

Nama : Angela Lisanthoni

NPM : 21083010032

Kelas : Sistem Operasi A

Keterangan : Laporan Tugas 8

MULTIPROCESSING

LATIHAN SOAL

- a. Buat file Bernama "Tugas_8.py" menggunakan nano

```
angela@angela-VirtualBox:~/tugas8$ nano Tugas_8.py
```

- b. inputkan kodingan berikut

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

1. Import modul yang diperlukan.

getpid digunakan untuk mendapatkan proses id

time digunakan untuk mengambil waktu pada proses dijalankan atau diakhiri

sleep digunakan untuk menunda waktu eksekusi

cpu_count digunakan untuk menghitung berapa banyak cpu yang tersedia

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

```
def cetak(i):
    if (i+1)%2==0:
        print(i+1, "genap - ID Process", getpid())
    else:
        print(i+1, "ganjil - ID Process", getpid())
        sleep(1)
```

2. Terdapat fungsi Bernama 'cetak' dimana digunakan untuk melihat apakah angka yang masuk ganjil atau genap. Disini ketika angka yang dimasukkan menghasilkan 0 ketika di modulo 2, maka angka tersebut genap dan jika menghasilkan angka lainnya, maka angka tersebut berarti ganjil.

```
n=int(input("Angka batasan? "))
```

3. User diminta menginputkan sebuah angka bulat yang digunakan sebagai Batasan

```
#SEKUENSIAL
sekuensial_awal = time()
print("Sekuensial")
for i in range(n):
    cetak(i)
sekuensial_akhir=time()
```

4. Pertama, gunakan sekuensial processing.

- sekuensial_awal dan sekuensial_akhir adalah variable untuk menyimpan waktu durasi selama proses sekuensial processing berlangsung.

- lakukan Looping sebanyak angka yang dimasukkan oleh user, dan gunakan fungsi cetak yang sudah terisi di awal untuk mencetak setiap angka ganjil atau genap dengan proses idnya masing – masing

```
#MULTIPROCESSING DENGAN KELAS PROCESS
process_awal=time()
print("Multiprocess.process")
for i in range(n):
    p=Process(target=cetak, args=(i, ))
    p.start()
    p.join()
process_akhir=time()
```

5. kedua, gunakan multiprocessing dengan kelas process

- process_awal dan process_akhir adalah variable menyimpan waktu awal mulainya proses dijalankan dan waktu berakhirnya proses
- lakukan Looping sebanyak angka yang dimasukkan oleh user, dan gunakan fungsi cetak yang sudah terisi di awal untuk mencetak setiap angka ganjil atau genap dengan proses idnya masing – masing
- p.start() digunakan untuk mengeksekusi fungsi cetak di kelas process
- p.join() digunakan agar proses ditunggu hingga proses sebelumnya selesai. Sehingga akan menghasilkan proses id yang berbeda – beda tiap prosesnya.

```
#MULTIPROCESSING DENGAN KELAS POOL
pool_awal=time()
pool = Pool()
print("Multiprocess.pool")
pool.map(cetak, range(0,n))
pool.close()
pool_akhir=time()
```

6. Ketiga, gunakan multiprocessing dengan kelas pool

- pool_awal dan pool_akhir adalah variable menyimpan waktu awal mulainya proses dijalankan dan waktu berakhirnya proses
- definisikan pool sebagai fungsi pool
- fungsi map disini digunakan untuk memetakan fungsi cetak ke dalam setiap CPU yang tersedia sebanyak 0-n kali (n adalah inputan Batasan dari user)

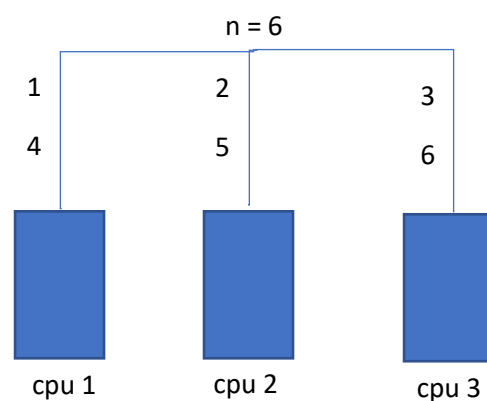
```
#BANDINGKAN WAKTU EKSEKUSI
print("Perbandingan waktu")
print("Sekuensial:", sekuensial_akhir - sekuensial_awal, "detik")
print("Kelas Process:", process_akhir - process_awal, "detik")
print("Kelas Pool:", pool_akhir - pool_awal, "detik")
```

7. Kemudian, bandingkan setiap jenis eksekusi dengan waktu akhir – waktu awal untuk melihat berapa lama pemrosesan berlangsung.

c. Eksekusi file dengan python3 Tugas_8.py

```
angela@angela-VirtualBox:~/tugas8$ python3 Tugas_8.py
Angka batasan? 6
Sekuensial
1 ganjil - ID Process 2428
2 genap - ID Process 2428
3 ganjil - ID Process 2428
4 genap - ID Process 2428
5 ganjil - ID Process 2428
6 genap - ID Process 2428
Multiprocess.process
1 ganjil - ID Process 2429
2 genap - ID Process 2430
3 ganjil - ID Process 2431
4 genap - ID Process 2432
5 ganjil - ID Process 2433
6 genap - ID Process 2434
Multiprocess.pool
1 ganjil - ID Process 2435
2 genap - ID Process 2436
3 ganjil - ID Process 2437
4 genap - ID Process 2436
5 ganjil - ID Process 2435
6 genap - ID Process 2437
Perbandingan waktu
Sekuensial: 6.05465030670166 detik
Kelas Process: 6.182013988494873 detik
Kelas Pool: 2.036794900894165 detik
```

- Disini user memasukkan angka 6 sebagai Batasan
- Pada sekuensial, terlihat bahwa setiap ID processnya itu sama. Hal ini karena sekuensial akan eksekusi pada pemroses yang sama.
- Pada multiprocessing dengan kelas process, terlihat bahwa setiap ID processnya itu berbeda dan beruntunt. Hal ini menunjukkan bahwa tiap pemanggilan fungsi cetak ditangani oleh satu proses saja
- Pada multiprocessing dengan kelas pool, terlihat ada 3 ID process yang berbeda namun, berulang. Karena di laptop saya ada 3 CPU maka dipetakan menjadi 3 pemrosesan di tiap CPU yang berbeda. Ilustrasinya bisa dilihat pada gambar dibawah



- Terlihat dari durasi waktunya, multiprocessing dengan kelas pool adalah yang terbaik dengan durasi yang paling singkat.