

## Практична робота № 4

### Створення, відслідковування і знищення процесів в ОС Linux

**Мета:** дізнатися, як керувати процесами в Linux. Навчитися переводити процеси із фонового режиму в пріоритетний і навпаки, слідкувати за працюючими процесами, зупиняти процеси і запускати їх так, щоб вони продовжували працювати у вашу відсутність.

#### Завдання.

Запустити цифровий годинник `xclock`.

Перевести його в пріоритетний і фоновий режим роботи.

Запустити аналоговий годинник `xclock`.

Провести моніторинг процесів:

    вивести інформацію про статус процесів;

    вивести інформацію про статус фонових процесів.

Проаналізувати стан пам'яті комп'ютера.

Вивести інформацію:

    про поточний час;

    про час безперервної роботи системи;

    про кількість підключення користувачів;

    про середню завантаженість за останні 1,5 і 15 хвилин.

Вивести список процесів, що постійно оновлюється.

### Теоретичні відомості

**Команда `ps`.** Назва: `ps` - видача інформації про стан процесів.

Синтаксис: `ps [-i] [- d] [- a] [- f] [- l] [- n` файл з системою] `[- t` список терміналів] `[-p` список ідентифікаторів процесів]; `[- u` список ідентифікаторів користувачів] `[- g` список ідентифікаторів лідерів груп]

Опис. Команда `ps` видає інформацію про активні процеси. За замовчуванням інформація дається тільки про процеси, що асоціюються з цим терміналом. Виводяться ідентифікатор процесу, ідентифікатор терміналу, витрачений до цього моменту час ЦП і ім'я команди. Якщо потрібна інша інформація, слід користуватися опціями. Деякі опції мають один аргумент або список аргументів. Аргументи в списку можуть бути або відокремлені один від

одного комами, або всі разом поміщені в подвійні лапки і відокремлені пропусками або комами.

Аргументи в списку процесів і в списку груп мають бути числами.

Командою `ps` обробляються наступні опції: *-i* - вивести інформацію про усі процеси; *-d* - вивести інформацію про усі процеси, окрім лідерів груп; *-a* - вивести інформацію про всі процеси, що найчастіше запрошуються, тобто про усі процеси, окрім лідерів груп і процесів, що не асоціюються з терміналом; *-f* - генерувати повний лістинг (див. нижче роз'яснення сенсу колонок); *-l* - генерувати лістинг в довгому форматі (див. нижче); *-n* - файл з системою (вважати, що операційна система завантажена з файлу з системою, а не з файлу `/unix`); *-t* - список терміналів; видавати інформацію тільки про процеси, що асоціюються з терміналами із заданого списку терміналів.

*Термінал* - це або ім'я файлу-пристрою, наприклад номер або `console`, або просто номер, якщо ім'я файлу починається з `ttu`.

*-p* - список ідентифікаторів процесів: видавати інформацію тільки про вказані процеси; *-u* - список ідентифікаторів користувачів: видавати інформацію тільки про процеси із заданими ідентифікаторами або вхідними іменами користувачів. Ідентифікатор користувача виводиться в числовому виді, за наявності опції *-f* - в символному.

*g* список ідентифікаторів лідерів груп: виводити інформацію тільки про процеси, для яких вказані ідентифікатори лідерів груп. *Лідер групи* - це процес, номер якого ідентичний його ідентифікатору групи. *Shell*, що запускається при вході в систему, є стандартним прикладом лідера групи.

При вказівці опції *-fps* намагається визначити ім'я команди і аргументи, з якими був створений процес, досліджуючи призначений для користувача блок процесу. Якщо це не вдається, ім'я процесу виводиться так само, як і за відсутності опції *-f*, тільки полягає в квадратних дужках.

Нижче наводяться заголовки колонок видачі і пояснюється значення їх вмісту. Букви *l* або *f* в дужках означають, що ця колонка з'являється відповідно при довгому або повному форматі видачі; відсутність букв означає, що ця колонка виводиться завжди. Відмітимо, що опції *-l* і *-f* впливають тільки на формат видачі, але не на список процесів, інформація про які буде надана.

*F(1)* - Прапори(шістнадцятирічні), логічна сума яких дає наступні відомості про процес : *00* - процес термінує; елемент таблиці процесів вільний.

*01* - системний процес: завжди в основній пам'яті.

02 - процес трасується батьківським процесом.

04 - батьківський сигнал трасування зупинив процес; батьківський процес чекає.

08 - процес не може бути розбуджений сигналом.

- процес в основній пам'яті.

- процес в основній пам'яті; блокований до завершення події.

- поступає сигнал до видаленої системи.

- процес в черзі на введення/виведення.

-  $S(l)$  - статус процесу :  $0$  – активний, обробляється процесором;  $S$  – сплячий, чекає завершення події;  $R$  – готовий, стоїть в черзі на виконання;  $I$  - що народжується, процес створюється;  $Z$  - стан "зомбі", процес завершений, але батьківський процес не чекає цього;  $T$  – трасований, процес зупинений сигналом, оскільки батьківський процес трасує його;  $X$  - процес що росте, чекає отримання більшого об'єму основної пам'яті.

$UID(f, l)$  - Ідентифікатор власника процесу; при вказівці опції -  $f$  видається вхідне ім'я користувача.

$PID$  - Ідентифікатор процесу(потрібний для того, що термінує процесу).

$PPID(f, l)$  - Ідентифікатор батьківського процесу.

$C(f, l)$  - Доля виділеного планувальником часу ЦП.

$STIME(f)$  - Час запуску процесу(години:хвилини:секунди). Якщо процес запущений більш ніж 24 години тому, видається місяць і день запуску.

$PRI(l)$  - Пріоритет процесу; більше число означає менший пріоритет.

$NI(l)$  - Поправка до пріоритету.

$ADDR(l)$  - Адреса процесу в пам'яті.

$SZ(l)$  – Розмір (у блоках по 512 байт) образу процесу в пам'яті.

$WCHAN(l)$  - Адреса події, на яку чекає процес. У активного процесу ця колонка порожня.

$TTY$  - термінал(зазвичай - термінал, з якого був запущений процес; див. нижче СЮРПРИЗИ), що управляє. Якщо такого немає, видається символ «?».

$TIME$  - Витрачений процесом час ЦП.

$COMMAND$  - Ім'я програми; якщо вказана опція -  $f$ , то виводиться повне ім'я команди і її аргументи.

Процес, що закінчив виконання своєї програми, і батьківський процес, що не дочекався ще завершення, в якості імені програми отримує <defunct>.

**Файли.**  $/dev/kmem$  - Віртуальна пам'ять ядра.

*/dev/mem* - Пам'ять.

*/etc/passwd* - Файл паролів, з якого береться інформація про ідентифікатори користувачів.

*/etc/psdata* - Внутрішня структура даних. */unix* - Ім'я файлу, звідки була завантажена операційна система. **Сюрпризи.** Під час виконання команди *ps* стан речей може змінитися;

таким чином, *ps* дає лише наближення до дійсності. Деякі відомості, що відносяться до процесів, відмічених як *<defunct>*, не відповідають дійсності.

Якщо не вказана жодна з опцій - *t*, -*p*, -*u*, -*g*, команда *ps* перевіряє стандартне введення, стандартне виведення і стандартний протокол (у вказаному порядку) для визначення терміналу, що управляє і намагається вивести інформацію про процеси, які асоціюються з цим терміналом. За ситуації, коли усі три стандартні напрями введення/виведення перепризначувалися, *ps* не може знайти і видати ім'я терміналу, що управляє.

**Системний виклик *fork*.** Назва: *fork* - створення нового процесу. Синтаксис: *int fork( )*.

Опис. Виклик *fork* призводить до створення нового процесу (породженого процесу) - точної копії процесу, що зробив виклик (батьківського процесу). Точніше, породжений процес наслідує у батьківського процесу наступні характеристики:

Оточення.

Прапор "закрити при виконанні виклику *exec*".

Способи обробки сигналів (тобто *SIG DFL*, *SIG IGN*, *SIGHOLD*, адреси функцій обробки сигналів).

Дозвіл встановлювати заново діючий ідентифікатор користувача.

Дозвіл встановлювати заново діючий ідентифікатор групи.

Стан профілізації(включено/вимкнено).

Значення поправки до пріоритету.

Усі приєднані сегменти пам'яті, що розділяються.

Ідентифікатор групи процесів.

Ідентифікатор групи терміналу.

Поточний робочий каталог.

Кореневий каталог.

Маска режиму створення файлів.

Обмеження на розмір файлу.

Породжений процес відрізняється від батьківського процесу наступним.

Породжений процес має свій унікальний ідентифікатор процесу.

Породжений процес має інший ідентифікатор батьківського процесу, рівний ідентифікатору процесу, його породжувача.

Породжений процес має свої власні копії батьківських дескрипторів файлів. Кожен дескриптор файлу породженого процесу розділяє з відповідним батьківським дескриптором файлу загальний покажчик поточної позиції у файлі.

Усі `semadj` значення скидаються [див. `semop(2)`].

Породжений процес не наслідуює у батьківського процесу ознак утримання в пам'яті сегменту команд, даних або усього процесу цілком [див. `rlock(2)`].

Обнуляються лічильники часу, витраченого для обслуговування цього процесу (`tms utime`, `tms stime`, `tms cutime`, `tms cstime`). Відміняється запит до будильника.

## Хід роботи

### Управління пріоритетними і фоновими завданнями

Коли ви запускаєте команду у вікні терміналу, вона виконується в пріоритетному режимі. Багато команд виконуються швидко, але припустимо, що ви працюєте в графічному режимі і хочете бачити на робочому столі цифровий годинник.

Використаємо утиліту `xclock`. Для того щоб запустити цифровий годинник на вашому робочому столі потрібно виконати в терміналі наступну команду: `$ xclock -d -update 1`

Опція `-update 1` задає оновлення годинника кожної секунди; в іншому випадку оновлення буде виконуватися кожної хвилини. Ви можете побачити годинник, зображений на рисунку 1, а вміст вашого вікна термінала має бути схожим на вміст лістингу 1.

*Примітка.* На момент написання цієї роботи у роботі `xclock` була присутня програмна помилка, яка виявлялася при ввімкнутих ефектах робочого столу. Вона виявлялася, перш за все, у тому, що панель заголовку не змінювалася навіть при перемиканні на неї. Якщо у вас виникає така помилка, то можна тимчасово вимкнути ефекти робочого столу.

### Лістинг 1. Запуск xclock

```
daniel@daniel-desktop: ~  
Файл Правка Вид Поиск Терминал Справка  
daniel@daniel-desktop:~$ xclock -d -update 1
```

Тепер у вашому вікні термінала не відображається запрошення, тому необхідно повернути управління.

командному інтерпретаторі bash є комбінація клавіш для призупинення Ctrl-z. Натиснувши цю комбінацію клавіш, ви знову побачите запрошення термінала, як показано в *лістингу 2*.

### Лістинг 2. Призупинка xclock за допомогою Ctrl-z.

```
daniel@daniel-desktop: ~  
Файл Правка Вид Поиск Терминал Справка  
daniel@daniel-desktop:~$ xclock -d -update 1  
^Z  
[1]+ Остановлено xclock -d -update 1  
daniel@daniel-desktop:~$
```

Годинник все ще знаходиться на вашому робочому столі, але із-за призупинення він не йде. Зверніть увагу на повідомлення “[1]+Остановлено”

вікні терміналу. Цифра 1 в цьому повідомлення означає номер завдання. Ви можете перезапустити годинник, набравши команду *fg %1*. Тут ви також можете використати ім'я команди або його частину, набравши, наприклад, *fg %xclock* або *fg %?clo*. Нарешті якщо ви запустите команду *fg* без яких-небудь параметрів, то буде перезапущено останнє зупинене завдання, в нашому випадку це завдання 1. Перезапуск завдання за допомогою *fg* поновлює його роботу в пріоритетному режимі, і ви знову не побачите запрошення. В цій ситуації нам потрібно перевести завдання у фоновий режим. Саме для цього призначена команда *bg*, параметри якої аналогічні параметрам команди *fg*.

лістингу 3 показано, як поновити роботу програми xclock в пріоритетному режимі, призупинити її, і знову запустити, використовуючи дві форми команди *fg*. Після призупинки ви можете поновити роботу xclock в фоновому режимі — годинник продовжить працювати, а ви зможете продовжити роботу у вікні терміналу.

```
daniel@daniel-desktop: ~  
Файл Правка Вид Поиск Терминал Справка  
daniel@daniel-desktop:~$ xclock -d -update 1  
^Z  
[1]+ Остановлено xclock -d -update 1  
daniel@daniel-desktop:~$ fg %1  
xclock -d -update 1  
^Z  
[1]+ Остановлено xclock -d -update 1  
daniel@daniel-desktop:~$ fg %?clo  
xclock -d -update 1  
^Z  
[1]+ Остановлено xclock -d -update 1  
daniel@daniel-desktop:~$ bg  
[1]+ xclock -d -update 1 &  
daniel@daniel-desktop:~$
```

### Лістинг 3. Переведення xclock в пріоритетний або фоновий режим роботи.

Ви могли помітити, що при переведенні xclock в фоновий режим в кінці команди використовується *&*, а повідомлення “Остановлено”

зникає. На справді не обов'язково призупиняти процес для його переведення в фоновий режим; для цього потрібно просто додати `&` в кінець команди, і інтерпретатор запустить її (або список команд) в фоновому режимі.

Використаємо це для запуску аналогічного годинника. Ви маєте побачити годинник, зображений на рисунку 1, а вміст вашого вікна термінала має бути схожим на вміст лістингу 4.



Рисунок 1 – Аналоговий годинник *xclock*

**Лістинг 4.** Запуск аналогового годинника *xclock* в фоновому режимі за допомогою `&`.

```
daniel@daniel-desktop: ~
Файл Правка Вид Поиск Терминал Справка
daniel@daniel-desktop:~$ xclock -bg wheat -update 1 &
[2] 2625
daniel@daniel-desktop:~$
```

Цього разу повідомлення дещо відрізняється. Воно містить номер завдання і ідентифікатор процесу *process id (PID)*. Давайте запусимо команду *jobs* і

подивимось які завдання виконуються в нашій системі. Додайте опцію `-l` для відображення ідентифікаторів *PID*, і ви побачите, що завдання 2 має ідентифікатор *PID* 2625, як показано на лістингу 5. Також зверніть увагу на те, що завдання 2 має знак `+`, який означає, що це завдання є поточним. Якщо ніяке інше завдання не призначене на виконання командою *fg*, то поточне завдання буде виконуватися в пріоритетному режимі.

**Лістинг 5.** Вивід інформації про запущені завдання і процеси, що виконуються.

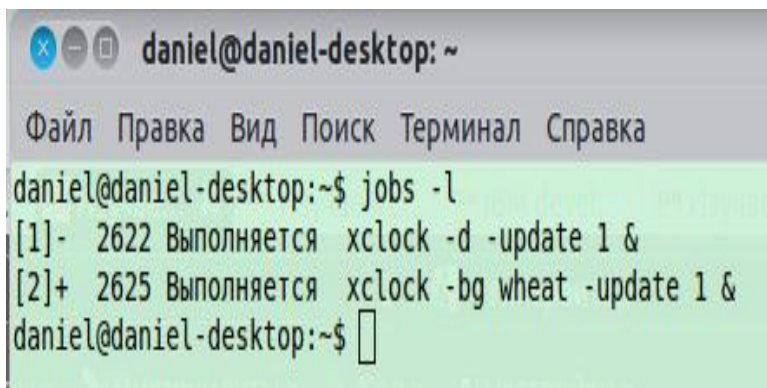
```
daniel@daniel-desktop: ~
Файл Правка Вид Поиск Терминал Справка
daniel@daniel-desktop:~$ jobs -l
[1]- 2622 Выполняется xclock -d -update 1 &
[2]+ 2625 Выполняется xclock -bg wheat -update 1 &
daniel@daniel-desktop:~$
```

Далі створимо у вікні терміналу простий цифровий годинник. Ми будемо використовувати команду *sleep*



для створення двохсекундної паузи і команду *date* для відображення поточної дати і часу. Ми розташуємо ці дві команди в циклі *while* з блоком *do/done* для створення нескінченного циклу. Цю конструкцію розташуємо в дужках для створення списку команд і запустимо цей список на виконання в фоновому режимі, використовуючи *&*.

### Лістинг 6. Простий цифровий годинник.



```
daniel@daniel-desktop: ~
Файл Правка Вид Поиск Терминал Справка
daniel@daniel-desktop:~$ jobs -l
[1]- 2622 Выполняется xclock -d -update 1 &
[2]+ 2625 Выполняется xclock -bg wheat -update 1 &
daniel@daniel-desktop:~$
```

Наш список команд запускається у вигляді завдання з номером 2 і ідентифікатором *PID* 2969.

Кожні дві секунди запускається команда *date*, і на екран виводиться поточна

дата і час. Данні, що вводяться нами в лістингах, виділені жирним шрифтом. Якщо ви друкуєте повільно, то перш ніж ви закінчите набирати команду, між символами, що ви вводите можуть з'явитися рядки виводу. Дійсно, ви можете помітити, що символи *'f'* і *'g'*, які ми надрукували для переведення списку в пріоритетний режим, розділені декількома рядками. Коли нарешті команда *fg* надрукована, інтерпретатор *bash* показує поточну команду, що працює в ньому, а саме наш список команд, який продовжує виводити час кожні дві секунди.

Як тільки ви перевели завдання в пріоритетний режим, ви можете завершити його або виконати над ним яку-небудь іншу дію. У нашому випадку давайте натиснемо *Ctrl-C* і зупинимо наш годинник.

## Моніторинг процесів

### Команда *ps*

Існує і інша команда, що відображає різну інформацію про статус процесу — команда *ps* ("*process status*" - *статус процесу*). Команда *ps* приймає якості аргументів нуль або більше ідентифікаторів *PID* і відображає статус відповідних процесів. Якщо запустити команду *jobs* з опцією *-p*, то її вивід буде представляти собою ідентифікатор лідера групи процесів кожного завдання. Ми будемо використовувати цей вивід в якості аргументів команди *ps*, як показано лістингу 7.



## Лістинг 7. Статус фонових процесів.

```
daniel@daniel-desktop: ~
Файл Правка Вид Поиск Терминал Справка
daniel@daniel-desktop:~$ jobs -p
2622
daniel@daniel-desktop:~$ ps $(jobs -p)
  PID TTY          STAT       TIME COMMAND
 2622 pts/0    S          0:01 xclock -d -update 1
daniel@daniel-desktop:~$
```

Якщо ви запустите команду `ps` без яких-небудь опцій, то ви побачите список процесів, для яких ваш термінал є керуючим терміналом.

Деякі опції, включаючи `-f` (повний), `-j` (`jobs` — завдання) і `-l` (`long` — довгий) дозволяють керувати кількістю інформації, що виводиться. Якщо ви не вкажете ніякі ідентифікатори `PID`, то іншою корисною опцією є опція `-forest`, що відображає команди у вигляді дерева і показує, які процеси є батьківськими для інших процесів. Зокрема можна побачити, що команди `sleep` з попередніх лістингів є дочірніми процесами сценаріїв, запущених вами в фоновому режимі. Приклади використання деяких опцій наведені в лістингу 8.

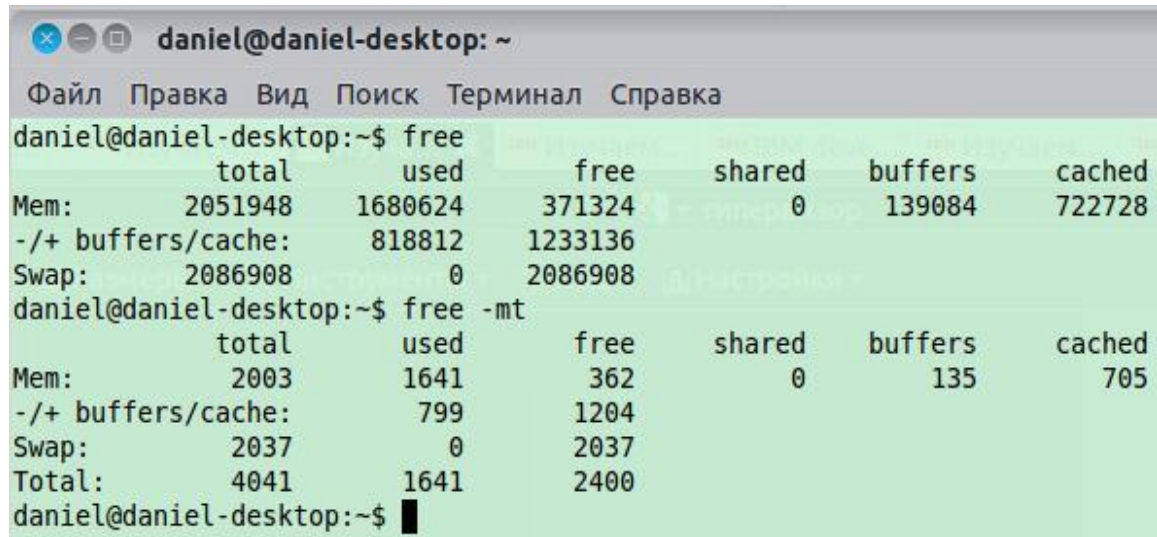
## Лістинг 8. Більш детальна інформація про статуси процесів.

```
daniel@daniel-desktop: ~
Файл Правка Вид Поиск Терминал Справка
daniel@daniel-desktop:~$ ps -f
UID          PID  PPID  C  STIME TTY          TIME CMD
daniel      2321  2316  0  11:10 pts/0        00:00:00 bash
daniel      2622  2321  0  12:12 pts/0        00:00:01 xclock -d -update 1
daniel      3301  2321  0  14:19 pts/0        00:00:00 ps -f
daniel@daniel-desktop:~$ ps -j --forest
  PID  PGID  SID  TTY          TIME CMD
 2321  2321  2321 pts/0        00:00:00 bash
 2622  2622  2321 pts/0        00:00:01 \_ xclock
 3302  3302  2321 pts/0        00:00:00 \_ ps
daniel@daniel-desktop:~$
```

### Команда `free`.

Команда `free` відображає кількість вільної пам'яті і пам'яті, що використовується комп'ютером. По замовчуванню значення виводяться в кілобайтах, але можна змінити це за допомогою різних опцій: `-b` (байт) `-k` (кб) `-m` (мб) `-g` (гб). Опція `-t` виводить сумарний рядок, а опція `-s` оновлює інформацію з вказаною періодичністю (час вказується в секундах). В лістингу 9 наведені два приклади.

## Лістинг 9. Використання команди free.

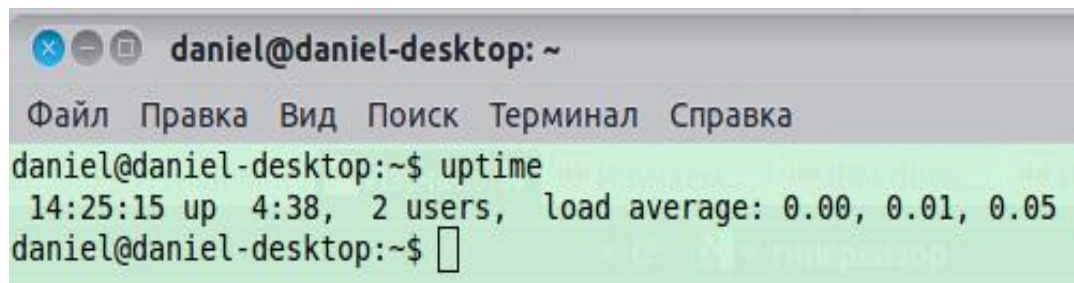


```
daniel@daniel-desktop: ~
Файл Правка Вид Поиск Терминал Справка
daniel@daniel-desktop:~$ free
              total        used          free      shared    buffers     cached
Mem:          2051948    1680624     371324           0     139084     722728
-/+ buffers/cache:    818812    1233136
Swap:          2086908              0     2086908
daniel@daniel-desktop:~$ free -mt
              total        used          free      shared    buffers     cached
Mem:             2003         1641           362           0           135         705
-/+ buffers/cache:           799         1204
Swap:             2037              0         2037
Total:            4041         1641         2400
daniel@daniel-desktop:~$
```

### Команда uptime.

Команда `uptime` виводить рядок, де вказано поточний час, час безперервної роботи системи, кількість підключених користувачів і середня завантаженість за останні 1, 5 і 15 хвилин. Приклад команди показаний в лістингу 10.

### Лістинг 10. Інформація, що виводиться командою uptime.



```
daniel@daniel-desktop: ~
Файл Правка Вид Поиск Терминал Справка
daniel@daniel-desktop:~$ uptime
 14:25:15 up 4:38, 2 users, load average: 0.00, 0.01, 0.05
daniel@daniel-desktop:~$
```

### Використання команди top.

Якщо ви хочете відслідковувати зміни і для цього запускаєте команду `ps` декілька раз, то, можливо, кращім рішенням буде команда `top`. Ця команда відображає список процесів, що постійно оновлюється, а також корисну сумарну інформацію. В лістингу 11 показані перші декілька рядків виводу команди `top`. Використовуйте підкоманду `q` для закриття `top`.

## Лістинг 11. Вивід інформації про процеси за допомогою *top*.

```
daniel@daniel-desktop: ~
Файл Правка Вид Поиск Терминал Справка
top - 14:40:12 up 4:53, 2 users, load average: 0.00, 0.01, 0.05
Tasks: 150 total, 1 running, 148 sleeping, 0 stopped, 1 zombie
Cpu(s): 10.3%us, 4.3%sy, 0.0%ni, 85.4%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 2051948k total, 1674556k used, 377392k free, 141112k buffers
Swap: 2086908k total, 0k used, 2086908k free, 711820k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM   TIME+  COMMAND
 1792 daniel    20   0  519m 242m  23m  S   12  12.1 32:43.71 plugin-containe
 1729 daniel    20   0  700m 276m  33m  S    8  13.8 16:59.75 firefox
 1118 root      20   0  70592 24m  10m  S    3  1.2  7:06.77 Xorg
 1540 daniel    9  -11  160m  10m 8668  S    3  0.5  2:32.19 pulseaudio
 1536 daniel    20   0 60920 15m  9.8m  S    1  0.8  1:55.07 compiz
 1849 daniel    20   0  292m  73m  23m  S    1  3.7  3:38.36 skype
 1324 rtkit     21   1 18888 1188 1000  S    0  0.1  0:00.14 rtkit-daemon
 2316 daniel    20   0  95032 14m  10m  S    0  0.7  0:04.31 gnome-terminal
    1 root      20   0  2920 1768 1216  S    0  0.1  0:00.90 init
daniel@daniel-desktop:~$
```

Команда *top* має декілька підкоманд, з яких найбільш корисними є:

*h* — вивід довідкової інформації; *q* — завершення роботи команди *top*; *f* — додавання і видалення полів, що відображаються; *o* — сортування порядку виводу інформації; *F* — вибір полів, по яким виконується сортування.

### Контрольні запитання

Що таке процеси? Які основні причини створення нових процесів у обчислювальних системах?

Що називається образом процесів? Назвіть атрибути процесів?

Які дії виконує ОС при створенні процесів?

Які дії виконує ОС при перемиканні процесів?

Які є види планування процесів?

Назвіть основні критерії короткострокового планування.

Які Ви знаєте стратегії короткострокового планування процесів?