“Київський коледж зв’язку”

Циклова комісія Комп’ютерної інженерії

ЗВІТ ПО ВИКОНАННЮ

ЛАБОРАТОРНОЇ РОБОТИ №2

з дисципліни: «Операційні системи»

Тема: “Знайомство з базовими командами **CLI**-**режиму** в Linux”

Виконали

студенти

групи КСМ-83а

Лісовий М.І.\_\_\_\_\_\_

Колосовський Р.Ю.\_\_\_\_\_\_

Перевірив викладач

Повхліб В.С.

Київ 2020

Мета роботи:

1. Отримання практичних навиків роботи з командними інтерпретаторами Bash та PowerShell.

2. Знайомство з базовими текстовими командами в термінальному режимі роботи в різних ОС.

Матеріальне забезпечення занять

1. ЕОМ типу IBM PC.

2. ОС сімейства Windows (Windows 7).

3. Віртуальна машина – Virtual Box (Oracle).

4. Операційна система GNU/Linux – CentOS.

5. Сайт мережевої академії Cisco netacad.com та його онлайн курси по Linux

**Завдання для попередньої підготовки:**

**5.** **Командний інтерпретатор** — програма, яка забезпечує взаємодію користувача з операційною системою.

Командний інтерпретатор звичайно підтримує командний рядок, змінні оточення, історію виконаних команд, власні конфігураційні файли. Основне призначення командного інтерпретатора полягає в виконанні команд користувача. Командний інтерпретатор одночасно є середовищем програмування.

**Консо́ль** (англ. *console*) — пристрій, який забезпечує взаємодію оператора комп'ютера з операційною системою. Як правило, як консоль використовується дисплей і клавіатура, або окремий комп'ютерний термінал.

Також під словом консоль часто мають на увазі інтерфейс командного рядка.

**Комп'ютерний термінал**, або просто **термінал** — електронний або елекромеханічний прилад, призначений для введення людиною даних у комп'ютерну систему, а також для відображення інформації комп'ютером. Прикладом електромеханічного терміналу може слугувати телетайп, а повністю електронного — прилад з екраном (монітором), клавіатурою і комунікаційним інтерфейсом.

**Інтерфейс командного рядка** (англ. Command line interface, CLI) - різновид текстового інтерфейсу (TUI) між людиною і комп'ютером, в якому інструкції комп'ютера даються в основному шляхом введення з клавіатури текстових рядків (команд), в UNIX-системах можливе застосування миші [1 ]. Також відомий під назвами «консоль» і «термінал».

Інтерфейс командного рядка протиставляється системам управління програмою на основі меню, а також різним реалізаціям графічного інтерфейсу. Формат виведення інформації в інтерфейсі командного рядка не регламентується; зазвичай це також простий текстовий висновок, але може бути і графічним, звуковим і т. д.

**6.** За назвою команди можна дізнатися для чого вона застосовується, як використовується і доступні для неї опції.

Для цього використовують конструкцію назва\_команди –help. Або можна використати альтернативний варіант - команду man

-a - відображати всі файли, включаючи приховані, це ті, перед ім'ям яких стоїть крапка;

-A - не відображати посилання на поточну папку і кореневу папку. і ..;

--author - виводити творця файлу в режимі докладного списку;

-b - виводити Escape послідовності замість недрукованих символів;

--block-size - виводити розмір каталогу або файлу в певній одиниці виміру, наприклад, мегабайтах, гігабайтах або кілобайтах;

-B - не виводити резервні копії, їх імена починаються з ~;

-c - сортувати файли за часом модифікації або створення, спочатку будуть виведені нові файли;

-C - виводити колонками;

--color - включити кольоровий режим виведення, автоматично активована в багатьох дистрибутивах;

-d - виводити тільки директорії, без їх вмісту, корисно при рекурсивном виведення;

-D - використовувати режим виведення, сумісний з Emacs;

-f - НЕ сортувати;

-F - показувати тип об'єкта, до кожного об'єкта буде додано один із спеціалізованих символів \* / => @ |;

--full-time - показувати детальну інформацію, плюс вся інформація про час в форматі ISO;

-g - показувати детальну інформацію, але крім власника файлу;

--group-directories-first - спочатку відображати директорії, а вже потім файли;

-G - не виводити імена груп;

-h - виводити розміри папок в зручному для читання форматі;

-H - відкривати символічні посилання при рекурсивном використанні;

--hide - не відображати файли, які починаються з вказаного символу;

-i - відображати номер індексу inode, в якій зберігається цей файл;

-l - виводити докладний список, в якому буде відображатися власник, група, дата створення, розмір та інші параметри;

-L - для символічних посилань відображати інформацію про фото, на який вони посилаються;

-m - розділяти елементи списку коми;

-n - виводити UID і GID замість імені і групи користувача;

-N - виводити імена як є, не обробляти контролюючі послідовності;

-Q - брати імена папок і файлів в лапки;

-r - зворотний порядок сортування;

-R - рекурсивно відображати вміст піддиректорій;

-s - виводити розмір файлу в блоках;

-S - сортувати за розміром, спочатку великі;

-t - сортувати за часом останньої модифікації;

-u - сортувати за часом останнього доступу;

-U - НЕ сортувати;

-X - сортувати за алфавітом;

-Z - відображати інформацію про розширення SELinux;

-1 - відображати один файл на один рядок.

Це були всі основні параметри команди ls, які нам може коли-небудь знадобитися використовувати. Їх дуже багато, але на ділі будуть потрібні тільки деякі з них, ви їх швидко запам'ятайте, а всі інші зможете завжди знайти в цій статті або на сторінці довідки man ls. А тепер давайте розглянемо кілька прикладів використання команди ls linux.

**pwd** (англ. present working directory - поточний робочий каталог [1] [2]; або print working directory [3] - вивести робочий каталог) - консольна утиліта в UNIX-подібних системах, яка виводить повний шлях від кореневого каталогу до поточного робочого каталогу : в контексті якого (за замовчуванням) будуть виконуватися вводяться команди.

**more**

Найпростіша програма для посторінкового перегляду тексту more використовується з ім'ям файлу (ів) в якості параметра або як остання команда в ланцюжку перенаправлень вводу / виводу. За допомогою опцій + <число> або + / <зразок тексту> можна почати перегляд з цікавить рядки.

more вміє гортати текст вперед. Текст з файлу, в принципі, можна перегорнути назад, але це нетипове використання more.

Основні інтерактивні команди more схожі на команди редактора vi:

h - help

q - вихід

SPACE - перегорнути один екран

d - перегорнути підлогу екрана

RETURN - перегорнути один рядок

/ Pattern - перегорнути до рядка містить підрядок pattern

n - повторити пошук підрядка

= - показати номер поточного рядка

: N - перейти до наступного файлу

: P - повернутися до попереднього файла

: F - надрукувати ім'я поточного файлу і номер рядка в ньому

! - запуск shell-команди

Тільки при перегляді файлів:

b - перегорнути екран назад

v - викликати текстовий редактор, встановивши курсор на поточний рядок

**less**

less - істотно більш розвинена команда для гортання тексту. При читанні даних зі стандартного вводу вона створює буфер, який дозволяє перегортати текст як вперед, так і назад, а також шукати як у напрямку до кінця, так і у напрямку до початку тексту. Заповнення буфера йде блоками по 64KB, тому (якщо текст на стандартне введення подається повільно) можлива ситуація "підвисання", коли less показує чергові надійшли рядки, але не дозволяє виконати будь-яку команду.

less розуміє все команди more плюс свої.

h - виклик довідки

q - вихід

SPACE - на екран вперед

b - перегорнути екран назад

k і j або стрілки вгору і вниз - вертикальна прокрутка по рядку

g - перейти на 1-шу рядок

<N> g - перейти на рядок N

G - перейти на останній рядок

F - перейти на останній рядок файлу і очікувати записи нових рядків (аналог tail -f, не працює зі стандартним вводом)

/ Pattern - пошук за шаблоном вперед

? Pattern - пошук за шаблоном назад

n - наступне збіг

N - попереднє збіг

mбуква - відзначити позицію буквою

'Буква (апостоф і буква) - перейти на зазначену позицію

! - запуск shell-команди (% - ім'я поточного файлу, наприклад:! Cat%> /tmp/foobar.txt)

Тільки для файлу

v - запуск редактора

Тільки для стандартного введення

s filename - зберегти виведений текст в файл filename

Хід роботи.

- Запрошення є сигналом для користувача про правильність функціонування операційної системи та її готовність виконувати команди. Для знаходження маршруту до файлу операційній системі потрібні дані про ім’я каталогу, який містить файл, та ім’я самого файлу. Якщо файл знаходиться в активному каталозі, то немає необхідності вказувати ім’я каталогу, оскільки система продивляється активний каталог автоматично. Коли файла немає в активному каталозі, то вказівка маршруту до файлу необхідна. Для будь-якої команди можна завжди передивитись її призначення та параметри (ключі), ввівши в командний рядок ім’я відповідної команди з параметром “--h” або “--help” . Наприклад, dir – help. Для білш детальної інформації використовується команда man. Наприклад, man dir. - Командою echo $PS1 можна вивести поточне значення змінної PS1. Не лякайтесь. Насправді існує маса можливостей для настройки рядка запрошення. Нижче ми розглянемо базові методи налаштування.

4. **Змінні середовища** — змінні, набір динамічних значень, які можуть вплинути на те як запущені процеси будуть себе поводити на комп'ютері.

В усіх [Unix](https://uk.wikipedia.org/wiki/Unix) і Unix-подібних системах, кожен процес має свій власний набір змінних середовища. За замовчуванням, створений процес успадковує дубльоване середовище свого батьківського процесу за винятком змін, внесених батьківським процесом під час створення даного процесу. На рівні системного програмування, ці зміни робляться між [fork](https://uk.wikipedia.org/wiki/Fork) і [exec](https://uk.wikipedia.org/w/index.php?title=Exec&action=edit&redlink=1). З іншого боку, в командних оболонках, таких, як [Bash](https://uk.wikipedia.org/wiki/Bash), можна змінити значення змінних середовища для конкретної команди безпосереднім запусканням через [env](https://uk.wikipedia.org/wiki/Env) або з використанням нотації ЗМІННА\_СЕРЕДОВИЩА = ЗНАЧЕННЯ <команда>.

Всі операційні системи подібні до Unix, а також [DOS](https://uk.wikipedia.org/wiki/DOS) і [Microsoft Windows](https://uk.wikipedia.org/wiki/Microsoft_Windows) мають змінні середовища, однак не всі вони використовують одні і ті ж назви змінних. Програми, що стартували можуть отримати доступ до значень змінних середовища для цілей конфігурації. Приклади змінних середовища включають в себе:

* PATH в якому перераховані каталоги де знаходяться програми команд, назви яких користувач може ввеси.
* HOME вказує, де знаходиться домашній каталог користувача у файловій системі.
* TERM (Unix-подібні) визначає тип комп'ютерного терміналу або емулятора терміналу (наприклад vt100).
* CVS\_RSH (Unix-подібні) використовується для яких повідомити CVS RSH-подібну програму використовувати.
* MAIL (Unix-подібні) використовується, щоб вказати місце пошти користувача.

Скрипти оболонки та командні файли використовують змінні середовища для зберігання тимчасових значень з метою вживання пізніше в цьому скрипті, а також щоб повідомити дочерні процеси про зміну даних або наладок.

В більшості систем зміна значення змінної середовища в скрипті або в програмі тільки впливає на процес цього скрипту або програми і, можливо, на дочерні процесі. Базовий процес системи і будь-які інші процеси не будуть зачеплені.

В Unix змінні середовища в основному ініціалізуються під час запуску системи системним скриптом і успадковуються усіма іншими процесами в системі. Користувачі додають їх в профільний скрипт (.profile). У Microsoft Windows, змінні середовища звичайно зберігаються в реєстрі Windows або в autoexec.bat.

Для того щоб змінити значення змінних середовища застосується:

**set** ЗМІННА=значення

Однак, це тимчасово. Постійна зміна значення змінної середовища може бути досягнута шляхом редагування реєстру (не рекомендується для новачків), а також за допомогою Windows Resource Kit і застосування setx.exe. З впровадженням Windows Vista, команда setx стала частиною Windows. Найпоширенішим методом визначення змінних середовища в Windows є <Панель управління:Система:Додатково:Змінні середовища>.

Для UNIX подібних є env, set, і printenv команди щоб показати всі змінні середовища і їхні значення. env, set також використовується для встановлення змінних середовища і найчастіше включаються безпосередньо в оболонку. printenv може також використовуватися для друку однієї змінної за умови, що ім'я змінної є єдиним аргументом для команди.

У UNIX також можуть бути використані наступні команди в залежності від певної оболонки.

Body

Запрошення в командному середовищі bash змінюється за допомогою зміни змінних PS1, PS2, PS3, PS4 і PROMPT\_COMMAND.

PS1 - змінна, яка задає вид рядка запрошення командного рядка, яке bash виводить, коли очікує введення чергової команди користувачем.

PS2 - змінна, що відповідає за вторинний тип запрошення введення.

PS3 - третинне запрошення, виводиться тоді, коли користувач повинен зробити вибір в операторі select.

PS4 - запрошення четвертого рівня, виводиться при налагодження сценаріїв (скриптів) на початку кожного рядка перед командою перед що виконується командою.

PROMPT\_COMMAND **-** вміст цієї змінної виконується перед кожним рядком запрошення.

Bash

Умовні оператори, думаю, знайомі практично кожному, хто хоч раз намагався на щось писати програми. В bash умови пишуться слід. чином (як зазвичай на прикладі):

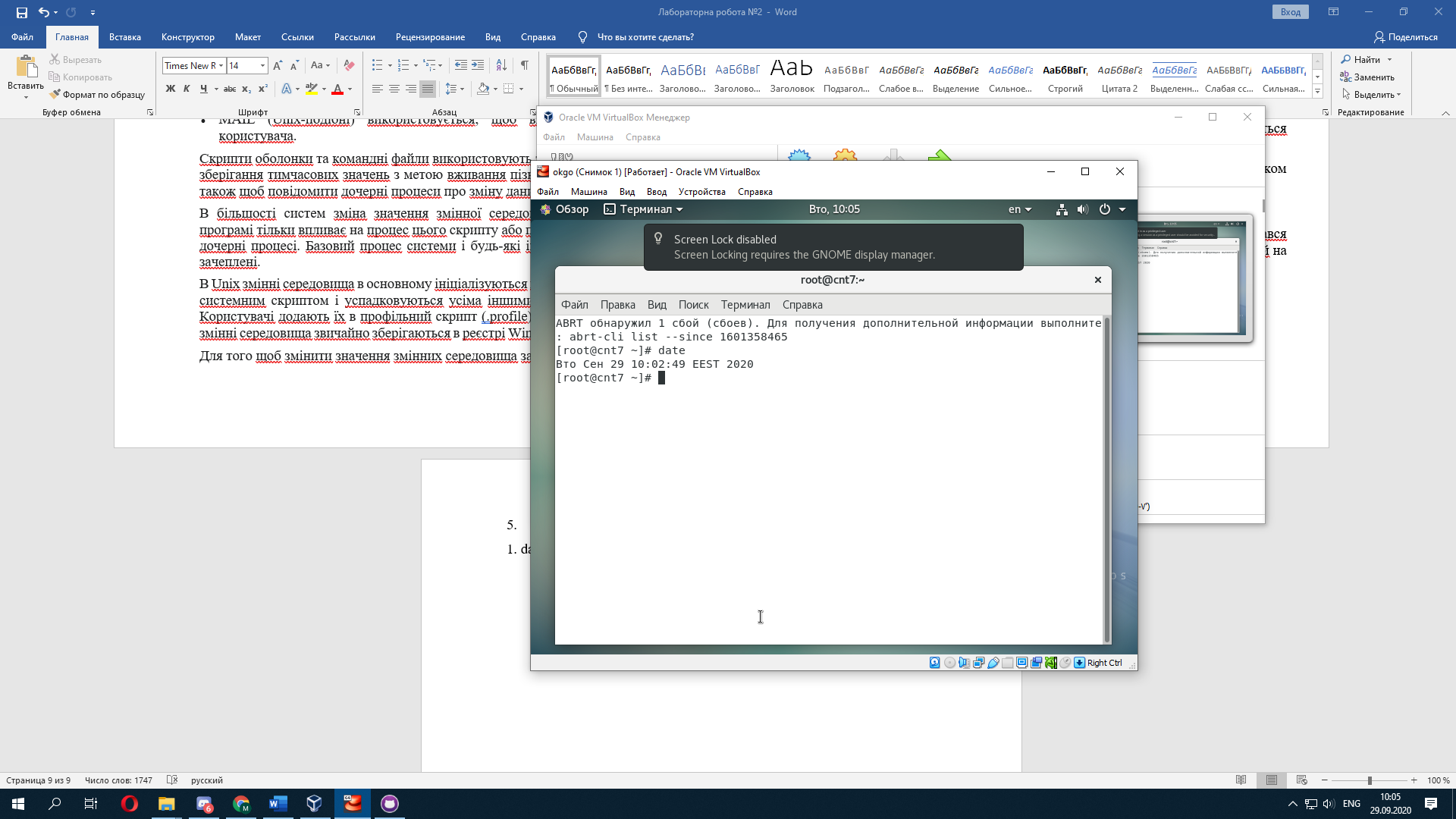
#! / Bin / bash

source = $ 1 # в змінну source засовуємо перший параметр скрипта

dest = $ 2 # в змінну dest засовуємо другий параметр скрипта

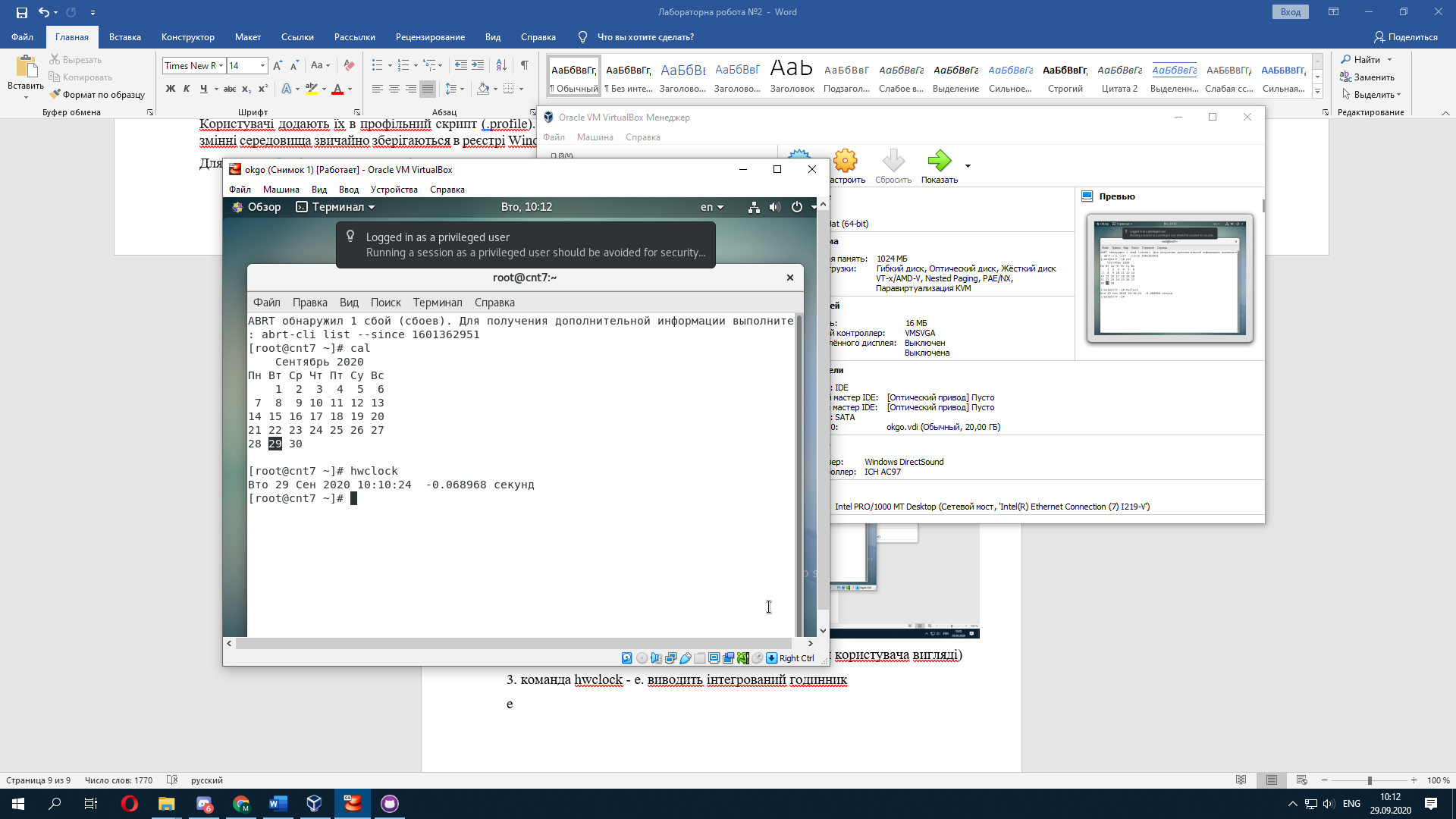
5.

1. date - g. показує нинішні дату і час, по системним годинах ядра

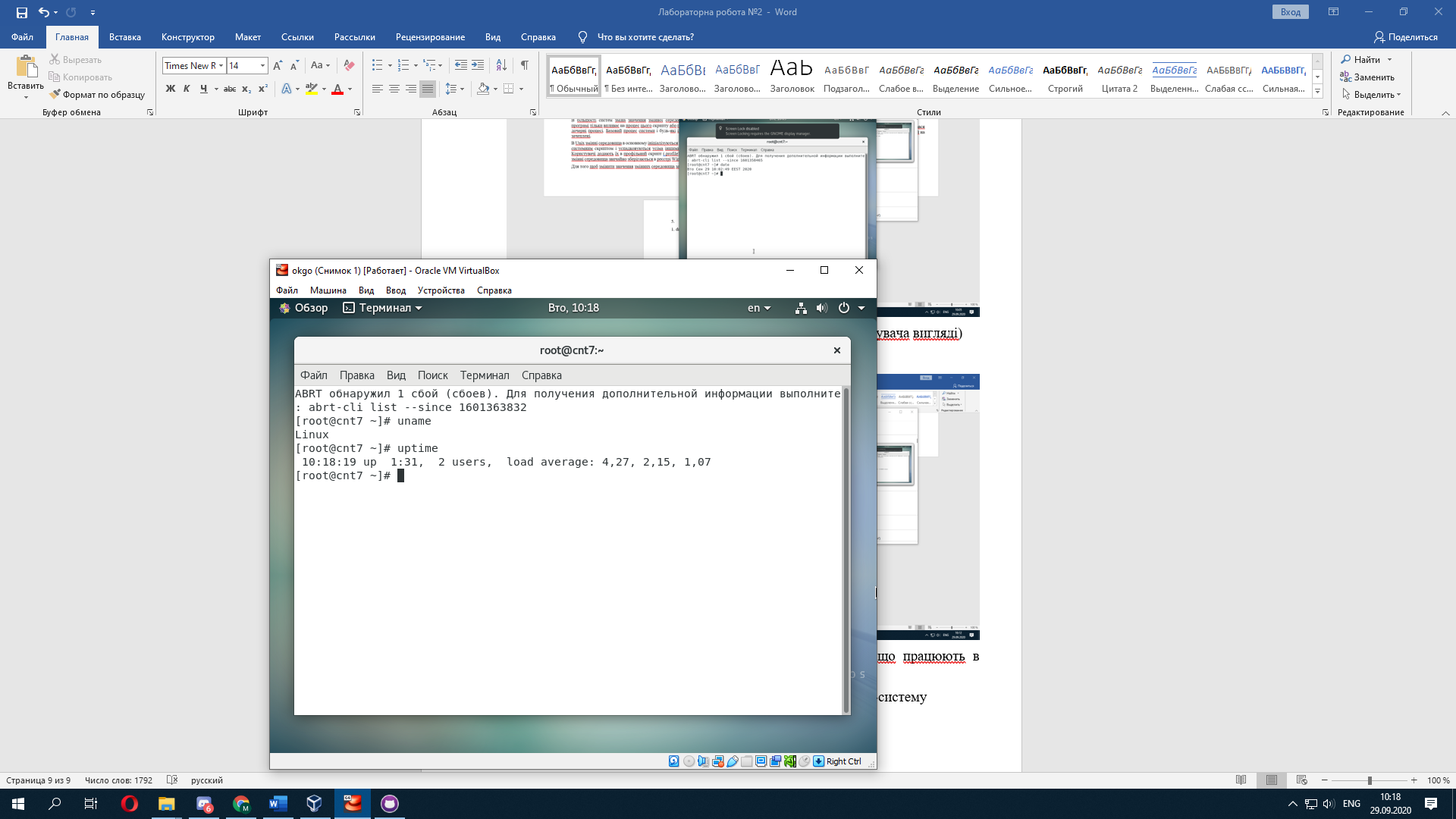


2. команда cal - d. виводить календар (у зручному для користувача вигляді)

3. команда hwclock - e. виводить інтегрований годинник

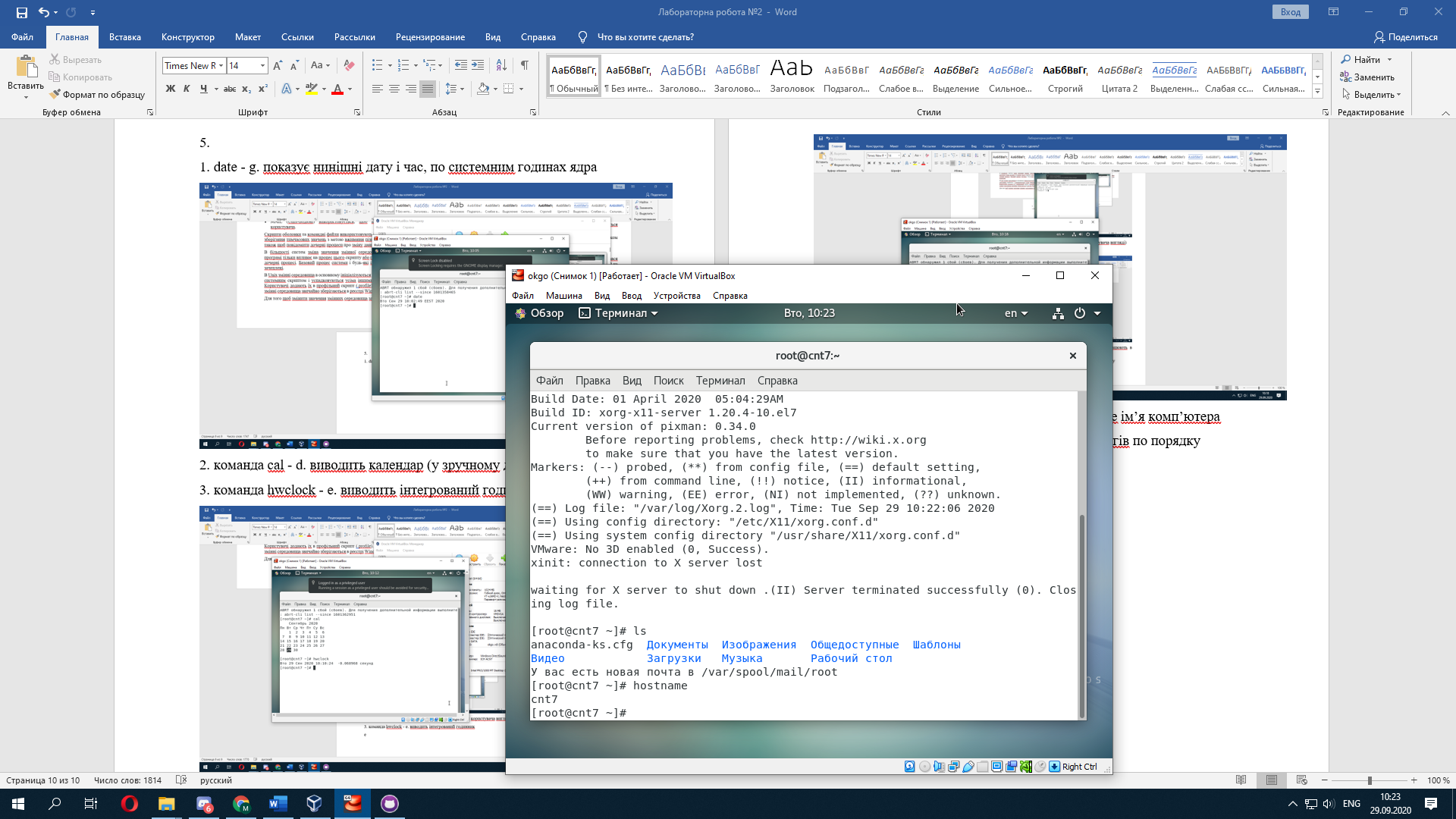


4. команда uptime - g. показує нинішні дату і час, по системним годинах ядра5. команда uname - i. виводить інформацію про поточну unix-систему



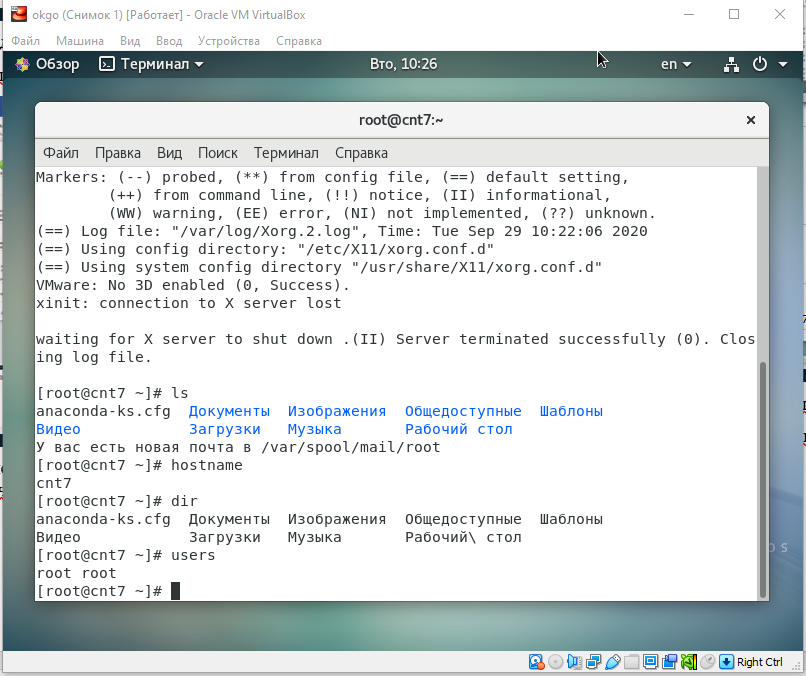
6. команда hostname - a. команда показує мережне ім’я комп’ютера

**7**. команда ls - h. виводить список файлів і каталогів по порядку

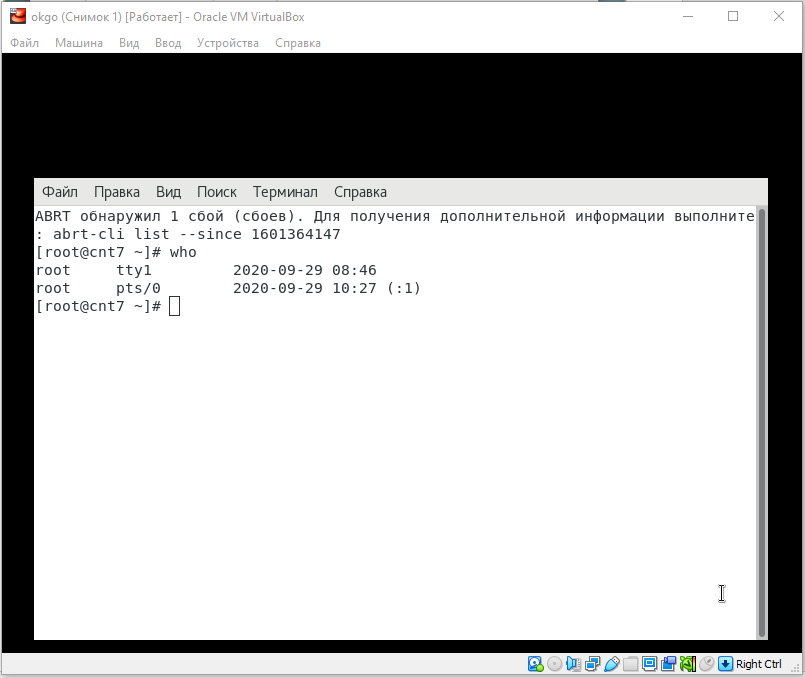


8. команда dir - h. виводить список файлів і каталогів по порядку

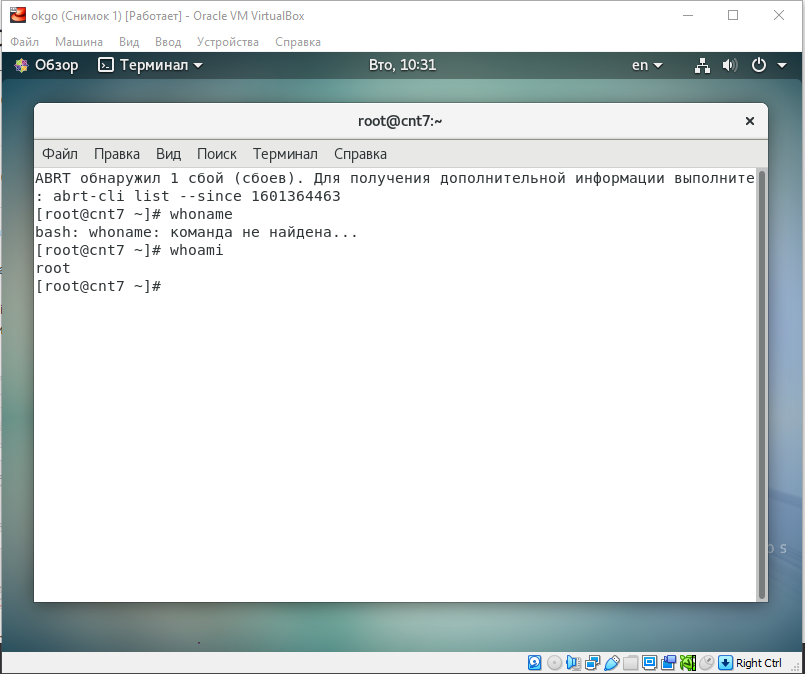
9. команда users - c. відображає перелік користувачів, що працюють в поточний сеанс



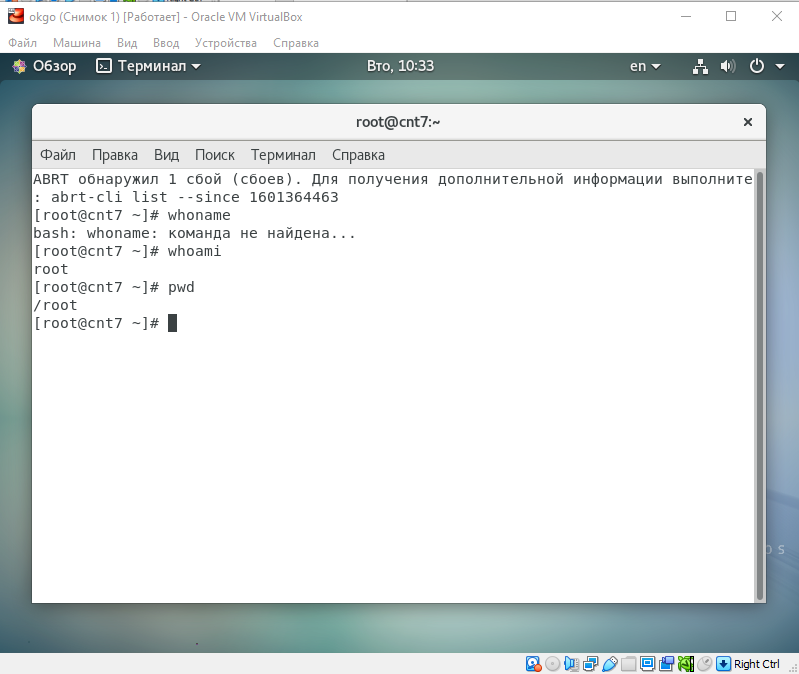
10. команда who - l. показує поточний час і роботу системи (тривалість сеансу, число користувачів і інше) без перезавантаження і виключення.



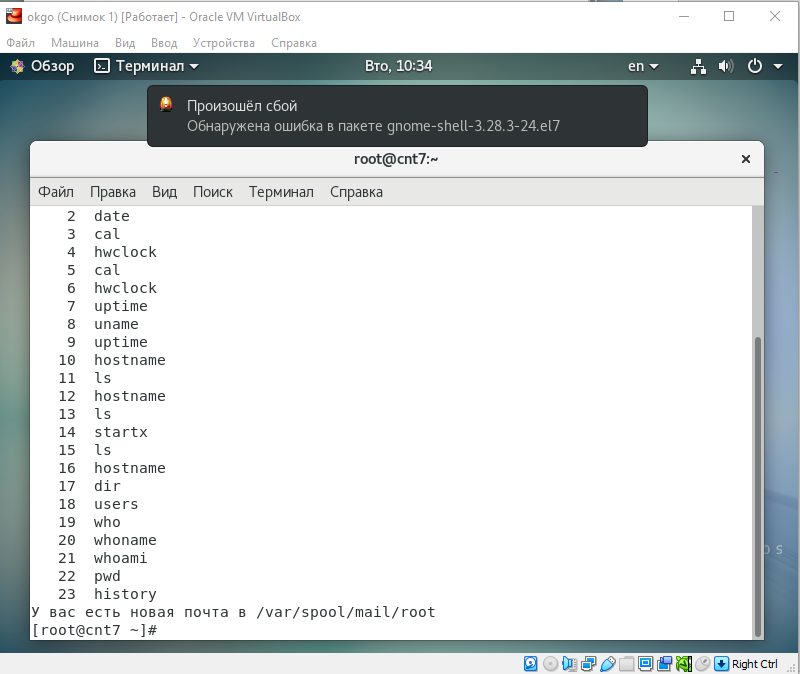
11. команда whoami – c. відображає перелік користувачів, що працюють в поточний сеанс



12. команда pwd - j. показує користувачів системи

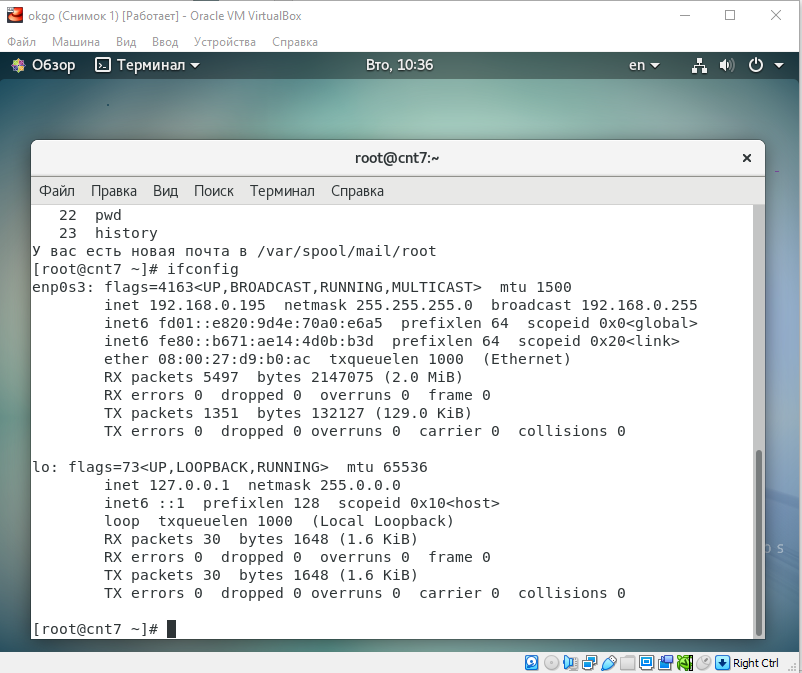


13. команда history – k. демонструє пронумерований перелік команд, які Ви виконували в даному і минулому сеансі.

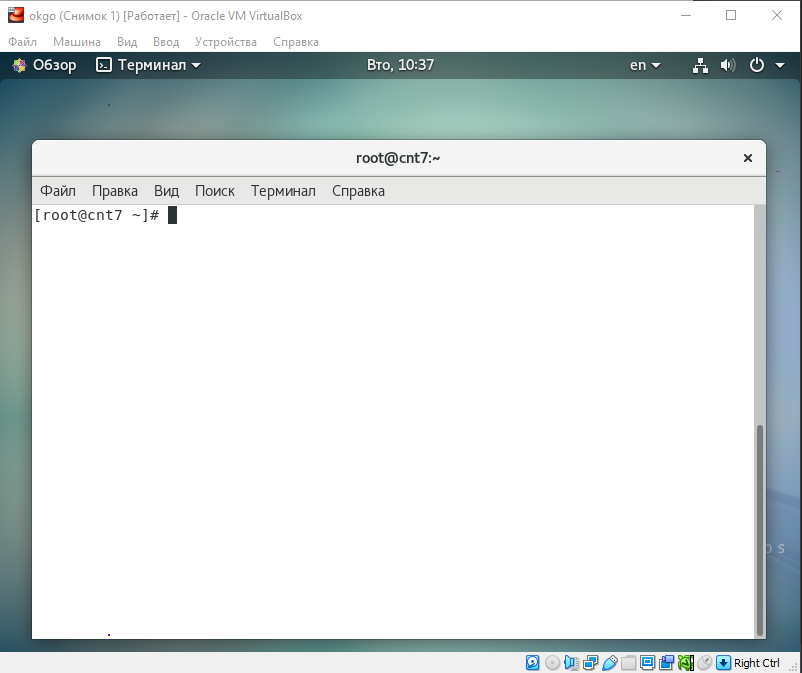


14. команда ifconfig – n. відображає стан поточної конфігурації мережі або ж налаштовує

мережевий інтерфейс.

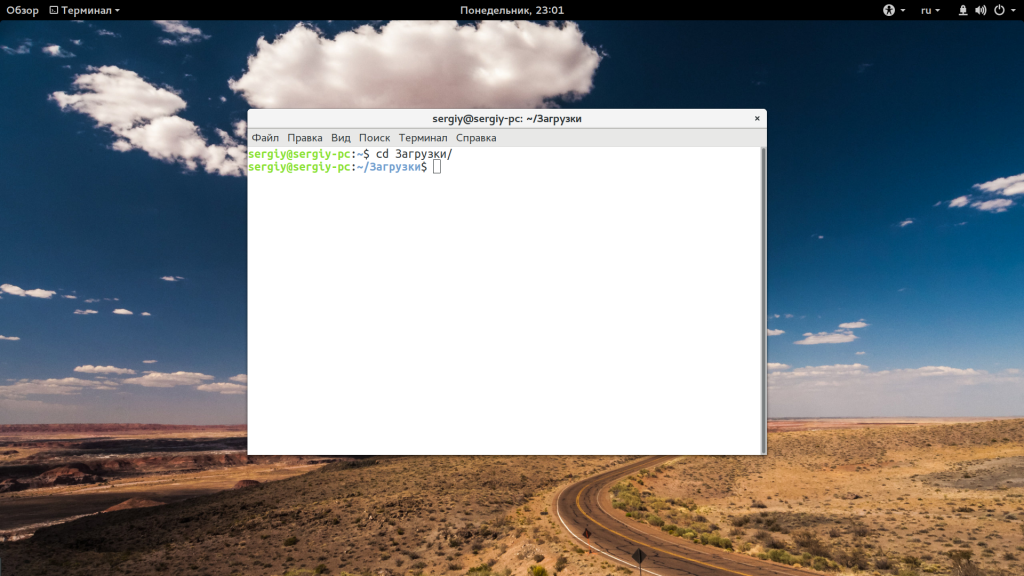


15. clear - очитска



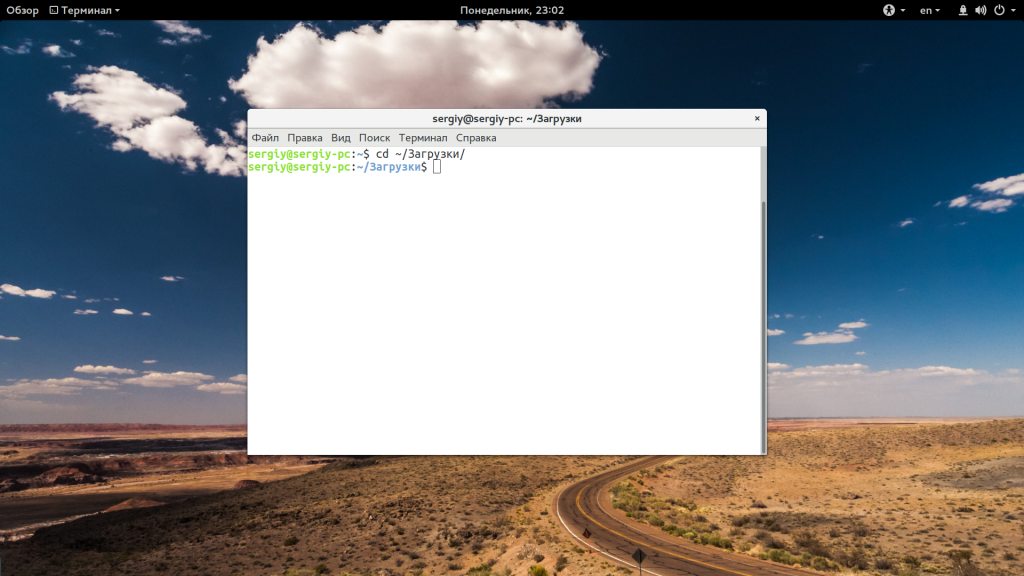
**6.**

Я не буду тут описувати якими бувають шляху в Linux. Ми розглядали цю тему в окремій статті. За замовчуванням, в якості робочого каталогу використовується домашня папка користувача. Давайте спочатку перейдемо в одну з підпапок домашньої папки:



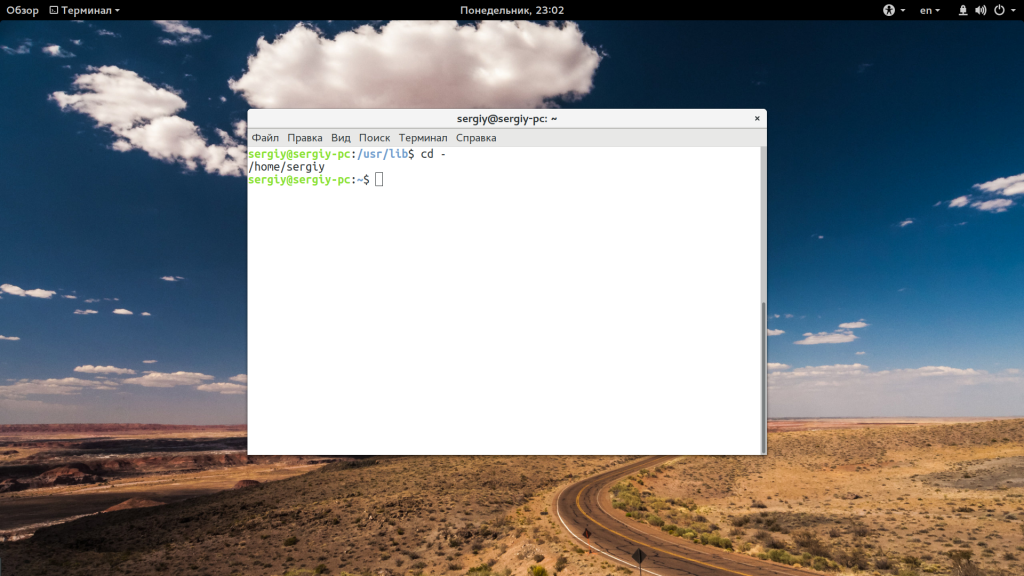
Домашня папка позначається як ~ /. Тому наступна команда виконає аналогічну дію:

cd ~ / Завантаження /



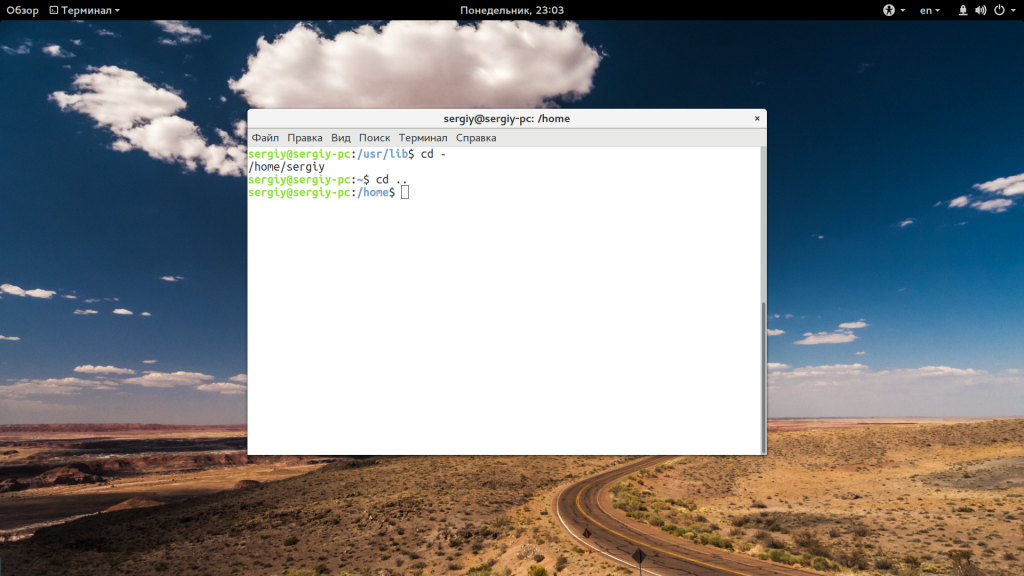
За допомогою символу риси "-" ви можете повернутися в попередню папку:

cd –



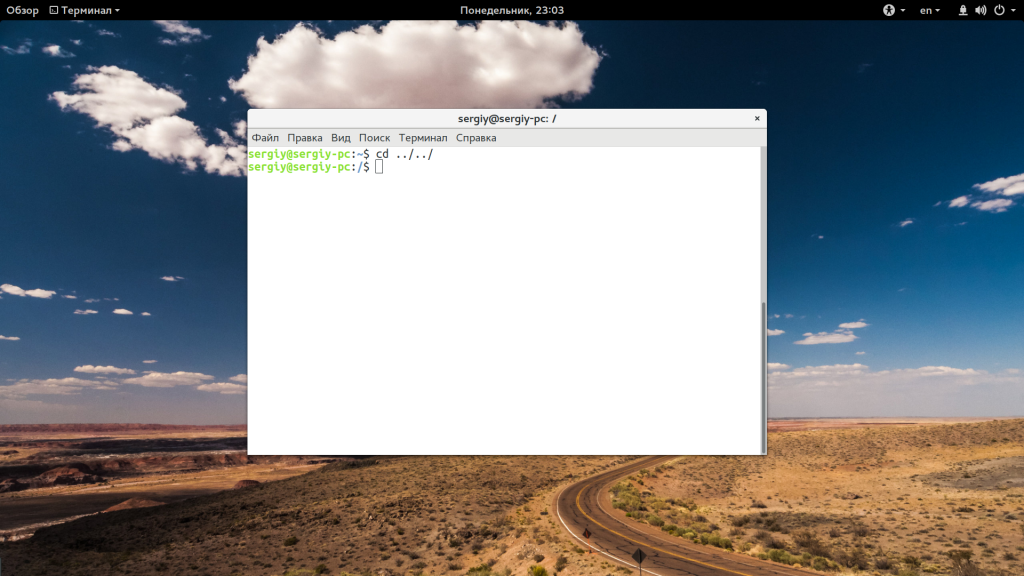
Використовуючи подвійну точку ".." можна перейти в батьківський каталог:

cd ..



Bи можете використовувати кілька блоків з точками для переміщення на кілька рівнів вгору:

cd ../../



**7.** 1. reboot – b Перезавантаження комп’ютера

2. shutdown – a Вимкнення комп’ютера о 17.00

3. poweroff – c Термінове вимкнення комп’ютера

4. halt - c Термінове вимкнення комп’ютера

Найкраща команда для вимкнення – poweroff, адже вона вимикає комп’ютер одразу, але декому команда – shutdown, може здаватися більш српучнішою, адже вона вимикає в той час, який задав користувач

**Контрольні запитання.**

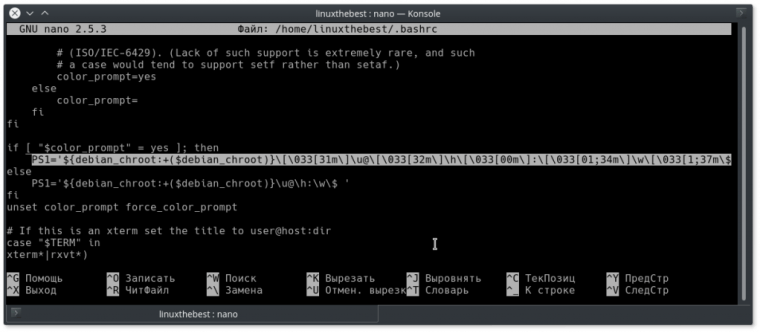
1. За замовчуванням термінал або консоль (KDE), виглядає не дуже привабливо, хоча можна змінити колірні схеми, які доступні в налаштуванні терміналу, але ми можемо змінити чуть-ли не кожен символ в терміналі, і в цій статті я покажу як це зробити.

Ми можемо редагувати колір імені поточного користувача або зробити відступ, або взагалі прибрати ім'я користувача, а поставити замість нього якийсь символ, і все це можна зробити, просто відредагувавши файл bashrc, який знаходиться в нашій домашній директорії.

Для редагування файлу, відкрийте термінал (Ctrl + Alt + T), і введіть команду

nano ~ / .bashrc

І знайдіть ось таку строчку

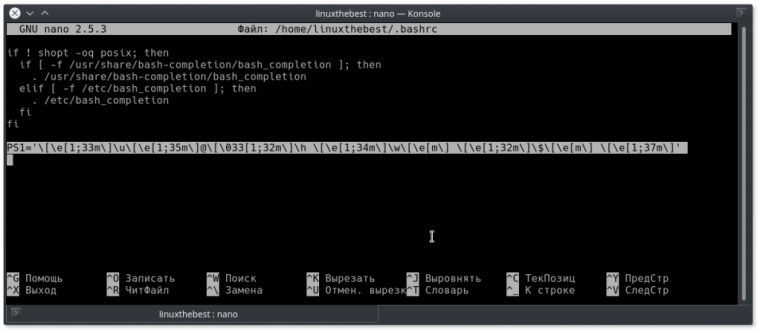


У цьому рядку вказані всі наші кольори для імені користувача, імені хоста і ін.

Для призначення свого кольору, раджу вам скопіювати ось цей рядок

PS1 = '\ [\ e [1; 33m \] \ u \ [\ e [1; 35m \] @ \ [\ 033 [1; 32m \] \ h \ [\ e [1; 34m \] \ w \ [\ e [m \] \ [\ e [1; 32m \] \ $ \ [\ e [m \] \ [\ e [1; 37m \] '

і вставити в кінець вашого файлу bashrc



Зберігаємо за допомогою поєднання клавіш Ctrl + O, і для закриття файлу використовуйте Ctrl + X.

А тепер давайте трохи розберемо, з чого складається цей рядок.

PS1 = '\ [\ e [1; 33m \] \ u - цей шматочок рядка означає що ми присвоюємо імені нашого користувача жовтий жирний колір.

\ [\ E [1; 35m \] @ - далі йде символ собачки з фіолетовим жирним кольором

\ [\ 033 [1; 32m \] \ h - ім'я комп'ютера в мережі жирного зеленого кольору

\ [\ E [1; 34m \] \ w - поточна директорія - жирним блакитним кольором

\ [\ E [1; 32m \] \ $ - знак долара розфарбований жирним зеленим кольором

\ [\ E [1; 37m \] '- і останній білий колір який позначає те що ми будемо вводити в терміналі.

1. Отже, ви придбали перший досвід роботи в текстовому, або "консольному", режимі системи Linux. Поняття "терміналу" і "консолі", які зустрінуться нам ще не раз, потрібно, ймовірно, додатково пояснити.

Коли створювалася система UNIX, комп'ютери були великими (мейнфреймами), і користувачі працювали на них через безліч послідовних інтерфейсів для підключення віддалених терміналів. Термінал - це пристрій, який призначений для взаємодії користувача з комп'ютером і складається з монітора і клавіатури. До вашого персонального комп'ютера напевно не підключені віддалені термінали, але є клавіатура і монітор, які і виконують роль терміналу користувача (тільки в його склад додалася миша).

У мейнфреймів був особливий термінал, який призначався для системного адміністратора і називався консоллю. Консоль зазвичай під'єднують до комп'ютера не по послідовному інтерфейсу, а через окремі роз'єми (іноді в якості пристрою виводу до її складу замість монітора входило принтер).

Оскільки в UNIX-системах зазвичай дотримуються традиції, клавіатура і монітор персонального комп'ютера поводяться так само, як раніше консоль. Перевага такого рішення полягає в тому, що всі старі програми, що створювалися для адміністраторів UNIX, без проблем працюють і на новому типі системної консолі.

Але, крім консолі, Linux дозволяє підключати до комп'ютера і віддалені термінали і, більш того, забезпечує можливість роботи з декількома віртуальними терміналами з одного консолі. Натисніть комбінацію клавіш <Ctrl> + <Alt> + <F2>. Ви знову побачите запрошення login :. Однак це не повернення до початку роботи з системою - ви просто переключилися в інший віртуальний термінал. Тут ви можете зареєструватися під іншим ім'ям. Спробуйте увійти в систему під ім'ям щойно заведеного користувача. Після цього натисніть комбінацію клавіш <Ctrl> + <Alt> + <F1>. Ви повернетеся до першого екрану. За замовчуванням Red Hat Linux відкриває при запуску 6 паралельних сеансів роботи (віртуальних терміналів), і цим іноді дуже зручно користуватися. Для перемикання між віртуальними терміналами використовуються комбінації <Ctrl> + <Alt> + <F1> - <Ctrl> + <Alt> + <F6>. (Зауважимо, що при роботі в текстовому режимі той же результат можна отримати, використовуючи комбінації <Alt> + <F1> - <Alt> + <F6>, проте в графічному режимі без клавіші <Ctrl> не обійтися, так що краще відразу звикати до комбінаціям з 3 клавіш). До речі, якщо в процесі роботи ви забули, в якому терміналі перебуваєте в даний момент, скористайтеся командою tty, яка виводить ім'я терміналу в наступному форматі: / dev / tty2.

Відразу ж скажемо, що, якщо ви хочете завершити сеанс роботи з системою в одному з терміналів, ви можете зробити це натисканням комбінації клавіш <Ctrl> + <D>. Це не призведе ні до зупинки роботи комп'ютера, ні до перезавантаження системи. Не забувайте, що Linux - багатозадачна і розрахована на багато користувачів система. Завершення роботи одного користувача не означає, що треба вимикати комп'ютер. Просто завершується сеанс роботи одного з користувачів, і система знову виводить в даному терміналі запрошення, яке ви вже бачили. Можна завершити сеанс роботи і ввівши одну з команд logout або exit.

Знаючи тепер як відкрити і закрити сеанс роботи в системі, виконайте наведені вище рекомендації, т. Е. Заведіть себе як рядового користувача (без суперпользовательскіх прав), завершіть всі сеанси роботи, відкриті від імені root, і знову увійдіть в систему під своїм новим ім'ям .

Тепер треба сказати кілька слів про оболонці. Оболонка, або просто shell (це слово часто не перекладають, а залишають в англійському написанні), - це програма, яка здійснює всі спілкування з користувачем. Саме оболонка сприймає всі команди, що вводяться користувачем з клавіатури, і організовує виконання цих команд. Тому оболонку можна назвати ще командним процесором (більш звичний термін для користувача DOS, чи не так?). Строго кажучи, коли вище говорилося, наприклад, "система виводить запрошення", це неправильно, оскільки запрошення виводить саме оболонка, чекаючи введення користувачем черговий команди. Кожен раз, коли черговий користувач входить в систему, команда login запускає для нього командний процесор - оболонку. Якщо ви Залогуватися з другого терміналу під ім'ям користувача jim (або під іншим обраним вами іменем), то зверніть тепер увагу на відмінність в запрошеннях у користувачів root і jim. У користувача root запрошення закінчується символом #, а у всіх інших користувачів - символом $.

Оболонку може запускати не тільки команда login. Ви можете просто ввести команду bash (саме так називається програма-оболонка в системі Red Hat Linux) і тим самим запустити новий екземпляр оболонки. Виходячи з нього (по команді exit або по комбінації клавіш <Ctrl> + <D>) ви повернетеся до попереднього примірнику оболонки.

Оболонка bash є не тільки командним процесором, а й потужним мовою програмування. У ній є цілий ряд вбудованих (внутрішніх)

1. Якщо стиль KDE витримано в традиціях Microsoft Windows, то GNOME пропонує інтерфейс користувача, який основано на використанні менеджера вікон Enlightenment, (тепер його замінив sawfish). Для GNOME базовим менеджером дисплея є gdm, а графічною оболонкою, яка включає файловий менеджер - Nautilus.
2. Linux, як і деякі інші версії UNIX, забезпечує доступ до декількох консолей, які дозволяють увійти в систему під декількома іменами в один час. Свіжоінстальований Linux дозволяє працювати з шістьма-родину консолями, перемикання між якими виконується за alt-F1 - alt-F7. Але можливо забезпечити роботу з 12-ю - по одній на кожну функціональну клавішу. Як бачите, ви можете працювати на декількох консолях одночасно. Поки ви працюєте на консолі #1, ви можете переключитися на консоль #2 і почати роботу над чим іншим
3. Тty (від англ. teletypewriter - телетайп) - Unix-утиліта, що виводить ім'я терміналу, з'єднаного зі стандартним вводом. У разі, якщо стандартний ввід не є терміналом, виводить повідомлення «not a tty».

Віртуальні машини є частиною абстракції tty. Подивимося на загальну структуру tty-комплексу і виділимо три компоненти:

**Dev / ttyX** - віртуальний пристрій консолі в файлової системі, яке зайняло місце UART-драйвера і з яким ми вже знайомі. На цьому ж рівні розташовуються пристрої / dev / vcsX і / dev / vcsaX, спілкування з ними здійснюється безпосередньо через / dev / ttyX.

**TTY Line Discipline** - драйвер, який робить ECHO набирємої команди і дає нам можливості її редагування. Також драйвер цього шару генерує сигнали при наборі керуючих послідовностей (^ C, ^ Z і т.д.). За замовчуванням тут панує N\_TTY, однак цей модуль можна замінити, наприклад, своїм драйвером;

**TTY driver** - драйвер, який надає набір методів ініціалізації і відкриття консолі, а також методи, обробні операції введення / виводу, припинення консолі при перемиканні і відновлення її роботи і, звичайно, забезпечує «передачу» отриманої від користувача команди активного процесу.