# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Національний університет “Львівська політехніка”**



**Інститут післядипломної освіти**

**ЗВІТ**

**Про виконання лабораторної роботи №1**

**«Ознайомлення з процесами в ОС Windows»**

**з дисципліни «Операційні системи»**

Виконав:

слухач групи ПЗС-11

Гринчук Тарас

Прийняв:

доц. Яковина В.С.

« »\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2014 р.

∑ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

ЛЬВІВ – 2014

**Тема роботи**: Ознайомлення з процесами в ОС Windows.

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

## 1. Теоретичні відомості

Процес - це програма, завантажена в пам'ять разом з усіма приналежними до неї ресурсами. Кожен процес має свій власний віртуальний адресний простір (4Gb). Процес складається з коду, даних та інших системних ресурсів, таких як відкриті файли, канали (pipes), синхронізуючі об'єкти. Однак процес - статичний об'єкт, який сам по собі дії не виробляє.

  Потік (thread) - це базовий об'єкт, якому операційна система розприділяє час центрального процесора. Потік виконує команди програми з урахуванням заданого йому маршруту. Кожен процес являє собою один початковий потік, який іноді називають первинним потоком.

Первинний потік здатний створити вторинні потоки. Всі потоки, приналежні одному процесу, мають спільний доступ до його ресурсів. Всі вони працюють під управлінням команд однієї і тієї ж програми, звертаються до одних і тих же глобальних змінних, записують інформацію в одну і ту ж область пам'яті і мають доступ до одних і тих же об'єктів.

В цілому слід зазначити, що програма може виконувати поставлені завдання і без організації потоків, проте в даному випадку для запуску «дочірнього» процесу необхідно тимчасово припиняти основний процес, що приводить до уповільнення виконання програми в цілому. Додаткові потоки створюють в першу чергу в тому випадку, коли програма повинна виконувати асинхронні операції, працює одночасно з декількома вікнами.

На системному рівні кожен потік являє собою об'єкт, створений системним менеджером об'єктів. Аналогічно іншим системним об'єктам, потік містить дані (атрибути) і методи (функції). Схематично об'єкт-потік може бути представлений у вигляді, представленому на рис. 1.1. Для більшості методів потоку є відповідні API - функції Win32. Windows захищає свої внутрішні структури від прямого втручання користувача програм. На відміну від більш привілейованих програм, що функціонують на рівні ядра операційної системи, користувацькі не можуть прямо аналізувати або змінювати параметри системних об'єктів. Всі операції з ними виконуються за допомогою функцій Win32 API. Windows надає описувач (дескриптор), що ідентифікує об'єкт. При виконанні операцій з об'єктом його дескриптор передається в якості аргументу однієї з API - функцій. Свої дескриптори мають потоки, процеси, семафори, файли та інші об'єкти. Внутрішня структура об'єктів доступна тільки менеджеру об'єктів. Функція, що створює потік, повертає дескриптор нового об'єкта. За допомогою цього дескриптора можна виконати наступні операції:

- підвищити або знизити плановий пріоритет потоку;

- призупинити потік і відновити його виконання;

- припинити виконання потоку;

- визначити код завершення потоку.

В ОС Windows потоки, процеси, семафори і виключаючі семафори можуть мати кілька різних дескрипторів . Завершивши роботу з об'єктом, необхідно викликати функцію CloseHandle(), яка, закривши останній дескриптор, сама знищить об'єкт. Загалом у Windows кожен процес не може одночасно підтримувати більш 65536 відкритих дескрипторів.

Робота з потоками не зводиться тільки до їх запуску і зупинці, необхідно забезпечити спільне функціонування потоків. Для організації ефективної взаємодії між декількома потоками необхідно робити контроль за їх часовими параметрами. Контроль здійснюється:

  - Встановленням пріоритетів;

  - Синхронізацією.



Рис 1.1. Схема об’єкту потоку

Пріоритет потоку визначає , наскільки часто даний потік отримує доступ до центрального процесора. Синхронізація регулює порядок звернення потоків до загальних ресурсів. Коли системна програма-планувальник зупиняє один потік і шукає інший, який повинен бути запущений наступним, вона віддає перевагу потокам, які мають найбільш високий пріоритет. Оброблювачі системних переривань завжди мають вищий пріоритет у порівнянні з користувача процесами. Кожному процесу притаманний власний пріоритет. Базовий плановий пріоритет потоку визначається на основі пріоритету процесу, який є власником цього потоку. Усього розрізняють 32 рівні пріоритету: від 0 до 31. При цьому пріоритети рівня від 0 до 15 називаються змінними пріоритетами, а від 16 до 31 - фіксованими пріоритетами.

## 2. Хід роботи

1. За допомогою «Диспетчера задач» отримати повну інформацію про процеси: ідентифікатор процесу, завантаження ЦП (центрального процесора), час ЦП, пам'ять-використання, пам'ять-зміни, пам'ять-максимум, помилок сторінки, об’єкти USER, код сеансу, об’єм віртуальної пам’яті, лічильник дескрипторів, лічильник потоків.

Запустимо «Диспетчер задач» комбінацією клавіш «Ctrl+Alt+Delete» та перейдемо на вкладку «Процессы». Для того, щоб вивести інформацію по всіх параметрах процесу, що нас цікавлять - виконаємо команду: «Вид –> Выбрать столбцы» та виберемо відповідні стовбці таблиці (рис. 2.1). В результаті отримаємо таблицю, яка зображена на рис. 2.2.

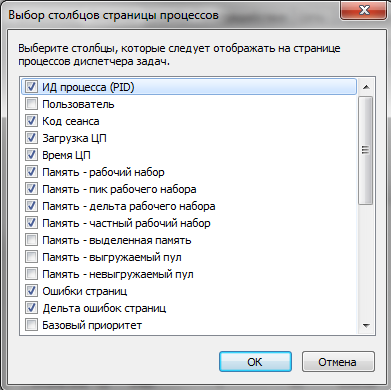


Рис. 2.1. Меню «Вид –> Выбрать столбцы»

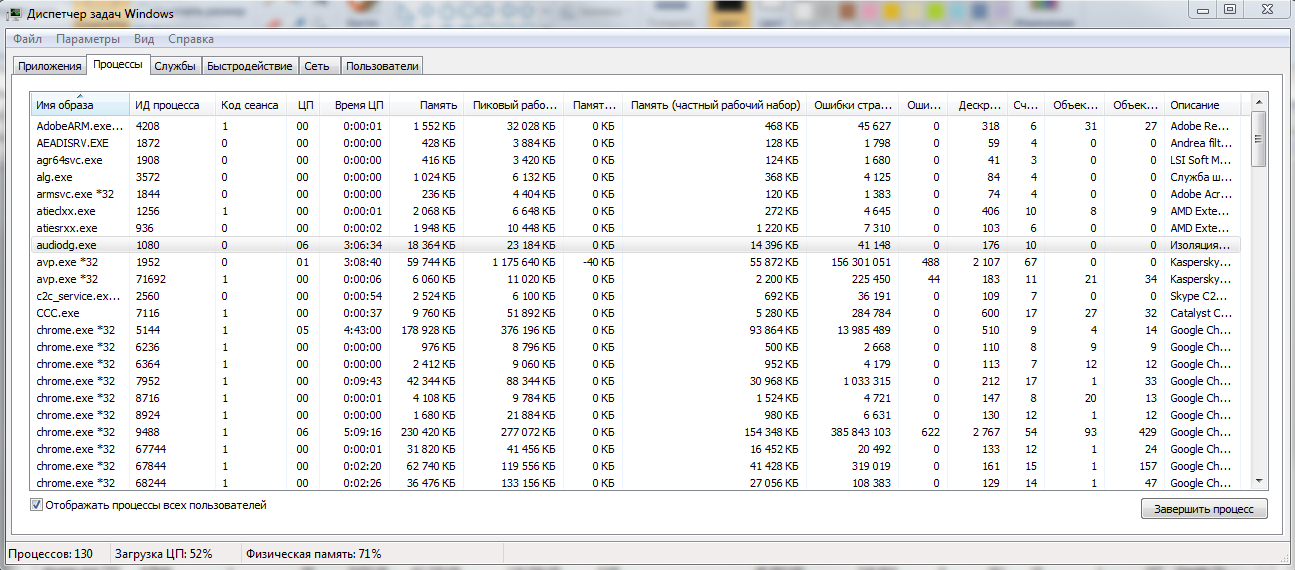


Рис. 2.2. Закладка «Процесы» Диспетчера задач

Наведемо коротко, опис кожного стовбчика:

* **Ідентифікатор процесу ( PID ) -** число , унікально ідентифікує що виконується процес .
* **Код сеансу -** число, що ідентифікує власника процесу. Якщо кілька користувачів здійснили вхід в систему , кожному користувачеві привласнюється свій унікальний код сеансу.
* **Завантаження ЦП -** відсоток часу, протягом якого процесом використовувався центральний процесор ( ЦПУ) з часу останнього оновлення (відображається як ЦПУ в заголовку стовпця).
* **Час ЦП -** загальний час процесора в секундах , виділене процесу з початку його роботи. Об'єм пам'яті в приватному робочому наборі плюс обсяг пам'яті, використану процесом, яку не можна використовувати спільно з іншими процесами.
* **Пам'ять-пік робочого набору -** максимальний об'єм пам'яті робочого набору, використовуваної процесом.
* **Пам'ять-дельта робочого набору -** зміна обсягу пам'яті робочого набору, використаної процесом .
* **Пам'ять - виділена пам'ять** **- о**бсяг віртуальної пам'яті, виділеної процесу.
* **Помилки сторінок** - кількість спроб читання даних процесу з диска в разі, якщо його не вдалося виявити в пам'яті. Кількість помилок сторінки накопичується з моменту запуску процесу.
* **Дельта помилок сторінок -** зміна кількості помилок сторінки з часу останнього оновлення.
* **Дескриптори -** кількість дескрипторів об'єктів в таблиці об'єктів процесу.
* **Лічильник потоків** **-** кількість виконуваних гілок процесу.
* **Об'єкти USER -** кількість об'єктів, що використовуються процесом в даний момент. Об'єкти USER - об'єкти диспетчера вікон, включаючи вікна, меню, курсори, значки, обробники, прискорювачі, монітори, розкладки клавіатури, а також інші внутрішні об'єкти.

1. За допомогою утиліти Process Explorer отримати повну інформацію про процеси: ідентифікатор процесу, завантаження ЦП (центрального процесора), час ЦП, пам'ять-використання, пам'ять-зміни, пам'ять-максимум, помилок сторінки, об’єкти USER, код сеансу, об’єм віртуальної пам’яті, лічильник дескрипторів, лічильник потоків.

Запустимо програму «Process Explorer». Для того, щоб вивести інформацію по всіх параметрах процесу, що нас цікавлять - виконаємо команду: «View –> Set columns» та виберемо відповідні стовбці таблиці процесів на закладках «Process image» (рис. 2.3), «Process perfomance» (рис. 2.4), «Process memory» (рис. 2.5) та інших. В результаті отримаємо таблицю, яка зображена на рис. 2.6.

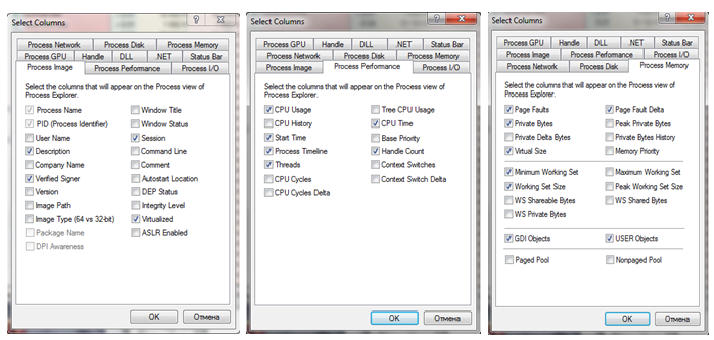


Рис. 2.3. Process image Рис. 2.4. Process performance Рис. 2.5. Process memory

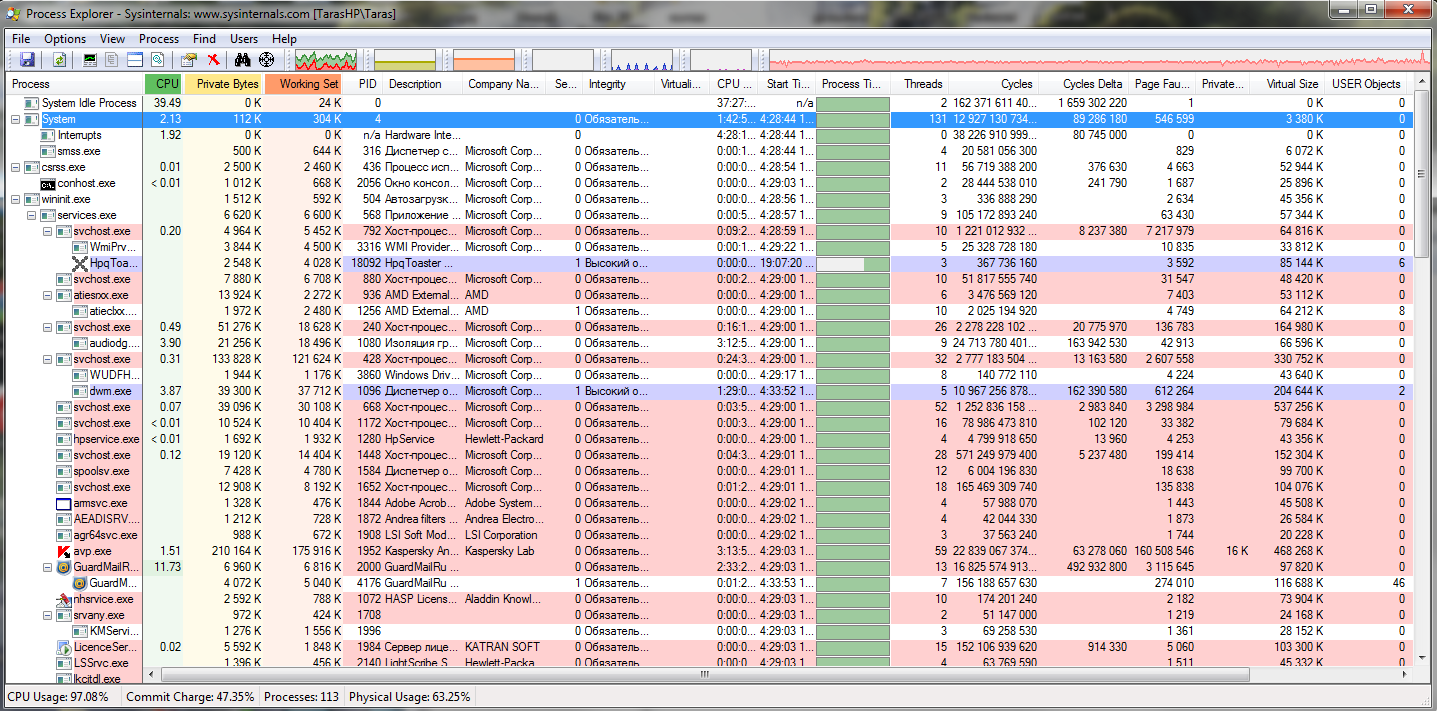


Рис. 2.6. Вигляд Process Explorer

після вибору стовбців

1. Використовуючи «Диспетчер задач» та Process Explorer змінити пріоритет будь-якого процесу, від низького до «реального часу». Виконати завершення процесу, використовуючи попередні утиліти.

Для того, щоб в «диспетчері задач» змінити пріоритет процесу – викличемо контекстне меню, потрібного нам процесу та в підменю «Приоритет» встановимо перемикач в положення: «Реального времени» (рис. 2.7).

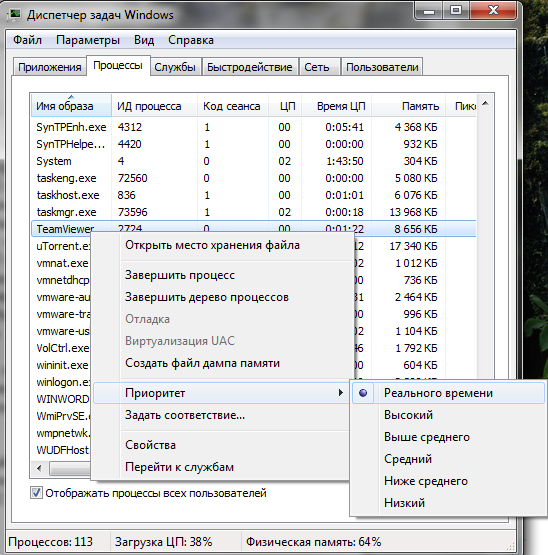


Рис. 2.7. Зміна пріоритету процесу

у диспетчері задач

Для того, щоб в «диспетчері задач» завершити процес – викличемо контекстне меню, потрібного нам процесу та виберемо команду «Завершить процесс» (рис. 2.8), у вікні підтвердження, яке з’явиться після цього, підтвердимо це рішення.

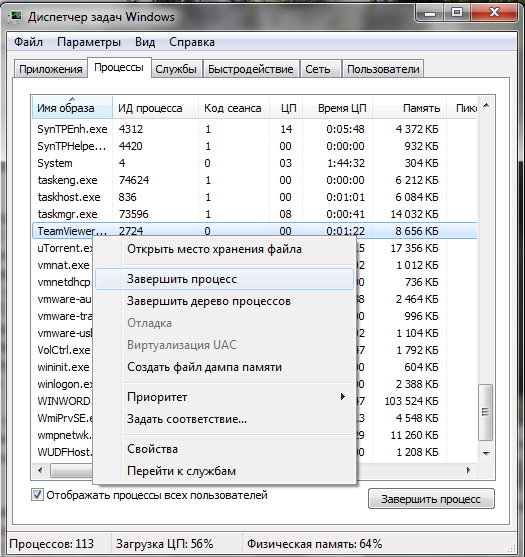


Рис. 2.8. Завершення процесу

у диспетчері задач

Для того, щоб в «Process Explorer» змінити пріоритет процесу – викличемо контекстне меню, потрібного нам процесу та в підменю «Set Priority» встановимо перемикач в положення: «Realtime: 24» (рис. 2.9).

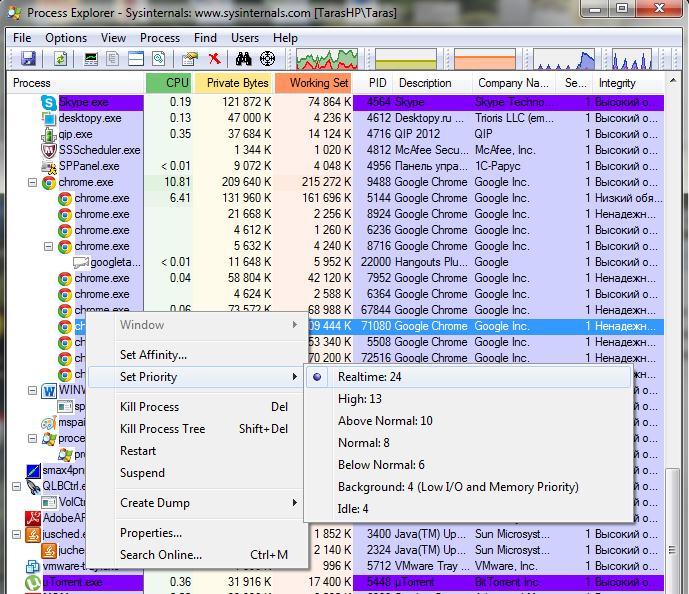
****

Рис. 2.9. Зміна пріоритету процесу

Process Explorer

Для того, щоб в «Process Explorer» завершити процес – викличемо контекстне меню, потрібного нам процесу та виберемо команду «Kill Pocess» (рис. 2.10), у вікні підтвердження, яке з’явиться після цього, підтвердимо це рішення.

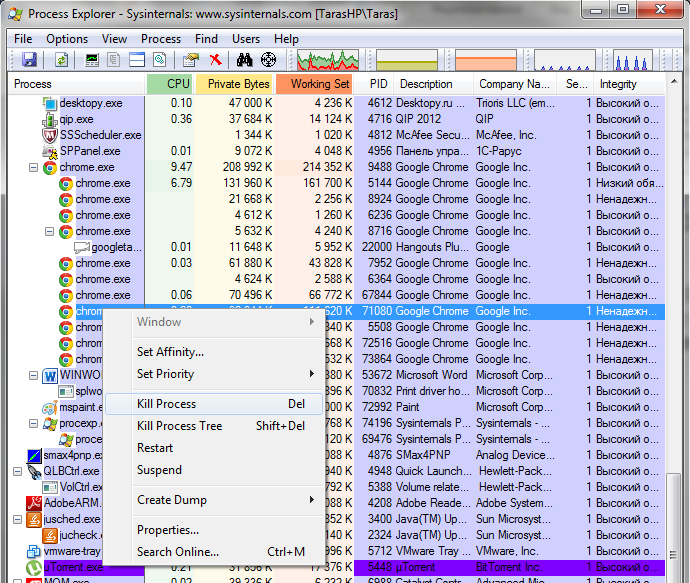


Рис. 2.10. Завершення процесу

у Process Explorer

1. Використовуючи «Диспетчер задач» та Process Explorer задати відповідність виконання процесів на окремих ядрах центрального процесора.

Задамо виконання процесу explorer.exe на ЦП 0, за допомогою «Диспетчера задач»: для цього активуємо цей процес у списку та викличемо контекстне меню. Виберемо пункт: «Задать соответствие…» (рис. 2.11) та у вікні, яке з’явиться встановимо прапорець тільки навпроти поля «ЦП 0» (рис. 2.12).

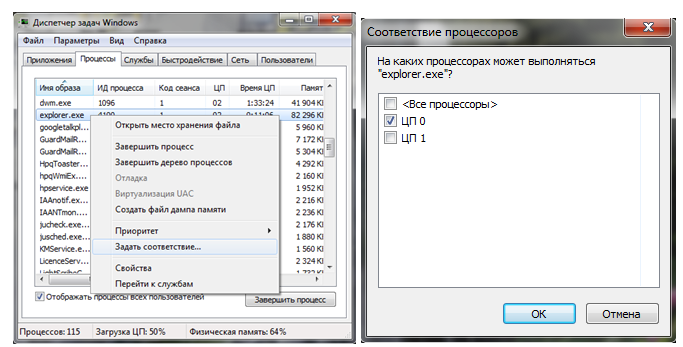


Рис. 2.11 Рис. 2.12

Задамо виконання процесу uTorrent.exe на ЦП 1, за допомогою «Process Explorer» : для цього активуємо цей процес у списку та викличемо контекстне меню. Виберемо пункт: «Set Affinity…» (рис. 2.13) та у вікні, яке з’явиться встановимо прапорець тільки навпроти поля «CPU 1» (рис. 2.14).

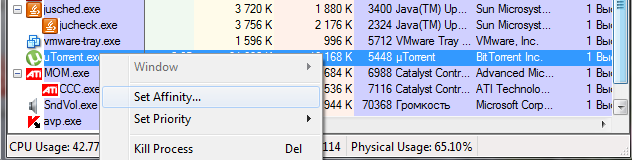


Рис. 2.13

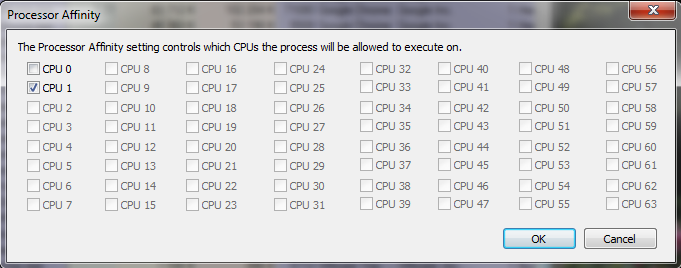


Рис. 2.14

## ВИСНОВКИ

На даній лабораторній роботі я дослідив роботу Диспетчера задач Windows 7 та утиліти Process Explorer, зокрема виводити потрібну мені інформацію по процесах, змінювати пріоритети процесів та задавати порядок виконання їх на ядрах центрального процесора.