# Робота №1. Структура файлової системи Linux,

# основні команди, команди роботи з файлами

## Мета

**Оволодіння практичними навичками роботи в системі Linux. Знайомство із структурою файлової системи, основними командами роботи з файлами.**

## Завдання для самостійної підготовки

1. Ознайомитись з документацію, звернувши увагу на такі питання:

 команди входу в систему, зміни пароля, одержання системної підказки, виводу календаря і зміни дати;

 організацію і структуру файлової системи Linux, обмеження на імена файлів;

 типи файлів, каталоги і посилання;

 системні каталоги;

 створення, видалення, копіювання і перегляд умісту файлів.

1. Рекомендовані джерела: [2, 3]. Зокрема, [3] рекомендується скачати і роздрукувати (2 сторінки). Хоча ці джерела розроблені для Debian Linux, значні їх частини є вірними і для інших сучасних UNIX-подібних систем. Однак, для кожної конкретної системи рекомендується застосовувати документацію, розроблену конкретно для неї (за наявності)
2. Ознайомитись з такими командами UNIX-подібних систем (рекомендується застосовувати сторінки man):

man, passwd, date, cat, more, wc, who, ls, cd, cal, cp,

mv, mkdir, rm, rmdir

1. Відповідно до завдання підготувати послідовність команд для його виконання.

## Довідковий матеріал

При встановленні ОС UNIX/Linux на персональному комп’ютері, вона налаштовується на роботу з графічною оболонкою. При цьому вхід користувача в систему здійснюється вже засобами графічної оболонки. Оскільки у цьому практикумі нам буде потрібне інтенсивне застосування терміналу з інтерфейсом командного рядка, розглянемо його особливості.

У системі підтримується певна кількість так званих віртуальних терміналів, між якими можна переключатись комбінаціями клавіш Alt+F#, де F# – одна з функціональних клавіш (при цьому вихід з багатовіконного графічного інтерфейсу в один з текстових віртуальних терміналів може здійснюватись комбінацією клавіш Alt+Ctrl+F#). Користувач може працювати в системі, використовуючи одночасно кілька терміналів. Для здійснення входу в систему в текстовому терміналі, користувач повинен спочатку ввести свій ідентифікатор (login). Як правило, він вводиться у відповідь на запрошення системи такого вигляду:

**Login:**

Якщо цього запрошення не екрані немає (і не діє екранна заставка – screensaver), то це означає, що даний термінал не очікує входу користувача. Три найтиповіші причини:

1. Вхід вже здійснено. Для виходу можна ввести команду logout. Або можна ввести команду login – тоді після завершення сеансу нового користувача термінал повернеться до роботи з попереднім.
2. Термінал апаратно заблоковано клавішею “Scroll Lock” (це можна визначити за відповідним індикатором) – розблокуйте термінал.
3. Термінал зайнятий, тобто з ним пов’язана деяка програма – слід вийти з цієї програми і з командної оболонки, для чого потрібен певний досвід (можуть спрацювати клавіша q, комбінації клавіш Ctrl+C, Ctrl+D, але іноді це не допоможе – див. Роботу № 4 з цього практикуму). У загальному випадку не слід застосовувати комбінацію Ctrl+Z – при цьому дійсно з’явиться запрошення на введення команди, але програма, що виконувалась, не буде зупинена, а лише перейде у фоновий режим, забираючи ресурси вашого комп’ютера. Якщо таких програм буде багато, вони можуть суттєво погіршити працездатність вашого комп’ютера.
4. Термінал використовується винятково для виведення на екран важливих системних подій, при цьому на ньому можна вводити команди і будь-які символи, але реакції на це не буде – слід перейти на іншу консоль комбінацією клавіш Alt+F#, де F# – одна з функціональних клавіш, крім F1.

Після введення ідентифікатора система запитує в користувача пароль:

**Password:**

Під час введення пароля символи на екрані не відображаються. Якщо ідентифікатор і пароль користувача були введені правильно, система здійснює авторизацію користувача, тобто, надає йому певні повноваження, необхідні для роботи в системі. У текстовому терміналі після цього користувач зазвичай опиняється в середовищі командної оболонки (англ. – shell). При цьому на екрані з’являється так зване запрошення командної оболонки (найчастіше – символ ‘$’ або ‘>’, також можна довільно змінити запрошення). Командна оболонка приймає команди, що вводяться з клавіатури, інтерпретує їх і виконує відповідні дії. Ці дії можуть полягати у запуску певних утиліт із заданими у командному рядку параметрами. Крім того, командна оболонка надає користувачеві певний додатковий сервіс, наприклад, дозволяє виконувати редагування команди (курсор можна переміщати вправо чи вліво, додавати або знищувати символи під курсором), в деяких оболонках можна легко відтворити попередні команди (клавіші переміщення курсору вгору та вниз), а також користуватись підказками щодо імен наявних файлів (клавіша Tab). Докладніше про ці та інші сервісні можливості можна дізнатись в довідковій системі man, а також в будь-якій доступній книзі про системи UNIX чи Linux.

В кожній системі UNIX є в наявності кілька різних командних оболонок. Практично в кожній системі можна знайти звичну оболонку, або подібну до неї. Найбільш стандартною, що присутня в усіх системах, є оболонка Bourne (Bourne shell), яка стала основою стандарту POSIX shell. Ця оболонка пропонує мінімальний сервіс для користувача, і для інтерактивної роботи незручна. Її файл – /bin/sh. Існують альтернативні оболонки. Одна з них – C shell (/bin/csh), вона досить сильно відрізняється синтаксисом багатьох команд, але дуже зручна для користувача і програміста, особливо для тих, хто добре знайомий з мовою програмування С. Як правило, нею користуються адміністратори систем BSD. У сучасних системах, зокрема в Linux і в Mac OS X, стандартною оболонкою є Bourne again shell (/bin/bashабо /usr/bin/bash) – розвиток Bourne shell, що зберігає програмну сумісність з sh, але включає в себе багато нових можливостей.

В подальшому ми будемо використовувати синтаксис оболонки sh (запрошення має вигляд ‘$’), а в окремих випадках порівнювати її з csh (запрошення має вигляд ‘>’).

Командна оболонка, в якій розпочинає роботу користувач після входу в систему, визначається з файлу /etc/passwd. Це один з найголовніших конфігураційних файлів системи, який містить параметри облікових записів користувачів. Кожний рядок файлу відповідає певному користувачу, точніше, обліковому запису користувача, що розрізняється за ідентифікатором login або userid. Користувач в процесі роботи може запустити іншу командну оболонку, просто набравши її ім’я.

Усі команди, які можна ввести у рядку запрошення оболонки, належать до одної з таких категорій:

* вбудовані функції,
* функції, що визначені користувачем,  зовнішні програми й утиліти.

Вбудовані функції реалізуються фрагментами програмного коду оболонки і виконуються найшвидше. Користувач може визначити свої функції (хоча таку можливість використовують нечасто). Якщо команда не є вбудованою функцією і не визначена як функція користувачем, тоді оболонка буде шукати файл з відповідним ім’ям і намагатись запустити його на виконання. Якщо ім’я файлу задано із шляхом до нього, то система намагається знайти його саме у тому каталозі, який явно задано у команді, але якщо ім’я файлу задано без шляху, то пошук файла здійснюється лише у тих каталогах, які задані системною змінною PATH. Для перегляду значення змінних їх слід набрати зі знаком $ попереду. Зокрема, якщо у змінній PATH не задано пошук у поточному каталозі, і поточний каталог не є одним із тих, що явно задано у цій змінній, то оболонка не побачить виконуваних файлів з поточного каталогу.

Розглянемо деякі основні команди. Слід зазначити, що немає штатного засобу, який надавав би користувачеві перелік доступних йому команд, тому основні команди необхідно пам’ятати.

Команда man форматує і відображає на терміналі сторінки довідкової системи. Відповідно до номерів розділів даються посилання на ту чи іншу сторінку довідника. Якщо є необхідність, можна вказати, в якому розділі треба шукати потрібну сторінку (Приклад 2). Приклади використання (системні запрошення не показані):

### man passwd man 7 mdoc

19 Команда passwd, що викликає однойменну утиліту, дозволяє користувачеві змінювати свій пароль входу в систему. Спочатку необхідно підтвердити свою автентичність, набравши свій пароль, а далі можна ввести новий пароль (це здійснюється двічі для уникнення випадкових помилок).

Команда who дозволяє визначити, хто ще працює в поточний момент в системі. Команда who am i нагадає вам, який ваш login.

Існують зручні команди визначення поточної дати й часу (date), а також виводу на екран календаря на будь-який місяць будь-якого року (cal).

Для того, щоби переглянути вміст текстового файлу, можна скористатись командою cat <ім’я файлу>, або more <ім’я

файлу> (остання команда призначена для виводу інформації на екран посторінково, вона надає можливість “перегортати сторінки” вперед і назад). Існує команда wc <ім’я файлу> (word count – підрахувати слова), яка дозволяє підрахувати кількість рядків (wc -l), слів (wc -w) і символів (wc -c) у файлі. Створити текстовий файл можна командою touch <ім’я файлу>.

Розглянемо особливості файлової системи UNIX. Вся файлова система поєднується в єдине дерево каталогів, які починаються з кореневого каталогу, що має позначення ‘/’. Всі зовнішні файлові системи (змінні носії інформації, мережеві диски і таке інше) монтуються у визначенні місця єдиного дерева файлової системи.

Як і в інших ієрархічних файлових системах, у файловій системі UNIX ім’я файлу повинно бути унікальним лише в межах одного каталогу (на відміну від MS-DOS/Windows, UNIX розрізняє великі і малі літери в назвах файлів). Для однозначної ідентифікації файлу в дереві каталогів слід указувати повний шлях до файлу. Якщо шлях починається з символу ‘/’ (наприклад, /usr/bin/cal), то він відраховується від кореневого каталогу (абсолютний шлях), а якщо з іншого символу – то від поточного каталогу, тобто того, в якому користувач знаходиться в поточний момент (відносний шлях). Крім того, поточний каталог позначається символом ‘**.**’ (крапка), каталог, що знаходиться на один рівень вище, тобто батьківський каталог – символом ‘**..**’ (дві крапки). Крім того, існує спеціальне позначення для так званого домашнього каталогу користувача, тобто каталогу, з якого він починає свою роботу – ‘~’ (тильда). Домашній каталог для кожного користувача також задається у файлі /etc/passwd, за умовчанням це /home/<login>.

Для переходу з каталогу в каталог існує команда cd <новий каталог> (change directory – змінити каталог). Якщо використати цю команду без параметрів, відбудеться перехід в домашній каталог користувача.

Слід зазначити, що відносні шляхи слід використовувати з обережністю. Для того, щоби перевірити, в якому каталозі знаходиться користувач, можна скористатись командою pwd.

Перегляд вмісту каталогів здійснюється за допомогою команди ls, а розширений варіант цієї команди ls -l дає також інформацію з таблиці індексних дескрипторів (див. Роботу №2). Щоби скопіювати файл, використовується команда cp **<***файл-джерело***> <***призначення***>**. Для перенесення файлу з каталогу в каталог, а також для перейменування файлу, використовується команда mv **<***файл-джерело***> <***призначення***>**. В обох командах в якості параметра **<***призначення***>** може задаватись каталог призначення або ім’я файлу призначення. Крім того, число параметрів може бути більше двох. В такому випадку всі параметри, крім останнього, розглядаються як список імен файлів-джерел, а останній параметр може бути лише каталогом призначення. Створити каталог можна командою mkdir, видалити файл – командою rm, видалити каталог – командою rmdir або rm -r.

Крім звичайних файлів існують різні типи спеціальних файлів. З одним із них ми вже познайомились – це каталоги. Ще одним типом спеціальних файлів є так звані зв’язки або посилання (рос. – ссылка, англ. – link). В системі UNIX розрізняють два принципово різних типи посилань, хоча створюються вони однією командою – ln. Перший тип – це так звані жорсткі посилання. Фактично вони є абсолютно рівноправними новими іменами вже існуючого файлу. Після створення такого посилання система не розрізняє, яке ім’я було первинне, а яке було створене як посилання. Спроба видалити такий файл призводить до того, що одне з його імен (те, за яким ми намагаємось його видалити), знищується, а інші (як і сам файл) залишаються. Тільки після видалення останнього з імен фактично знищується сам файл. Другий тип посилання – символічне посилання, яке створюють командою ln –l. Це спеціальний тип файлу, який містить в собі ім’я того файлу (або каталогу), на який він посилається. Символічні посилання дуже широко застосовуються в системах UNIX і Linux для забезпечення сумісності програм з різними системами.

Більшість команд, що застосовуються по відношенню до символічного посилання, діють безпосередньо на файл, на який посилання здійснене. Натомість, деякі команди, наприклад mv і rm, діють не на файл, а на посилання. При цьому деякі послідовності команд можуть привести до небажаних наслідків. Наприклад, маємо файл oldfile і бажаємо перейменувати його в newfile. Це можна зробити як командою **mv oldfile newfile**

так і послідовністю команд **cp oldfile newfile rm oldfile**

Результати будуть однакові. До речі, в одному командному рядку можна задати декілька команд, розділивши їх знаком ‘;’, ці команди будуть виконуватись послідовно:

### cp oldfile newfile; rm oldfile

Тепер уявимо, що маємо файл targetfile і посилання на нього oldfile. Команда **mv oldfile newfile**

перейменує посилання, тобто тепер newfileбуде посилатись на targetfile. Команда **cp oldfile newfile**

скопіює не посилання, а сам файл targetfile, тобто під іменем newfileбуде створено новий файл – копію targetfile.

Наступна команда **rm oldfile**

знищить старе посилання, не пошкодивши при цьому файл targetfile. Тобто замість одного файлу з посиланням на нього у нас утворилися два ідентичних файла, які абсолютно не пов’язані між собою.

## Завдання до виконання

1. Завантажтеся в систему під вашим користувацьким ім'ям.
2. Поміняйте ваш пароль. Ваш новий пароль повинен включати в себе як частину номер Вашої залікової книжки.
3. Виведіть системну дату.
4. Підрахуйте кількість рядків у файлі:

|  |  |
| --- | --- |
| Варіант | Файл |
| 1 | **/etc/passwd** |
| 2 | **/etc/group** |
| 3 | **/etc/profile** |
| 4 | **/etc/pam.conf** |
| 5 | **/etc/rsyslog.conf** |
| 6 | **/etc/crontab** |
| 7 | **/etc/modules** |
| 8 | **/etc/networks** |
| 9 | **/etc/protocols** |
| 10 | **/etc/fstab** |

1. Виведіть на екран вміст відповідного файлу.
2. Виведіть календар на <1995+№варіанту> рік.
3. Виведіть календар на 1752 рік. Чи не помічаєте що-небудь цікаве у вересні? Поясніть.
4. Визначте, хто ще завантажений у систему.
5. Наберіть команду ping. Поясніть результат.
6. Скопіюйте (скопіюйте, а не перемістіть, бо система перестане працювати коректно!) файли

|  |  |  |
| --- | --- | --- |
| **варіант** | **файл 1[[1]](#footnote-1)** | **файл 2** |
| 1 | **/bin/cat** | **/bin/grep** |
| 2 | **/bin/echo** | **/bin/chmod** |
| 3 | **/bin/ls** | **/bin/chown** |
| 4 | **/bin/touch** | **/bin/kill** |
| 5 | **/bin/more** | **/bin/gzip** |
| 6 | **/bin/date** | **/bin/more** |
| 7 | **/bin/cp** | **/bin/ps** |
| 8 | **/bin/mv** | **/bin/bash** |
| 9 | **/bin/login** | **/bin/sh** |
| 10 | **/bin/less** | **/bin/rm** |

у ваш домашній каталог різними способами.

1. Створіть каталог lab\_1.
2. Скопіюйте в нього з вашого домашнього каталогу копію файлу 1, яку ви отримали в п.10, під ім'ям my\_<ім’я файлу 1>. Перемістіть в цей каталог з вашого домашнього каталогу копію файлу 2, яку ви отримали в п.10, перейменувавши його при цьому в my\_<ім’я вихідного файлу 2>. За ім’я вихідного файлу слід брати саме ім’я файлу, без імен каталогів і шляху до файлу (інакше символ "/" буде проінтерпретований операційною системою зовсім не так, як Ви очікуєте).
3. Перейдіть у свій домашній каталог і переконайтеся в тому, що все зроблено правильно.
4. Створіть каталог lab\_1\_<№варіанту> і перейдіть в нього.
5. Скопіюйте в каталог lab\_1\_<№варіанту> файл з п.4 під ім'ям n<ім’я вихідного файлу>.
6. За допомогою команд cat і less перегляньте його вміст.
7. Перейдіть у свій домашній каталог.

Видаліть каталог lab\_1\_<№варіанту>.

1. Якщо файл 1 або 2 не знайдено в каталозі /bin, шукайте його в каталогах /usr/bin, /sbin або /usr/sbin [↑](#footnote-ref-1)