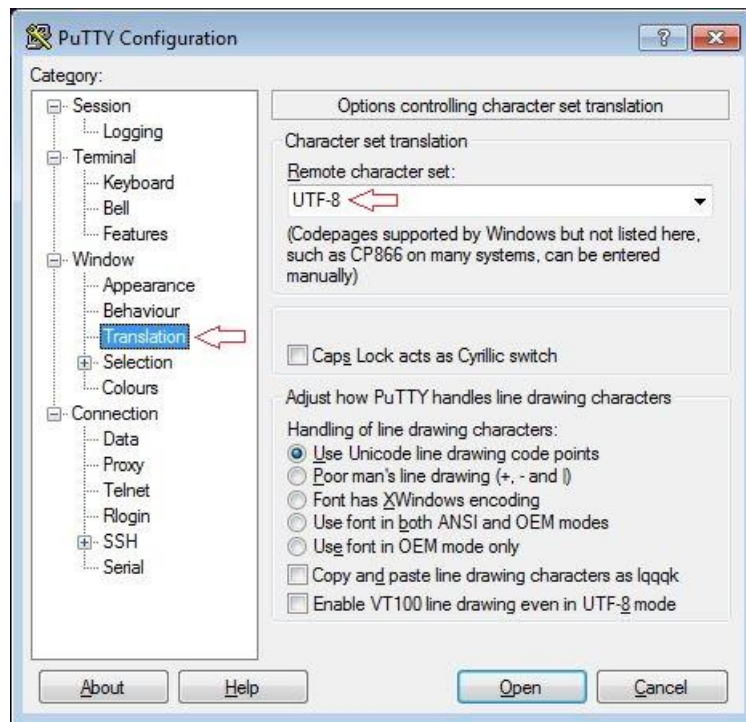
- б. Встановити трансляцію кодів клавіатури:



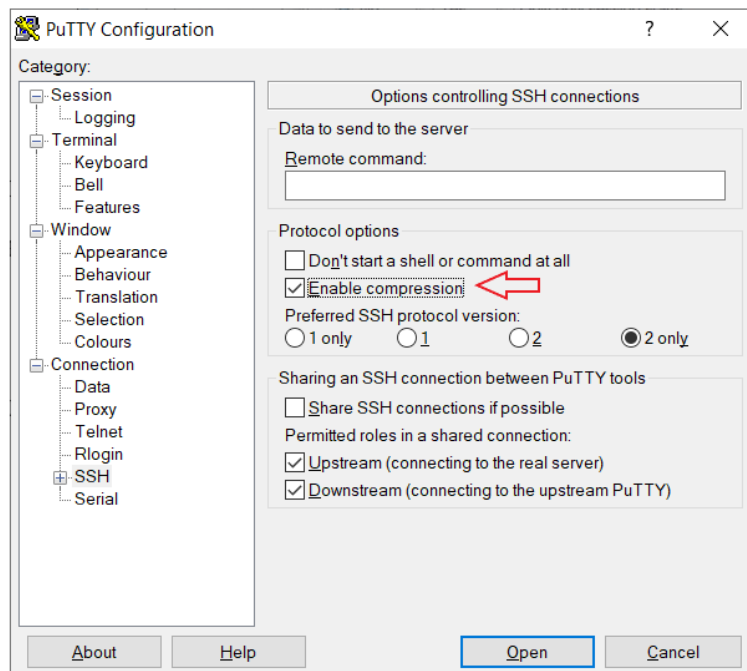
с. Встановити бажаний тип та розмір шрифту для консолі:



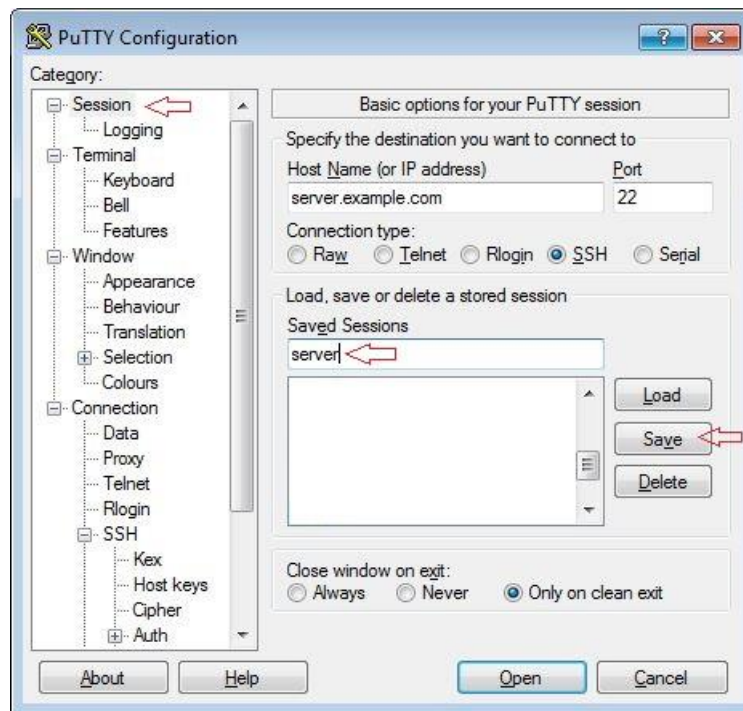
д. Встановити трансляцію кодування:



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



ф. 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.

Користувач, привілейований користувач

Ядро ОС *UNIX* ідентифікує 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 вирішенні задач користувача.

Зверніть увагу, що на відміну від деяких інших ОС, для відокремлення імен каталогів використовується символ "/", 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 (промттер), яке з'являється на екрані. У найпростішому випадку це долар ("\$").

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 промттер. 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> <ключі> <аргумент(и)>

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.

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, підміняючи 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**.

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*.

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] [им'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 в ім'я файлу, у вісімковому вигляді (**\ddd**).
- q** Видавати недруковані символи, що входять в ім'я файлу, у вигляді символу **?**.
- i** Видавати в першій колонці номера 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 символним 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.

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

При використанні команди

```
ls -l /util/by
```

результат може виглядати наступим 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 *s*, 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 «*rxw*» (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**'а). 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*.

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 сім'ю битами символ.

З опцією -v можна використовувати наступні опції:

-t Візуалізація символів табуляції у вигляді ^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.

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.

vi

vi, *view*, *vedit* – екранно-орієнтований редактор (*visual editor*)

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, типу «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 – *пицати* 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 а 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.

joe

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

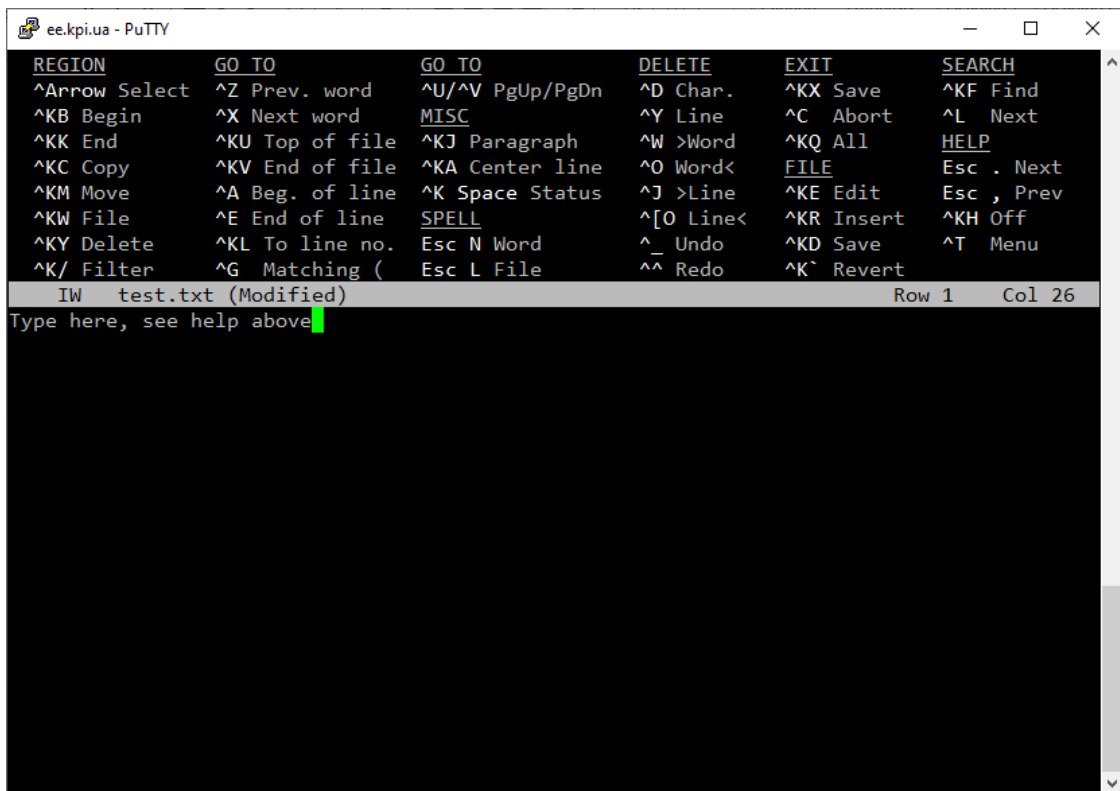
Потужний текстовий редактор, який можна зустріти у багатьох системах. У найпростішому випадку запускається командою

```
joe ім'я.файла
```

У нижньому рядку відображається підказка

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

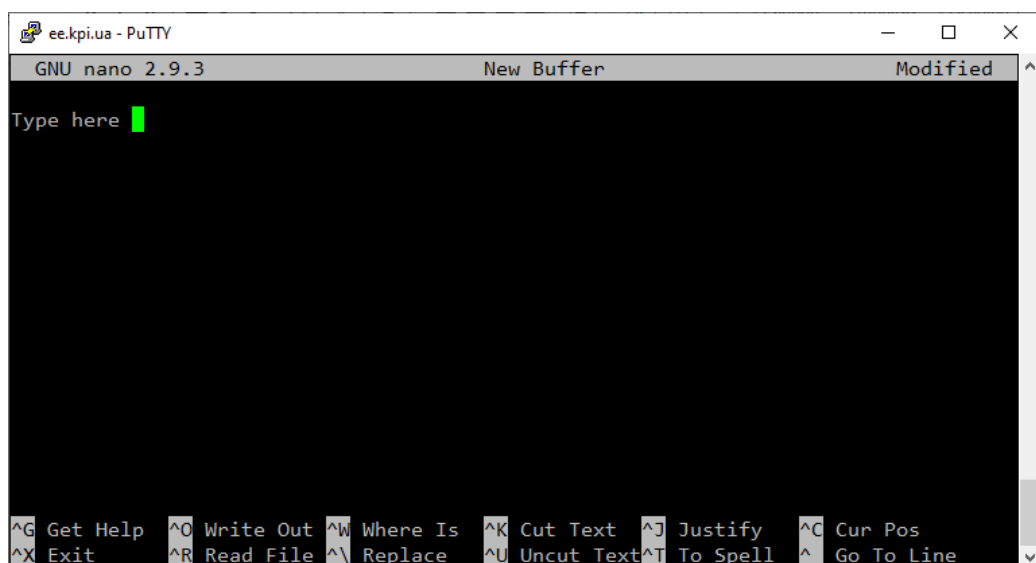
Новачкам варто скористатися порадою і увімкнути відображення підказок комбінацією «Ctrl-K H». Вікно редактора матиме вигляд:



Символ «^» означає клавішу Ctrl, а екран допомоги відображає підказки на найпоширеніші команди.

nano

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



Midnight Commander

Інколи, при роботі із термінальними клієнтами, зручно користуватися файловими менеджерами. Прикладом такого менеджера може слугувати *GNU Midnight Commander (MC)* – це двохпанельний файловий менеджер з текстовим інтерфейсом типу *Norton Commander* для *UNIX*-подібних операційних систем. Створено в 1994 році Мігелем де Ікас.

Midnight Commander є частиною проекту *GNU*, і всі розробники передали свої права на програму на користь *Free Software Foundation*. Зараз *Midnight Commander* є складовою частиною більшості дистрибутивів *Linux*.

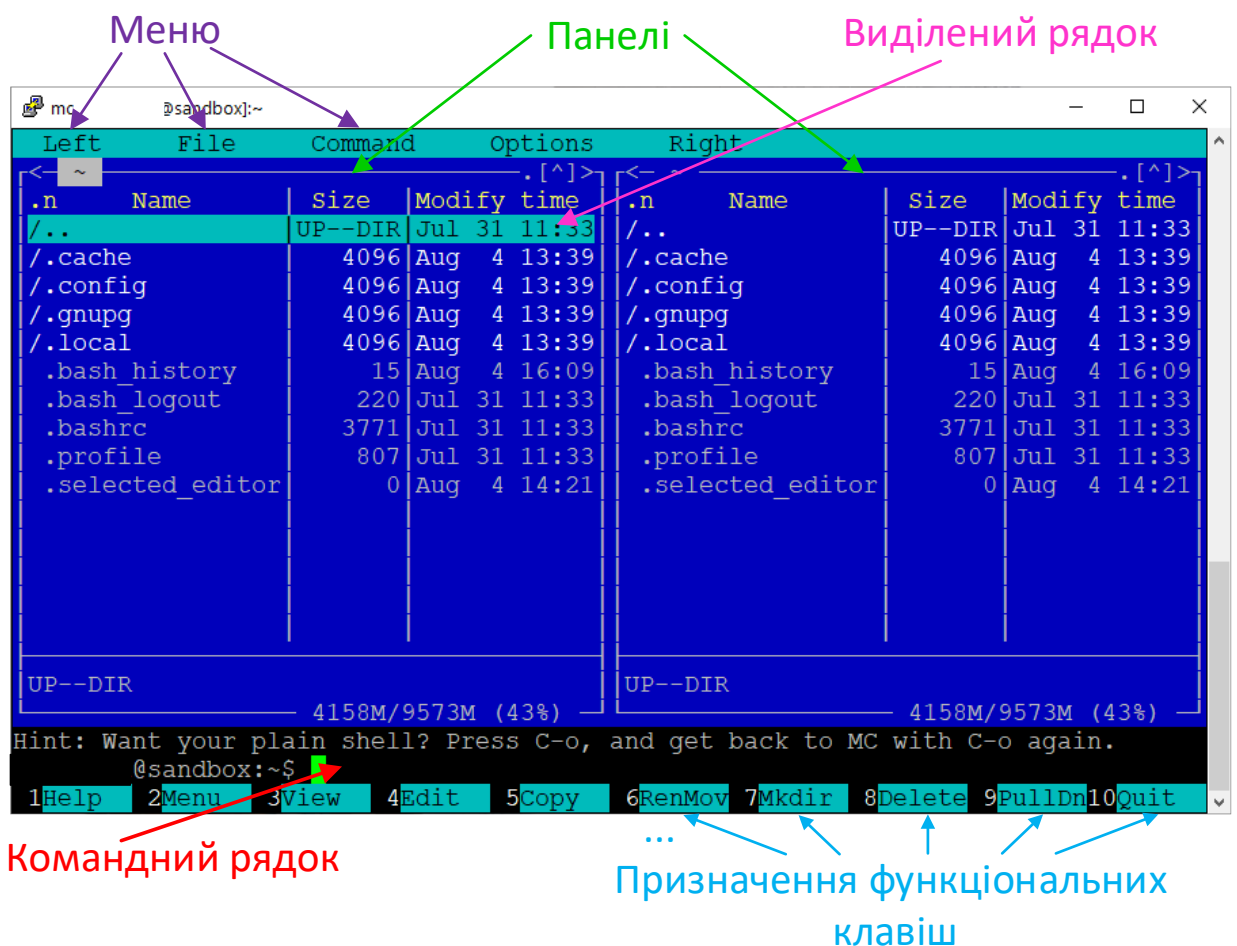
Midnight Commander надає цілий ряд можливостей при роботі із файлами та каталогами:

- *MC* дозволяє роботу із різними архівами та образами файлових систем, таким самим чином, якщо б вони були звичайними каталогами,
- Є можливість роботи з мережевими файловими системами *Samba*, *FTP*, *SFTP*,
- Можна виконувати копіювання та переміщення файлів у фоновому режимі,
- *MC* в своєму складі має вбудований текстовий редактор з підсвічуванням синтаксису,
- Є текстовий режим, який заснований на бібліотеках *ncurses* та *S-Lang*. Це дає можливість програмі працювати як в консолі, так і в різних емуляторах терміналу і через віддалені з'єднання, наприклад, за протоколом *SSH*,
- Підтримується багатомовний інтерфейс,
- Підтримується кодування *UTF-8*.

Для запуску *Midnight Commander (MC)* в термінальному клієнті *PuTTY*, достатньо набрати команду *mc* в командному рядку. Для виходу із цього файлового менеджера потрібно натиснути **F10**.

Екран *MC* ділиться на чотири частини. Майже весь екран зайнятий двома панелями. За замовчуванням, другий рядок знизу є командним, а в самому нижньому рядку відображається призначення функціональних клавіш. Самий верхній рядок – рядок меню. Він може бути невидимим, але при натисканні клавіші *F9* або при натисканні миші цей рядок відображається у верхній частині екрану.

MC дозволяє одночасно спостерігати вміст двох каталогів за допомогою двох панелей. Одна з панелей є поточною (в цій панелі знаходиться виділений рядок). Практично всі команди виконують дії над поточною панеллю. Хоча деякі файлові операції, такі як *Rename* або *Copy*, за замовчуванням використовують каталог неактивної панелі в якості каталогу призначення (при виконанні подібних команд завжди з'являється запит на підтвердження).



Можна виконувати системні команди прямо з *MC* просто шляхом їх набору на клавіатурі. Все, що набирається на клавіатурі, з'являється в

командному рядку, а при натисканні клавіші *Enter*, *МС* виконає команду, яка була набрана в командному рядку.

МС поставляється з підтримкою миші. Миша підключається, при запуску *МС* з *xterm(1)* терміналу або з *Linux* консолі при наявності *gpm mouse* сервера.

При натисканні лівою кнопкою миші на файлі, цей файл стає поточним. При натисканні правою кнопкою, файл позначається (або позначка знімається, в залежності від попереднього стану).

При подвійному натисканні на файлі, буде зроблено спробу його запуску, якщо це файл програми. Якщо розширення файлу пов'язано з якою-небудь програмою, то ця програма буде запущена.

Якщо *МС* налаштований з підтримкою миші, то ви можете вирізати і склеювати фрагменти тексту, використовуючи клавішу *Shift*.

Робота з файлами в *МС*

Клавіши	Дія
<i>F3</i>	Переглянути файл
<i>F4</i>	Редагувати файл
<i>Shift + F4</i>	Створити новий файл
+	Викликати вікно для виділення групи
\	Зняти позначку з групи

Робота з командним рядком в *МС*

Клавіши	Дія
---------	-----

<i>Alt + Enter</i>	Вставити файл або каталог, на якому встановлений курсор, в командний рядок
<i>Alt + a</i>	Вставити в командний рядок повний шлях в активній панелі
<i>Alt + p</i>	Повернути попередню виконану команду в командний рядок
<i>Ctrl + x, t</i>	Вставка всіх виділених об'єктів в командний рядок
<i>Ctrl + x, Ctrl + p</i>	Вставити в командний рядок повний шлях, встановлений на неактивній панелі
<i>Alt + t</i>	Зміна режиму відображення панелі
<i>Alt + .</i>	Приховати або показати приховані файли і каталоги, які починаються на точку
<i>Alt + a</i>	Вставити в командний рядок повний шлях, встановлений на активній панелі
<i>Alt + c</i>	Зміна поточного каталогу
<i>Alt + h</i>	Перегляд останніх виконаних команд
<i>Alt + i</i>	Зміна каталогу на неактивній панелі на каталог, в якому знаходиться користувач
<i>Alt + o</i>	Неактивна панель переходить в каталог, на який вказує активна панель
<i>Ctrl + x, q</i>	Перегляд вмісту файлу на другий панелі
<i>Ctrl + x, i</i>	Перегляд інформації про поточний об'єкт на другий панелі
<i>Ctrl + o</i>	Приховати або показати панелі
<i>Ctrl + r</i>	Передивитися вміст каталогу
<i>Ctrl + Space</i>	Показати розмір поточного каталогу

Робота з пошуком

Клавіши	Дія
<i>Alt + Shift + ?</i>	Розширений пошук
<i>Ctrl + s</i>	Пошук файлу або каталогу

Робота з правками

Клавіши	Дія
<i>Ctrl + x, o</i>	Перегляд або зміна власника, групи
<i>Ctrl + x, c</i>	Перегляд або зміна прав доступу

Робота з посиланнями

Клавіши	Дія
<i>Ctrl + x, l</i>	Створити жорстке посилання
<i>Ctrl + x, s</i>	Створити символічне посилання <i>symlink</i>
<i>Ctrl + x, Ctrl + s</i>	Редагувати символічне посилання

У *Midnight Commander* є вбудований текстовий редактор *mcedit* (за допомогою клавіши **F4**). Серед можливостей *mcedit* є підсвітка синтаксису для різних мов програмування, здатність працювати у режимах *ASCII* та *HEX*.

Для використання цього редактора за замовчуванням, потрібно виконати певні налаштування: в меню обрати *Options: Configuration...*

The screenshot shows the Midnight Commander (MC) interface. The 'Options' menu is open, and 'Configuration...' is highlighted. The interface is split into two panes, 'Left' and 'Right', both displaying a directory listing of files and folders. The status bar at the bottom shows the current directory as 'golubeva@here.8503' and the disk usage as '4158M/9573M (43%)'. A hint at the bottom suggests pressing 'C-o' to switch to a plain shell.

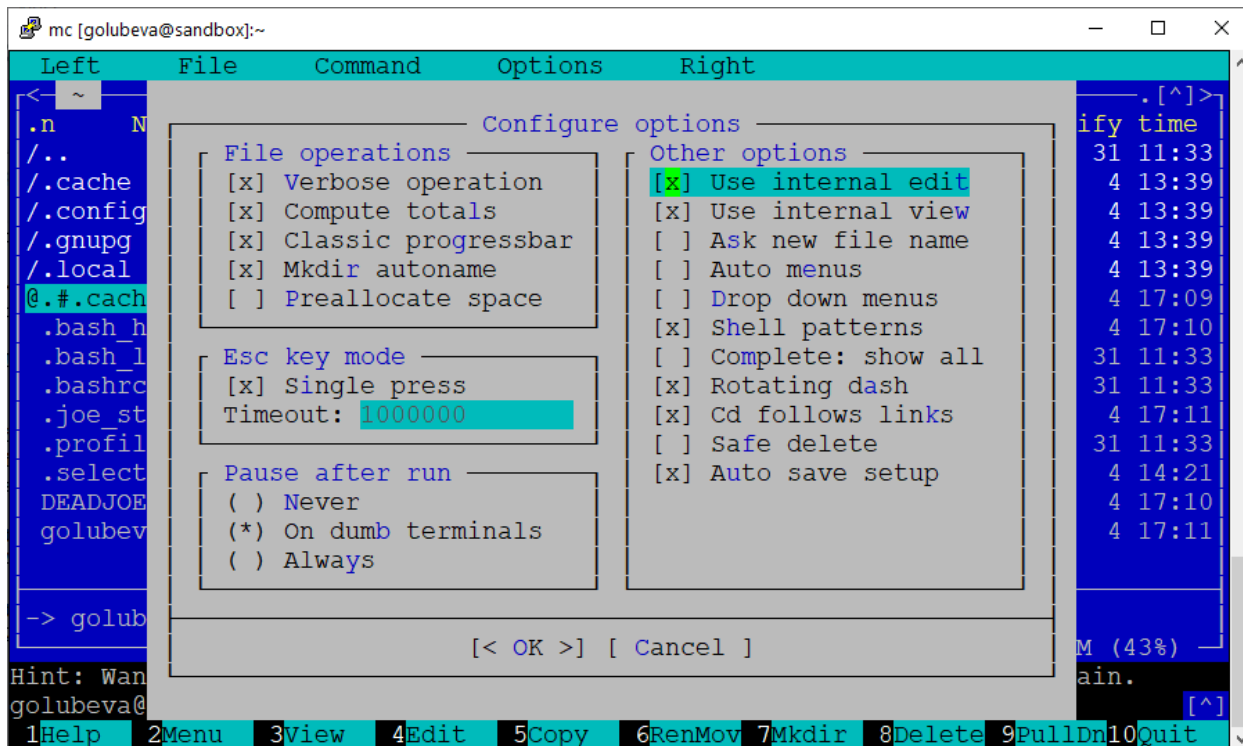
```

mc [golubeva@sandbox]:~
Left      File      Command  Options  Right
<- ~     Name      Size     Mod     Configuration...  e      Size     Modify time
/..      UP--DIR   Jul      Layout...          e      4096    Aug  4 13:39
/.cache  4096     Aug     Panel options...   e      4096    Aug  4 13:39
/.config 4096     Aug     Confirmation...    e      4096    Aug  4 13:39
/.gnupg  4096     Aug     Appearance...      e      4096    Aug  4 13:39
/.local  4096     Aug     Display bits...    e      4096    Aug  4 13:39
@.#.cache 18       Aug     Learn keys...      e      18      Aug  4 17:09
.bash_history 25      Aug     Virtual FS...      tory   25      Aug  4 17:10
.bash_logout 220     Jul     Save setup         out    220     Jul 31 11:33
.bashrc   3771    Jul     .profile           e      242     Aug  4 17:11
.joe_state 242     Aug     .profile           807    Jul 31 11:33
.profile  807     Jul 31 11:33 | .profile           807    Jul 31 11:33
.selected_editor 0      Aug  4 14:21 | .selected_editor   0      Aug  4 14:21
DEADJOE  375     Aug  4 17:10 | DEADJOE           375    Aug  4 17:10
golubeva~re.8503 28     Aug  4 17:12 | golubeva~re.8503  28     Aug  4 17:12

-> golubeva@here.8503      UP--DIR
4158M/9573M (43%)      4158M/9573M (43%)

Hint: Want your plain shell? Press C-o, and get back to MC with C-o again.
golubeva@sandbox:~$ [^]
1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn 10Quit
  
```

І далі в меню що випало поставити галочку навпроти напису *Use internal edit*, як це показано на рисункунижче:



Зміна виду редактора в *mcedit*

Клавіши	Дія
<i>Alt + n</i>	Показати або приховати нумерацію рядків
<i>Ctrl + s</i>	Включити або вимкнути підсвічування синтаксису
<i>Alt + Shift + -</i>	Включити або вимкнути відображення табуляцій та пробілів
<i>Alt + Shift ++</i>	Включити або вимкнути автовирівнювання за допомогою повернення каретки

Робота з текстом в *mcedit*

Клавіши	Дія
<i>F3</i>	Почати виділення тексту, повторне натискання <i>F3</i> закінчить виділення

<i>Shift + F3</i>	Почати виділення блоку тексту, повторне натискання <i>F3</i> закінчить виділення
<i>F5</i>	Копіювати виділений текст
<i>F6</i>	Перемістити виділений текст
<i>F8</i>	Видалити виділений текст
<i>Ctrl + k</i>	Видалити частину рядка до кінця рядка
<i>Ctrl + y</i>	Видалити рядок
<i>Ctrl + x</i>	Перейти в кінець слова
<i>Ctrl + z</i>	Перейти на початок попереднього слова
<i>Alt + l</i>	Перехід на рядок по номеру рядка
<i>Alt + u</i>	Виконати команду і вставити друк результату команди в позицію під курсором
<i>Alt + t</i>	Сортування рядків виділеного тексту
<i>Tab</i>	Зсунути праворуч виділений текст
<i>Ctrl + u</i>	Скасувати дії

Робота з файлами в *mcedit*

Клавіши	Дія
<i>Ctrl + n</i>	Створити новий файл
<i>Ctrl + f</i>	Копіювати виділений текст в файл
<i>Shift + F5</i>	Вставка тексту з файлу

Більш детальну інформацію про *Midnight Commander* можна отримати за посиланням:

<https://midnight-commander.org/>

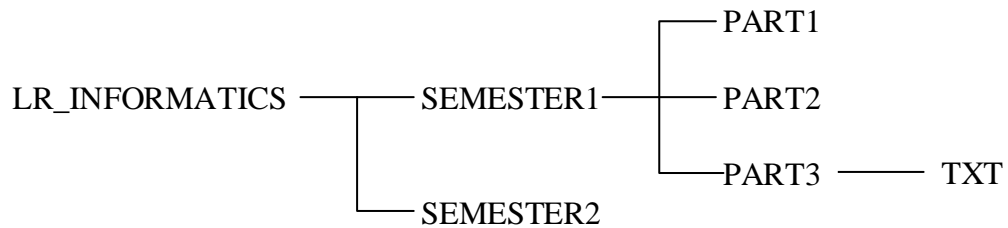
Завдання для виконання

Частина 1

В пунктах завдання 1-18 цієї лабораторної роботи ми НЕ використовуємо файловий менеджер *Midnight Commander*.

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 свій пароль.
3. Вивчіть довідку (команда *man*) по команді *uname*. Визначте тип і версію операційної системи.
4. За допомогою команди *apropos 'date'* вивчити основні команди роботи із системною датою. Визначте системні дату і час за допомогою потрібної команди.
5. За допомогою команди *apropos 'working directory'* вивчити основні команди роботи з робочим каталогом.
6. Вивчіть керівництво по команді *ls*, звертаючи увагу на вплив ключів на роботу команди.
7. Перегляньте вміст кореневого каталогу файлової системи в декількох форматах і 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.
8. Перегляньте вміст каталогу будь-якого підкаталога свого домашнього каталогу або іншого каталога у системі.

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



10. До каталогу *PART1* скопіюйте декілька будь-яких файлів, а в *PART2* та *TXT* - копії цих файлів (наприклад, можна скопіювати кілька текстових файлів налаштувань з каталогу */etc*, наприклад, конфігураційних).
11. Виведіть вміст цих каталогів.
12. Видаліть рекурсивно каталог *PART3*.
13. Виведіть на екран і перегляньте один з скопійованих текстових файлів.

Частина 2

14. Скопіюйте з консолі 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).
15. Скопіюйте цей файл під іншим ім'ям.
16. Перейменуйте копію так, щоб вона мала розширення *txt*.
17. 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.
- 18.*** Зі створеного файлу виведіть рядки, які містять деякий зразковий текст (наприклад «DM» чи «DP»).
18. За допомогою *Midnight Commander* виконайте пункти 7-13 цієї роботи.
19. В *Midnight Commander* () створіть файл та запишіть в нього номер групи, прізвище, ім'я. Збережіть файл у домашньому каталозі.

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. Який екранно-орієнтовані редактори в *UNIX*-подібних системах вам відомий?