

NAMA : Nada Salsabila
NPM : 21083010089
MATKUL : Sistem Operasi A

DOKUMENTASI TUGAS KE-8 “*Multiprocessing*”

1. Pada tahapan awal kita membuat file py menggunakan command nano namafile.py

```
nada@nada:~$ nano Tugas_8.py
```

2. Selanjutnya, muat built-in libraries. Getpid, time, sleep, cpu_count, pool, dan process

```
GNU nano 6.2
from os import getpid
from time import time,sleep
from multiprocessing import cpu_count, Pool, 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 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

3. Menginisialisasi fungsi yang akan digunakan

```
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)
```

- Fungsi di atas digunakan untuk mencetak angka dari variabel i beserta ID proses sejumlah parameter yang diberikan. Fungsi sleep dipanggil untuk memberi jeda waktu(detik) sebanyak parameter yang diberikan.

4. Melakukan Pemrosesan Sekuensial

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

5. Tahapan Multiprocessing dengan Kelas Proses

```
#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()
```

- Process_awal digunakan untuk mendapatkan waktu sebelum eksekusi
- Saat proses berlangsung
- Dapat diperhatikan dengan seksama bahwa ID proses tiap memanggil fungsi cetak adalah berbeda-beda. Ini menandakan bahwa tiap pemanggilan fungsi cetak ditangani oleh satu proses saja. Kemudian untuk pemanggilan selanjut'nya ditangani oleh proses yang lain.
- Kumpulan proses harus ditampung dan digabung menjadi satu(p.join()) agar tidak merambah ke proses selanjutnya.

6. Tahapan Multiprocessing dengan kelas Pool

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

- Jumlah ID proses terbatas pada empat saja karena jumlah CPU pada komputer saya hanyalah 4. Jangan risaukan urutan angka yang dicetak jika tidak berurutan, kan emang ini pemrosesan paralel. Fungsi map() itu memetakan pemanggilan fungsi cetak ke dalam 4 CPU sebanyak n kali.

7. Membandingkan Waktu Eksekusi

```
#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")
```

8. Selanjutnya untuk menjalankan program, simpan file nano terlebih dahulu dengan menggunakan *ctrl+x, y* dan, *Enter*

9. Jalankan program dengan menggunakan command python3 namafile.py

```
nada@nada:~$ python3 Tugas_8.py
Angka batasan? 3
Sekuensial
1 ganjil - ID Process 2971
2 genap - ID Process 2971
3 ganjil - ID Process 2971
Multiprocess.process
1 ganjil - ID Process 2973
2 genap - ID Process 2974
3 ganjil - ID Process 2975
Multiprocess.pool
1 ganjil - ID Process 2976
2 genap - ID Process 2976
3 ganjil - ID Process 2976
Perbandingan waktu
Sekuensial: 3.0051279067993164 detik
Kelas Process: 3.1945300102233887 detik
Kelas Pool: 3.1966865062713623 detik
nada@nada:~$
```

- Perbandingan proses sekuensial ternyata tidak berbeda jauh dengan multiprocessing.