“Київський коледж зв’язку”

Циклова комісія Комп’ютерної інженерії

ЗВІТ ПО ВИКОНАННЮ

ЛАБОРАТОРНОЇ РОБОТИ №3

з дисципліни: «Операційні системи»

Тема: «Ознайомлення з робочим середовищем

віртуальних машин та операційних систем

різних сімейств»

Виконали

студенти

групи КСМ-83а

Колосовський Р.Ю.\_\_\_\_\_\_

Лісовий М.І.\_\_\_\_\_\_

Перевірив викладач

Повхліб В.С.

Завдання для попердньої підготовки

1. Команди та їх параметри:

|  |  |
| --- | --- |
| */* | Віртуальний каталог, де зазвичай файли не розміщуються |
| /bin | Двійковий каталог, де зберігається багато службових програм GNU на рівні користувача |
| /boot | Каталог, де зберігаються завантажувальні файли |
| /dev | Каталог пристроїв, де Linux створює вузли пристрою |
| /etc | Каталог файлів конфігурації системи |
| /home | Каталог, де Linux створює каталоги користувачів |
| /lib | Каталог бібліотеки, де зберігаються файли бібліотеки системи та додатків |
| /media | Загальне місце для точок монтування, що використовується для знімних носіїв |
| /mnt | Каталог монтажу, ще одне загальне місце для точок монтування, що використовується для знімних носіїв |
| /opt | Необов’язковий каталог, який часто використовується для зберігання сторонніх програмних пакетів та файлів даних |
| /proc | Процес каталог, де зберігається поточне обладнання та інформація про процес |
| /root | root домашній каталог |
| /sbin | Двійковий каталог системи, де зберігається багато службових програм GNU на рівні адміністратора |
| /run | Каталог, де дані про час зберігання зберігаються під час роботи системи |
| /srv | Каталог служб, де місцеві служби зберігають свої файли |
| /sys | Системний каталог, де зберігаються інформаційні файли системного обладнання |
| /tmp | Тимчасовий каталог, де можна створювати та знищувати тимчасові робочі файли |
| /usr | Користувацький двійковий каталог, де зберігається основна частина службових програм GNU та файлів даних |
| /var | Каталог змінних для файлів, які часто змінюються, наприклад файлів журналів |

2)

Файлова структура Unix характеризується такими властивостями:

Чітка побудова;

Звернення до даних файлу без суперечностей;

Захист даних файлу;

Цей стандарт вимагає, щоб операційна система, яка працюватиме з СПФС узгодженою файловою системою, підтримуватиме ці базові риси надійності, які є в більшості файлових систем Unix. Цей стандарт не намагається узгоджуватися з кожним можливим аспектом в усіх конкретних реалізаціях Unix-систем. Проте, багато аспектів цього стандарту базується на ідеях, що є в Unix та інших Unix подібних системах.

Це — після уважного розгляду інших факторів, включає:

Традиційна та добре сприйнята практика в Unix-подібних системах;

Впровадження побудов інших файлових структур;

Застосовні стандарти;

Визначається дві незалежні категорії файлів: загальні (shareable) на противагу приватним (unshareable) та змінні на противагу постійним.

Загальні дані

це ті, що можуть бути спільними для декількох головних машин;

Приватні дані

мають бути специфічними для кожного головного комп'ютера.

Наприклад, домашні каталоги користувачів — загальні данні, але файли контролю пристроїв — ні.

Постійні файли

двійкові, бібліотеки, документація та все інше, що має змінюватися тільки адміністиратором системи;

Змінні

все, що може бути змінено без втручання адміністиратора системи.

Для полегшення резервування, керування та спільного використання файлів в мішаних мережах з машин із різними архітектурами та операційними системами, бажано щоб було просто та легко розуміти зв'язки між каталогами (певні каталоги ро зглядаються як потенційні точки монтування) та типом даних, що вони містять.

Скрізь в цьому документі та в усіх добре спланованих файлових системах, знання цих базових принципів допоможе побудувати структуру та надасть їй додаткової логічності.

Різниця між загальними та приватними даними потребує деяких пояснень:

В мережевому середовищі (тобто коли є декілька головних машин в одному місці), є гарною практикою мати дані загальні для декількох головних машин, з метою збереження місця та полегшення завдань супроводу.

В мережевому середовищі, деякі файли містять дані, що стосуються винятково конкретної головної системи. Такі частини файлової системи не можуть бути узагальнені (без вживання певних заходів).

Попередні реалізації файлових систем Unix-типу, перемішують загальні та приватні дані в одній структурі, що ускладнює узагальнення великих частин файлової системи.

Особливість «загальності» може бути використана для спрощення підтримки, наприклад:

Розділ /usr (або його частини) монтується (тільки на читання) через мережу (використовуючи NFS).

Розділ /usr (або його частини) монтується з носія, що може тільки читатися. CD-ROM з однією копією, що розповсюджується між багатьма користувачами через пошту або іншими засобами, може бути змонтований як файлова система тільки на читання, що узагальнюється з іншими файловими системами узгодженими з СПФС яким-небудь типом «мережі».

Властивості «постійний» та «змінний» справляють на файлову систему вплив головним чином двома шляхами:

Оскільки / містить як змінні так і постійні дані, його доводиться монтувати на читання та запис.

Хоч зазвичай /usr містить як змінні так і постійні дані, все ж буває потрібно монтувати його тільки для читання; тоді треба мати метод монтувати /usr тільки на читання. Це робиться через створення структури /var, що монтується на читання та запис (або як частина іншого розділу, що дозволяє читання та запис як /), перебираючи на себе значну частину функціональності /usr розділу.

Linux зберігає файли в одній структурі каталогів, яка називається віртуальною

каталог. Віртуальний каталог містить файлові шляхи з усіх накопичувачів, встановлених на комп'ютері, об'єднаних у

єдину структуру каталогів.

Структура віртуального каталогу Linux містить єдиний базовий каталог, який називається кореневим. Каталоги та файли

під кореневим каталогом перелічено на основі шляху до каталогу, використовуваного для потрапляння до них, подібно до того, як це робить Windows.

Ви помітите, що Linux використовує косу риску (/) замість косою рискою (\) для позначення каталогів у файлі

стежки. Символ зворотної косої риски в Linux позначає символ втечі та викликає всілякі проблеми при його використанні

у шляху до файлу. Це може зайняти деяке звикання, якщо ви походите із середовища Windows.

У Linux ви побачите файлові шляхи, подібні до таких:

/home/Rich/Documents/test.doc

Це вказує на те, що файл test.doc знаходиться в каталозі Документи під розширеним каталогом, який міститься в

додому каталог. Зверніть увагу, що шлях не надає жодної інформації щодо того, на якому фізичному диску зберігається файл.

Хитра частина віртуального каталогу Linux полягає в тому, як він включає кожен пристрій зберігання даних.

Перший жорсткий диск, встановлений у системі Linux, називається кореневим диском. Кореневий диск містить ядро ​​віртуального каталогу. Все інше будується звідти**.**

На кореневому диску Linux може використовувати спеціальні каталоги як точки монтування. Точки монтування - це каталоги у віртуальному каталозі, де ви можете призначити додаткові пристрої зберігання. Linux призводить до того, що в цих каталогах точок монтування з’являються файли та каталоги, хоча вони фізично зберігаються на іншому диску.

Часто системні файли фізично зберігаються на кореневому диску. Файли користувача зазвичай зберігаються на окремому диску або дисках, як показано

b) стандарт, прийнятий для уніфікації розташування файлів і каталогів загального призначення у файловій системі операційних систем UNIX. Сьогодні, більшість UNIX-подібних систем, в тій або іншій мірі, слідують цим правилам. Наприклад, типова база даних про користувачів системи Linux завжди зберігається у файлі /etc/passwd.

с) Каталог для монтування (від англ. mount) - тимчасового підключення файлових систем, наприклад, на знімних носіях (флеш-диски, CD-ROM та інші).***mount(special pathname,directory pathname,options);***

де **special pathname** - ім'я спеціального файлу пристрою (у загальному випадку ім'я розділу), відповідного дисковому розділу з вмонтовуваною файловою системою, **directory pathname** - каталог в існуючій ієрархії, де вмонтовуватиметься файлова система (іншими словами, крапка або місце монтування), а **options** указує, чи слід вмонтовувати файлову систему "тільки для читання" (при цьому не виконуватимуться такі функції, як write і creat, які проводять запис у файлову систему). Потім ОС повинна переконатися, що пристрій містить дійсну файлову систему очікуваного формату з суперблоком, списком індексів і кореневим індексом.

Деякі ОС здійснюють монтування автоматично, як тільки зустрінуть диск вперше (жорсткі диски на етапі завантаження, гнучкі, - коли вони вставлені в дисковод), ОС шукає файлову систему на пристрої. Якщо файлова система на пристрої є, вона вмонтовується на кореневому рівні, при цьому до ланцюжка імен абсолютного імені файлу (pathname) додається буква розділу.

d) 1. для створення директорії(папки) використовуєтсья команда - */mkdir* ;

для створення файлу, використовується команда - */touch* ;

2. для видалення директорії потрібно ввести команду - */rm –r* ;

для видалення файлу, просто - */rm* ;

3. для копіювання директорії потрібно вписати команду - */cp –r* ;

для копіювання файлу, просто - */cp ;*

4. для переміщення як директорій так і файлів використовується команда - */mv;*

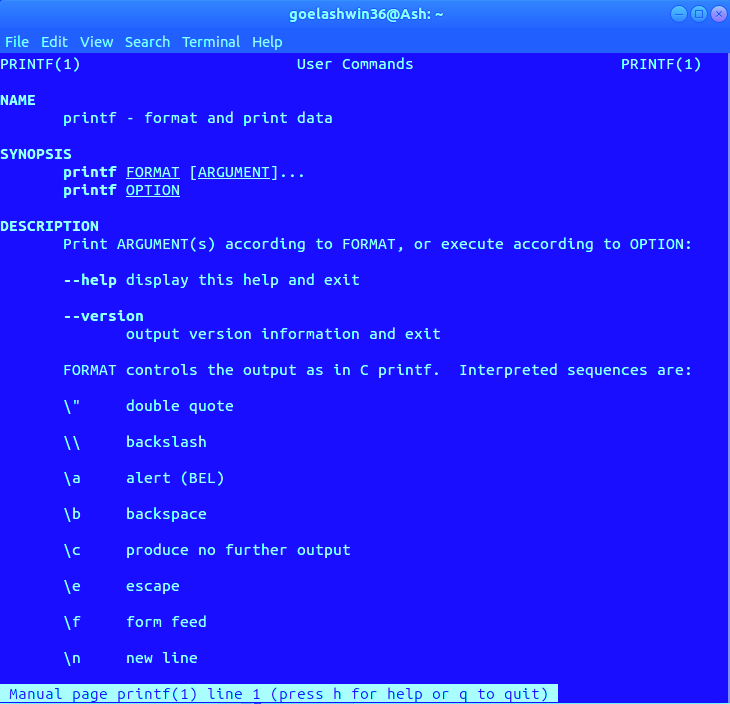
5. для перегляду вмісту можна використати команду */cat* або */less* .

2.

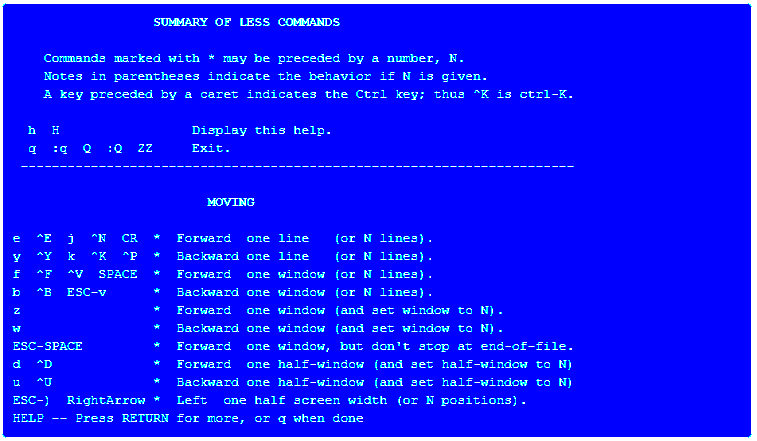
Date - відобразити сьогоднішню дату:



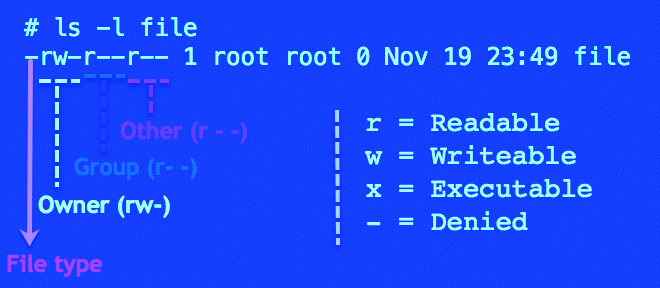
Щоб дізнатися більше про команди, перейдіть на сторінку керівництва для команди з командою *man*. Наприклад, виконайте таку команду, щоб дізнатися більше про команду *date*:



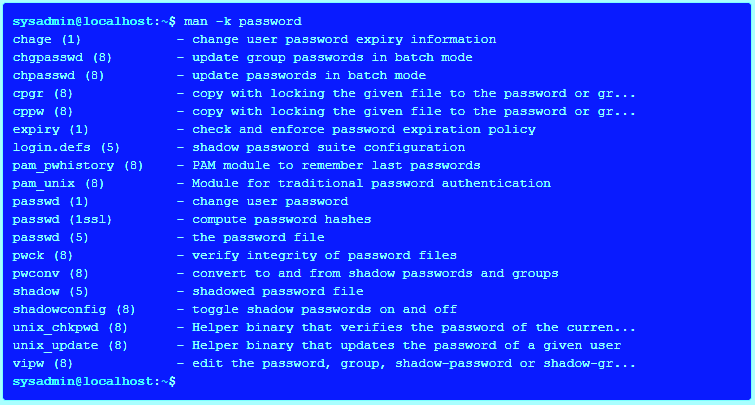
Введіть літеру *h*, щоб переглянути список команд руху. Прочитавши команди руху, введіть літеру *q*, щоб повернутися до документа.



Пошукові запити не чутливі до регістру та не “обертаються” знизу вгору або навпаки. Почніть прямий пошук слова “file”, набравши */file*

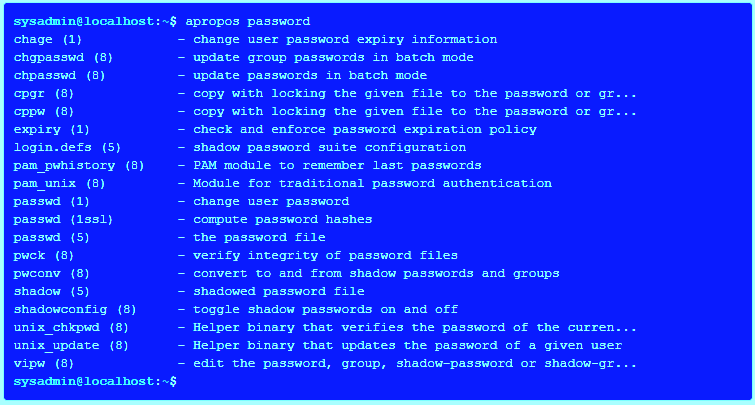


-*k password*



Зауважте, що команда apropos - це ще один спосіб перегляду резюме сторінок з ключовими словами. Введіть таку команду

*apropos password*



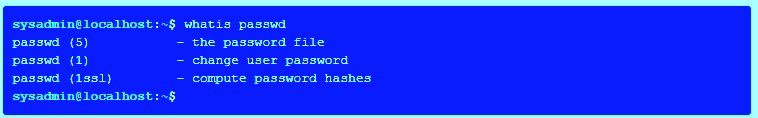
Часто існує декілька сторінок із однаковими назвами. Наприклад, наступна команда показує три сторінки для passwd. Виконайте наступну команду для перегляду сторінок керівництва для слова passwd:

*man -f passwd*



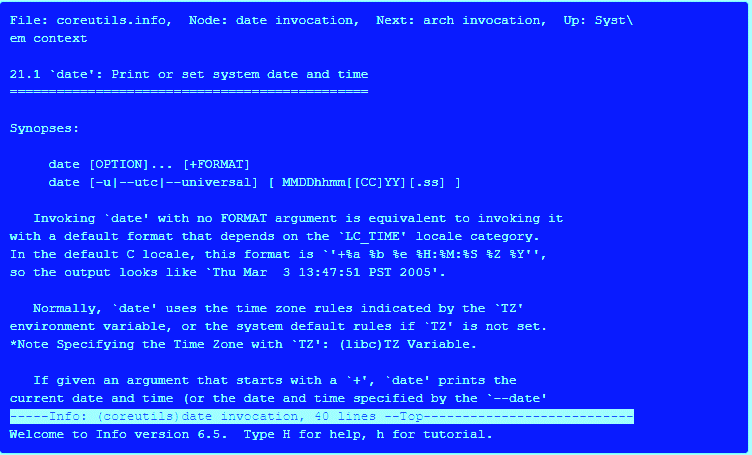
Замість того, щоб використовувати man -f для відображення всіх розділів сторінки man для імені, ви також можете використовувати команду whatis:

*whatis passwd*



Майже всі системні функції (команди, системні файли тощо) мають сторінки користувача. Деякі з цих функцій також мають вдосконалену функцію, яка називається інформаційними сторінками. Наприклад, виконайте таку команду:

*into date*



Найпростіший спосіб пошуку файлу - це використання команди locate. Наприклад, ви можете знайти розташування файлу crontab, виконавши таку команду:

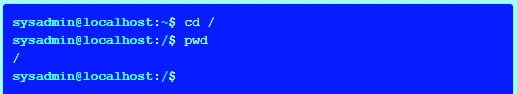
*locate crontab*



Ви можете використовувати команду cd із шляхом до каталогу, щоб змінити поточний каталог. Введіть таку команду, щоб зробити кореневий каталог вашим поточним робочим каталогом і підтвердити за допомогою команди pwd:

*cd/*

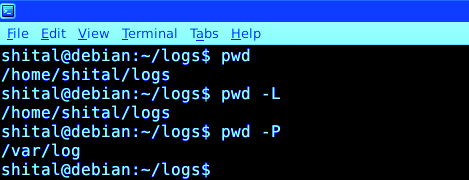
*pwd*



Щоб повернутися до домашнього каталогу, команду cd можна виконати без шляху. Поверніться до домашнього каталогу та перевірте, ввівши такі команди:

*cd*

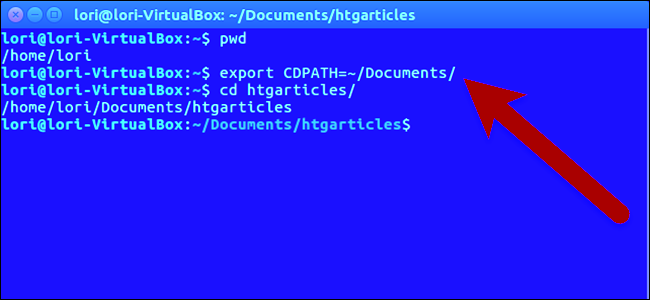
*pwd*



Команда cd може бути введена із шляхом до каталогу, вказаного як аргумент. Виконайте команду cd з каталогом / home як аргумент, набравши наступне:

*cd /home*

*pwd*



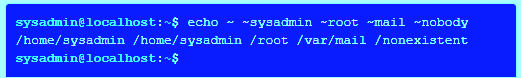
Поверніться до домашнього каталогу, використовуючи команду cd з тильдою ~ як аргумент:

*cd ~*

*pwd*

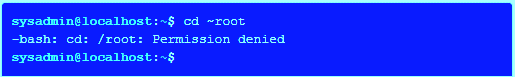
Використовуйте команду echo нижче, щоб відобразити деякі інші приклади використання тильди як частини шляху:

*echo ~ ~sysadmin ~root ~mail ~nobody*



Спробуйте перейти до домашнього каталогу кореневого користувача, набравши таку команду:

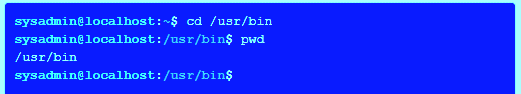
*cd ~root*



Використовуючи абсолютний шлях, перейдіть до каталогу / usr / bin та відображте робочий каталог, використовуючи такі команди:

*cd /usr/bin*

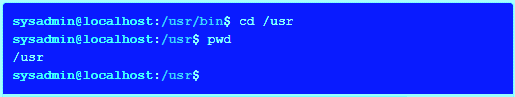
*pwd*



Використовуйте абсолютний шлях, щоб перейти до каталогу / usr і відобразити робочий каталог, виконавши такі команди:

*cd /usr*

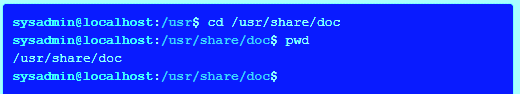
*pwd*



Використовуйте абсолютний шлях, щоб перейти до каталогу / usr / share / doc та відобразити робочий каталог, виконавши такі команди:

*cd /usr/share/doc*

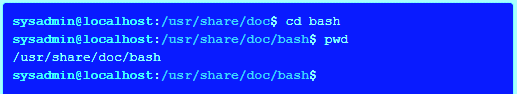
*pwd*



Використовуючи відносний шлях, перейдіть до каталогу / usr / share / doc / bash і відображте робочий каталог, виконавши такі команди:

*cd bash*

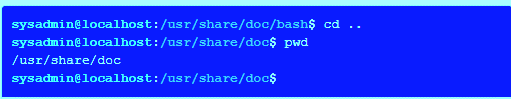
*pwd*



Використовуйте відносний шлях, щоб перейти до каталогу над поточним каталогом:

*cd ..*

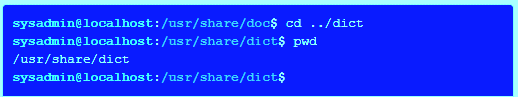
*pwd*



Використовуйте відносний шлях, щоб перейти на один рівень з поточного каталогу, а потім вниз у каталог dict:

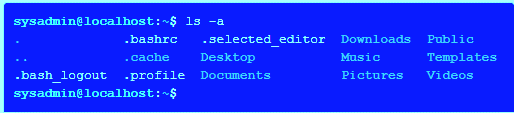
*cd ../dict*

*pwd*



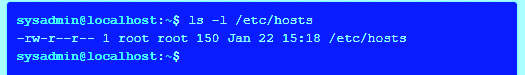
Не всі файли відображаються за замовчуванням. Є файли, які називаються прихованими файлами, які за замовчуванням не відображаються. Щоб відобразити всі файли, включаючи приховані файли, використовуйте опцію -a до команди ls:

*ls -a*



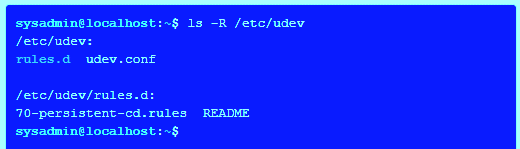
Сама по собі команда ls просто надала імена файлів та каталогів у зазначеному (або поточному) каталозі. Виконайте наступну команду, щоб побачити, як параметр -l надає додаткову інформацію про файл:

*ls -l /etc/hosts*



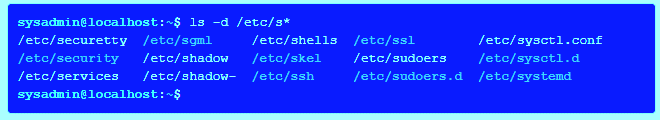
Іноді хочеться побачити не лише вміст каталогу, але й вміст підкаталогів. Ви можете використовувати параметр -R для цього:

*ls -R /etc/udev*



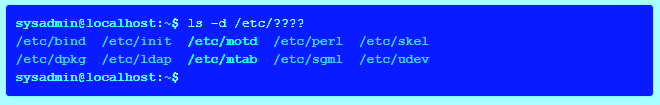
Ви можете використовувати глобалізацію файлів (узагальнюючі символи), щоб обмежити, які файли або каталоги ви бачите. Наприклад, символ \* може збігатися з "нулем або більше будь-яких символів" в назві файлу. Виконайте наступну команду, щоб відобразити лише файли, що починаються на букву s в каталозі / etc:

*ls -d /etc/s\**



? Символ може використовуватися для відповідності рівно 1 символу в назві файлу. Виконайте наступну команду, щоб відобразити всі файли в каталозі / etc, які мають рівно чотири символи:

*ls -d /etc/????*

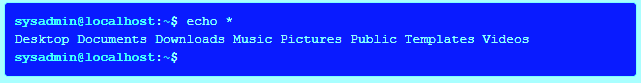


За допомогою квадратних дужок [] ви можете вказати один символ, який збігатиметься з набору символів. Виконайте наступну команду, щоб відобразити всі файли в каталозі / etc, які починаються на літери a, b, c або d:

ls -d /etc/[abcd]\*

Використовуйте таку команду echo, щоб відобразити всі імена файлів у поточному каталозі, які відповідають шаблону глобуса \*:

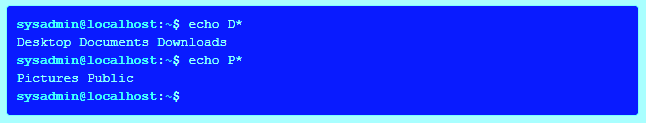
*echo \**



Наступні команди відображатимуть усі файли в поточному каталозі, що починаються з літери D та літери P:

*echo D\**

*echo P\**



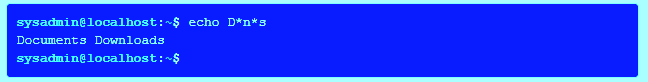
Зірочку \* можна використовувати де завгодно в рядку. Наступна команда покаже всі файли у вашому поточному каталозі, які закінчуються на букви s:

*echo \*s*



Зверніть увагу, що зірочка також може відображатися кілька разів або в середині декількох символів:

*echo D\*n\*s*



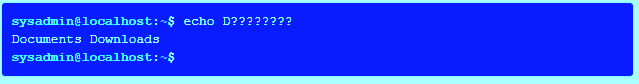
Оскільки кожен знак запитання відповідає одному невідомому символу, введення шести з них буде відповідати іменам файлів із шести символів. Введіть наступне, щоб відобразити імена файлів, які мають рівно шість символів:

*echo ??????*



Використання знака питання з іншими символами обмежить збіги. Введіть наступне, щоб відобразити імена файлів, які починаються з літери D і мають рівно дев'ять символів:

*echo D????????*



Узагальнюючі символи або символи глобуса можна поєднувати разом. Наступна команда відобразить імена файлів, які мають принаймні шість символів і закінчуються літерою s.

*echo ?????\*s*



Наступний глобус подібний до глобуса із запитанням, щоб вказати один символ.

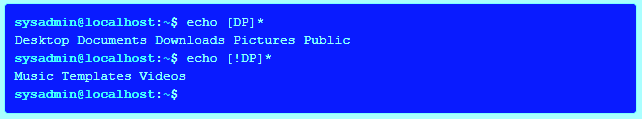
Цей глобус використовує пару квадратних дужок [], щоб вказати, який символ буде дозволений. Дозволені символи можуть бути вказані як діапазон, список або як відомий як клас символів.

Дозволені символи також можна заперечувати знаком оклику!.

У першому прикладі першим символом імені файлу може бути D або P. У другому прикладі першим символом може бути будь-який символ, крім D або P:

*echo [DP]*\*

*echo[!DP]\**



У наступних прикладах буде вказано діапазон символів. У першому прикладі першим символом імені файлу може бути будь-який символ, що починається з D і закінчується на P. У другому прикладі цей діапазон символів заперечується, тобто будь-який окремий символ збігатиметься, якщо він не знаходиться літери D і P:

*echo [D-P]\**

*echo [!D-P]\**

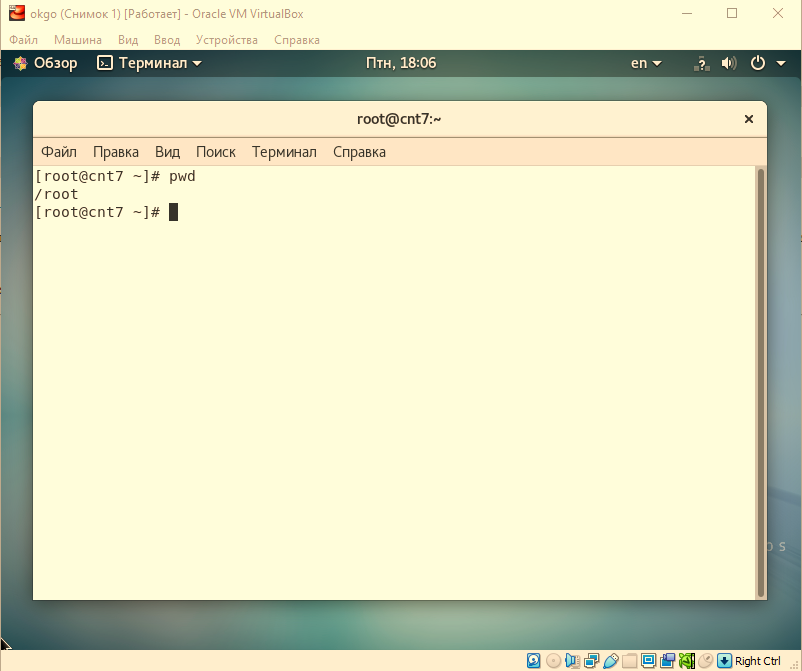


3.

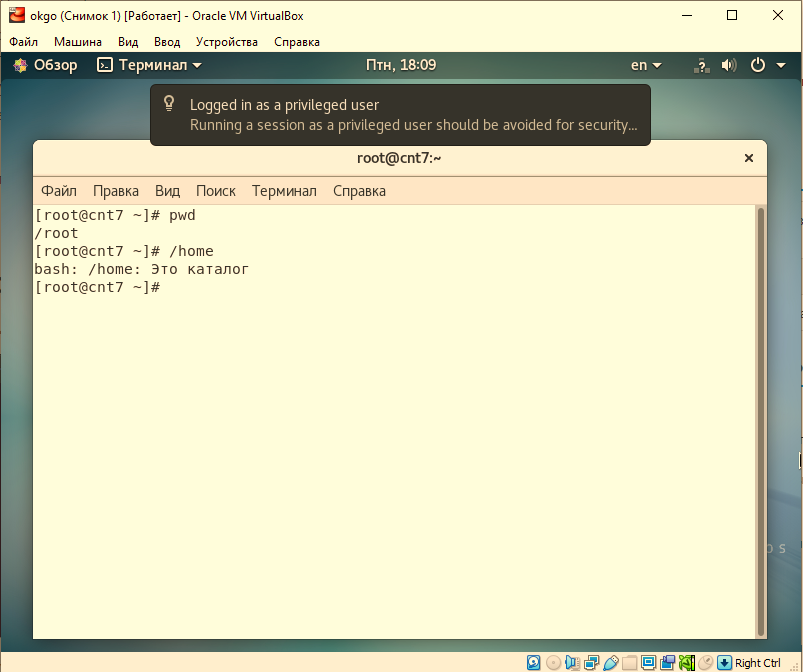
|  |  |
| --- | --- |
| Назва команди | Її призначення та функціональність |
| date | Відображає сьогодняшню дату |
| man | Показує інструкцію до програм і командам Linux. man (від англ. manual - керівництво) - команда Unix, призначена для форматування і виведення довідкових сторінок. |
| whatis | Виводить короткий опис програми. |
| locate | Швидкий пошук файлів по базі даних updatedb для шаблонів імен файлів. Ця БД працює зі знімком файлової системи, що багаторазово прискорює процес пошуку. Але тут є і істотний мінус. Точність пошуку залежить від того, наскільки актуальні дані про файлову систему в поточній версії знімка. |
| ls | (List). За допомогою цієї утиліти можна подивитися, що міститься в папці. Без вказівки конкретного шляху показує поточний каталог. Якщо шлях невідомий, то перехід відбутися в кінцевий його каталог.  -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 ..;  -d - виводити тільки директорії, без їх вмісту, корисно при рекурсивном виведення;  -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;  -r - зворотний порядок сортування;  -R - рекурсивно відображати вміст піддиректорій;  -s - виводити розмір файлу в блоках;  -S - сортувати за розміром, спочатку великі; |
| echo | -n - не виводити новий рядок;  -e - включити підтримку виведення Escape послідовностей;  -E - відключити інтерпретацію Escape послідовностей.  Це все опції, якщо включена опція -e, то ви можете використовувати такі Escape послідовності для вставки спеціальних символів:  / C - видалити новий рядок;  / T - горизонтальна табуляція;  / V - вертикальна табуляція;  / B - видалити попередній символ;  / N - новий рядок;  / R - символ повернення каретки на початок рядка. |
| pwd | Показує на екрані поточний каталог. |
| cd | cd  (Change Directory). Змінює поточний каталог, в якому працює термінал на вказаний. Коли термінал запущений, він використовує за замовчуванням кореневий каталог (root), повернутися в який можна, набравши:  cd ~  Щоб встановити іншу для роботи з файлами, потрібно ввести:  cd адрес\_папкі  Щоб піднятися по древу каталогів на рівень вгору набираємо:  cd ..  Щоб повернутися до попередньої директорії:  cd - |
| Apropos\whatis | Команда apropos здійснює пошук за фрагментами слів. Для того, щоб отримати потрібну інформацію, завжди потрібно знати, що шукати. У таких випадках можуть допомогти команди whatis ( `` Що є '') і apropos ( `` ??? ''). Команда whatis виробляє контекстний пошук заданого ключового слова (шаблону) в базі даних, що містить перелік системних команд з коротким їх описом. Виводяться тільки точні співпадіння з ключовим словом. Команда apropos здійснює пошук за фрагментами слів. Аналогічно команді apropos працює команда man з прапором -k:  man -k net  Для того, щоб команди man -k, whatis і apropos працювали, необхідно попередньо створити базу даних про системні командах, для чого ввести:  makewhatis  В іншому випадку у відповідь на будь-який запит буде видано повідомлення `` nothing appropriate ''. Правом запустити команду makewhatis володіє тільки привілейований користувач root. У разі безперервної роботи комп'ютера найкраще запускати цю команду як завдання для процесу cron. |

4.

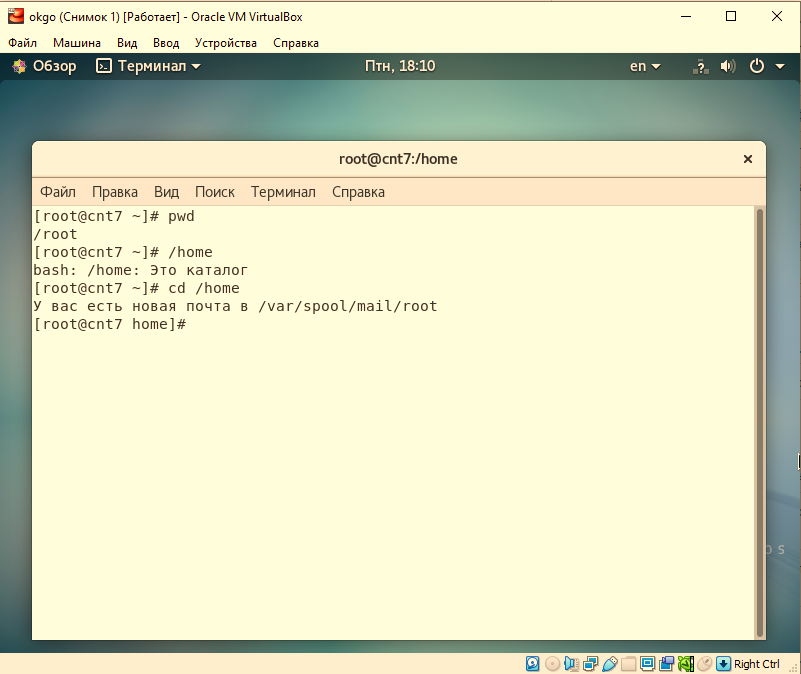
-pwd



* /home



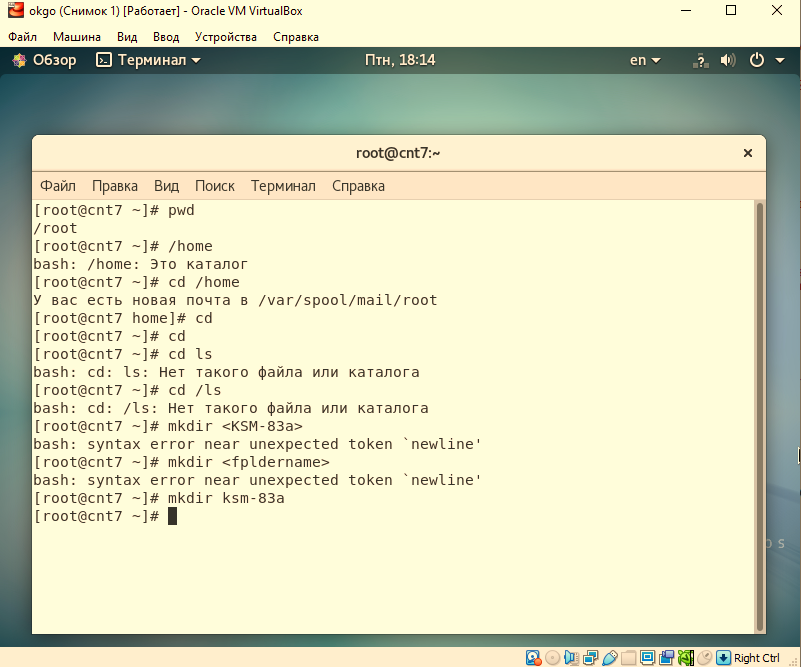
* Cd / home



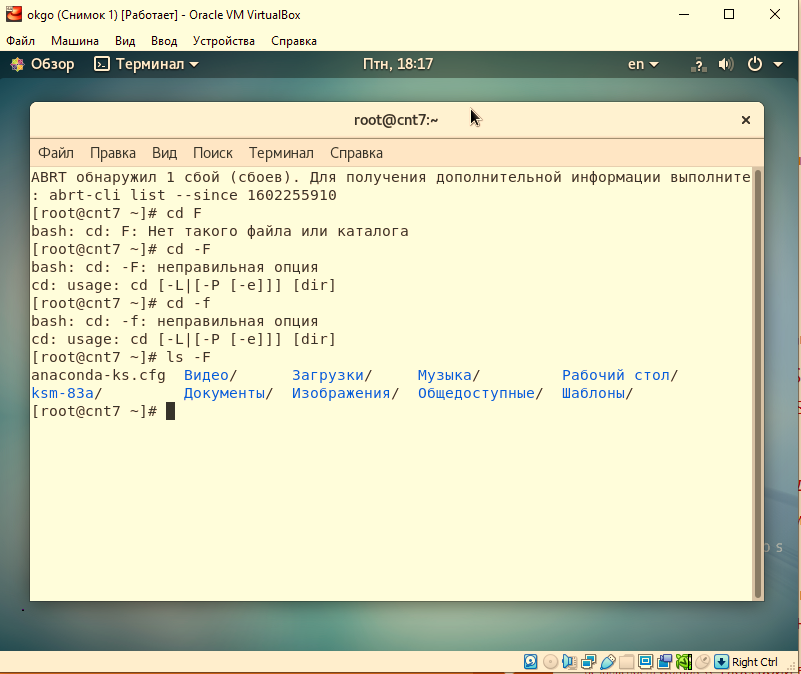
-cd

-mkdir

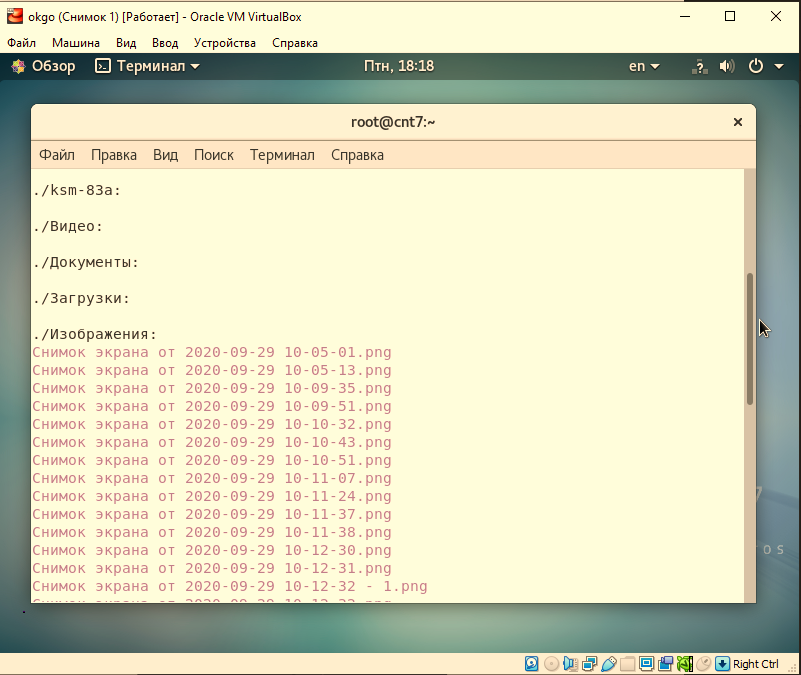
-



-



-



Контрольні запитання

1. cat - одна з найбільш часто використовуваних команд в Linux. Вона зчитує дані з файлів і виводить їх вміст. Це найпростіший спосіб відображення вмісту файлу в командному рядку.

синтаксис

cat [ОПЦИИ] ... [Фото] ...

Її можна використовувати для відображення текстових файлів, копіювання текстових файлів в новий документ, приєднання вмісту текстового файлу в кінці іншого текстового файлу для їх об'єднання.

Відображення текстових файлів

Найпростіший спосіб використання cat - задати команді ім'я файлу. Вміст текстового файлу буде виведено на екран. Наприклад, наступна команда прочитає вміст файлу mytext.txt і виведе його в стандартний потік виводу (екран вашого терміналу):

cat mytext.txt

Якщо файл занадто довгий, він перегорнув, і ви побачите тільки останній екран. При необхідності переглянути документ посторінково або прокручувати його вперед і назад можна скористатися такими засобами перегляду як pg, more або less.

Коли вказано кілька файлів, cat буде виводити їх послідовно, об'єднуючи вміст в потоці стандартного виводу. Тому наступна команда виведе вміст двох текстових файлів, як якщо б вони були одним:

cat mytext.txt mytext2.txt

Копіювання текстового файлу

Зазвичай копіювання здійснюється командою cp, але для копіювання текстових файлів ви можете аналогічним чином використовувати команду cat. cat відправляє результат в stdout (стандартний потік виводу), зазвичай це екран терміналу. Однак, можна перенаправити результат в файл за допомогою символу перенаправлення ">".

Наприклад, команда:

cat mytext.txt> newfile.txt

вважає вміст mytext.txt і відправити його в стандартний потік виведення, але замість відображення тексту оболонка перенаправить результат в файл newfile.txt. Якщо цей файл не існує, він буде створений. Якщо він буде ж існує, то буде перезапісантс втратою попереднього вмісту, тому будьте обережні.

Аналогічно можна об'єднувати кілька файлів в один. Наприклад, команда

cat mytext.txt mytext2.txt> newfile.txt

вважає вміст mytext.txt і mytext2.txt і запише об'єднаний текст в newfile.txt. При відсутності файлу він буде створений, за наявності - перезаписан.

Приєднання вмісту текстового файлу в кінець іншого текстового файлу

Замість перезапису файлу можна приєднати текст файлу-джерела до іншого файлу за допомогою оператора перенаправлення «>>».

Наприклад, команда:

cat mytext.txt >> another-text-file.txt

Прочитає вміст mytext.txt і запише його в кінець файлу another-text-file.txt. Якщо цей файл не існує, він буде створений, і вміст mytext.txt буде записано в новий файл.

Це працює також для декількох файлів. Наступна команда послідовно запише вміст mytext.txt і mytext2.txt в кінець файлу another-text-file.txt:

cat mytext.txt mytext2.txt >> another-text-file.txt

Додавання стандартного введення в висновок cat

Якщо в якості імені файлу задати дефіс ( «-«), cat здійснює зчитування з стандартного потоку введення. Наприклад, якщо у вас є файл list.txt з наступним текстом:

яблука

апельсини

олія

хліб

ви можете скористатися командою echo для виведення тексту, направити цей текст команді cat і вказати їй об'єднати його з вмістом файла:

echo "Мій список покупок" | cat - list.txt

На виході отримаємо наступний текст:

Мій список покупок

яблука

апельсини

олія

хліб

Опції

Дані опції доступні для GNU cat, стандартної для більшості дистрибутивів Linux. Для уточнення опцій, доступних конкретно у вашій системі, зверніться до керівництва.

-n, -number - виводити номер кожного рядка результату.

-b, -number-nonblank - виводити номер кожного рядка результату, крім порожніх. Ця опція скасовує -n.

-E, -show-ends - відображати «$» в кінці кожного рядка.

-s, -squeeze-blank - виключати з виведення повторювані порожні рядки.

-T, -show-tabs - показувати символи табуляції як ^ I.

-v, -show-nonprinting - показувати недруковані символи, крім табуляції і кінця рядка.

-t - показувати недруковані символи, крім кінця рядки.

-e - показувати недруковані символи, крім табуляції.

-A, -show-all - показувати всі недруковані символи.

-help - вивести довідку і вийти.

-version - вивести інформацію про версію і вийти.

2.

### **Вставить текст в файл через утилиту cat в Unix/Linux**

Утилита cat, позволяет записывать данные (текс) в файл. Приведу несколько примеров.

# cat > /etc/yum.repos.d/docker.repo <<EOF

TEXT HERE

EOF

Где:

* TEXT HERE — произвольный текст, который нужно записать в файл.
* EOF — это команда для завершения записи в файл.

$ cat >> filename.txt

TEXT HERE

^D

Где:

* TEXT HERE — произвольный текст, который нужно записать в файл.
* ^D — Сочетание клавиш «CTRL-D» для завершения записи в файл.

Можно использовать:

$ sudo sh -c 'cat << EOF >> file\_name

TEXT

HERE

EOF'

Или можно использовать что-то типа:

$ cat > tmp.txt <<TXT

> fdsfsdf

> dsf

> f

> s

> ff

> sf

> f

> s

> TXT

Т.е вхождение у меня — TXT. Начало и конец записи регулируется именно этими символами.

3

Копіювання файлів і каталогів

Для копіювання файлів використовуйте команду cp ( `` copy ''). Вона приймає два аргументи: вихідний файл, який існує і повинен бути скопійований, і цільової файл, який визначає ім'я файлу-копії. cp створює ідентичну копію файлу, привласнюючи їй вказане ім'я цільового файлу. Якщо файл з таким ім'ям вже існує, cp перезаписує його. Вона не змінює вихідний файл.

Щоб скопіювати файл `my-copy 'в файл` neighbor-copy', наберіть:

$ Cp my-copy neighbor-copy [Enter]

Ця команда створює новий файл з ім'ям `neighbor-copy ', який є ідентичним файлу` my-copy' у всіх відносинах, крім імені файлу, імені власника, групи і часу модифікації - новий файл має час модифікації, яке вказує момент його копіювання. Файл `my-copy 'не змінюється.

Щоб зберегти всі атрибути оригінального файлу, включаючи його час модифікації, ім'я власника, групу і права доступу, використовуйте ключ `-p '(` `preserve' ') (зберегти).

Щоб скопіювати файл `my-copy 'в файл` neighbor-copy', зберігши всі атрибути вихідного файлу в копії, наберіть:

$ Cp -p my-copy neighbor-copy [Enter]

Ця команда копіює файл `my-copy 'в новий файл` neighbor-copy', повністю ідентичний вихідному з усіх аспектів, крім імені файлу.

Щоб скопіювати каталог разом з усіма підкаталогами і файлами, які він містить, використовуйте опцію -R - вона створює рекурсивную копію зазначеного каталогу і його вмісту.

Щоб скопіювати каталог `public\_html 'разом з усіма його файлами і підкаталогами, в новий каталог` private\_html', введіть:

$ Cp -R public\_html private\_html [Enter]

Опція `-R 'не копіює файли, які є символічним посиланням, і не зберігає оригінальні права доступу до файлів. Щоб скопіювати каталог рекурсивно, включаючи всі посилання і враховуючи всі права доступу, використовуйте опцію `-a '(` `archive' '). Це корисно для резервного копіювання великих за обсягом дерев каталогів.

Щоб зробити резервну копію дерева каталогів `public\_html 'в каталог` private\_html', введіть:

$ Cp -a public\_html private\_html [Enter]