“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №4**

з дисципліни: «Операційні системи»

**Тема: «Команди Linux для управління процесами»**

Виконали:

студенти групи РПЗ-93б

**Команда 1**: Усенко Б.О.,

Мельнічук М.О

Перевірив викладач

Сушанова В.С.

Київ 2022

**Мета роботи:**

1. Отримання практичних навиків роботи з командною оболонкою Bash.
2. Знайомство з базовими командами для управління процесами.

**Матеріальне забезпечення занять**

1. ЕОМ типу IBM PC.

2. ОС сімейства Windows (Windows 7).

3. Віртуальна машина – Virtual Box (Oracle).

4. Операційна система GNU/Linux – CentOS.

5. Сайт мережевої академії Cisco netacad.com та його онлайн курси по Linux

**Завдання для попередньої підготовки.**

***Готувала матеріал студентка Мельнічук М.О.***

1. Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеличкий словник базових англійських термінів з питань призначення команд та їх параметрів.

|  |  |
| --- | --- |
| **Термін** | **Переклад** |
| ps | Може надати багато інформації про всі програми, запущені у вашій системі |
| PID | ID програм |
| ps -A | Показує всі процеси |
| ps -ef | Комбінація параметрів для виведення інформації про програми. |
| ps -l | Отримання додаткової інформації. |
| top command | Відображає інформацію про процес подібно до команди ps, але робить це в режимі реального часу |
| TERM | Припиняє процес, якщо можливо |
| kill command | Надсилає сигнал TERM всім PID, переліченим у командному рядку |
| killall command | Спосіб зупинити процеси, використовуючи їх імена, а не номери PID |

1. На базі розглянутого матеріалу дайте відповіді на наступні питання:
   1. Які команди для моніторингу стану процесів ви знаєте. Як переглянути їх можливі параметри?

Команди для моніторингу стану процесів: **ps**(з різними ключами) та **top**.

За замовчуванням команда **ps** показує лише процеси, які належать поточному користувачеві та запущені на поточному терміналі.

*The ps Command Unix Parameters*

|  |  |
| --- | --- |
| Parameter | Description |
| -A | Shows all processes |
| -N | Shows the opposite of the specified parameters |
| -a | Shows all processes except session headers and processes without a terminal |
| -d | Shows all processes except session headers |
| -e | Shows all processes |
| -U userlist | Shows processes owned by a userid listed in *userlist* |
| -l | Displays a long listing |
| -L | Shows process threads |
| -V | Displays the version of ps |

Команда **top** відображає інформацію про процес подібно до команди ps, але вона робить це в режимі реального часу.

* 1. Чи може команда ps у реальному часі відслідковувати стан процесів?

Команда **ps** чудово підходить для збору інформації про процеси, що виконуються в системі, але вона має один недолік. Команда **ps** може відображати інформацію лише для певного моменту часу. Якщо ви намагаєтеся знайти тенденції щодо процесів, які часто змінюються в пам’яті та з неї, це важко зробити за допомогою команди **ps**.

Натомість команда **top** може вирішити цю проблему. Команда **top** відображає інформацію про процес подібно до команди **ps**, але вона робить це в режимі реального часу.

* 1. За якими параметрами можливе сортування процесів в команді top? Як переключатись між ними?

За замовчуванням, коли ви запускаєте **top**, він сортує процеси на основі значення **% CPU**. Ви можете змінити порядок сортування, використовуючи одну з кількох інтерактивних команд під час роботи **top**. Кожна інтерактивна команда — це один символ, який ви можете натиснути під час запуску **top** і змінює поведінку програми. Натискання **f** дозволяє вибрати поле для сортування виводу, а натискання **d** дозволяє змінити інтервал опитування. Натисніть **q,** щоб вийти з верхнього дисплея.

* 1. Які команди для завершення роботи процесів ви знаєте?

Команди для завершення роботи процесів: **kill** та **killall**.

Команда **kill** дозволяє надсилати сигнали процесам на основі їх ідентифікатора процесу (PID). За замовчуванням команда **kill** надсилає сигнал TERM всім PID, перерахованим у командному рядку.

Команда **killall** — це потужний спосіб зупинити процеси, використовуючи їх імена, а не номери PID. Команда **killall** також дозволяє використовувати символи підстановки, що робить її дуже корисним інструментом, коли у вас система, яка зійшла з ладу.

1. Вивчіть матеріали онлайн-курсів академії Cisco:

* NDG Linux Unhatched (Chapter 14 - 18 all Topics)

1. Дайте відповіді на такі питання (на базі вивченого курсу):

**3.1**. Які команди-фільтри ви знаєте?

Команда **grep** — це текстовий фільтр, який шукатиме введені та повертає рядки, які містять відповідність заданому шаблону.

grep [*OPTIONS*] *PATTERN* [*FILE*]

**3.2.** Що таке регулярні вирази та базові патерни, для чого вони використовуються?

Регулярні вирази Linux — це спеціальні символи, які допомагають шукати дані та відповідати складним шаблонам. Регулярні вирази скорочуються як «**re-gexp**» або «**regex**». Вони використовуються в багатьох програмах Linux, таких як grep, bash, re-name, sed тощо. Регулярні вирази — це шаблони, які можуть інтерпретувати лише певні команди. Регулярні вирази можна розширити, щоб відповідати певній послідовності символів у тексті.

**3.3.** Які базові команди мережевої конфігурації ви знаєте?

Команда **ifconfig** означає «конфігурація інтерфейсу» і використовується для відображення інформації про конфігурацію мережі.

ifconfig [OPTIONS]

Команду **ifconfig** також можна використовувати для тимчасової зміни параметрів мережі. Зазвичай ці зміни мають бути постійними, тому використання команди **ifconfig** для внесення таких змін досить рідко.

Команда **ping** використовується для перевірки з'єднання між двома комп'ютерами. Він робить це шляхом надсилання пакетів на іншу машину в мережі. Якщо відправник отримує відповідь, має бути можливість підключитися до цієї машини.

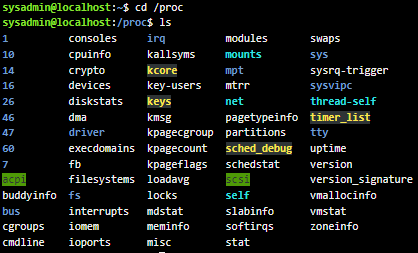
**3.4.** Які системи управління пакетами ви знаєте, для чого вони потрібні?

Управління пакетами – це система, за допомогою якої програмне забезпечення може *встановлюватися, оновлюватися, запитуватися або видалятися з файлової системи*. У Linux існує багато різних систем керування пакетами програмного забезпечення, але дві найпопулярніші — це від Debian і Red Hat. Віртуальні машини для цього курсу використовують Ubuntu, похідну від Debian.

**Хід роботи. *Готувала матеріал студентка Усенко.Б.О.***

* 1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:
  2. Запустіть віртуальну машину Ubuntu\_PC (якщо виконуєте завдання ЛР через академію netacad)
  3. Запустіть термінал, та в командному рядку виконайте наступні дії для ознайомлення з роботою з каталогами:
* вивести вміст директорії /proc. Де вона знаходиться та для чого призначена? Охарактеризуйте інформацію про її вміст.

У цьому каталозі всі файли віртуальні. Вони розташовуються не на диску, а в ОП. У цих файлах міститься інформація про програми (про процеси), які виконуються в даний момент в системі.

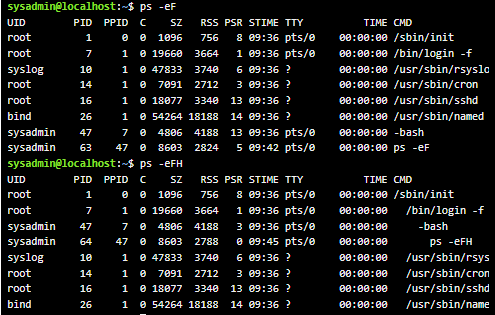


* вивести поточні сеанси користувачів. Якою командою це можна зробити?



* вивести інформацію про всі процеси, що виконуються. Які параметри при цьому треба використати?

Найпростіший спосіб подивитися список процесів, запущених в поточній командою оболонці, використовувати команду ps без параметрів.Щоб подивитися всі процеси, додайте опцію -e , а для максимально докладної інформації - опцію -F. Щоб подивитися список процесів у вигляді дерева, і розуміти який процес має якісь дочірні процеси додайте ще опцію **H.**

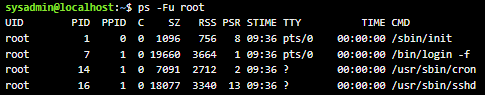


Щоб вивести тільки активні процеси, використовуємо параметр r.



* вивести інформацію про процеси одного користувача. Які параметри при цьому треба використати?

Щоб подивитися список процесів певного користувача, наприклад, root використовуйте опцію -u

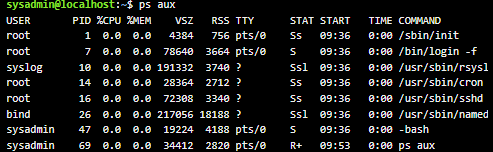


* вивести інформацію тільки про системні процеси. Які параметри при цьому треба використати?

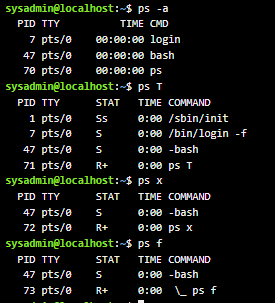
Щоб вивести системні процеси потрібно використати параметр grep, та шукаємо syslog.



* вивести інформацію про процеси за обраним вами критерієм (5 прикладів). Які параметри при цьому використані?



ps aux -  всі процеси з додатковою інформацією



ps -a - процеси запущені в даному терміналі, окрім головних системних

ps T - все на конкретному терміналі

ps x - від’єднані від терміналу

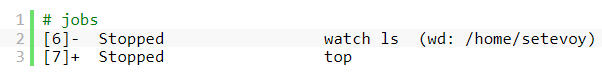
ps f - дерево процесів з батьками.

* 1. При роботі з процесами досить часто виникає необхідність запуску та роботи з фоновими процесами. Дайте відповіді на такі питання:
* **Чим відрізняється фоновий процес від звичайного. Де вони використовуються?**

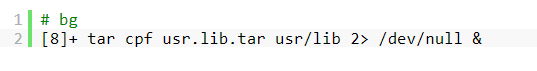
Фонові процеси - не пов'язані з конкретними користувачами процеси, але виконують ряд специфічних функцій. Основну частину часу такі процеси проводять в сплячому режимі, активізуючись тільки в міру появи активної діяльності (електронна пошта, веб-сторінки, виведення інформації на друк тощо). Такі фонові процеси досить часто називають демонами.

* **Опишіть наступні команди та поясніть що вони виконують – команда jobs, bg, fg.**

Відобразити список поточних фонових завдань можна командою **jobs**:



Тепер, щоб продовжити виконання у фоновому режимі – введіть команду **bg (background):**



Що б вивести задачу з фонового режиму - використовується команда **fg (foreground)**, якій можна або передати номер завдання в якості аргументу, або - запустити без аргументів.



* **Якою командою можна переглянути інформацію про запущені в системи фонові процеси та задачі?**

в Linux подивитися процеси потрібно скористатися командою ps

* **Як призупинити фоновий процес, як його потім відновити та при необхідності перезапусти?**

Для примусового завершення активного і фонових процесів використовуються різні способи. Як вже раніше відмічалось, активний процес можна ліквідувати, натиснувши клавіші <CTRL><C> або клавішу DEL. Для завершення фонового процесу використовується команда kill, яка має кілька форматів: kill PID kill - signal PID kill%n Ця команда може брати як аргумент номер роботи, або ідентифікатор процесу. Наприклад, для завершення процесу із ідентифікатором PID=237 необхідно виконати команду kill 237, а для завершення роботи із номером 20 необхідно виконати команду kill%20 Для перевірки ліквідації вказаного процесу, можна виконати команду ps в результаті чого на екрані дисплея отримаємо відповідь: 237 Terminated А якщо виконати команду jobs тоді теж одержимо аналогічне підтвердження: [20] + Terminated Ключ “-signal” змушує команду kill виконати ряд додаткових послуг, тобто послати процесу певний сигнал. Може бути послано понад 20 сигналів, кожний з яких має свій номер. При виході користувача із системи, Linux посилає всім його процесам сигнал 1, що змушує всі процеси завершити роботу. За замовчуванням усім процесам посилається сигнал 15. Якщо ввести команду kill 0, то можна ліквідувати всі фонові процеси. Якщо який-небудь процес “завис”, тоді потрібно перейти до іншої консолі, і з її допомогою ввести команду kill для ліквідації “завислого” процесу. Гарантовано можна знищити процес за допомогою сигналу 9, наприклад: kill - 9 125 Звичайний користувач має право припиняти тільки процеси, запущені з його термінала. Для завершення процесу використовується системний виклик ехit(), при якому звільняються усі використовувані ресурси, зокрема такі, як пам'ять і структури таблиць ядра. Крім того, завершуються і процеси-нащадки, породжені даним процесом. Потім з пам'яті вилучаються сегменти коду і даних, а сам процес переходить у стан “зомбі” (у полі Stat такі процеси позначаються буквою “Z”). “Зомбі” не займає процесорного часу, але рядок у таблиці процесів залишається, і відповідні структури ядра не звільняються. Якщо батьківський процес з якоїсь причини завершиться раніше дочірнього, останній стає “сиротою” (orphaned process). “Осиротілий” “зомбі” на короткий час стає нащадком init, після чого вже остаточно “помирає”. Також, процес може впасти в “сон”, що не вдається перервати (у полі Stat це позначається буквою “D”). Процес, що знаходиться в такому стані, не реагує на системні запити і може бути знищений тільки перезавантаженням системи.

**Відповіді на контрольні запитання:**

***Готували матеріал студентки Мельнічук М.О, Усенко Б.О.***

1. Яке призначення директорії */proc* в системах Linux. Яку інформацію вона зберігає?

**/proc** - це не справжня файлова система. Вона віртуальна. Її основне *завдання* - отримання стану системи і частково виконання керуючих дій. Інформація про процеси зберігається в директорії / proc / N, де N - числовий ідентифікатор процесу. У цій директорії містяться різні псевдо-файли, які містять інформацію про сам процес і пов'язаному з ним оточенні.

Більше інформації читати [тут](https://mnorin.com/nemnogo-o-direktorii-proc-v-linux.html).

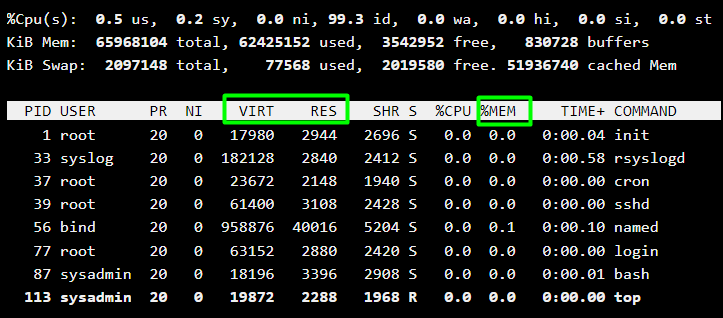
1. Як серед будь-яких трьох процесів динамічно визначати, який з них в поточний момент часу використовує найбільший обсяг пам'яті? Який відсоток пам’яті він споживає від загального обсягу?

Команда **top** чудово підходить для збору інформації про процеси, що виконуються в системі в режимі реального часу(запустивши цю команду можна динамічно визначити, яка з команд в поточний момент часу використовує найбільш обсяг пам’яті).

■ VIRT: загальна кількість віртуальної пам'яті, яку використовує процес

■ RES: обсяг фізичної пам'яті, який використовує процес

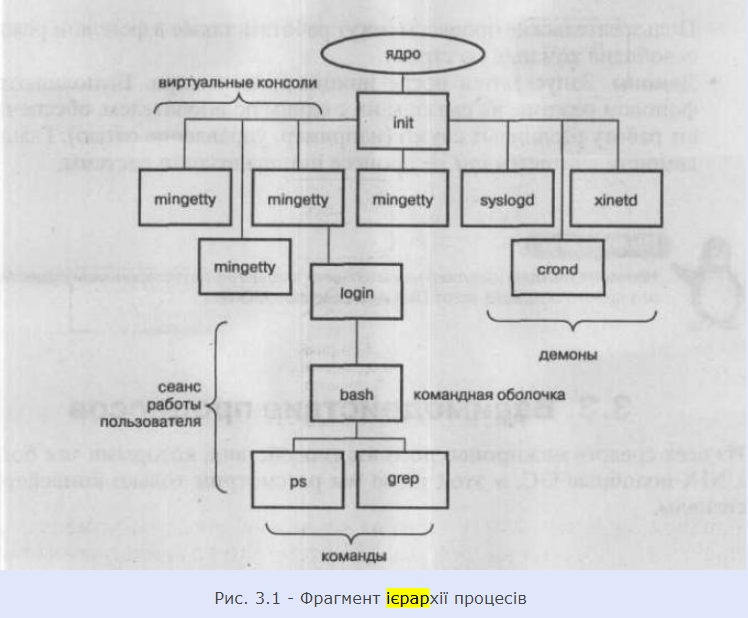
■ %MEM: частка доступної фізичної пам'яті, яку використовує процес



**Рисунок 1** – результат роботи команди top

1. Як отримати ієрархію батьківських процесів в системах Linux? Наведіть її структуру та охарактеризуйте.

 Linux реалізована чітка ієрархія процесів в системі. Кожен процес в системі має всього одного з батьків і може мати один або декілька породжених процесів.



**Рисунок 2** – фрагмент ієрархії процесів

Коли система завантажується, ядро ​​самостійно запускає кілька процесів. Найбільш важливий з них - демон **init**, ідентифікатор якого завжди дорівнює 1. Демон init відповідає за виконання сценаріїв запуску системи. Все процесу, крім тих, що створюються ядром, є нащадками демона init.

1. Чим відрізняється команда *top* від *ps*?

За замовчуванням команда **ps** показує лише процеси, які належать поточному користувачеві та запущені на поточному терміналі.

Команда **top** відображає інформацію про процес подібно до команди ps, але вона робить це в режимі реального часу. При виконанні **top** у верхній частині вікна відображається час, що минув з моменту запуску системи, число користувачів в системі, кількість запущених процесів і кількість процесів, що знаходяться в різних станах, дані про використання ЦП та пам'яті.

1. Які додаткові можливості реалізує *htop* в порівнянні з *top*?

**htop** показує динамічний список [системних процесів](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)), список зазвичай вирівнюється за використанням [процесора](https://uk.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BD%D1%82%D1%80%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80). На відміну від **top**, **htop** показує всі процеси в системі. Також показує час безперервної роботи, використання процесорів і пам'яті.

**htop** часто застосовується в тих випадках, коли інформації, що дається утилітою **top**, недостатньо, наприклад при пошуку витоків пам'яті в процесах.

1. Опишіть компоненти вашої мобільної ОС, які дозволяють здійснювати моніторинг запущених в системі процесів?

**Activity Monitor Touch** перевіритьпоточніпроцесина **Android**

1. Чи підтримує Ваша мобільна система термінальне керування роботою процесів. Якщо так, то опишіть як саме.

НІ, не підтримує

1. Чи можливо поставити сторонні програмні засоби, що дозволять організувати управління та моніторинг роботою процесесів у Вашому мобільному телефоні. Коротко опишіть їх.

**Activity Monitor Touch** перевіритьпоточніпроцесина **Android**

Виможетевідкритисписокзапущенихдодатківіпроцесівіінформаціюпрозавантаженістьпам**'**яті**.** Знизитинавантаженнянаресурсисвогосмартфонаможна**,** простозакрившидодатки**.** Дляцьогопотрібновсьоголишедвічіклікнутинакнопку **Home,** апотімутриматипалецьнаіконціпрограми**,** якепотрібнозакрити**.**

**Висновок**: В ході виконання лабораторної роботи нами було досліджено базові команди для управління процесами, отримано практичні навички роботи з командною оболонкоюBash.