

Raport 5: Procesy i wątki

Zadanie 1 – Wątki w Pythonie

Stwórz w Pythonie program, który będzie:

1. Pobierał od użytkownika liczbę wątków, jaka ma zostać utworzona.
2. Tworzył odpowiednią liczbę wątków oraz zapisywał je do listy.
3. Startował wszystkie utworzone wątki.
4. Czekał na zakończenie wątków.

Każdy utworzony wątek powinien 100 razy wykonać następujące operacje:

1. Wypisać na ekran "Hello, this is thread number X"
2. Począkać 10 sekund (z użyciem `time.sleep()`)

W raporcie umieścić wynik działania komendy `ps` połączonej z komendą `grep` (zarówno w wersji dla procesów jak i dla wątków) oraz informację ile procesów i wątków zostało uruchomionych.

Działanie komendy z procesami:

```
sansforensics@siftworkstation: ~
$ ps aux | head -n 1;ps aux | grep zadanie.py
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
sansfor+    99069  0.3  0.3 243112 13344 pts/1    SL   18:44   0:00 /bin/python
/home/sansforensics/.vscode-oss/extensions/ms-python.python-2022.4.1/pythonFile
s/lib/python/debugpy/launcher 36599 -- /home/sansforensics/Documents/zadanie.py
sansfor+    99077  0.0  0.2  91592  8620 pts/1    SL+  18:44   0:00 /bin/python
/home/sansforensics/Documents/zadanie.py
sansfor+    99136  0.0  0.0   9036   660 pts/0    S+   18:45   0:00 grep --color
=auto zadanie.py
```

Zostały uruchomione 2 procesy: PID 99069 i PID 99077

Działanie komendy dla wątków:

```
sansforensics@siftworkstation: ~
$ ps aux -T | head -n 1;ps aux -T | grep zadanie.py
USER          PID  SPID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
sansfor+    99069  99069  0.2  0.3 243112 13344 pts/1    SL   18:44   0:00 /bin/python /home/sansforensics/.vscode
-oss/extensions/ms-python.python-2022.4.1/pythonFiles/lib/python/debugpy/launcher 36599 -- /home/sansforensics/Doc
uments/zadanie.py
sansfor+    99069  99074  0.0  0.3 243112 13344 pts/1    SL   18:44   0:00 /bin/python /home/sansforensics/.vscode
-oss/extensions/ms-python.python-2022.4.1/pythonFiles/lib/python/debugpy/launcher 36599 -- /home/sansforensics/Doc
uments/zadanie.py
sansfor+    99069  99076  0.0  0.3 243112 13344 pts/1    SL   18:44   0:00 /bin/python /home/sansforensics/.vscode
-oss/extensions/ms-python.python-2022.4.1/pythonFiles/lib/python/debugpy/launcher 36599 -- /home/sansforensics/Doc
uments/zadanie.py
sansfor+    99069  99078  0.0  0.3 243112 13344 pts/1    SL   18:44   0:00 /bin/python /home/sansforensics/.vscode
-oss/extensions/ms-python.python-2022.4.1/pythonFiles/lib/python/debugpy/launcher 36599 -- /home/sansforensics/Doc
uments/zadanie.py
sansfor+    99077  99077  0.0  0.2  91592  8620 pts/1    SL+  18:44   0:00 /bin/python /home/sansforensics/Documen
ts/zadanie.py
sansfor+    99077  99081  0.0  0.2  91592  8620 pts/1    SL+  18:44   0:00 /bin/python /home/sansforensics/Documen
ts/zadanie.py
sansfor+    99169  99169  0.0  0.0   9036  2608 pts/0    R+   18:45   0:00 grep --color=auto zadanie.py
sansforensics@siftworkstation: ~
```

Liczba uruchomionych wątków wynosi 4 – dla procesu o PID 99069 są to wątki i SPID 99074, 99076 i 99078, a dla procesu o PID 99077 jest to wątek o SPID 99081.

Wątki o PID i SPID = 99069 i 99077 to procesy (inaczej wątki główne), a wątki o tym samym PID, ale różnych SPID to pozostałe wątki w tym procesie.

Zadanie 2 – tworzenie nowych procesów w Pytonie

Stwórz analogiczny program co w zadaniu 1, ale tym razem wykorzystaj procesy zamiast wątków. Po uruchomieniu programu i podaniu wymaganej liczby sprawdź ile procesów jest uruchomionych w systemie. Możesz to zrobić z użyciem komendy `ps aux` połączonej z komendą `grep`.

Po uruchomieniu programu i podaniu liczby wątków sprawdź ile procesów oraz ile wątków jest uruchomionych przez Twoją aplikację. Możesz to zrobić z użyciem komendy `ps aux` połączonej z komendą `grep`. Aby zobaczyć wątki możesz użyć flagi `-T` w komendzie `ps -T aux`.

Działanie komendy z procesami:

```
sansforensics@siftworkstation: ~
$ ps aux | head -n 1;ps aux | grep zadanie2.py
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
sansfor+  102231  0.3  0.3 243112 13364 pts/0    SL   19:04   0:00 /bin/python /home/sansforensics/.vscode-oss/exten
sions/ms-python.python-2022.4.1/pythonFiles/lib/python/debugpy/launcher 45179 -- /home/sansforensics/Documents/zadanie2.py
sansfor+  102236  0.1  0.2 19656 11048 pts/0    S+   19:04   0:00 /bin/python /home/sansforensics/Documents/zadanie2.py
sansfor+  102284  0.0  0.2 19656 9156 pts/0    S+   19:04   0:00 /bin/python /home/sansforensics/Documents/zadanie2.py
sansfor+  102377  0.0  0.0 9036 2572 pts/3     S+   19:04   0:00 grep --color=auto zadanie2.py
```

Zostały uruchomione 3 procesy z czego pierwszy jest procesem głównym a pozostałe są realizowane na podstawie tego procesu. Mają one numery PID 102231, 102236, 102284.

Działanie komendy dla wątków:

```
sansforensics@siftworkstation: ~
$ ps aux -T | head -n 1;ps aux -T | grep zadanie2.py
USER      PID  SPID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
sansfor+  102231 102231 0.5  0.3 243112 13364 pts/0    SL   19:04   0:00 /bin/python /home/sansforensics/.vscode-oss/extensions/ms-python.python-2022.4.1/pythonFiles/lib/python/debugpy/launcher 45179 -- /home/sansforensics/Documents/zadanie2.py
sansfor+  102231 102233 0.0  0.3 243112 13364 pts/0    SL   19:04   0:00 /bin/python /home/sansforensics/.vscode-oss/extensions/ms-python.python-2022.4.1/pythonFiles/lib/python/debugpy/launcher 45179 -- /home/sansforensics/Documents/zadanie2.py
sansfor+  102231 102235 0.0  0.3 243112 13364 pts/0    SL   19:04   0:00 /bin/python /home/sansforensics/.vscode-oss/extensions/ms-python.python-2022.4.1/pythonFiles/lib/python/debugpy/launcher 45179 -- /home/sansforensics/Documents/zadanie2.py
sansfor+  102231 102237 0.0  0.3 243112 13364 pts/0    SL   19:04   0:00 /bin/python /home/sansforensics/.vscode-oss/extensions/ms-python.python-2022.4.1/pythonFiles/lib/python/debugpy/launcher 45179 -- /home/sansforensics/Documents/zadanie2.py
sansfor+  102236 102236 0.3  0.2 19656 11048 pts/0    S+   19:04   0:00 /bin/python /home/sansforensics/Documents/zadanie2.py
sansfor+  102284 102284 0.0  0.2 19656 9156 pts/0    S+   19:04   0:00 /bin/python /home/sansforensics/Documents/zadanie2.py
sansfor+  102352 102352 0.0  0.0 9036 2408 pts/3     R+   19:04   0:00 grep --color=auto zadanie2.py
```

Zostało uruchomionych 6 wątków z czego 4 dla procesu głównego i 2 dla pozostałych procesów.

Zadanie 3 – porównanie wątków i procesów

Porównaj (w formie tabeli) współpracę kilku wątków pochodzących z jednego procesu do pracy kilku procesów pod kątem:

- czy pamięć jest współdzielona pomiędzy wątkami/procesami?
- czy z punktu widzenia systemu zachodzi zmiana kontekstu w przypadku przełączania między wątkami/procesami?

- czy wątki/procesy współdzielą między sobą zasoby systemowe?

	Współpraca wątków pochodzących z jednego procesu	Współpraca procesów
Współdzielenie pamięci	Odnosząc się do zmiennej statycznej lub globalnej, różne wątki tego samego procesu odwołują się do tego samego miejsca w pamięci. Zmienne na stosie są natomiast lokalne dla wątku, ponieważ każdy wątek posiada własny stos, rezydujący w innym miejscu pamięci	Jest to najszybszy sposób komunikacji pomiędzy procesami. Jeden z procesów tworzy segment pamięci współdzielonej, a każdy proces zyskuje do niej dostęp względem miejsca wyznaczonego przez jego adres dowiązania
Zmiana kontekstu	Przełączanie między procesami powoduje zmianę kontekstu, ale są to mniejsze koszty niż przy przełączaniu procesów	Przełączanie między procesami powoduje zmianę kontekstu
Współdzielenie zasobów systemowych	Wątki współdzielą zasoby systemowe w ramach procesu, w którym są utworzone	Procesy nie dzielą między sobą zasobów, ale w przypadku tworzenia procesu przez inny proces początkowo współdzielą ze sobą większość zasobów

Zadanie 4 – analiza pamięci procesu

Uruchom poniższy kod w Pythonie oraz podaj jakieś wymyślane hasło zawierające przynajmniej 4 znaki (i zapamiętaj je!).

```
sansforensics > Documents > kod3.py > ...
import time

password = input("Input password (but pls remember it)\n")
while True:
    print("I am still alive and still remember your password")
    time.sleep(10)
```

Ten program po prostu przechowuje w pamięci podane przez Ciebie hasło. Naszym celem jest znalezienie tego hasła w pamięci procesu. W tym celu najpierw musisz znaleźć PID uruchomionego procesu (np. z wykorzystaniem komendy `ps aux`). Następnie wykonaj zrzut całej pamięci procesu za pomocą komendy: `$gcore <PID>`

```
sansforensics@siftworkstation: ~/Documents
$ ps aux | head -n 1;ps aux | grep kod3.py
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
sansfor+ 105216  0.0  0.2 17608  8700 pts/3    S+   19:25   0:00 python kod3.py
sansfor+ 105220  0.0  0.0  9032   656 pts/5    S+   19:26   0:00 grep --color=auto kod3.py
sansforensics@siftworkstation: ~/Documents
```

Wybrany PID: 105216

Output z komendy `cat`:

```
sansforensics@siftworkstation: ~/Documents
$ cat 105216out.txt | grep "Wisloka"
Wisloka
Wisloka
sansforensics@siftworkstation: ~/Documents
```

Wynik działania komendy file na pliku wynikowym z komendy gcore:

```
sansforensics@siftworkstation: ~/Documents
$ file core.105216
core.105216: ELF 64-bit LSB core file, x86-64, version 1 (SYSV), SVR4-style, from
'python', real uid: 1000, effective uid: 1000, real gid: 1000, effective gid: 1000
, execfn: '/usr/bin/python', platform: 'x86_64'
sansforensics@siftworkstation: ~/Documents
```