Міністерство освіти і науки України

Національний технічний університет України  
"Київський політехнічний інститут імені Ігоря Сікорського"

Навчально-науковий інститут атомної та теплової енергетики

Кафедра цифрових технологій в енергетиці

Лабораторна робота №6 з дисципліни «Операційна система UNIX»

Робота з процесами ОС Linux

Група ТР-25

Колісніченко Анни

Перевірила д.т.н., проф. Левченко Л.О.

КИЇВ 2023

**Мета роботи**: набуття навичок управління процесами в оболонці Bash, опанування команд ps, top, pstree, bg, fg, nice, renice, kill, killall.

**Теоретична частина**

Процеси – це одна з найбільш фундаментальних абстракцій в системах UNIX після файлів. Від оптимального налаштування підсистеми управління процесами та числа одночасно виконуваних процесів залежить завантаження ресурсів процесора, що безпосередньо впливає на продуктивність системи в цілому. Задача ядра – управління процесами. Необхідно чітко розуміти відмінності між процесом і програмою.

Процес – це середовище виконання завдання (оточення), яке містить виконуваний код, системні дані, дані користувача і, а також набір додаткових ресурсів, отриманих під час виконання (ресурси пам'яті, можливість доступу до пристроїв введення/виведення та різних системних ресурсів, включаючи послуги ядра. Програма – це файл, який містить виконуваний код, дані для ініціалізації та дані користувача.

Процес можна розглядати як сукупність даних ядра системи (Kernel), необхідних для опису образу програми в пам'яті і управління її виконанням, або як програму в стадії її виконання, тому що усі програми Unix представлені у вигляді процесів. Ядро ОС обробляє переривання від пристроїв, виконує запити системних процесів та додатків користувача, розподіляє віртуальну пам'ять, створює і знищує процеси, забезпечує багатозадачність за допомогою перемикання між ними, містить драйвери пристроїв, обслуговує файлову систему. Процес складається з інструкцій, які виконуються процесором, даних та інформації про виконуване завдання, а саме: виділена пам'ять, відкриті файли і статус процесу.

**Завдання**

1. Ознайомитися з теоретичними матеріалом по лабораторній роботі. Набути навичок по роботі з процесами.
2. Опанувати команди для управління процесами.
3. Підготувати звіт для викладача про виконання лабораторної роботи і представити його.

**Хід роботи**

1. Виведемо на екран лістинг характеристик (у довгому і короткому форматах) процесів, ініціалізованих з нашого терміналу. Запишемо їх у файл. Проаналізуємо і пояснимо вміст кожного поля повідомлення.
2. Виведемо за допомогою команди ps, яка без будь-яких аргументів відображає процеси для поточної оболонки.

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

1. Команда ps з ключем -А дозволяє вивести дані про всі процеси у системі Linux.

Зображення, що містить текст, знімок екрана, меню

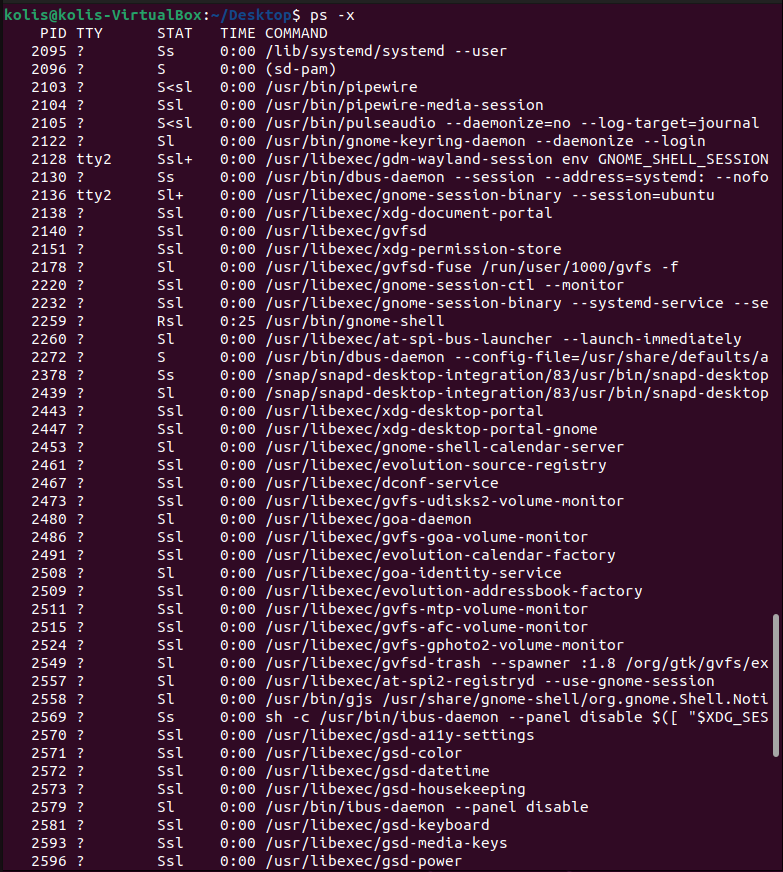
Автоматично згенерований опис

1. Команда ps з ключем au дозволяє отримати інформацію про процеси у форматі BSD (з розширеним виведенням параметрів).

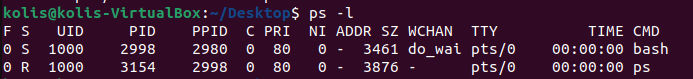
Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

1. Команда ps з ключем -х дозволяє відобразити запущені користувацькі процеси.



1. Команда ps з ключем -l виводить розширений список у форматі BSD.



1. Перенаправляємо цю інформацію до файлу за допомогою операції перенаправлення виводу. У результаті отримуємо файл з інформацією про процеси в форматі BSD.

Зображення, що містить текст, знімок екрана, Шрифт, Мультимедійне програмне забезпечення

Автоматично згенерований опис

Опис полів характеристик процесів:

F – flags

S – статус процесу (sleep, running, zombie і т.д.)

UID – ідентифікатор користувача

PID – ідентифікатор процесу

PPID – ідентифікатор батьківського процесу

C – використання процесору

PRI – пріоритет процесу

NI – значення “NICE”, яке впливає на пріоритет процесу

ADDR – адреса пам’яті процесу

SZ – використання віртуальної пам’яті

WCHAN – ім’я функції ядра, в якій спить процес

TTY – ідентифікатор терміналу

TIME – час ЦП

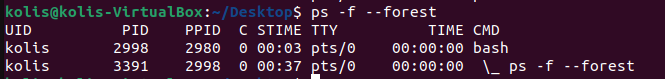
CMD – ім’я процесу

1. Виведемо всю ієрархію процесів поточної оболонки разом полями pid та ppid.
2. Використаємо ключ -f, який відобразить процеси разом із їх батьківськими ID.

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

1. Побудуємо дерево процесів, які визначені у попередньому пункті. Результат виконання виведемо на екран і допишемо в файл.



1. Запишемо у файл

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

1. Переглянемо список процесів нашого користувача.
2. Покажемо перелік процесів, які належать поточному користувачеві, за допомогою команди ps з параметрами -u для реального ідентифікатора користувача та -U для ефективного ідентифікатора користувача.

Зображення, що містить текст, знімок екрана, меню

Автоматично згенерований опис

1. Виведемо дерево процесів поточного користувача.

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, меню

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, меню

Автоматично згенерований опис

1. Переглянемо список сигналів вашого користувача. Запишемо у окремий файл.
2. Для створення переліку сигналів ми скористаємося командою kill з параметром -l.

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

1. Запишемо цю інформацію в окремий файл.

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

1. За допомогою команди history виведемо команди, які ми використовували.

Зображення, що містить текст, знімок екрана

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, меню

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

**Висновок:** Під час виконання лабораторної роботи №6 я набула навичок управління процесами в оболонці Bash, а також опанувала команди ps, top, pstree, bg, fg, nice, renice, kill, killall. У результаті я вивела на екран лістинг характеристик процесів, записала їх у файл, і пояснила вміст кожного поля повідомлення. А також вивела всю ієрархію процесів, побудувала дерево процесів, переглянула список процесів користувача, вивела його у вигляді дерева, переглянула список сигналів, записала його у окремий файл, і за допомогою команди history вивела усі команди, які були використані.