

КОМАНДНИЙ ІНТЕРПРЕТАТОР ОПЕРАЦІЙНОЇ СИСТЕМИ UNIX

Мета роботи – вивчення складу та основних команд операційної системи UNIX (LINUX).

Основи ОС UNIX

Оригінальна система UNIX була розроблена у кінці 1960-х років співробітниками Bell Labs Кеном Томпсоном, Деннісом Рітчі та Дугласом Макілроєм. У 1973 р. було прийняте рішення переписати ядро системи на щойно створеній мові Сі. UNIX став першою операційною системою, написаною на мові програмування високого рівня, що суттєво спростило перенесення системи на інші архітектури.

Операційна система UNIX складається із трьох основних компонентів: підсистеми керування процесами, файлової підсистеми та підсистеми введення-виведення. Підсистема керування процесами контролює створення та знищення процесів, розподілення системних ресурсів між ними, міжпроцесову взаємодію, керування пам'яттю.

Файлова підсистема забезпечує єдиний інтерфейс доступу до даних, розташованих на дискових накопичувачах, і до периферійних пристроїв. Такий інтерфейс є однією з найважливіших особливостей UNIX. Одні й ті самі системні виклики використовують як для обміну даними із диском, так і для виведення на термінал або принтер (програма працює із принтером так само, як із файлом). При цьому файлова система переадресовує запити відповідним модулям підсистеми введення-виведення, а ті – безпосередньо периферійним пристроям. Крім того, файлова підсистема контролює права доступу до файлів, які значною мірою визначають привілеї користувача в системі.

Підсистема введення-виведення виконує запити файлової підсистеми, взаємодіючи з драйверами пристроїв. В UNIX розрізняють два типи пристроїв: символні (наприклад, принтер) і блокові (наприклад, жорсткий диск). Основна відмінність між ними полягає в тому, що блоковий пристрій допускає прямий доступ. Для підвищення продуктивності роботи із блоковими пристроями використовують буферний кеш – ділянку пам'яті, у якій зберігаються останні дані, зчитані з диска. Під час наступних звертань до цих даних, вони можуть бути отримані з кеша.

Сучасні UNIX-системи дещо відрізняються за своєю реалізацією, але мають схожу архітектуру. У цих системах підтримується багатопроцесорна обробка, а також багатопотоковість. У них виділено окремий менеджер пам'яті, відповідальний за підтримку віртуальної пам'яті. Стандартом для реалізації інтерфейсу файлової системи є віртуальна файлова система, що абстрагує цей інтерфейс і дає змогу організувати підтримку різних типів файлових систем.

Так, в ОС Linux можна виділити три основні частини:

- ядро, яке реалізує основні функції ОС (керування процесами, пам'яттю, введенням- виведенням тощо);
- системні бібліотеки, що визначають стандартний набір функцій для використання у застосуваннях (виконання таких функцій не потребує переходу в привілейований режим);
- системні утиліти (прикладні програми, які виконують спеціалізовані задачі).

В ОС UNIX реалізовано технологію монолітного ядра. Весь код і структури даних ядра перебувають в одному адресному просторі. У ядрі можна виділити кілька функціональних компонентів:

- Планувальник процесів – відповідає за реалізацію багатозадачності в системі (обробка переривань, робота з таймером, створення і завершення процесів, перемикання контексту).
- Менеджер пам'яті – виділяє окремий адресний простір для кожного процесу і реалізує підтримку віртуальної пам'яті.
- Віртуальна файлова система – надає універсальний інтерфейс взаємодії з різними файловими системами та пристроями введення-виведення.
- Драйвери пристроїв – забезпечують безпосередню роботу з периферійними пристроями. Доступ до них здійснюється через інтерфейс віртуальної файлової системи.
- Мережний інтерфейс – забезпечує доступ до реалізації мережних протоколів і драйверів мережних пристроїв.
- Підсистема міжпроцесової взаємодії – пропонує механізми, які дають змогу різним процесам у системі обмінюватися даними між собою.

Вхід у систему

Кожен користувач має свої ім'я (для ідентифікації і взаємодії користувачів) та пароль (для контролю і захисту). Користувачі можуть бути об'єднані в групи під відповідними іменами. Один з користувачів (superuser), який має ім'я root є адміністратором системи. Він створює всіх інших користувачів.

Знак \$ є запрошенням системи для введення команди. Для входу в систему на запит login задається ім'я, а на запит password- пароль. Для виходу з сеансу вказується ^D(Ctrl+D) або команда logout.

Формат команд

Команда в UNIX - це послідовність слів розділених пропусками. Перше слово - назва команди, решта - її параметри. Команди розділяються закінченням рядка, або крапкою з комою.

Структура команди:

ім'я_команди опції вираз імена_файлів

де опція (ключ) - уточнює команду (може бути зі знаками + - або без знаку); вираз - рядок символів;

ім'я_файла - ідентифікатор(символи a-z, A-Z, 0-9, _, ., -, *, ?)

Символ * означає підстановку будь-якого рядка символів, а символ ? - будь-якого одного символа в ім'я файлу. Службові символи .(крапка) та _ (підкреслення) не можуть бути на початку ідентифікатора. Крапка використовується для виділення в імені файлу так званого "розширення", яке характеризує тип файлу:

- .c- програма на Сі;
- .h- файл заголовку, що долучається до програми на Сі;
- .p- програма на Паскалі;
- .o- об'єктний код отриманий після трансляції з мов;
- .a- бібліотечний архівний файл.

Приклади команд:

```
$ who- виведення списку всіх активних користувачів;  
$ date- виведення поточної дати;  
$ rm text.c prog.c – перейменування файлу;  
$ grep -o "mary" people.
```

Службові комбінації символів:

- ^C- припинення виконання команди;
- ^S- призупинка виведення на екран;
- ^Q- продовження виведення на екран;
- ^D- кінець файлу для стандартного введення.

Імена команд майже завжди записуються малими літерами, ключі, як правило, складаються з однієї літери. На відміну від багатьох інших операційних систем в UNIX враховується різниця між малими і великими літерами.

Якщо необхідно задати два або більше однолітерних ключа, більшість команд дозволяє їх об'єднувати. Наприклад, дві наступні команди ідентичні:

```
$ ls -la a.out  
$ ls -l -a a.out
```

Деякі ключі вимагають наявності параметра. В цьому випадку параметр задається після ключа, в цьому випадку останній не можна об'єднувати з іншим ключем.

Структура директорій

Файлова система UNIX має ієрархічну структуру; вона описується у вигляді дерева. Вершина цього дерева – це директорія **root**. Вона позначається символом **/**. Всі інші директорії і файли обов'язково містяться у директорії **root**.

Один зі шляхів з **root** веде у вашу власну директорію. Ви можете організовувати і зберігати інформацію у вашій власній ієрархії директорій і файлів. Інші шляхи ведуть до системних директорій і доступні всім користувачам. Щоб одержати перелік усіх директорій і файлів у директорії **root**, введіть команду:

```
$ ls -l /
```

Для переміщення по файловій системі, слід використовувати команду **cd** та потрібні імена шляхів. Наприклад, для того, щоб переміститися в директорію **/usr/bin**, потрібно ввести наступну команду:

```
$ cd /usr/bin
```

Основні директорії системи UNIX:

/ - директорія **root**;

/dev - містить спеціальні файли периферійних пристроїв, наприклад консолі (**console**), або принтера (**lp**);

/etc - містить файли конфігурації і бази даних;

/home - домашні директорії користувачів;

/tmp - містить тимчасові файли, наприклад, буфери для редагування файлу;

/var - піддерево для змінюваних файлів (наприклад, файлів реєстрації);

/usr - містить інші директорії, наприклад, **bin**, **lib**;

/usr/bin - містить виконувані програми (наприклад: **cat**, **grep**, **date**, **mkdir**, **login**, **who**);

/usr/lib - містить бібліотеки для програм і мов програмування.

Каталоги та файли

Перші версії UNIX мали обмеження на довжину файлу у 14 символів. В Linux це обмеження знято – імена файлів можуть мати до 256 символів. Деякі з існуючих версій UNIX (наприклад BSD), дозволяють використовувати імена довжиною до 64 символів, але тільки перші 14 з них є значущими. Для забезпечення сумісності між операційними системами рекомендується обмежити імена файлів 14 символами.

Під час реєстрації користувача йому призначається власний каталог (**home directory**). В системі використовується ієрархічна деревовидна структура

каталогів, в які записуються файли. Послідовність імен каталогів та файлів розділених знаком / задає повне ім'я файла, або шлях доступу. Повне ім'я (шлях) файлу може мати необмежену кількість символів.

- /- ім'я кореневого каталога - може бути упущене;
- /lib- каталог бібліотек підпрограм;
- /bin- каталог часто виконуваних команд;
- /tmp- каталог тимчасових файлів;
- /usr- каталог каталогів користувачів;
- /usr/user1- каталог користувача 1;
- /usr/user1/file1.txt- файл file1.txt в каталозі першого користувача.

Команди для роботи з каталогами та файлами

- \$ pwd - виведення повного імені каталога;
- \$ ls - виведення поточного каталогу;
- \$ ls -al – повна інформація (-l) про всі файли та каталоги (-a);
- \$ ls /- виведення кореневого каталога;
- \$ ls /dev- виведення каталога драйверів;
- \$ ls /users/user1 - виведення каталога користувача 1;
- \$ cd /dev - (change directory) - перехід у каталог dev;
- \$ cd - повернення до власного (home) каталога;
- [\$ cd;] \$ mkdir pc pf – створення підкаталогів рста рфу власному каталозі;
- \$ cat /users/user1/file1- виведення файла file1;
- \$ cat file1 - виведення файла file1 з власного каталога;
- \$ tail /file1 - виведення кінця файлу file1(10 останніх стрічок);
- \$ tail -7 /file1 - виведення останніх 7 стрічок файлу file1;
- \$ tail +5 /file1- виведення решти стрічок файлу file1, починаючи з п'ятої;
- \$ file /file1 - виведення типу файлу file1;
- \$ cp [/users/user1/]prog1 [/users/user1/]prog2 - копіювання файлу prog1 в prog2;
- \$ mv prog1 user2/prog1 - перенесення файла prog1 з біжучого каталога в каталог user2 без перейменування;
- \$ mv prog1 prog2 - перейменування файла prog1 в prog2;
- \$ rm [-i] prog1 - видалення файла prog1 (-i-запит підтвердження);
- \$ rmdir user1 - видалення порожнього каталога user1;

Встановлення прав доступу до файлів та каталогів

Оскільки UNIX – багатокористувацька система, тому щоб захистити файли кожного користувача від впливу інших користувачів, UNIX використовує потужний механізм розподілу прав доступу до файлів. Цей механізм дозволяє кожному файлу задати конкретного власника. UNIX дозволяє також спільно використовувати файли декільком користувачам і групам користувачів. У більшості систем за замовчуванням іншим користувачам дозволяється читати ваші файли, але забороняється їх змінювати

і видаляти. Кожен зареєстрований користувач стає членом як мінімум однієї групи користувачів. Системний адміністратор може дати користувачу доступ до більш, ніж однієї групи.

Права доступу поділяються на три типи: читання (read), запис (write) і виконання (execute). Ці типи прав доступу можуть бути надані трьом класам користувачів: власнику файлу, групі, у яку входить власник, і всім (іншим) користувачам. Дозвіл на читання дозволяє користувачу читати вміст файлів, а у випадку каталогів – переглядати перелік імен файлів у каталозі (використовуючи, наприклад, ls). Дозвіл на запис дозволяє користувачу писати у файл і змінювати його. Для каталогів це надає право створювати в каталозі нові файли і каталоги, чи видаляти файли в цьому каталозі. Нарешті, дозвіл на виконання дозволяє користувачу виконувати файли (як бінарні програми, так і командні файли).

Встановлено три категорії користувачів файлами та каталогами:

- власник (user - u);
- користувачі з групи власника (group - g);
- всі інші (other - o).

Встановлено три рівні доступу до файлів та каталогів:

- тільки читання файла чи каталога (read - r);
- читання та запис (write - w);
- виконання файла як командного (execute - x).

Встановлені три операції зміни режиму доступу:

- призначити (=);
- дозволити (+);
- заборонити (-).

\$ chmod <режим> <файли> - встановлення режиму доступу, наприклад:

\$ chmod rwxr----- prog1 prog2;

\$ chmod u-w,g+x,o=r prog1 prog2 - зміна режиму доступу.

Режим доступу встановлюється по категоріях користувачів в такому порядку: власник- група- решта. Розглянемо приклад, що демонструє роботу з правами доступу. Використовуючи команду ls з опцією -l можна одержати на екрані перелік файлів даного каталогу в "довгому" форматі, що включає інформацію про права доступу.

```
/home/student/foo# ls -l stuff
```

```
-rw-r--r-- 1 student users 505 Mar 13 19:05 stuff
```

Перше поле у виведеному рядку представляє права доступу. Третє поле – власника файлу (student) і четверте – групу (users). Очевидно, що останнє поле є ім'я файлу (stuff), а інші полюючи ми обговоримо пізніше. Цим файлом володіє student, і він належить групі users. Давайте подивимось на права доступу. У рядку -rw-r--r-- один по одному зазначені права власника, групи і всіх інших.

Перший символ цього рядка прав доступу ("-") представляє тип файлу.

Символ "-" означає, що це звичайний файл (на противагу чи каталогу спеціальному файлу якогось пристрою). Наступні три позиції ("rw-") представляють права доступу, що має власник файлу student. Символ "r" означає "read"(читати), "w" – "write"(писати). Тобто student може читати файл stuff і писати в нього.

Як ми вже згадували, крім дозволів на читання і запис існує дозвіл на виконання "execute" – представляється символом "x". Але в даному випадку на цій позиції "-", так що в student немає прав на виконання цього файлу. І це правильно, файл stuff не є програмою. Наступні три символи r-- представляють права доступу групи для цього файлу. Ця група має ім'я users. Оскільки отут є тільки "r", будь-який користувач цієї групи може тільки читати файл.

Останні три символи представляють ту ж комбінацію r--, тобто для всіх інших доступно читання цього файлу і заборонені запис та виконання.

От ще кілька прикладів визначення прав доступу.

-rwxr-xr-x – власник файлу може читати, змінювати і виконувати файл.

Члени групи і всіх інших користувачів можуть читати і виконувати файл.

-rw----- – власник файлу може читати і записувати у файл. Всім іншим доступ до файлу закритий.

-rwxrwxrwx – усі можуть читати, змінювати і виконувати файл.

Команди та стандартні файли

У системі є такі стандартні файли, які можуть використовуватись у командах: Standard Input (S.I.) – стандартний ввід (клавіатура), має дескриптор 0; Standard Output (S.O.) – стандартне виведення (екран), має дескриптор 1; Diagnostic Output (D.O.) – виведення діагностики (екран), має дескриптор 2.

Інформація виводиться у стандартні файли по замовчуванню. Однак є можливість при потребі перенаправити введення-виведення у інші файли. Для перенаправлення стандартного виведення використовується значок >, для перенаправлення стандартного введення – значок <. Для перенаправлення стандартного виведення з дозаписом до існуючого файлу використовується значок >>.

Цей варіант рекомендується використовувати завжди, як безпечніший.

Наприклад:

\$ ls – виведення поточного каталога на стандартний пристрій виведення (екран);

\$ ls > catal – перенаправлення інформації з стандартного виведення в файл catal;

\$ ls -l >> catal – дозапис перенаправленого системного виведення у файл catal;

\$ cat < file1 – перенаправлення стандартного введення (інформація на стандартний ввід перенаправляється з файлу file1 і виводиться командою cat (еквівалент \$ cat file1));

\$ cat -u > file2 – створення нового файлу file2 і запис в нього інформації з клавіатури (без буферизації);
\$ cat -u >> file2 – дозапис інформації з клавіатури до файлу.
\$ cat file1 > file2 2> fileerr – перенаправлення стандартного виведення у файл

file2, а виведення діагностики (дескриптор 2) – у файл

Система забезпечує організацію конвейєрів команд, як і в ОС Windows. Конвейєром називається група команд об'єднаних програмними каналами. Програмний канал утворюється призначенням стандартного виведення однієї команди стандартним вводом наступної. Для формування програмного каналу використовується символ |.

Наприклад: \$ who | wc -l – отримання списку активних користувачів (команда who) та підрахунок їх кількості (команда wc).

В системі забезпечена можливість обробки команд у фоновому (background) режимі:

\$ cp file1 file2 & cat file3 – копіювання файлу file1 у фоновому режимі під час виконання команди \$ cat file3

\$ ps [-l][-a] – виведення статусу фонового процесу;

\$ kill 2042 – безумовне завершення процесу з Program Identifier (PID) 2042 (2042 – ідентифікатор фонового процесу).

Редагування тексту

Для редагування тексту в системі UNIX використовується текстовий редактор, який може працювати як екранно-орієнтований (викликається командою vi), або стрічково-орієнтований (викликається командою ex). Робота з цими редакторами проводиться у двох режимах: командному та режимі введення тексту. Перемикання режимів здійснюється клавішею Esc.

\$ vi - редагування нового тексту;

\$ vi file1 - редагування файлу file1 (курсор на початку);

\$ vi +file2 - редагування файлу file2 (курсор в кінці);

\$ vi +10 file3 - редагування файлу file3 (курсор на стрічці 10).

Вихід з редактора здійснюється за двома варіантами: Esc:wq!Enter - зміни запам'ятовуються; Esc:q!Enter - зміни не запам'ятовуються.

Позиціонування курсора здійснюється клавішами зі стрілками а також відповідними символами:

H - початок

екрана; M-
середина
екрана; L-
кінець екрана;
w- переміщення курсора на початок наступного слова;
e- переміщення курсора на кінець біжучого слова;
b- переміщення курсора на початок поточного слова.

Дописування, зміна та видалення тексту здійснюється встановленням курсора в потрібну позицію і натисканням таких клавіш:
x- видалення букви зі посунанням тексту;
r- заміна відміченої курсором букви введеною з клавіатури після букви r;
i- вставити довільний текст перед курсором; a - додати довільний текст після курсора; Esc- кінець введення тексту.

Редактор забезпечує такі операції над словами (початок яких відмічається курсором) та стрічками (курсор визначає позицію в стрічці):

dw - видалення слова; ndw - видалення n слів;
cw <слово> Esc - заміна слова вказаним;
ncw <n слів> Esc - заміна n слів вказаними;
d\$ або D- видалення частини стрічки (від курсора до кінця);
d^ - видалення частини стрічки (від початку до курсора);
c\$ або C <текст заміни>- заміна до кінця стрічки;
c^ <текст заміни> - заміна початку стрічки до курсора;
dd- видалення стрічки;
ndd- видалення n стрічок;
cc <текст заміни> - заміна однієї стрічки;
ncc <текст заміни> - заміна n стрічок;
o - порожня стрічка після біжучої;
O- порожня стрічка перед біжучою;
^іабо I- ввід до біжучої стрічки;
\$аабо A- ввід після біжучої стрічки.

Для роботи з фрагментами тексту (їх розмноження та перенесення) вони можуть зберігатися в буферах. Буферів може бути кілька. Один з них - анонімний (створюється і використовується без імені редактором), решта – поіменовані буквами a-z створюються при потребі користувачем. Для занесення в буфер фрагменти тексту (слова чи стрічки) відмічаються курсором по початку. В анонімному буфері завжди залишається останній фрагмент видалений командою ndd. Команда заміни (c) псує вміст анонімного буфера. Для роботи з буферами використовуються такі команди:

uw - занести відмічене слово в

анонімний буфер; уу- занести відмічену стрічку в анонімний буфер; пуу - занести n стрічок в анонімний буфер; "а5уу- занести 5 стрічок в буфер а; р- вставити вміст анонімного буфера після біжучої стрічки; Р- вставити вміст анонімного буфера перед біжучою стрічкою; "ар- вставити вміст буфера апісля біжучої стрічки; "аР- вставити вміст буфера аперед біжучою стрічкою. "аAdd- додати до буфера аостанню видалену командою ddстрічку (з анонімного буфера).

Зв'язок між користувачами

Система UNIX передбачає можливості організації колективної роботи групи користувачів. Для зв'язку між користувачами використовуються команди write та mail. Команда write видає повідомлення адресату в момент її передачі. Команда mail забезпечує накопичення повідомлень в поштових скриньках повідомляючи користувачів про надходження після закінчення поточної задачі. Про наявність надходжень система попереджає і при початку сеансу.

Варіанти використання команди write:

\$ write <login адресата> <повідомлення> [^D]

\$ write user3 Результати готові?Enter— передача повідомлення з продовженням діалогу (немає ^D);

\$ write user4 Робота завершена^DEnter— передача повідомлення без діалогу;

\$ write user3<file1— передача повідомлення із файлу;

\$ write user3 Go away !whu^d — формування повідомлення з використанням команд ОС;

\$ mesg n — блокування прийому повідомлень;

\$ mesg y — дозвіл прийому повідомлень;

Використання команди mail:

\$ mail <login адресата> <лист>^D — передача листа;

\$ mail — отримання переданих надходжень;

?— запит системи на наступні дії;

??— вивести меню;

х— вийти без зміни пошти; р— друкувати;

— друкувати попередні;

s file1 — зберегти в file1;

w file2 — зберегти в file2 без заголовка;

d — видалити;

+— перейти на наступне (без видалення);

!cmd — виконати команду cmd;

m user1 — переслати користувачу user1;

q— завершити роботу з поштою.

Додаток. Команди користувача ОС UNIX

AC	- виводить статистику про час підключення користувачів до системи.
ALIAS	- команда інтерпретатора bash.
AR	- створює, модифікує і розгортає архіви.
ARCH	- виводить архітектуру машини.
AS	- GNU асемблер.
ASCITOPGM	- перетворює ASCII-графіку у формат portable graymap.
AT	- ставить в чергу, перевіряє або видаляє завдання для подальшого виконання.
ATQ	- ставить в чергу, перевіряє або видаляє завдання для подальшого виконання.
ATRM	- ставить в чергу, перевіряє або видаляє завдання для подальшого виконання.
AUMIX	- встановлює параметри аудіо-мікшера.
AWK	- мова пошуку й обробки шаблонів.
BASENAME	- видаляє імена каталогів і суфікс з імені файлу.
BASH	- командний інтерпретатор мови Shell, що виконує команди їх читанням зі стандартного введення або з файлу.
BATCH	- встановлює в чергу, перевіряє або видаляє завдання для подальшого виконання.
BC	- калькулятор, що дозволяє проводити обчислення з довільною точністю.
BIND	- команда інтерпретатора bash.
CAL	- виводить календар.
CAT	- поєднує файли і виводить на стандартний потік виведення.
CDRECORD	- записує дані на компакт-диск.
CHARSET	- встановлює кодування для терміналу.
CHATTR	- змінює атрибути файлів.
CHROOT	- запускає команду зі зміненої кореневої директорії.
CHVT	- переключає віртуальні термінали.
CJPEG	- стиснення графічних файлів у форматі jpeg.
CKSUM	- виводить контрольну суму і кількість байт у файлі.
CLEAR	- очищає екран терміналу.
CMP	- порівнює два файли.

COL	- фільтрує символи переведення рядка в стандартному потоці
COLRM	- видаляє стовпці з зазначеного файлу.
COLUMN	- формує текст, розбиваючи його на кілька стовпців.
COMM	- виконує порядкове порівняння відсортованих файлів.
COMPRESS	- стискає файл.
CONSOLETYPE	- виводить тип консолі.
CONTINUE	- команда інтерпретатора bash.
CP	- копіює файли і директорії.
CRONTAB	- обслуговує <i>crontab-файли</i> для окремих користувачів.
CSPLIT	- розбиває файл на секції, визначені контекстними рядками.
CSH	- Shell інтерпретатор скриптів.
CUT	- видаляє секції з кожного рядка файлів.
DATE	- виводить поточну дату і час.
DC	- виводить калькулятор.
DD	- перетворить файл під час його копіювання.
DEALLOCVT	- відключає невикористовувані віртуальні термінали.
DF	- видає звіт про використання дискового простору.
DIFF	- шукає розходження між двома файлами.
DIF3	- робить порівняння трьох файлів і пошук розходжень між ними подібно
DIFFP	- перетворить дані у форматі виведення команди diff у формат команди <i>enscript</i> .
DIFSTAT	- створює гістограму з даних у форматі виведення команди diff.
DIG	- налагоджує роботу DNS-сервера.
DIRCOLORS	- налаштовує колір для команди 'ls'.
DIRNAME	- видаляє ім'я файлу з повного імені файлу (залишається тільки ім'я каталогу, де знаходиться файл).
DISOWN	- команди інтерпретатора bash.
DJPEG	- розгорнути JPEG-файл у графічне зображення.
DNSKEYGEN	- генерує відкриті і секретні ключі для DNS.
DOEXEC	- виконує програму.
DU	- оцінка простору, зайнятого файлом.
DUMPKEYS	- виводить інформацію про драйвер клавіатури.
ECHO	- виводить рядок тексту.
ED	- текстовий редактор.
EGREP	- робить пошук заданого фрагмента тексту в одному або декількох файлах.
EJECT	- видає змінний носій.
ENABLE	- команда інтерпретатора bash.
ENSCRIPT	- перетворить текстові файли у файли мовою PostScript.
ENV	- встановлює значення змінних оточення на час виконання зазначеної команди, або виводить значення змінних оточення на екран.
EQN	- форматує документи для системи troff.
EVAL	- команда інтерпретатора bash.

EXEC	команда інтерпретатора bash.
EXIT	- команда інтерпретатора bash.
EXPAND	- перетворить символи табуляції в пробіли.
EXPR	- обчислює значення виразу.
FALSE	- нічого не робить, повертає хибне значення.
FGCONSOLE	- виводить кількість запущених терміналів
FGREP	- робить пошук заданого фрагмента тексту в одному або декількох файлах.
FOR	- команда інтерпретатора bash.
FILE	- визначає тип зазначеного файлу.
FIND	- здійснює пошук файлів.
FLEX	- лексичний аналізатор.
FM	- формує текст, вирівнюючи його по правій межі, видаляючи символи нового рядка.
FOLD	- вирівнює текст по правій межі, розриваючи, якщо необхідно, слова.
FTP	- дозволяє з'єднатися з віддаленою системою з використанням протоколу FTP.
FUNZIP	- фільтр для розпакування ZIP-архіву в потоці.
FUSER	- виводить програмний ідентифікатор (PID) процесів, що використовують
GAWK	- мова пошуку й обробки шаблонів.
GETKEY	- повертає один символ зі стандартного введення.
GETTY	- установлює режим роботи терміналу.
GREP	- робить пошук заданого фрагмента тексту в одному або декількох файлах.
GUNZIP	- розпаковує архіви, створені командами gzip, compress або pack.

GZIP	- архівує файли з використанням методу <i>lempel-Ziv coding</i> .
HEAD	- виводить початок файлу.
HOST	- здійснює пошук імен, використовуючи DNS-сервер.
ID	- виводить реальні й ефективні ідентифікатори <i>UID</i> і <i>GID</i> .
IGAWK	- варіант командного інтерпретатора AWK.
INSTALL	- копіює файли і встановлює їхні атрибути.
IPCALC	- робота з IP-адресами.
JOIN	- поєднує рядки з двох файлів у загальне поле.
KERNELVERSI	- виводить версію ядра.
KILL	- завершує процес.
KILLALL	- завершує процеси по імені команди.
LAST	- відображає список останніх термінальних сесій з вибіркою по користувачі

LN	- створює посилання між файлами.
LOADKEYS	- розкладка клавіатури.
LOGIN	- починає сесію в системі.
LOGNAME	- виводить реєстраційні імена користувачів.
LPR	- локальний друк.
LS	- виводить вміст директорії.
MACID	- виводить інформацію про поточну мандатну мітку і привілеї.
MAN	- форматує і відображає сторінки довідника.
MC	- менеджер файлової системи.
MDEL	- видаляє файли <i>MS-DOS</i> .
MKDIR	- створює директорії.
MKFIFO	- створює <i>FIFO</i> (іменовані канали).
MKNOD	- створює спеціальні блокові або символічні файли.
MKTEMP	- створює тимчасовий файл.
MORE	- посторінково виводить файл на екран.
MKTEMP	- створює тимчасовий файл.
MV	- перейменовує файли.
NICE	- запускає програму зі зміненим пріоритетом для
NL	- нумерує рядка файлів.
PASSWD	- змінює пароль користувача.
PASTE	- поєднує рядки файлів.
PR	- перетворює текстові файли для друку.
PRINTENV	- виводить все оточення або його частину.
PS	- виводить інформацію про процеси.
PSTREE	- виводить дерево процесів.
PWD	- виводить ім'я поточної робочої директорії.
REPLACE	- потоковий редактор, аналогічний sed.
REV	- виконує реверс рядка файлу.
RM	- видаляє файли або каталоги.
RMDIR	- видаляє порожні директорії.
SED	- редактор потоку (Stream EEditor).
S	- командний інтерпретатор мови Shell, що виконує команди їх читанням з стандартного введення або з файлу.
SLEEP	- затримка на задану кількість часу.
SMBSH	- дозволяє робити доступ до файлових систем Windows NT, використовуючи команди UNIX.
SMBSTATUS	- дозволяє робити доступ до файлових систем Windows NT, використовуючи команди UNIX.
SORT	- сортує рядки в текстовому файлі.
STTY	- змінює і виводить налаштування термінала.
SU	- змінює ідентифікатор користувача.
SYNC	- скидає на диск вміст буферів файлової системи.
TAR	- створює архів.

TCSH	- C shell цілком сумісний з csh.
TE	- читає зі стандартного пристрою введення і пише на стандартний пристрій
TEST	- перевіряє типи файлів і порівнює значення.
TOUCH	- змінює тимчасові мітки файлу.
TR	- перетворює або видаляє символи.
TRUE	- нічого не робить, повертає "істину".
TSORT	- робить топологічне сортування.
TT	- виводить ім'я файлу терміналу, асоційованого зі стандартним потоком введення.
UMASK	- встановлює маску режиму створення файлів.
UNAME	- виводить інформацію про систему.
UNIQ	- видаляє повторювані рядки із сортованого файлу.
UPTIME	- показує час роботи системи з останнього запуску.
USERS	- виводить імена користувачів, що у даний момент зареєстровані на поточному комп'ютері.
VI	- текстовий редактор Vi.
W	- показує, які користувачі знаходяться в системі і що вони
WALL	- відправляє повідомлення на всі термінали.
WATCH	- запускає і стежить за програмою через фіксовані інтервали
WC	- виводить кількість байт, слів і рядків для файлу.
WHO	- видає список користувачів, зареєстрованих у системі.
WHOAMI	- виводить ефективний ідентифікатор користувача.
WRITE	- посилає повідомлення іншому користувачеві.
ZCAT	- розпаковує архів, створений командою gzip.