

Nama : Radya Ardi Ninang Pudyastuti

NPM : 21083010097

Kelas : Sistem Operasi A

BRIEF TUGAS 8

MULTIPROCESSING

Pada saat kita akan mengoperasikan syntax menggunakan multiprocessing, pertama-tama kita memanggil library. Seperti getpid, time sleep, cpu_count, pool, dan process.

- 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 digunakan untuk melakukan pemrosesan parallel dengan menggunakan proses sebanyak jumlah CPU pada komputer
- Process digunakan untuk melakukan pemrosesan parallel dengan menggunakan proses secara beruntun pada komputer.

```
from os import getpid
from time import time, sleep
from multiprocessing import cpu_count, Pool, Process

def cetak(i):
    if i % 2 == 0:
        print("%d Genap" %i, "-Punya ID proses", getpid())
    else:
        print("%d Ganjil" %i, "-Punya ID proses", getpid())
        sleep(1)

x = int(input("input :"))

print("Sekuensial")
sekuensial_awal = time()
for i in range(x):
    cetak(i+1)
sekuensial_akhir = time()
```

Setelah memanggil library kita menginisialisasikan fungsi yang akan digunakan. Fungsi def digunakan untuk mencetak angka dari variabel i beserta ID proses dari parameter yang diberikan. Disamping itu kita memberikan sebuah pernyataan menggunakan fungsi if else untuk menentukan angka tersebut tergolong bilangan ganjil atau genap. Selain itu, kita panggil pula fungsi sleep untuk memberi jeda waktu dalam hitungan detik. Disini saya memberikan jeda waktu 1 detik.

Sebelum memasuki fungsi multiprocessing, kita memberikan sebuah pernyataan untuk bisa mengisikan angka yang ingin kita inputkan. Disini saya misalkan dengan x.

Masuk ke dalam fungsi sekuensial. Pada pengeksekusian fungsi sekuensial, script sekuensial_awal = time() digunakan untuk mendapatkan waktu sebelum dieksekusi. Kemudian kita memproses fungsi i dalam range yang akan ditentukan nanti dan akan menghasilkan output i+1. Setelah itu kita mencari waktu setelah eksekusi dengan menggunakan syntax sekuensial_akhir = time().

```

print("Multiprocessing.Process")
kumpulan_proses = []
process_awal = time()
for i in range(x):
    p = Process(target=cetak, args=(i+1,))
    kumpulan_proses.append(p)
    p.start()

for i in kumpulan_proses:
    p.join()

process_akhir = time()

print("Multiprocessing.Pool")
pool_awal = time()
pool = Pool()
pool.map(cetak, range(1, x+1))
pool.close()

pool_akhir = time()

print("Waktu Eksekusi Sekuensial :", sekuensial_akhir - sekuensial_awal, "detik")
print("Waktu Eksekusi Multiprocessing.process :", process_akhir - process_awal, "detik")
print("Waktu Eksekusi Multiprocessing.pool :", pool_akhir - pool_awal, "detik")

```

Pada multiprocessing kelas process menggunakan perulangan for dengan range yang telah ditentukan diawal, yaitu menggunakan x.

Pada multiprocessing kelas Pool menggunakan fungsi map yang memetakan pemanggilan fungsi cetak ke dalam CPU sebanyak angka yang telah diinputkan.

Dan yang terakhir adalah membandingkan waktu proses dari sekuensial, multiprocessing kelas proses dan multiprocessing kelas pool.

```

radyaardi@radyaardi:~/Tugas8$ nano Tugas_8.py
radyaardi@radyaardi:~/Tugas8$ python3 Tugas_8.py
input :3
Sekuensial
1 Ganjil -Punya ID proses 2861
2 Genap -Punya ID proses 2861
3 Ganjil -Punya ID proses 2861
Multiprocessing.Process
1 Ganjil -Punya ID proses 2862
3 Ganjil -Punya ID proses 2864
2 Genap -Punya ID proses 2863
Multiprocessing.Pool
1 Ganjil -Punya ID proses 2865
2 Genap -Punya ID proses 2866
3 Ganjil -Punya ID proses 2865
Waktu Eksekusi Sekuensial : 3.1056740283966064 detik
Waktu Eksekusi Multiprocessing.process : 1.0542590618133545 detik
Waktu Eksekusi Multiprocessing.pool : 2.086296558380127 detik
radyaardi@radyaardi:~/Tugas8$

```

Untuk mengetahui hasil dari script yang telah kita bikin, kita dapat memanggilnya dengan script `python3 Tugas_8.py`.