Nama : Renaldy Al Ikhsan

NPM : 21083010072

Kelas : Sistem Operasi B

Pertama tuliskan code berikut ini pada terminal :

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

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)

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

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

#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()

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

#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")
```

Kita tuliskan Code di seperti di atas pada terminal Linux

Pertama kita melakukan Improt :

1. Getpid : Ketika beberapa proses terbentuk dan sedang berjalan, id unik diberikan padanya. Ini adalah id prosesnya. Fungsi ini membantu mengembalikan id dari proses yang sedang dipanggil.
2. Time : di gunakan untuk pengukuran Waktu
3. Multiprocessing : digunakan untuk menjalakan multiprocessing

Di code tersebut ada *sekuensial* dan *multi proces*sing menurut pengertian *sekuensial* dan *multi processing* adalah

1, *Sekuensial* adalah Seluruh "Runnable Software" pada komputer yang meliputi juga sistem operasi biasanya diorganisasi atau disusun menjadi sejumlah proses-proses sequential (berurutan)

2. *Multi-Processing* adalah dimana seluruh proses di computer di lakukan secara serentak, dalam prosesnya menggunakan lebih banyak *CPU* untuk membantu proses yang di lakukan serentak.

Setelah code itu di jalankan maka ada di minta memasukkan Batasan, Batasan :

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

Contoh waktu code di jalankan

Batasan proes yaitu 9

```
renaldy@renaldy-VirtualBox:~/archon$ python3 Tugas_8.py
Angka batasan? 9
```

Hasil Proses *Sekuensial*

```
Sekuensial
1 ganjil - ID Process 2472
2 genap - ID Process 2472
3 ganjil - ID Process 2472
4 genap - ID Process 2472
5 ganjil - ID Process 2472
6 genap - ID Process 2472
7 ganjil - ID Process 2472
8 genap - ID Process 2472
9 ganjil - ID Process 2472
```

Hasil Proses *Multi Prosess*

```
Multiprocess.process
1 ganjil - ID Process 2478
2 genap - ID Process 2479
3 ganjil - ID Process 2482
4 genap - ID Process 2484
5 ganjil - ID Process 2485
6 genap - ID Process 2486
7 ganjil - ID Process 2487
8 genap - ID Process 2488
9 ganjil - ID Process 2489
```

Hasil *Mutli Proses Pool*

```
Multiprocess.pool
1 ganjil - ID Process 2490
2 genap - ID Process 2490
3 ganjil - ID Process 2490
4 genap - ID Process 2490
5 ganjil - ID Process 2490
6 genap - ID Process 2490
7 ganjil - ID Process 2490
8 genap - ID Process 2490
9 ganjil - ID Process 2490
```

Hasil Perbandingan Waktu Penyelesaian Ke 3 proses tersebut.

```
Perbandingan waktu
Sekuensial: 9.008931159973145 detik
Kelas Process: 9.047573804855347 detik
Kelas Pool: 9.02871036529541 detik
```

Kesimpulan.

Kita dapat disimpulkan bahwa waktu yang diperlukan untuk menjalakan paling cepat yaitu *Sekuensial*. Kenapa karena berurutan dan hanya sedikit daya *CPU* yang di gunakan sedangkan *Multi-Process* dipengaruhi power *CPU.*