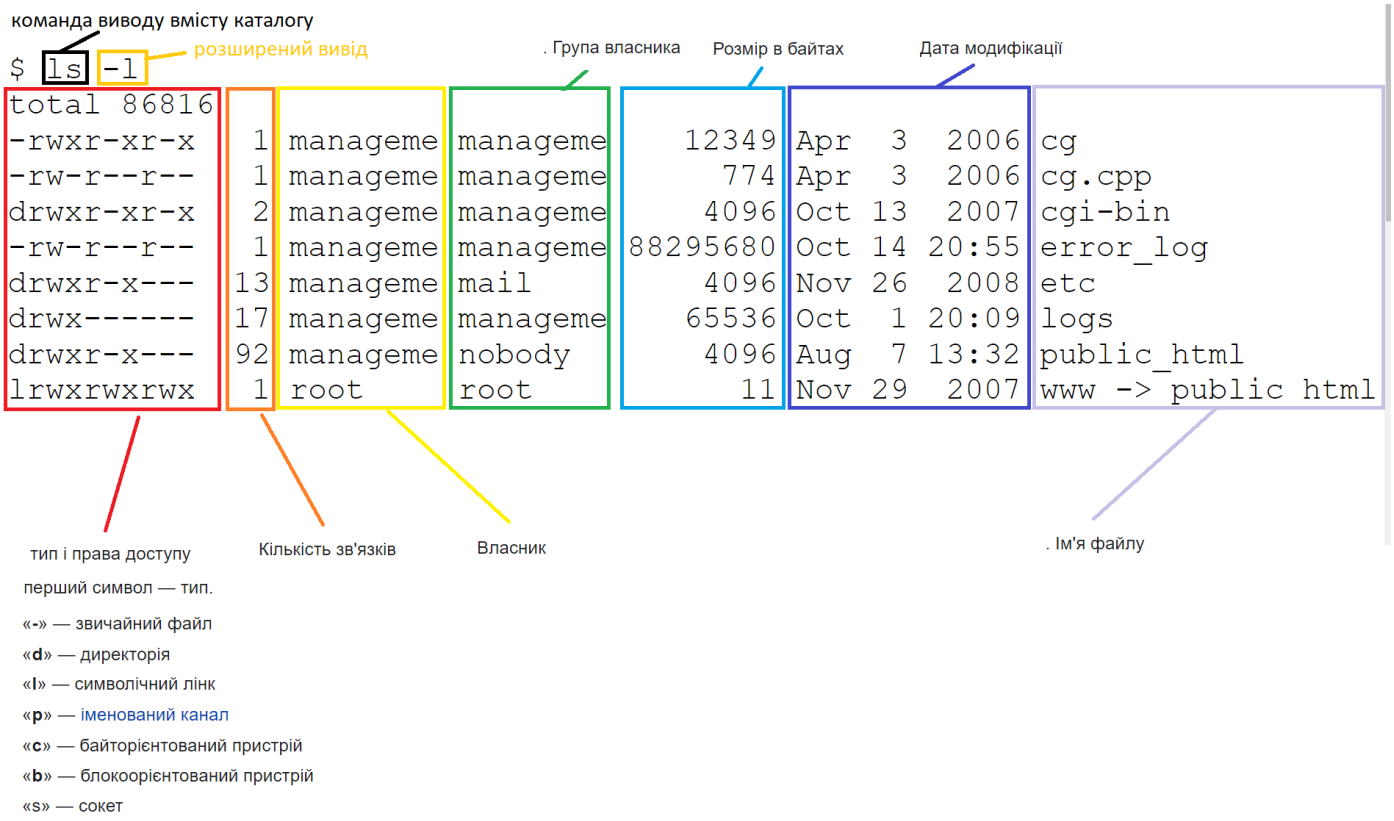
Права доступу до файлів у файловій системі Unix-подібних о/с. Атрибути файлів. Власник файлу. Зміна атрибутів.

ПІБ Бондар Денис Олегович Група ТК-31

1) Підписати елементи наступного знімку екрану:



2) Що означають літери «r, w, x» у першому стовпчику лістингу?

r - read, читання з файлу
w - write, запис до файлу
x - execute(run a program), виконання (програма, скрипт)
s - suid / sgid, в залежності від категорії, для якої ви його встановлюєте
S – якщо немає права для Виконання(execute) , але ми хочемо додати біти зміни ідентифікатора групи чи власника , то з'являється аппер кейс S

ls виводить букву «s» замість букви «x» в трійці «для власника». Точно так само, якщо відповідного x-атрибуту немає (що буває рідко), ls виведе «S» .)

Які особливості атрибуту «x» для каталогів?

Для каталогів, прапори r w x мають дещо відмінний зміст: r - дозволяє читати тільки імена файлів в каталозі; x - дозволяє робити каталог поточним і працює в парі з r; w має сенс тільки в поєднанні з x, і дозволяє маніпулювати з файлами в каталозі (створювати, видаляти і перейменовувати)

3) Які права доступу має кожний з наведених елементів каталогу?

Cg	читання з файлу запис до файлу виконання (програма, скрипт)	читання з файлу виконання (програма, скрипт)	читання з файлу виконання (програма, скрипт)
cg.cpp	читання з файлу запис до файлу	читання з файлу	читання з файлу
cgi-bin	читання з каталогу (випис вмісту) запис до каталогу (створення, видалення, перейменування файлів та підкаталогів) відкриття каталогу(робити поточним)	читання з каталогу (випис вмісту) відкриття каталогу(робити поточним)	читання з каталогу (випис вмісту) відкриття каталогу(робити поточним)
error_log	читання з файлу запис до файлу	читання з файлу	читання з файлу
etc	читання з каталогу (випис вмісту) запис до каталогу (створення, видалення, перейменування файлів та підкаталогів) відкриття каталогу(робити поточним)	читання з каталогу (випис вмісту) відкриття каталогу(робити поточним)	
logs	читання з каталогу (випис вмісту) запис до каталогу (створення, видалення, перейменування файлів та підкаталогів) відкриття каталогу(робити поточним)		
public_html	читання з каталогу (випис вмісту) запис до каталогу (створення, видалення, перейменування файлів та підкаталогів) відкриття каталогу(робити поточним)	читання з каталогу (випис вмісту) відкриття каталогу	
www	Читання посилання Зміна посилання відкриття посилання	Читання посилання Зміна посилання відкриття посилання	Читання посилання Зміна посилання відкриття посилання

4) Що означало б, якби перша літера у правах доступу dr--r--r-- була:

– звичайний файл

- Коли програма читає або записує дані з файлу, запити надходять у драйвер ядра. Якщо файл є звичайним, дані обробляються драйвером файлової системи, і вони, як правило, зберігаються в зонах на диску або на іншому носії інформації, а дані, які зчитуються з файлу, - це те, що раніше було записано в цьому місці.

d каталог, директорія

b block device - файл є спеціальним блоковим пристроєм

Коли дані зчитуються або записуються у файл пристрою, запит обробляється драйвером для цього пристрою. Кожен файл пристрою має відповідний номер, який ідентифікує драйвер для використання. Те, що пристрій робить з даними, - це його власна справа.

- ✓ Блокові пристрої зазвичай поводяться приблизно так само, як звичайні файли: вони являють собою масив байтів, і значення, яке зчитується в певному місці, є значенням, яке було останнє там записане. Дані з блочного пристрою можна кешувати в пам'яті та читати з кешу; записи можна буферизувати. Назва «блоковий пристрій» походить від того, що відповідне обладнання зазвичай читає і записує цілий блок за раз (наприклад, сектор на жорсткому диску).

c character device - файл є спеціальним символьним пристроєм

- ✓ Символьні пристрої поводяться як канали, послідовні порти тощо. Запис або читання на них є негайною дією. Запис байта на символьний пристрій може призвести до його відображення на екрані, виведення на послідовний порт, перетворення в звук. Читання байта з пристрою може призвести до того, що послідовний порт буде чекати введення, може повернути випадковий байт. Назва «пристрій символів» походить від того, що кожен символ обробляється індивідуально.

l лінк:

Символічне посилання(softlink) <u>(приклад в наступному питанні)</u>

- ✓ Вказує на цільовий файл або каталог.
- ✓ Фактично є невеликим файлом, що містить шлях до цільового файлу.
- ✓ Не містить в собі вмісту самого файлу.
- ✓ Містить шлях до цільового файлу.
- ✓ Має власні права доступу, які не поширюються на цільовий файл.
- ✓ Видалення / перейменування / переміщення цільового файлу не виконує автоматичне оновлення посилання. Посилання починає вказувати на неіснуючий файл, стає непрацюючим.
- ✓ Зміна прав доступу у цільового файлу не оновлює права доступу у посилання.
- ✓ Може бути створена для директорії.

- ✓ Посилання та цільовий файл мають різні файлові індекси (inode) в файловій системі.
- ✓ Може вказувати на неіснуючий файл.

Жорстке посилання(hardlink):

- ✓ Є свого роду ще одним ім'ям на файл.
- ✓ Не може вказувати на директорію.
- ✓ Не можна створювати жорсткі посилання між файлами різних файлових систем.
- ✓ Не може вказувати на неіснуючий файл.
- ✓ Жорстке посилання і файл, для якого воно створювалось, мають однакові індекси (inode) в файлової системі.

5) Якщо в назві файлу зустрічається "->" – це означає …

Посилання на файл , softlink
Символічні посилання використовуються для більш зручної організації структури файлів на комп'ютері, так як дозволяють для одного файлу або каталогу мати кілька імен і різних атрибутів; вільні від деяких обмежень, властивих жорстким посиланням.

6) Що таке setuid, setgid? На що вони впливають? Як їх змінити?

setuid- флажок , який робить так щоб файл запускався від імені власника
setgid- флажок , який робить так щоб файл запускався з правами групи

Є досить багато програм і файлів, які повинні належати користувачеві root, і в той же час прості користувачі повинні мати можливість виконувати його. Для прикладу - утиліта **passwd**, яка знаходиться в каталозі **/ usr / bin / passwd** і яка має справу з файлом **/ etc / passwd**, редагувати який може тільки користувач root:
Для того, щоб дати користувачам можливість змінювати свої паролі - на файл утиліти встановлений біт SUID:
SUID дає можливість на час виконання файлу (запущеного їм процесу) непривелігірованому користувачеві отримати права користувача - власника файлу, в даному випадку - root.
Біт SGID аналогічний SUID, але встановлюються права не користувача файлу, а групи - власника файлу. Так само, всі файли, створювані в каталозі з установленим SGID отримуватимуть ідентифікатор групи - власника каталогу, а не власника файлу.

Щоб встановити біт SUID виконайте:
Chmod u + s filename

Щоб встановити SGID - виконайте:

Chmod g + s filename

Використання SUID - потенційно небезпечне. Тому треба обов'язково встановлювати на розділи, в яких доступний запис всім користувачам, прапор **nosuid**.

Так само, на деяких серверах корисно встановити такий прапор на розділ, що містить домашні каталоги користувачів.

7) Що таке SUID, SGID, sticky bit? На що вони впливають та як змінюють поведінку файлів, для яких вони встановлені?

SUID- (Set User ID - біт зміни ідентифікатора користувача) програма працює з правами власника виконуваного файлу

SGID- (Set Group ID - біт зміни ідентифікатора групи), програма виконується з правами групи, що володіє файлом
Коли користувач або процес запускає виконуваний файл з встановленим одним з цих бітів, файлу тимчасово призначаються права його (файлу) власника або групи (в залежності від того, який біт заданий).

sticky bit –корисний для захисту файлів від випадкового видалення в середовищі, де кілька користувачів мають права на запис в один і той же каталог. Якщо застосовується закріплений sticky bit, користувач може видалити файл, тільки якщо він є користувачем-власником файлу або каталогу, в якому міститься файл. З цієї причини він застосовується в якості дозволу за замовчуванням для каталогу / tmp.

вказує ,що після завершення треба щоб програма повисіла в пам’яті , в сподіванні на те що її знову запустять , прискорює процес холодного старту(все що потрібно для запуску програми лежить в оперативній пам’яті)

8) Які задачі вирішують та які параметри наступних команд?

Chown ,chgroup
Створивши файл, ви автоматично стаєте його власником, але можете передати право володіння іншому користувачеві, у якого є запис у файлі / etc / passwd. Тільки системний адміністратор або фактичний власник може передавати права на файл іншому користувачеві. Для передачі прав власника призначена команда chown. Команда chgrp задає групу, якій належить файл. Загальний формат цих команд такий:
chown власник файл
chgrp власник файл
Наприклад:
-rw-r--r-- 1 root root 1 Feb 20 17:35 chownSample.txt
Змінимо власника з root на нового користувача з ім'ям daria
chown daria chownSample.txt

umask
umask – маска режиму створення користувацьких файлів – функція, що змінює права доступу, які присвоюються новим файлам і директоріям за замовчуванням. Режим повного доступу для директорій – 777, для файлів – 666.
Як працює: якщо встановити umask 027 , то для файлів права стануть такими: 666-27=640(rx-r-----)
А для директорій :777-27=750(rwxr-x---)

chmod
права можуть змінюватися або власником файлу, або суперкористувачем за допомогою цієї команди

Chmod опції категоріядіяпрапор файл

З найкорисніших опцій :

-R - рекурсивне призначення прав. Тобто призначити права всім об'єктам, керуючись регулярним виразом.

Наприклад: chmod -R 700 z * - Призначити повні права для власника і виключити права для групи і всіх інших для всіх об'єктів, які починаються іменуватися на z, що знаходяться в поточному каталозі і його підкаталогах.

1) Що означають оператори в командному рядку:

; - оператор послідовного виконання: вираз1; вираз2 (запуск вираз1, завершення виконання вираз1, далі запуск вираз2, виконання вираз2)

& -- оператор виконання програми у фоновому режимі : вираз1 & вираз2
(запуск вираз1, одразу після цього запуск вираз2 та повернення управління користувачу)

&& - Спочатку виконується вираз, що стоїть ліворуч, якщо він істинний, виконується вираз, що стоїть праворуч. Якщо вираз1 повертає БРЕХНЯ, то вираз2 не буде виконано. Якщо обидва вирази повертають істину, виконується наступний набір команд. Якщо будь-який з виразів неправдивий, наведений вираз вважається помилковим в цілому.
| | - Логічне АБО перевіряє помилковість вираження зліва, якщо це так - воно починає виконувати наступний оператор

2) Які номери, назви та яке призначення мають стандартні потоки вводу та виводу?

0 – стандартний потік вводу (stdin). З цього файлу процеси витягують свої вхідні дані. За замовчуванням вхідний потік асоційований з клавіатурою, але частіше за все він надходить по каналу від інших процесів або зі звичайного файлу

1 – стандартний потік виводу (stdout). У цей файл записуються всі вихідні дані процесу. За замовчуванням дані виводяться на екран терміналу, але їх можна також перенаправити в файл або надіслати по каналу іншому процесу

2 – стандартний потік помилок (stderr). У цей файл записуються повідомлення про помилки, що виникають в ході виконання команди. За замовчуванням повідомлення про помилки виводяться на екран термінал, але їх також можна перенаправити в файл.

3) Які функції виконують наступні оператори перенаправлення вводу/виводу?

> означає, що вивід команди, що знаходиться зліва від цього символу, буде записаний у файл, що знаходиться праворуч від символу, при цьому файл буде перезаписано.

< отримує дані з вказаного файлу в якості стандартного вхідного потоку

>> Якщо замість> вказано >>, то вихідний файл не буде перезаписано, а вивід команди додасться в кінець файлу

| - Існує можливість перенаправити вихідний потік однієї утиліти безпосередньо у вхідний потік іншої, без використання тимчасових файлів. Це так звані конвеєри (pipe). В системі UNIX всі утиліти, що поєднані в конвеєр, запускаються паралельно і обробляють інформацію по мірі її надходження. Конвеєр утворюється за допомогою символу '|'

4) Як перенаправити потік помилок програми prog у стандартний потік виводу при запуску? Написати командний рядок.

prog 2>&1

5) Як створити архів каталогу /home/user1/ за допомогою програм tar та gzip? Написати командний рядок.

Команда tar об'єднує всі файли в один архівний файл . tar не стискає сам файл. Якщо об'єднати tar с gzip, команда tar створить один єдиний архівний файл з папки, а gzip стисне цей файл. Можна об'єднати ці дії використовуючи опцію z.

tar -cvf archive.tar /home/user1/
де
z - стиснути архів в gzip (без цього параметра, tar не стискується, а створює так званий тарбол);
c - ключ на створення архіву;
v - verbose режим, тобто з виводом на екран процесу;
f - використовувати файл (обов'язково вказуємо, так як в більшій мірі працюємо саме з файлами).

6) Як створити tar-gzip архів каталогу /home/user1/ за допомогою програми tar? Написати командний рядок.

tar -czf archive.tar.gz /home/user1/
(tar -cv /home/user1/ | gzip > archive.tar.gz)

7) Що роблять наступні скрипти shell? Який їх результат роботи?

1
 	
`if [ -e somefile.txt ] ; then`
`cat somefile.txt`
`else`
`echo "no such file"`
`fi`
	Якщо існує somefile.txt
	вивести вміст файлу
	інакше
	показати "no such file"

2
 	
`for a in 1 2 3 4 5; do`
`touch file$a`
`done`
	Цикл буде створювати порожні файли з іменами file1
	file2,… file5

3
 	
`for a in $( seq 1 10 ) ; do`
`cat file_$a`
`done`
	Цикл від 1 до 10 буде виводити вміст file_1,file_2…..file_10

4
 	
`while [ -d mydirectory ] ; do`
`ls -l mydirectory >> logfile`
`echo -- SEPARATOR -- >> logfile`
`sleep 60`
`done`
	Поки існує каталог mydirectory, виводити довгу інформацію про вміст mydirectory з перенаправленням стандартного потоку виводу в файл logfile
	Та виводити повідомлення -- SEPARATOR – в файл logfile
	Команда sleep припиняє подальше виконання команд на 60 секунд
	Виконується нескінченно