**МІНІСТЕРСТВО ОСВІТИ І НАУКИ  УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

**ІКНІ**

Кафедра **ПЗ**

**ЗВІТ**

до лабораторної роботи № 2

**На тему:** *“ Ознайомлення та керування процесами в операційних системах для персонального комп’ютера. Linux та MacOS”*

**З дисципліни:** *“Операційні системи”*

**Лектор:**

ст. викладач каф. ПЗ

Грицай О.Д.

**Виконала:**

ст. гр. ПЗ-26

Матолінець Л.А.

**Прийняв:**

доцент каф. ПЗ

Горечко О.М.

« \_\_\_\_ » \_\_\_\_\_\_\_\_  2022 р.

∑= \_\_\_\_                              .

Львів – 2022

**Тема роботи:** Ознайомлення та керування процесами в операційних системах для персонального комп’ютера. Linux та MacOS.

**Мета роботи:** Ознайомитися з процесами та потоками в ОС Linux та MacOS. Навчитися працювати із системними утилітами, що дають можливість отримувати інформацію про процеси, потоки, використовувану ними пам’ять, та іншу необхідну інформацію.

**Індивідуальне завдання**

1. Встановити операційні системи Linux та MacOS

2. За допомого консольних засобів ОС Linux отримати повну інформацію про процеси.

3. За допомогою утиліт top, htop, qps, System Monitor отримати повну інформацію про процеси в ОС Linux та MacOS.

4. Використовуючи консольні засоби ОС Linux та утиліти змінити пріоритет виконання процесу.

5. Використовуючи консольні засоби ОС Linux та сторонні утиліти змінити стан виконання процесу, завершити виконання заданого процесу.

6. Скомпілювати файл main.cpp представлений у лабораторній роботі № 1 (на MacOS і Linux можна командою: g++ main.cpp -pthread) і запустити виконуваний файл на різній кількості активних процесорів (ядер). Знайти для даної програми величини A , S , p при різних вхідних значеннях величини n . Порівняти результати для різних операційних систем.

7. Результати лабораторної роботи оформити у звіт, у висновку надати порівняння моніторингу процесів у різних системах різними утилітами, відповідно до індивідуального варіанту.

1) Завантаження файлів за URL адресою з використанням wget\curl

**Теоретичні відомості**

Процес - це об'єкт ОС Linux, який складається з адресного простору пам'яті і набору структур даних. По суті, процес це запущена програма або служба.

Кожен запущений процес в ОС Linux може породити додаткові процеси. Процес, що запустив новий процес називається продительским процесом. Новий процес по відношенню до створив його процесу називається дочірнім.

 Ідентифікатор процесу (PID) . Кожен процес в системі має унікальний ідентифікатор. Кожен новий запущений процес отримує номер на одиницю більше ппредыдущего.

Ідентифікатор батьківського процесу (PPID). Даний атрибут процес отримує під час свого запуску і використовується для отримання статусу батьківського процесу.

Для управління процесом (наприклад, kill) використовуються реальні ідентифікатори. Всі ідентифікатори передаються від батьківського до дочірнього процесу.  
Для перегляду даних атрибутів можна скористатися командою [ps](https://life-prog.ru/.man/_ps.html), задавши бажаний формат відображення колонок.

Стан процесу. В ОС Linux кожен процес обов'язково знаходиться в одному з перерахованих нижче станів і може бути переведений з одного стану в інше системою або командами користувача. Розрізняють наступне стану процесів:

TASK\_RUNNING - процес готовий до виконання або виконується (runnable). Позначається символом R.

TASK\_INTERRUPTIBLE - чекаючий процес (sleeping). Цей стан означає, що процес ініціалізував виконання  
якої системної операції і чекає її завершення. До таких операцій відносяться введення/висновок, завершення дочірнього процесу  
пі т.д. Процеси з таким станом позначаються символом S.

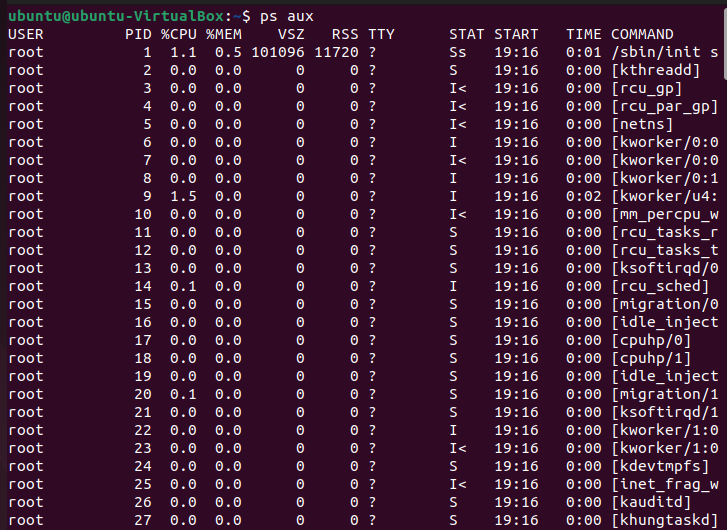
TASK\_STOPPED - виконання процесу зупинено (stopping). Будь-який процес можна зупинити. Це може робити як система,  
ptak і користувач. Стан такого процесу позначається символом Т.

TASK\_ZOMBIE - завершився процес (zombie). Процеси даного стану виникають у разі, коли батьківський процес  
не чекаючи завершення дочірнього процесу, продовжує паралельно працювати. Процеси з таким станом позначаються символом Z.  
пЗавершившиеся процеси більше не виконуються системою, але далі продовжують споживати її не обчислювальні ресурси.

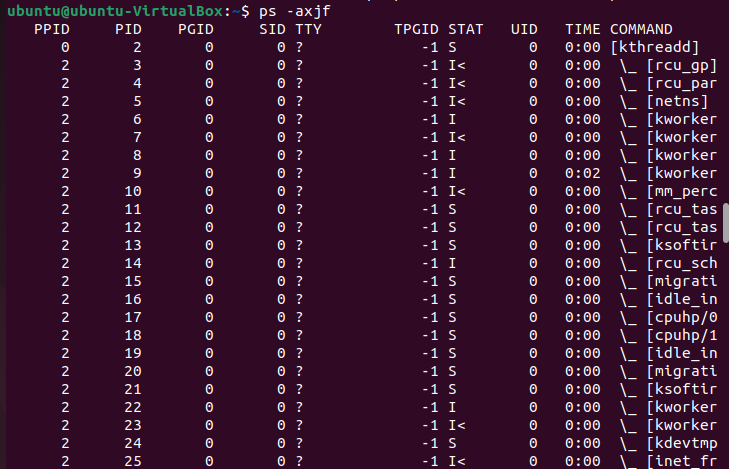
TASK\_UNINTERRUPTIBLE -неперерваний процес (uninterruptible). Процеси в цьому стані очікують завершення операції  
введення - виведення з прямим доступом в пам'ять. Такий процес не можна завершити, поки не завершиться операція введення/виводу.  
Процеси з таким станом позначаються символом D. Стан аналогічно TASK\_INTERRUPTIBLE, за винятком того, що процес  
не відновлює виконання при отриманні сигналу. Використовується у випадку, коли процес повинен чекати безперервно або коли очікується, що певна подія може виникати досить часто. Так як завдання в цьому стані не відповідає  
пна сигнали, TASK\_UNINTERRUPTIBLE використовується менш часто, ніж TASK\_INTERRUPTIBLE.

**Хід роботи**

1. Після встановлення Ubuntu, отримую повну інформацію про процеси за допомогою консольних засобів: через команди ps aux, ps –axjf та інші;

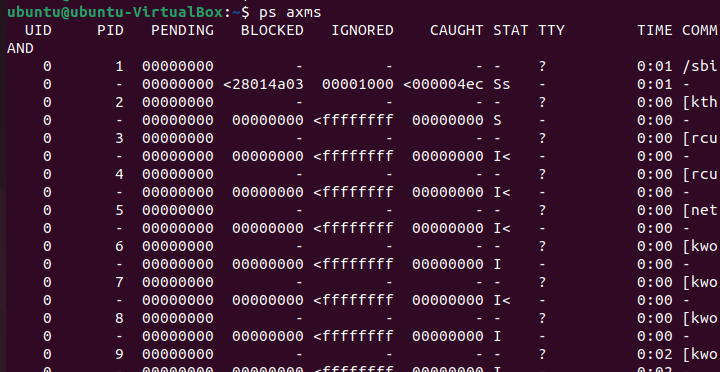


*Рис. 1 Інформація про процеси*



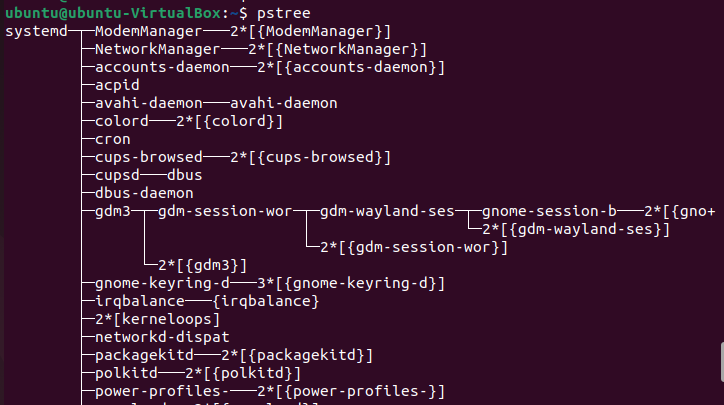
*Рис. 2 Інформація про процеси*

Також отримую інформацію про потоки:

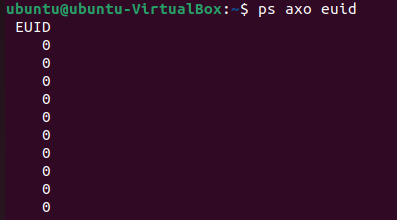


*Рис. 3 Інформація про процеси та потоки*

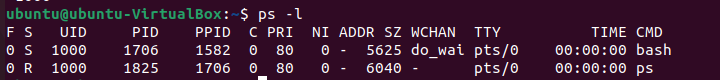
Отримую інформацію про дерева процесів :



*Рис. 4 Інформація про дерева процесів*



*Рис. 5 Effective UID*



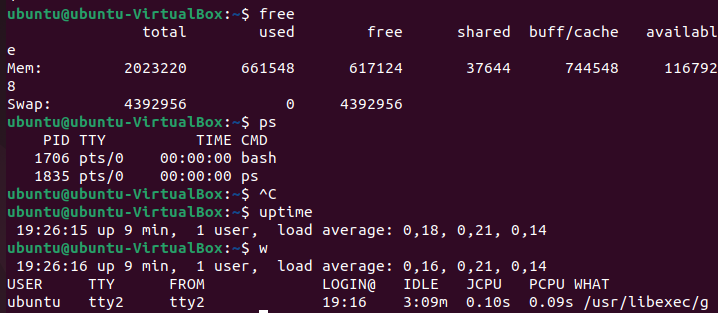
*Рис. 6 Інформація про процеси*

Отримую лічильний дескрипторів для окремого процесу:



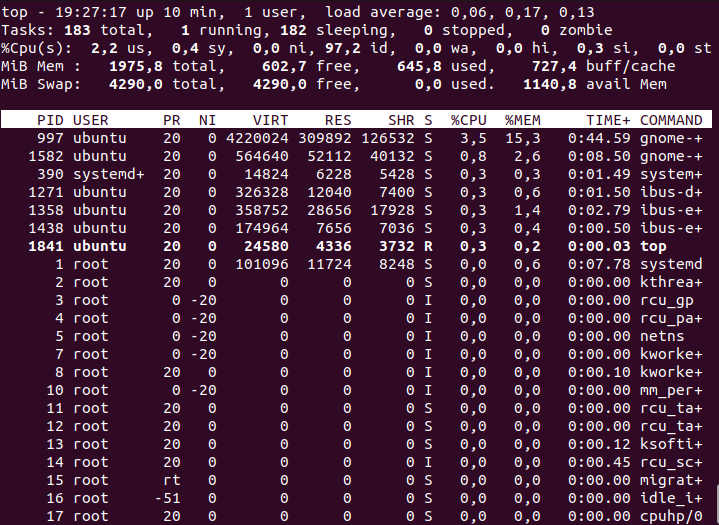
*Рис. 7 Лічильник дескрипторів*

Дані про пам’ять тощо:

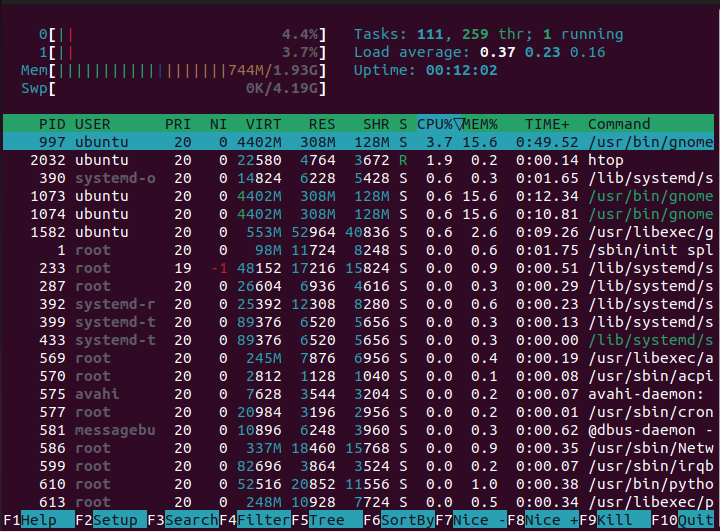


*Рис. 8 Дані про пам’ять*

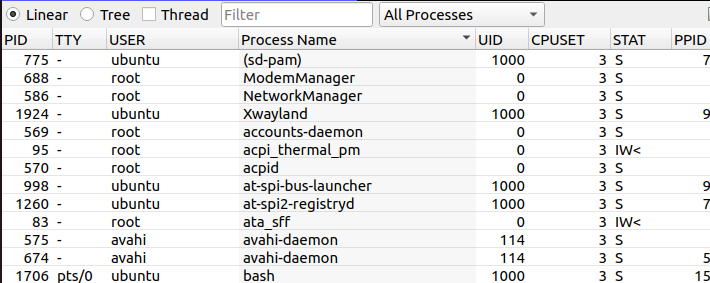
1. За допомогою утиліти top, htop, qps та System Monitor отримую повну інформацію про процеси:



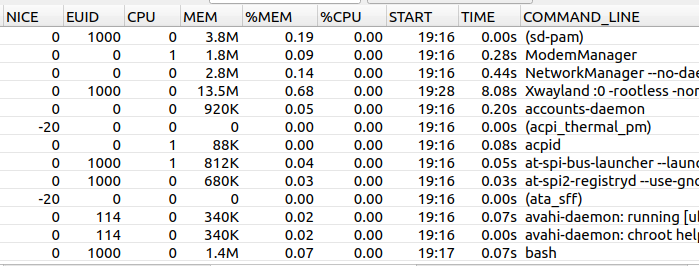
*Рис. 9 Отримання даних через утиліту top*



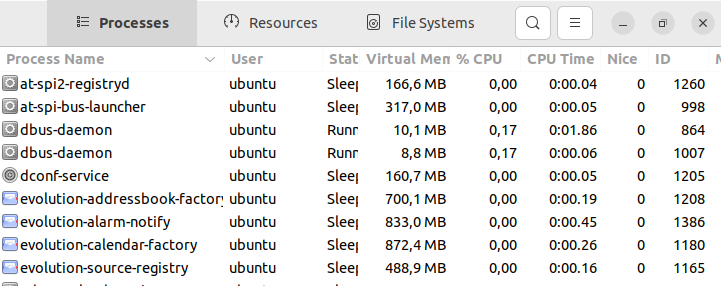
*Рис. 10 Отримання даних через утиліту htop*



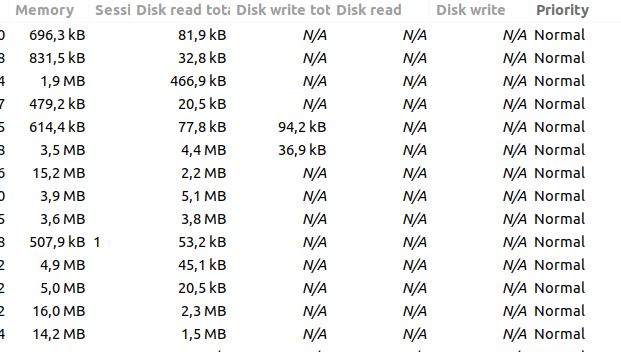
*Рис. 11 Отримання даних через утиліту QPS*

****

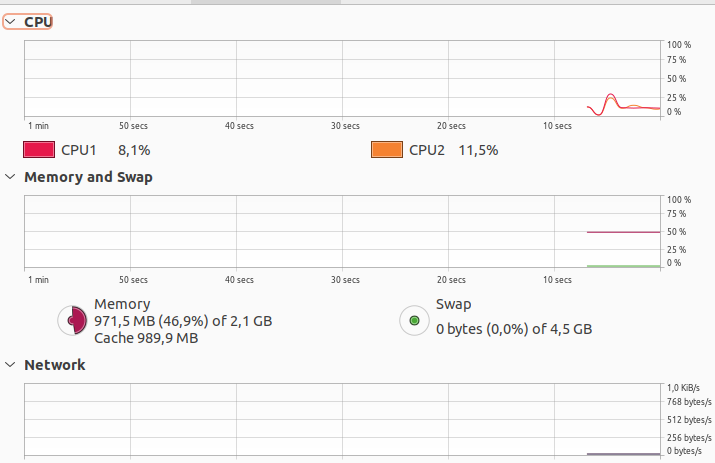
*Рис. 12 Отримання даних через утиліту QPS*

****

*Рис. 13 Отримання даних через утиліту System Monitor*

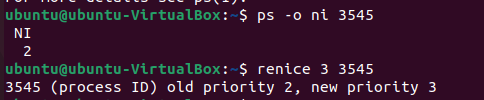
****

*Рис. 14 Отримання даних через утиліту System Monitor*

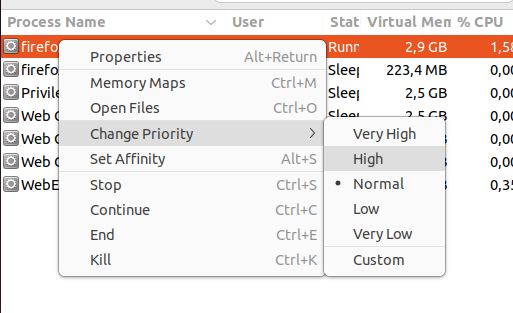
****

*Рис. 15 Отримання даних через утиліту System Monitor*

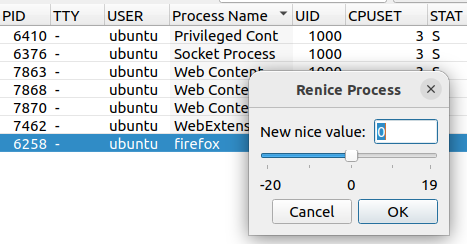
1. Змінюю пріоритет виконання процесу за допомогою консольних засобів та утиліт:

****

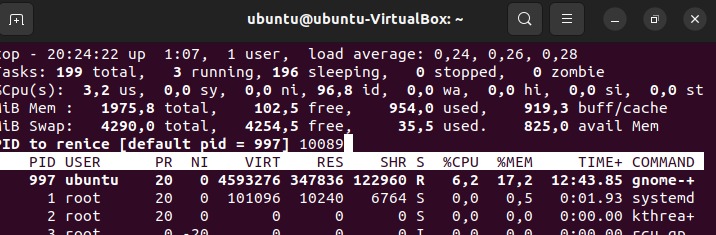
*Рис. 15 Зміна пріоритету процесу*



*Рис. 16 Зміна пріоритету процесу у System Monitor*

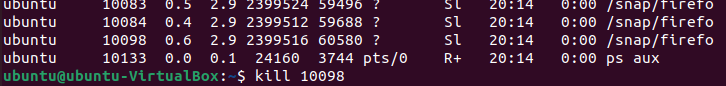


*Рис. 17 Зміна пріоритету процесу у QPS*

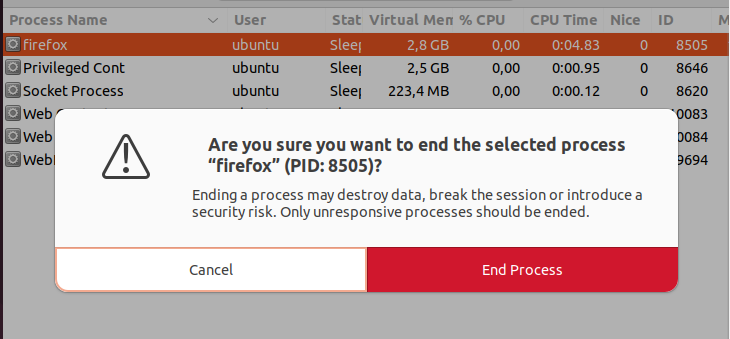


*Рис. 18 Зміна пріоритету процесу використовуючи top*

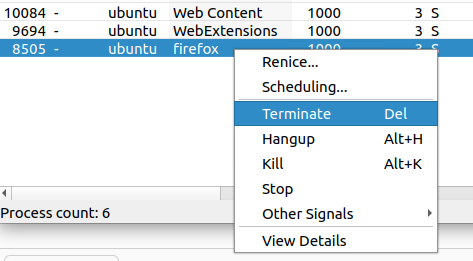
1. Завершую процеси за допомогою консольних засобів та утиліт:



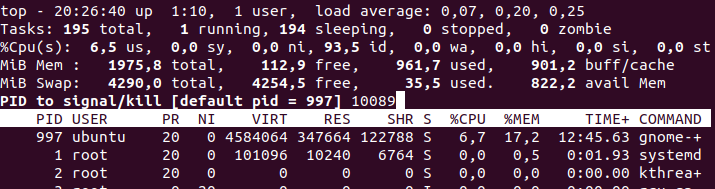
*Рис. 19 Завершення виконання процесу*



*Рис. 20 Завершення виконання процесу у System Monitor*

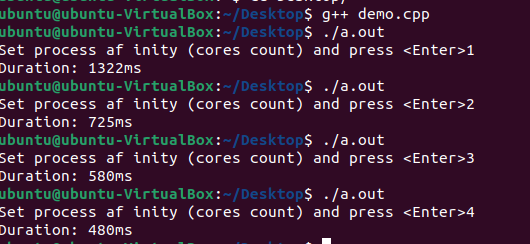


*Рис. 21 Завершення виконання процесу у QPS*



*Рис. 22 Завершення виконання процесу використовуючи top*

1. Компілюю файл main.cpp, запускаючи його на різній кількості ядер. Оскільки для Ubuntu я виділила 4 ядра, то максимальне n = 4.



*Рис. 23 Результати виконання програми*

При запуску з CPU 0, T1 = 1322мс . Для розрахунків використовувала ці формули:

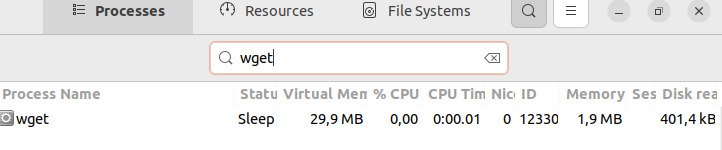
A = T1/Tn; A = S, p + (1-p)/n = 1/A => p = (n/A - 1) / (n-1);

Спочатку запускала файл на CPU 1, CPU 2. При n = 2, T2 = 725 ms , A = 1.82, p = 0.08;

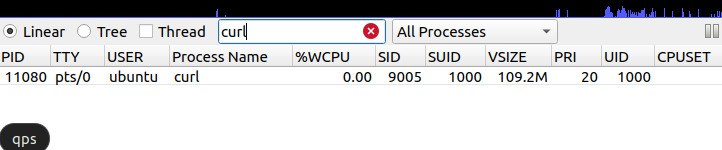
Запуск на CPU 1, CPU 2, CPU 3. При n = 3, T3 = 580 ms, A = 2.278, p = 0.1 ;

Запуск на CPU 1, CPU 2, CPU 3, CPU 4. При n = 4, T = 480 ms, A = 2.75 , p = 0.11;

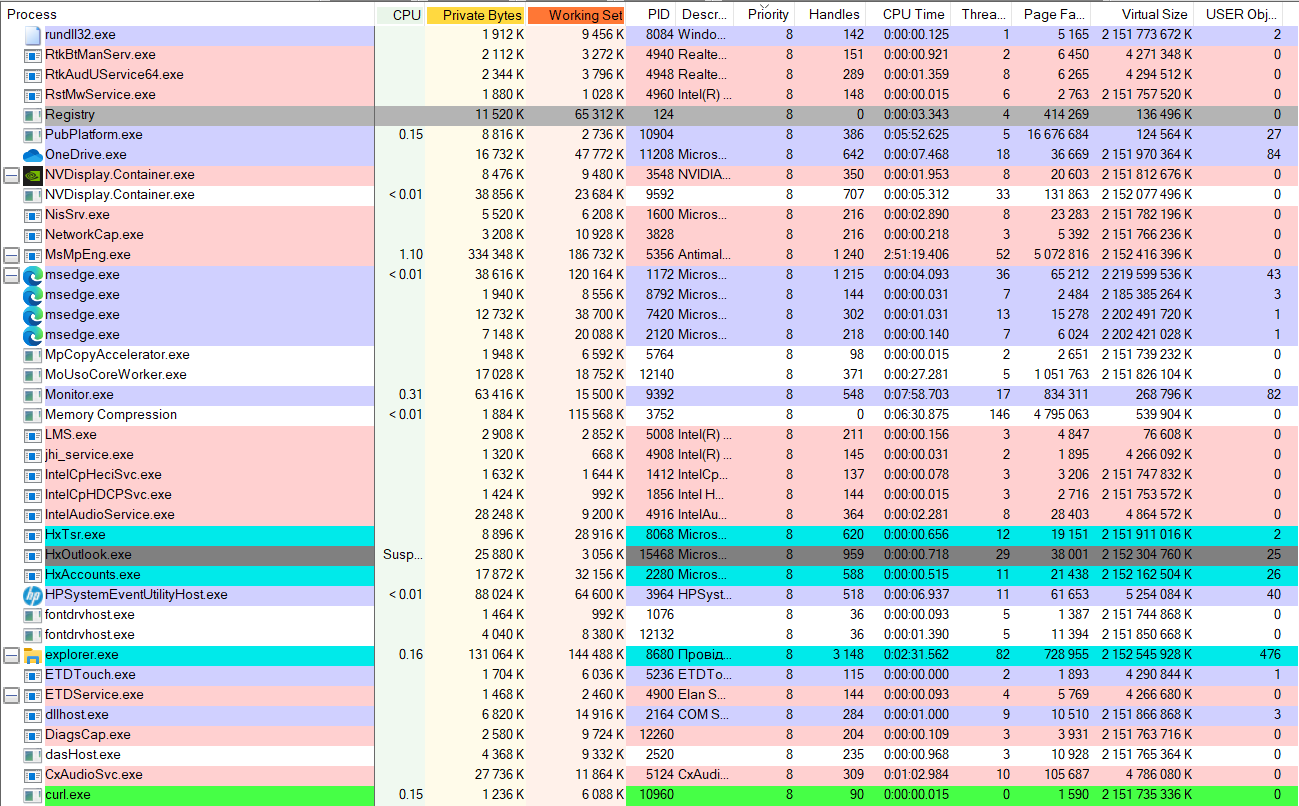
1. Завантаження з використанням wget, моніторинг відбувався у System Monitor, QPS та Process Explorer для порівняння роботи в різних системах.



*Рис. 24 Моніторинг завантажування у System Explorer*



*Рис. 25 Моніторинг завантажування у QPS*



*Рис. 26 Моніторинг завантажування у Process Explorer*

**Висновок**

На цій лабораторній роботі я познайомилася та попрацювала з консольними засобами та утилітами в дистрибутиві Linux Ubuntu, зокрема з QPS, System Monitor, top, htop та іншими. За допомогою них я провела моніторинг процесів, отримала дані про пам’ять та загальні властивості процесів. Також виконала індивідуальне завдання, де здійснювала завантаження файлу через команди curl / wget, які при запуску створюють процес curl.exe та wget.exe відповідно. Впродовж роботи, я побачила, що утиліта QPS дає більше можливостей отримати дані про процеси, ніж System Monitor, в обох утилітах зручно відслідковувати процеси сортуючи за певними категоріями. Також я порівнювала ці утиліти з Process Explorer у Windows, який є, на мою думку, найкращим навіть у порівнянні з Linux утилітами, бо є можливість працювати з потоками, є кольорове маркування процесів.