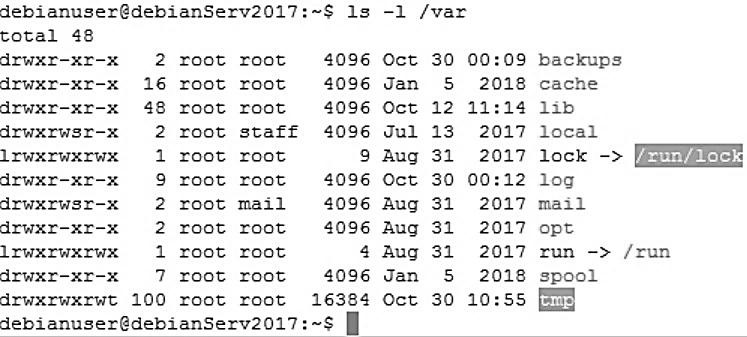
**Лекція № 14**

**4.3. Права доступу до файлів і каталогів**

У цьому розділі розглянемо, які права доступу до файлів і каталогів визначаються у системах, що є нащадками UNIX, як їх можна переглянути та змінити. Існують 3 різні права доступу, які можуть бути надані певній категорії користувачів системи (нагадаємо, що такими категоріями є власник файлу або каталогу; група, до якої належить власник; усі інші користувачі). Це такі права як:

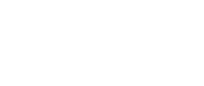
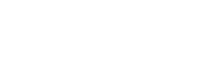
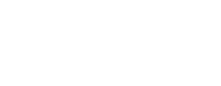
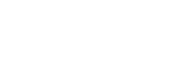
1) на читання; 2) на запис; 3) на виконання.

Для того, щоб переглянути поточні права на певний елемент файлової системи, необхідно скористатися вже знайомою нам командою ***ls*** з ключем ***-l***, яка виводить найбільш детальну інформацію щодо вмісту певного каталогу. До неї входять повідомлення про права доступу, що представляється у першому стовпчику виведення команди. Наприклад:



Інформація щодо прав доступу представляє 10 символів. Схематично вона показана на рис. 4.1.

Тип Привілеї для Привілеї Привілеї елемента власника для групи для інших



|  |  |  |  |
| --- | --- | --- | --- |
| - | rwx | rwx | rwx |

**Рис. 4.1.** Інформація про права доступу файлу

Перший символ не відноситься до прав доступу, він показує тип елемента. Значення першого символу та їх опис представлені у табл. 4.3.

Табл. 4.3. Значення першого символу та їх опис

|  |  |
| --- | --- |
| **Значення** | **Опис** |
| ***-*** | звичайний файл |
| ***d*** | каталог |
| ***l*** | символічне посилання |
| ***c*** | файл символьного пристрою (термінал, модем) |
| ***b*** | файл блочного пристрою (жорсткий диск, cd-rom і т. п.). |

Усі інші 9 символів представляють інформацію про права доступу. Їх можна розподілити на 3 групи по 3 символи, кожна з яких представляє інформацію щодо права доступу для різних категорій:

символи **2–4** – права для власника;



символи **5–7** – права для користувачів, що входять до основної

групи власника; символи **8–10** – права для всіх інших користувачів.

Права доступу представляються відповідними символами та завжди йдуть в одному й тому самому порядку:

***r*** – право на читання; ***w*** – право на запис; ***x*** – право на виконання.



За наявності відповідного права зображена відповідна літера, якщо ні – то символ ‘***-***’.

Але одне й те саме право у контексті файлів та каталогів може означати абсолютно різні привілеї, які детально описані у табл. 4.4.

Табл. 4.4. Привілеї, які дають права доступу відносно файлів та каталогів

|  |  |  |
| --- | --- | --- |
| **Право** | **Привілеї для файлів** | **Привілеї для каталогів** |
| ***r*** | Відкриття та читання вмісту файлу | Читання вмісту каталогу |
| ***w*** | Запис даних у файл або видалення інформації з нього (право не дає привілеїв перейменування та видалення файлів, потрібне відповідне право для каталогу, в якому файл розташовується) | Створення, видалення й перейменовування файлів, що розташовуються у каталозі |
| ***x*** | Інтерпретація файлу як програми і її виконання (файли, які є програмами на мовах сценаріїв, мають бути доступними для читання, інакше вони не будуть виконуватися) | Дозвіл переходу в каталог  (з використанням команди ***cd***). |

Зазначимо, що для символічних посилань завжди будуть відображуватись максимальні права (***rwxrwxrwx***), однак це не відображує дійсні права доступу. У випадку символічних посилань фактичні права доступу до файлу визначаються правами доступу самого файлу, на який вказує це символічне посилання.

Наведемо декілька прикладів встановлених прав доступу для різних елементів файлової системи у табл. 4.5.

Табл. 4.5. Приклади встановлених прав доступу до файлів

|  |  |
| --- | --- |
| **Права доступу** | **Опис** |
| ***-rwx------*** | Звичайний файл, на який повні права має власник. Інший користувач при цьому не має ніяких прав доступу до нього |
| ***-rw-------*** | Звичайний файл, який доступний власнику для читання і запису. Ніякий інший користувач при цьому не має прав доступу до нього |
| ***-rw-r--r--*** | Звичайний файл, який доступний власнику для читання і запису. Право на читання файлу мають члени основної групи власника та всі інші користувачі, але інших прав доступу на цей файл у них немає |
| ***-rwxr-xr-x*** | Звичайний файл, право на читання і виконання якого має власник, члени основної групи власника та всі інші користувачі.  Власнику файлу він доступний також для запису |
| ***-rw-rw-----*** | Звичайний файл, на який мають право читання та запису власник файлу та члени його основної групи. Усі інші не мають ніяких прав доступу до нього |
| ***lrwxrwxrwx*** | Символічне посилання. Це означає, що всі права доступу є недійсними, а фактичні права на нього визначаються правами  доступу файлу, на який вказує символічне посилання |
| ***drwxrwx---*** | Каталог, у який можуть входити, а також створювати, перейменовувати і видаляти файли власник та члени його основної групи. Усі інші не мають ніяких прав доступу до нього |
| ***drwxr-x---*** | Каталог, у який можуть входити власник та члени його основної групи. Власник може створювати, перейменовувати і видаляти файли всередині каталогу. Всі інші не мають ніяких прав доступу до нього. |

**4.3.1. Зміна режиму доступу до файлу**

Для зміни прав доступу до файлу або каталогу призначена команда ***chmod***. Запам’ятайте, що права доступу до файлу або каталогу може змінити тільки його власник. Ця команда має наступний синтаксис: ***chmod <права\_доступу> <елементи…>***

Під час виконання команди потрібно обов’язково задати хоча б один елемент, ними може бути файл або каталог, однак можна змінити права доступу і більш ніж одному елементу одразу. Права доступу можна задати

2 способами: за допомогою вісімкового та символічного представлення.

**Вісімкове представлення**

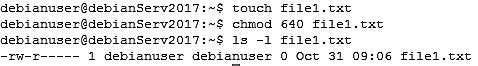
Використання вісімкової форми запису права доступу визначаються вісімковими числами. Оскільки кожна цифра у вісімковому числі визначається трьома двійковими розрядами, вона відображається в схемі зберігання режиму доступу до файлу (табл. 4.6).

Табл. 4.6. Режими доступу до файлу в двійковому та вісімковому представленні

|  |  |  |  |
| --- | --- | --- | --- |
| **Вісімкове** | **Двійкове** | **Режим доступу** | **Опис** |
| ***0*** | ***000*** | ***---*** | відсутність будь-яких прав |
| ***1*** | ***001*** | ***--x*** | тільки виконання |
| ***2*** | ***010*** | ***-w-*** | тільки запис |
| ***3*** | ***011*** | ***-wx*** | запис і виконання |
| ***4*** | ***100*** | ***r--*** | тільки читання |
| ***5*** | ***101*** | ***r-x*** | читання і виконання |
| ***6*** | ***110*** | ***rw-*** | читання і запис |
| ***7*** | ***111*** | ***rwx*** | читання, запис і виконання. |

Права доступу за допомогою вісімкового представлення визначаються трьома цифрами від ***0*** до ***7***, перша з яких представляє права для власника, друга – для членів його основної групи, а третя – для всіх інших користувачів.

У наступному прикладі створимо новий файл з назвою ***file1.txt*** та встановимо на нього наступні права доступу: власник може читати та писати, члени основної групи – тільки читати, а всі інші не мають на нього ніяких прав. Ці права у вісімковому представленні будуть закодовані числом ***640***.



Наведемо інший приклад. Створимо новий каталог ***dir1*** та встановимо для власника повні права доступу, членам основної групи власника права, що дозволяють входити до каталогу та розглядати його вміст, усім іншим тільки розглядати вміст каталогу без можливості входу до нього (як це не дивно, але можна продивитися вміст каталогу за допомогою команди ***ls***, вказавши його назву в якості аргументу, знаходячись при цьому абсолютно в іншому каталозі). Описані права доступу можна представити за допомогою числа ***754***.



Таким чином, з двох наведених вище прикладів ми побачили, що права доступу після їх встановлення командою ***chmod*** дійсно відповідають їх вісімковому кодуванню.

**Символічне представлення**

Іншим способом задати права командою ***chmod*** є символічне представлення, яке передбачає символічну форму запису, яка має наступний вигляд:

***<категорія\_користувачів><операція><права>***, де: ***<категорія\_користувачів>=<u, g, o, a>***, ***<операція>= <+, -, =>***, ***<права>=<r, w, x>***.

Стосовно категорії користувачів, для яких встановлюються права доступу, використовуються наступні символи:

***u*** – власник файлу або каталогу (**user**);

***g*** – члени основної групи власника файлу або каталогу (**group**); ***o*** – усі інші користувачі (**other**);



***a*** – усі користувачі системи: комбінація всіх трьох попередніх

символів (**all**).

Для зазначення операцій використовуються наступні символи:

***+*** – додати задані права;



***–*** – відняти задані права;

***=*** – встановити задані права (різниця між знаком ***+*** полягає у тому, що задані права встановлюються, а всі інші віднімаються).

На останок вказуються права доступу звичними літерами: ***r***, ***w*** та ***x***.

Наведемо деякі приклади:

***u+x*** – додати власнику право на виконання; ***g-w*** – відібрати у групи право на читання.



При зазначенні всіх користувачів, літеру ***a*** також можна опустити:

***+x*** – додати право на виконання всім користувачам у системі (власнику, членам основної групи та іншим користувачам).

Різні категорії та права доступу можна об’єднувати, вказавши їх разом. Наприклад:

 ***go=rw*** – встановити для основної групи та інших користувачів право на читання і запис. Але якщо вони мали право на виконання, то воно відбирається; ***o-rw*** – відібрати право на читання і запис в усіх інших користувачів, крім власника і групи.

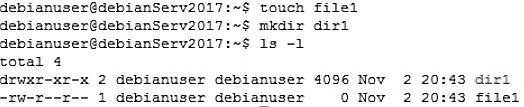
Нарешті, можна виконати одразу декілька операцій, перерахувавши їх через кому. Наприклад:

 ***u+x,go=rx*** – додати право на виконання для власника і водночас встановити права на читання і виконання для членів його основної групи та інших користувачів.

Порівняно з вісімковим представленням перевага символічного представлення прав доступу полягає у дозволі встановлення певного права, ніяк не впливаючи на інші.

**4.3.2. Права доступу до файлів і каталогів за замовчуванням**

Під час створення нових файлів та каталогів у своєму домашньому каталозі їм надаються певні права доступу. Наприклад, створимо новий файл та новий каталог у домашньому каталозі користувача **debianuser** та подивимось які права доступу були надані для них.



Таким чином, права доступу на новий файл та каталог є наступними:

 для файлу: власник може читати та писати, члени основної групи

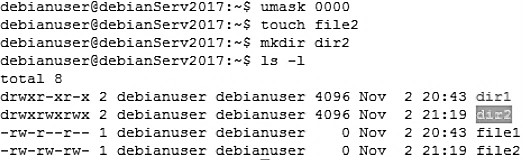
власника – тільки читати, інші користувачі – так само тільки читати;  для каталогу: власник може входити, читати його вміст, а також створювати, видаляти та перейменовувати файли; члени основної його групи – входити та читати вміст каталогу, однак не можуть створювати, видаляти та перейменовувати файли; всі інші користувачі – так само тільки входити та читати його вміст.

Може виникнути слушне запитання, чому для новоствореного файлу та каталогу були надані саме такі права доступу? Відповідь полягає у тому, що кожному новоствореному файлу та каталогу надаються певні права за замовчуванням. А визначити та змінити їх дає команда ***umask***.

Спочатку визначимо поточні права, що надаються за замовчуванням. Це можна здійснити за допомогою виконання команди ***umask*** без аргументу.



З огляду на це команда ***umask*** видала значення ***0022***. Що ж означають ці цифри? Для того, щоб було зрозуміло, проводимо експеримент. Виконаємо команду ***umask***, задавши їй як аргумент значення ***0000***, а потім створимо ще один новий файл та каталог, після чого переглянемо права доступу, які були йому надані.



Отже, нові файл та каталог після встановлення нової маски мають наступні права доступу: для каталогу всі користувачі мають повні права доступу, а щодо файлу, то користувачі можуть читати та писати, ніхто не має можливості його виконувати.

Встановивши маску ***0000***, ми тим самим фактично вимкнули її. Але для створених після її вимкнення файлу та каталогу, були надані максимальні права доступу, які можна надати за замовчуванням для файлу та каталогу відповідно. Таким чином, для каталогу максимальними правами доступу за замовчуванням є сукупність прав ***rwx*** для всіх користувачів, що відповідає вісімковому числу ***777***, а для файлу відповідна сукупність прав ***rw-***, що відповідає вісімковому числу ***666***. Таким чином, для файлу за замовчуванням не буде надано право на виконання. Зроблено це з точки зору безпеки, оскільки не кожен файл є виконуваною програмою. Написавши файл, що представляє програму, користувач завжди має власноруч надати право на його виконання.

Тепер нарешті розберемося, які права за замовчуванням надає маска. Для цього ми знову маємо повернутися до вісімкового представлення прав доступу. Нагадаємо, що початкова маска мала значення ***0022***. За цією маскою були надані права доступу за замовчуванням для файлу ***file1*** та каталогу ***dir1***. Представимо таблицю 4.7, у якій зазначимо максимальні права за замовчуванням, маску та результат (поточні права доступу). Зокрема поки не будемо розглядати першу цифру (***0***), далі буде описано, що вона означає.

Табл. 4.7. Вплив маски на призначення прав за замовчуванням

для файлу та каталогу

|  |  |
| --- | --- |
| **Для каталогу** |  |
| максимальні права за замовчуванням | rwx rwx rwx |
| маска | 000 010 010 |
| поточні права | rwx r-x r-x |
| **Для файлу** |  |
| максимальні права за замовчуванням | rw- rw- rw- |
| маска | 000 010 010 |
| поточні права. | rw- r-- r--. |

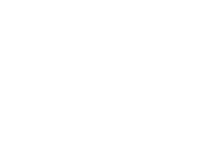
З таблиці досить легко побачити, що біт, який відповідає тій позиції, де в масці стоїть 1, був скинутий, – в цьому випадку право на запис членів основної групи власника та всіх інших користувачів. Тепер зрозуміло, що робить маска: за її допомогою ми можемо задати права, які будуть заборонені за замовчуванням для новостворених файлів та каталогів.

Задаючи нову маску, ми вказуємо інверсне значення, тобто ті права, які хочемо заборонити певній категорії користувачів. Тут навпаки цифра ***0***, буде означати повний доступ, а цифра ***7*** – відсутність усіх прав за замовчуванням.

**4.3.3. Розширені права доступу**

Зазвичай права доступу у вісімковому представленні позначаються трьома цифрами, кожна з яких означає права для певної категорії користувачів, однак правильно їх представляти чотирма цифрами. Четверта вісімкова цифра найстаршого розряду позначає спеціальні права доступу.

Існує 3 спеціальних права доступу, кожне з яких представляється відповідним бітом. На рис. 4.2. представлені біти четвертого вісімкового розряду.



---



біт

SetUID



біт

SetGID



біт

sticky

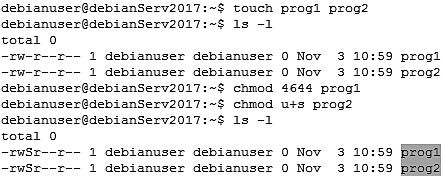
**Рис. 4.2.** Біти, що визначають спеціальні права доступу

Сукупність цих трьох бітів, як і у випадку прав доступу на читання, запис та виконання, можна представити однією вісімковою цифрою від ***0*** до ***7***. Однак у деяких випадках встановлений тільки один з трьох бітів, оскільки їх сукупність практично не дає ніяких ефектів. Так, встановлений біт **SetUID** дає цифру ***4*** у четвертому вісімковому розряді при кодуванні прав доступу, **SetGID** – цифру ***2***, а **sticky** – цифру ***1***.

Якщо встановити найстарший біт **SetUID** для виконуваного файлу (програми), то у якості ефективного ідентифікатора користувача для процесу встановлюється не його ідентифікатор, який фактично запустив програму, а ідентифікатор власника цієї програми. Найчастіше він встановлюється для програм, власником яких є привілейований користувач. Тобто коли звичайний користувач запускає виконуваний файл, для якого встановлений біт **SetUID**, ця програма виконується з привілеями суперкористувача. Однак через проблеми безпеки кількість таких програм у системі має бути зведена до мінімуму.

Для встановлення біту **SetUID** можна скористатися вісімковим або символічним представленням прав доступу в команді ***chmod***: у вісімковому представленні спеціальне право встановлюється за допомогою значення 4 у четвертому розряді, а у символічному за допомогою спеціального запису ***u+s***.

|  |
| --- |
|  |

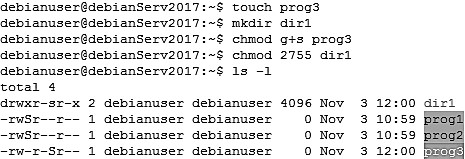
У наступному прикладі створимо два файли ***prog1*** та ***prog2*** та встановимо для них біт **SetUID** за допомогою віміскового представлення у першому випадку і символічного представлення у другому. Потім проаналізуємо як представляються права доступу з встановленим бітом.

При виведені інформації про файли, на які встановлений біт **SetUID**, у правах доступу як символ, призначеного для позначення дозволу виконання власником файлу, відображається літера ***S***.

Але другий, середній біт, що має назву **SetGID**, дозволяє підмінити основну групу власника. Його можна встановити як для файлу так і для каталогу. У випадку встановлення для файлу цей біт аналогічно до біту **SetUID** для власника файлу встановлює ефективний ідентифікатор основної групи для процесу, вибираючи замість ідентифікатора основної групи реального користувача ідентифікатор основної групи власника файлу. Під час встановлення цього біту для каталогу, новостворювані файли у ньому будуть належати основній групі власника каталогу, а не групі власника файлу, який його створив.

Для встановлення цього біту за допомогою вісімкового представлення у найстаршому розряді встановлюється значення 2, тоді як у символичному представленні необхідно скористатися спеціальним записом ***g+s***.

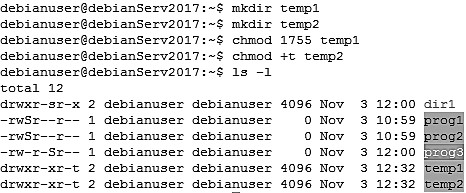
У наступному прикладі створимо новий файл та новий каталог, і встановимо для них біт **SetGID** двома різними способами, потім подивимося на права доступу до них.



У прикладі можна побачити, встановлення біту в інформацію щодо прав доступу на місці символу, який відповідає за право виконання для основної групи, встановлена літера **s**, але для каталогу вона маленька, а для файлу – велика.

Наостанок розглянемо молодший біт, що має назву **sticky**. У перших версіях UNIX він дозволяв позначити виконуваний файл таким, який неможливо видалити. Наразі для файлів система Linux ігнорує цей біт, однак встановлення його для каталогу має певний сенс. Якщо встановити біт **sticky** для каталогу, то він не дозволить видалення та перейменовування файлів та каталогів у представленому каталозі користувачу, який не є їх власником або суперкористувачем. Часто біт **sticky** встановлюється для управління доступом до загальних каталогів, наприклад, таких як ***/tmp***. У вісімковому представленні він встановлюється присвоюванням значення найстаршого розряду цифри 1, а за допомогою символічного представлення присвоюванням спеціального запису ***+t***.

У наступному прикладі створимо 2 каталоги з назвами ***temp1*** та ***temp2***, а потім встановимо для них біт **sticky** двома різними способами.



З прикладу видно, що встановлений біт **sticky** у правах доступу позначається літерою ***t*** на місці символу, який відповідає за право на виконання для всіх інших користувачів.