

2024

Лабораторна робота 4

Java/C/C++/C#/JS :
&& is and, || is or
Python:



Виконав: Федоренко Іван, група 556-ІПЗ

Перевірила: Денисюк Наталія

06.04.2024

C:\Users\Ivan> & C:\msys32\perl\bin\python.exe C:\Users\Ivan\Downloads\run.py

| Image Name | PID | Session Name | Session# | Mem Usage |
|---------------------|------|--------------|----------|-----------|
| System Idle Process | 0 | Services | 0 | 8 K |
| System | 4 | Services | 0 | 144 K |
| Secure System | 108 | Services | 0 | 31я624 K |
| Registry | 144 | Services | 0 | 55я188 K |
| smss.exe | 608 | Services | 0 | 1я092 K |
| csrss.exe | 960 | Services | 0 | 5я084 K |
| wininit.exe | 664 | Services | 0 | 6я060 K |
| services.exe | 1100 | Services | 0 | 10я336 K |
| LsaIso.exe | 1116 | Services | 0 | 3я360 K |
| lsass.exe | 1136 | Services | 0 | 26я276 K |
| svchost.exe | 1268 | Services | 0 | 32я136 K |
| WUDFHost.exe | 1296 | Services | 0 | 6я192 K |
| fontdrvhost.exe | 1316 | Services | 0 | 3я180 K |
| svchost.exe | 1436 | Services | 0 | 17я040 K |
| svchost.exe | 1496 | Services | 0 | 9я172 K |

```

3  # Функція для виведення всіх запущених процесів
4  def print_running_processes():
5      command = 'tasklist'
6      # command = 'ps aux' # for linux
7      result = subprocess.run(command, text=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
8      if result.returncode == 0:
9          print("Запущені процеси:")
10         print(result.stdout)
11     else:
12         print("Помилка при виведенні процесів:", result.stderr)

```

| | | | | |
|--------------------------|-------|----------|---|-----------|
| svchost.exe | 12032 | Services | 0 | 10я300 K |
| Microsoft.SharePoint.exe | 10908 | Console | 5 | 31я876 K |
| VSSVC.exe | 12144 | Services | 0 | 9я624 K |
| svchost.exe | 15900 | Services | 0 | 8я776 K |
| WmiPrvSE.exe | 12148 | Services | 0 | 11я372 K |
| svchost.exe | 12528 | Services | 0 | 7я828 K |
| svchost.exe | 1044 | Services | 0 | 10я484 K |
| smartscreen.exe | 17152 | Console | 5 | 10я764 K |
| chrome.exe | 4196 | Console | 5 | 29я392 K |
| Code.exe | 7468 | Console | 5 | 97я708 K |
| Code.exe | 12248 | Console | 5 | 28я076 K |
| Code.exe | 5828 | Console | 5 | 176я432 K |

Image Name: Цей стовпчик показує ім'я виконуваного файлу процесу.

PID (Process ID): Це унікальний ідентифікатор процесу, який використовується операційною системою для ідентифікації кожного процесу.

Session Name: Цей стовпчик вказує на сеанс, у якому запущений процес. В деяких операційних системах це може бути ім'я користувача або інше ідентифікуюче значення.

Session#: Ідентифікатор сеансу, у якому запущений процес.

Mem Usage: Це обсяг пам'яті, яку використовує процес наразі.

```

14 sites = {
15     "кіту нау": "kiutz.nau.edu.ua",
16     "гіт хаб": "git.hub",
17     "гугл": "google.com",
18     "лінкедін": "linkedin.com",
19 }
20 # Основна програма
21 if __name__ == "__main__":
22     print_running_processes()
23
24     # Запуск пінгування сайту коледжу
25     while True:
26         print("Введіть назву сайту для пінгування ('q' для виходу), або оберіть один із запропонованих нижче:")
27         for site in sites:
28             print(f'\t{site}')
29
30         website_input = input("Сайт: ")
31
32         if website_input.lower() == 'q':
33             break
34         elif website_input in sites:
35             website = sites[website_input]
36         else:
37             website = website_input
38
39         command = f'ping {website}'
40         result = subprocess.run(command, text=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
41         if result.returncode == 0:
42             print(f"Пінгування сайту {website} успішне:")
43             print(result.stdout)
44         else:
45             print(f"Помилка при пінгуванні сайту {website}:", result.stderr)
46

```

Введіть назву сайту для пінгування ('q' для виходу), або оберіть один із запропонованих нижче:

кіту нау
гіт хаб
гугл
лінкедін

Сайт: кіту нау

Пінгування сайту kiutz.nau.edu.ua успішне:

Pinging kiutz.nau.edu.ua [193.178.34.27] with 32 bytes of data:

Reply from 193.178.34.27: bytes=32 time=4ms TTL=58

Reply from 193.178.34.27: bytes=32 time=3ms TTL=58

Reply from 193.178.34.27: bytes=32 time=4ms TTL=58

Reply from 193.178.34.27: bytes=32 time=3ms TTL=58

Ping statistics for 193.178.34.27:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 3ms, Maximum = 4ms, Average = 3ms

Введіть назву сайту для пінгування ('q' для виходу), або оберіть один із запропонованих нижче:

кіту нау
гіт хаб
гугл
лінкедін

Сайт: q

PS C:\Users\Ivan> █

Pinging kiutz.nau.edu.ua [193.178.34.27] with 32 bytes of data: команда ping відправляє пакети на веб-сайт kiutz.nau.edu.ua і IP-адресою 193.178.34.27.

Reply from 193.178.34.27: bytes=32 time=509ms TTL=58: був отриманий відповідний пакет від IP-адреси 193.178.34.27. Параметри відповіді:

bytes=32: Розмір пакета в байтах.

time=509ms: Час, необхідний для відправлення та отримання відповіді в мілісекундах.

TTL=58: Кількість переадресацій, які пакет може здійснити, перед тим як бути викинутим з мережі.

Ping statistics for 193.178.34.27: Це загальна статистика пінгування для IP-адреси 193.178.34.27.

Packets: Sent = 4: Кількість відправлених пакетів.

Received = 4: Кількість отриманих пакетів.

Lost = 0 (0% loss): Кількість втрачених пакетів.

Approximate round trip times in milli-seconds: Оцінкові часи у мілісекундах для одного "кругового маршруту" пінгування.

Minimum = 12ms: Мінімальний час відправлення та отримання відповіді.

Maximum = 509ms: Максимальний час відправлення та отримання відповіді.

Average = 137ms: Середній час відправлення та отримання відповіді.

Відповіді на контрольні питання:

1. Що таке процес з точки зору операційного середовища Windows?

Процес з точки зору операційної системи Windows: В операційній системі Windows процес є інстанцією виконання програми, яка має власний набір ресурсів і виконується незалежно від інших процесів. Кожен процес має свій унікальний ідентифікатор, область пам'яті та набір даних, що включає стан процесу, включаючи код програми, дані стеку, змінні середовища, відкриті файли та інші ресурси.

2. Перелічить можливі функції для створення та знищення Win32 процесу.

Можливі функції для створення та знищення Win32 процесу: Для створення та керування процесами в операційній системі Windows використовують різні функції з Win32 API, такі як CreateProcess, OpenProcess, TerminateProcess та інші. Ці функції дозволяють створювати нові процеси, отримувати доступ до вже запущених процесів та керувати ними, включаючи їх завершення.

3. Перелічить компоненти процесу.

Компоненти процесу: Кожен процес в операційній системі Windows має ряд компонентів, включаючи область пам'яті, яка містить виконуваний код та дані процесу, область стеку для зберігання локальних змінних та адрес повернення, таблицю оброблених винятків, інформацію про стан процесу та його ресурси, такі як відкриті файли, вводу/виводу та інші.

4. За яких умов може завершитись процес

Умови завершення процесу: Процес може завершитися за різних умов, включаючи:

Завершення своєї роботи згідно зі своєю логікою.

Завершення через помилку або виняток.

Завершення користувачем, наприклад, через закриття вікна програми або через команду виходу з програми.

Завершення системою через обмеження ресурсів, наприклад, відсутність доступної пам'яті або перевищення часового ліміту виконання.

Висновки: Навчився пінгувати сайт – і розуміти отриману інформацію, що повідомляє про кількість отриманих пакетів, стабільність з'єднання. Також

переглядати задіяні в роботі операційної системи субпроцеси, скільки місця та пам'яті займає кожен.