

Galang Surya Ramadhan

21083010081

Sistem Operasi A

Tugas Week 8

Multiprocessing

Tugas di Week 8 kali ini adalah melakukan pemrograman paralel(yang merupakan salah satu konsep dasar system operasi) dengan menggunakan metode Multiprocessing. Pemrograman paralel adalah sebuah teknik eksekusi perintah yang mana dilakukan secara bersamaan pada CPU.

Manfaat Multipreprocessing

- Menggunakan CPU untuk komputasi
- Tidak berbagi sumber daya memori
- Memerlukan sumber daya memori dan waktu yang tidak Sedikit
- Tidak memerlukan sinkronisasi memori

Input

```
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
GNU nano 6.2                                     tugas8.py
from os import getpid
from time import time, sleep
from multiprocessing import cpu_count, Pool, Process

x = int(input("Batas perulangan: "))
def tampil(i):
    if i % 2 == 0:
        print(f"{i+1} Ganjil", "- ID proses", getpid())
    elif i % 2 != 0:
        print(f"{i+1} Genap", "- ID proses", getpid())
    else:
        print("selesai")
        sleep(1)

print("\nSekuensial")

# UNTUK MENDAPATKAN WAKTU SEBELUM EKSEKUSI
sekuensial_awal = time()

# PROSES BERLANGSUNG
for i in range(x):
    tampil(i)

# UNTUK MENDAPATKAN WAKTU SETELAH EKSEKUSI
sekuensial_akhir = time()
```

```
# UNTUK MENDAPATKAN WAKTU SETELAH EKSEKUSI
sekuensial_akhir = time()

print("\nKelas Process")

# UNTUK MENAMPUNG PROSES-PROSES
kumpulan_proses = []

# UNTUK MENDAPATKAN WAKTU SEBELUM EKSEKUSI
process_awal = time()

# PROSES BERLANGSUNG
for i in range(x):
    p = Process(target=tampil, args=(i,))
    kumpulan_proses.append(p)
    p.start()

# UNTUK MENGGABUNGKAN PROSES-PROSES AGAR TIDAK LONCAT KE PROSES SEBELUM'NYA
for i in kumpulan_proses:
    p.join()

# UNTUK MENDAPATKAN WAKTU SETELAH EKSEKUSI
process_akhir = time()

print("\nKelas Pool")
```

```
print("\nKelas Pool")

# UNTUK Mendapatkan Waktu Sebelum Eksekusi
pool_awal = time()

# Proses Berlangsung
pool = Pool()
pool.map(tampil, range(x))
pool.close()

# Untuk Mendapatkan Waktu Sebelum Eksekusi
pool_akhir = time()

print("\nSekuensial :", sekuensial_akhir - sekuensial_awal, "detik")
print("Kelas Process :", process_akhir - process_awal, "detik")
print("Kelas Pool :", pool_akhir - pool_awal, "detik")
```

Output:

Ini COntoh Output dengan perulangan sebbanyak 3 kali

```
galangsurya@galangsurya-VirtualBox:~/tugasweek8$ nano tugas8.py
galangsurya@galangsurya-VirtualBox:~/tugasweek8$ python3 tugas8.py
Batas perulangan: 3

Sekuensial
1 Ganjil - ID proses 2880
2 Genap - ID proses 2880
3 Ganjil - ID proses 2880

Kelas Process
2 Genap - ID proses 2882
3 Ganjil - ID proses 2883
1 Ganjil - ID proses 2881

Kelas Pool
1 Ganjil - ID proses 2884
2 Genap - ID proses 2884
3 Ganjil - ID proses 2884

Sekuensial : 3.5762786865234375e-05 detik
Kelas Process : 0.021821022033691406 detik
Kelas Pool : 0.059308767318725586 detik
```

Ini contoh Output perulangan sebanyak 7 kali

Batas perulangan: 7

Sekuensial

1 Ganjil - ID proses 2888
2 Genap - ID proses 2888
3 Ganjil - ID proses 2888
4 Genap - ID proses 2888
5 Ganjil - ID proses 2888
6 Genap - ID proses 2888
7 Ganjil - ID proses 2888

Kelas Process

3 Ganjil - ID proses 2891
1 Ganjil - ID proses 2889
2 Genap - ID proses 2890
4 Genap - ID proses 2892
6 Genap - ID proses 2894
5 Ganjil - ID proses 2893
7 Ganjil - ID proses 2895

Kelas Pool

1 Ganjil - ID proses 2896
2 Genap - ID proses 2896
3 Ganjil - ID proses 2896
4 Genap - ID proses 2896
5 Ganjil - ID proses 2896
6 Genap - ID proses 2896
7 Ganjil - ID proses 2896