

Tugas 8 - Multiprocessing

Nama : Alya Mirza Safira

NPM : 21083010039

Kelas : B

A. Materi

1. Membuat folder dengan perintah “mkdir SISOP_Tgs8”, kemudian masuk kedalam folder tersebut dengan perintah “cd SISOP_tgs8”

```
alya@alya-VirtualBox:~/SISOP$ mkdir SISOP_Tgs8
alya@alya-VirtualBox:~/SISOP$ cd SISOP_Tgs8
```

2. Karena, kita akan membuat file python, kita dapat mengecek apakah kita sudah menginstall python atau belum dengan mengecek versinya seperti di bawah ini. Jika sudah terinstall maka akan muncul output python version dalam linux kita

```
alya@alya-VirtualBox:~/SISOP/SISOP_Tgs8$ python3 --version
Python 3.10.4
```

3. Membuat file.py baru dengan perintah “nano latihan_8.py”

```
alya@alya-VirtualBox:~/SISOP/SISOP_Tgs8$ nano latihan_8.py
```

4. Kemudian menulis script .py seperti dibawah ini

```
alya@alya-VirtualBox: ~/SISOP/SISOP_Tgs8
File Edit View Search Terminal Help
GNU nano 6.2 latihan 8.py *
from os import getpid
from time import time, sleep
from multiprocessing import cpu_count, Pool, Process

# Inisialisasi Fungsi yang akan digunakan:
def cetak(i):
    print("Cetak angka", i+1, "-punya ID proses", getpid())
    sleep(1)

# 1-Pemrosesan Sekuensial
print("1-Pemrosesan Sekuensial")
## Untuk mendapatkan waktu sebelum eksekusi
sekuensial_awal = time()
## Pemrosesan berlangsung
for i in range(10):
    cetak(i)
## Untuk mendapatkan Waktu Setelah Eksekusi
sekuensial_akhir = time()
print()
```

```
alya@alya-VirtualBox: ~/SISOP/SISOP_Tgs8
File Edit View Search Terminal Help
GNU nano 6.2 latihan_8.py *

## 2-Multiprocessing dengan kelas process:
print("2-Multiprocessing dengan kelas Process")
## Untuk menampung proses-proses
kumpulan_proses = []
## Untuk mendapatkan waktu sebelum eksekusi
process_awal = time()
##Proses berlangsung
for i in range(10):
    p = process(target=cetak, args=(i,))
    kumpulan_proses.append(p)
    p.start()
## untuk menggabungkan proses-proses agar tidak loncat ke proses sebelumnya
for i in kumpulan_proses:
    p.join()
## Untuk mendapatkan waktu setelah eksekusi
process_akhir= time()
print()

## 3 -Multiprocessing dengan Kelas Pool:
print("3-Multiprocessing dengan kelas Pool")
## Untuk mendapatkan waktu sebelum eksekusi
pool_awal = time()
##Proses berlangsung
pool = Pool()
pool.map(cetak, range(0, 100))
pool.close
## Untuk mendapatkan waktu setelah eksekusi
pool_akhir = time()
print()

## Membandingkan waktu eksekusi:
print("Perbandingan waktu Eksekusi")
print("Sekuensial :", sekuensial_akhir - sekuensial_awal, "detik")
print("Kelas Process :", process_akhir - process_awal, "detik")
print("Kelas Pool :", pool_akhir - pool_awal, "detik")

^G Help      ^O Write Out ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste      ^J Justify
^C           ^_           ^/

[icon] [icon] [icon] [icon] [icon]
```

5. Simpan file dengan CTRL+X kemudian pilih y dan tutup dengan enter
6. Jalankan file latihan_8.py dengan perintah python3_latihan_8.py

File Edit View Search Terminal Help

alya@alya-VirtualBox:~/SISOP/SISOP_Tgs8\$ python3 latihan_8.py

1-Pemrosesan Sekuensial

Cetak angka 1 -punya ID proses 2313
Cetak angka 2 -punya ID proses 2313
Cetak angka 3 -punya ID proses 2313
Cetak angka 4 -punya ID proses 2313
Cetak angka 5 -punya ID proses 2313
Cetak angka 6 -punya ID proses 2313
Cetak angka 7 -punya ID proses 2313
Cetak angka 8 -punya ID proses 2313
Cetak angka 9 -punya ID proses 2313
Cetak angka 10 -punya ID proses 2313

2-Multiprocessing dengan kelas Process

Cetak angka 1 -punya ID proses 2314
Cetak angka 2 -punya ID proses 2315
Cetak angka 3 -punya ID proses 2316
Cetak angka 4 -punya ID proses 2317
Cetak angka 5 -punya ID proses 2318
Cetak angka 6 -punya ID proses 2319
Cetak angka 7 -punya ID proses 2320
Cetak angka 9 -punya ID proses 2322
Cetak angka 8 -punya ID proses 2321
Cetak angka 10 -punya ID proses 2323

3-Multiprocessing dengan kelas Pool

Cetak angka 1 -punya ID proses 2324
Cetak angka 2 -punya ID proses 2324
Cetak angka 3 -punya ID proses 2324
Cetak angka 4 -punya ID proses 2324
Cetak angka 5 -punya ID proses 2324
Cetak angka 6 -punya ID proses 2324
Cetak angka 7 -punya ID proses 2324
Cetak angka 8 -punya ID proses 2324
Cetak angka 9 -punya ID proses 2324
Cetak angka 10 -punya ID proses 2324

Perbandingan waktu Eksekusi

Sekuensial : 10.014171600341797 detik
Kelas Process : 1.1630043983459473 detik
Kelas Pol : 10.147743463516235 detik

B. Latihan Soal

Dengan menggunakan pemrosesan paralel buatlah program yang dapat menentukan sebuah bilangan itu ganjil atau genap!

Batasan:

- Nilai yang dijadikan argumen pada fungsi sleep () adalah satu detik.
- Masukkan jumlah'nya satu dan berupa bilangan bulat.
- Masukkan adalah batas dari perulangan tersebut.
- Setelah perulangan selesai program menampilkan waktu eksekusi pemrosesan sekuensial dan paralel.

Dengan output seperti dibawah ini:

Contoh input :

3

Contoh Output :

```
Sekuensial
1 Ganjil - ID proses ****
2 Genap - ID proses ****
3 Ganjil - ID proses ****

multiprocessing.Process
1 Ganjil - ID proses ****
2 Genap - ID proses ****
3 Ganjil - ID proses ****

multiprocessing.Pool
1 Ganjil - ID proses ****
2 Genap - ID proses ****
3 Ganjil - ID proses ****

Waktu eksekusi sekuensial : ** detik
Waktu eksekusi multiprocessing.Process : ** detik
Waktu eksekusi multiprocessing.Pool : ** detik
```

Jawaban:

1. Membuat file dengan perintah “nano Tugas_8.py”

```
alya@alya-VirtualBox:~/SISOP/SISOP_Tgs8$ nano Tugas_8.py
```

2. Isi script .py seperti dibawah ini

```
alya@alya-VirtualBox: ~/SISOP/SISOP_Tgs8
File Edit View Search Terminal Help
GNU nano 6.2 Tugas_8.py *
from os import getpid
from time import time, sleep
from multiprocessing import cpu_count, Pool, Process

# Input batas perulangan
batas = int(input("Masukkan batas perulangan : "))
print()
# Inisialisasi Fungsi yang akan digunakan:
def cetak(i):
    for i in range(1, batas+1):
        if i % 2 == 1:
            print(i, "Ganjil", "_ID proses", getpid())
            continue
        print(i, "Genap", "-ID proses", getpid())
        sleep(1)
# 1-Pemrosesan sekuensial
print("Sekuensial")
## Untuk mendapatkan waktu sebelum eksekusi
sekuensial_awal = time()
## Proses berlangsung
for i in range(1):
    cetak(i)
## Untuk mendapatkan waktu setelah eksekusi
sekuensial_akhir = time()
print()

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

- a. Kita import terlebih dahulu library-library dari modul di python
 - getpid digunakan untuk mengambil ID proses
 - time digunakan untuk mengambil waktu(detik)
 - sleep digunakan untuk memberi jeda waktu(detik)
 - cpu_count digunakan untuk melihat jumlah CPU
 - Pool adalah sebuah class pada library multiprocessing yang digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses sebanyak jumlah CPU pada komputer
 - Process adalah sebuah class pada library multiprocessing yang digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses secara beruntun pada komputer
- b. Lalu kita buat inputan untuk batasan perulangan dengan memasukkan bilangan bulat atau integer dan kita beri nama 'batas'
- c. Kita inisialisasi fungsi yang digunakan menjadi output dimana kita menggunakan def cetak(i), i merupakan range antara 1 hingga batas+1. Lalu kita berikan perulangan dibawahnya dengan menggunakan if continue, jika hasil sisa pembagian i dengan 2 adalah 1 maka akan mengprint ganjil lalu dilanjutkan genap dan akan kembali lagi ke ganjil hingga batas perulangan yang ditetapkan. Disini kita memakai sleep (1) untuk memberi jeda waktu sebanyak parameter yang diberikan.

d. Kita masuk pada Pemrosesan Sekuensial

- Kita memanggil time untuk mendapatkan waktu sebelum eksekusi di awal
- Dan ketika proses berlangsung yang akan dijalankan adalah untuk i dalam range 1, (kenapa 1? Karena kita menggunakan batasan semisal 3 untuk 1 perulangan). Lalu i akan dicetak
- Kemudian kita memanggil time lagi untuk mendapatkan waktu setelah eksekusi di akhir

```
# 2- Multiprocessing dengan kelas Process
print("Multiprocessing.Process")
## Untuk menampung proses-proses
kumpulan_proses = []
## untuk mebdapatkan waktu sebelum eksekusi
process_awal = time()
## proses berlangsung
for i in range(1):
    p = Process(target=cetak, args=(i, ))
    kumpulan_proses.append(p)
    p.start()
## Untuk menggabungkan proses-proses agar tidak loncat ke proses sebelumnya
for i in kumpulan_proses:
    p.join()
## Untuk mendapatkan waktu setelah eksekusi
process_akhir = time()
print()
```

e. Masuk ke multiprocessing kedua yakni Multiprocessing dengan Kelas Process

- Kita gunakan kurung [] sebagai tampungan proses-proses dalam list/index dengan nama kumpulan_proses
- Kita gunakan time untuk mendapatkan waktu sebelum eksekusi
- Prosesnya adalah untuk i dalam rentang 1, kita buat p = Process (target disamadengankan cetak, dan kita gunakan args agar fungsi memungkinkan menerima semua argumen tanpa mengetahui seberapa banyak) Lalu tampungan proses di awal tadi kita panggil dan ditambahkan dengan append untuk menambahkan proses p dan kita gunakan start untuk memulai proses p
- Untuk menggabungkan proses-proses agar tidak loncat ke proses selanjutnya, kita gunakan for i dalam kumpulan_proses dan kita panggil fungsi join
- Proses sudah selesai dan kita memanggil time kembali untuk mendapatkan waktu setelah eksekusi di akhir.

```

# 3-Multiprocessing dengan kelas Pool
print("Multiprocessing.Pool")
# Untuk mendapatkan waktu sebelum eksekusi
pool_awal = time()
## proses berlangsung
pool = Pool()
pool.map(cetak, range(0,1))
pool.close()
## untuk mendapatkan waktu setelah eksekusi
pool_akhir = time()
print()

# perbandingan waktu eksekusi
print("Waktu eksekusi sekuensial :", sekuensial_akhir - sekuensial_awal, "detik")
print("Waktu multiprocessing.Process :", process_akhir - process_awal, "detik")
print("Waktu multiprocessing.Pool", pool_akhir - pool_awal, "detik")

```

f. Yang ketiga adalah Multiprocessing dengan Kelas Pool

- Kita gunakan time untuk mendapatkan waktu sebelum eksekusi di awal
- Prosesnya:

Memanggil fungsi Pool (), kemudian gunakan map (cetak, dan rangenya dari 0, 1) lalu ditutup dengan close ()

- Kita dapatkan waktu setelah eksekusi dengan time

g. Dan yang terakhir adalah membandingkan waktu eksekusi

Dengan mengeprint satu-satu tiap langkah multiprocessing kemudian time akhir – time awal dengan keterangan detik

3. Lalu simpan dengan CTRL+X dan pilih y kemudian tekan enter untuk menutup

4. Eksekusi file py dengan perintah python3 Tugas_8.py dan akan menghasilkan hasil seperti di bawah ini. Maka hasil sudah sesuai dengan output yang diinginkan soal.

```

alya@alya-VirtualBox:~/SISOP/SISOP_Tgs8$ python3 Tugas_8.py
Masukkan batas perulangan: 3
Sekuenial
1 Ganjil - ID proses 3036
2 Genap - ID proses 3036
3 Ganjil - ID proses 3036
multiprocessing.Process
1 Ganjil - ID proses 3040
2 Genap - ID proses 3041
3 Ganjil - ID proses 3042
multiprocessing.Pool
1 Ganjil - ID proses 3043
2 Genap - ID proses 3043
3 Ganjil - ID proses 3043
Waktu eksekusi sekuensial : 3.0035133361816406 detik
Waktu eksekusi multiprocessing.Process : 3.0960123538970947 detik
Waktu eksekusi multiprocessing.Pool : 3.085005283355713 detik

```