

Nama : Fitri Indah Sari  
NPM : 21083010025  
Kelas : Sistem Operasi B

## TUGAS 8 (MULTIPROCESSING)

### SOAL LATIHAN

Dengan menggunakan pemrosesan paralel buatlah program yang dapat menentukan sebuah bilangan itu ganjil atau genap!

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.

a. Buat file python menggunakan nano dengan perintah nano namafile.py

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

b. Ketikkan script di bawah ini ke dalam file nano yang sudah dibuat

1. Import modul yang dibutuhkan

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

- getpid digunakan untuk mendapatkan ID proses
- time digunakan untuk mengambil waktu pada proses dijalankan atau diakhiri
- sleep digunakan untuk menunda waktu eksekusi
- cpu\_count digunakan untuk menghitung banyaknya CPU yang tersedia
- Pool digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses sebanyak jumlah CPU pada komputer
- Process digunakan untuk melakukan pemrosesan paralel dengan menggunakan proses secara beruntun pada komputer.

2. Mendefinisikan fungsi

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

Terdapat fungsi bernama cetak yang digunakan untuk melihat apakah angka yang diinputkan ganjil atau genap. Di sini ketika angka yang dimasukkan ketika di mod 2 hasilnya 0 maka termasuk angka genap, jika hasilnya  $\neq 0$  maka termasuk angka ganjil.

### 3. Membuat batasan

```
a=int(input("Masukkan batas perulangan: "))
```

User akan diminta untuk menginputkan angka sebagai batas perulangan.

### 4. Pemrosesan Sekuensial

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

- sekuensial\_awal dan sekuensial\_akhir adalah variabel untuk menyimpan waktu durasi selama pemrosesan sekuensial berlangsung.
- Dilakukan perulangan sebanyak angka batasan yang diinputkan oleh user dan menggunakan fungsi cetak untuk mencetak setiap angka ganjil atau genap dengan ID proses masing-masing.

### 5. Multiprocessing dengan Kelas Process

```
# Multiprocessing dengan Kelas Process
process_awal = time()
print("multiprocessing.Process")
for i in range(a):
    p=Process(target=cetak, args=(i, ))
    p.start()
    p.join()
process_akhir = time()
```

- process\_awal dan process\_akhir adalah variabel untuk menyimpan waktu awal dan waktu akhir proses tersebut.
- Dilakukan perulangan sebanyak angka batasan yang diinputkan oleh user dan menggunakan fungsi cetak untuk mencetak setiap angka ganjil atau genap dengan ID proses 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 ID proses yang berbeda-beda tiap prosesnya.

## 6. Multiprocessing dengan Kelas Pool

```
# Multiprocessing dengan Kelas Pool
pool_awal = time()
pool = Pool()
print("multiprocessing.Pool")
pool.map(cetak, range(0,a))
pool.close()
pool_akhir = time()
```

- pool\_awal dan pool\_akhir adalah variabel untuk menyimpan waktu awal dan waktu akhir dari proses tersebut.
- Didefinisikan pool sebagai fungsi pool.
- Fungsi map di sini digunakan untuk memetakan fungsi cetak ke dalam setiap CPU yang tersedia sebanyak 0-a kali (a adalah inputan batasan angka dari user)

## 7. Membandingkan waktu eksekusi dari setiap proses

```
# Membandingkan waktu eksekusi
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")
```

Perbandingan dilakukan dengan mengurangi waktu akhir dan waktu awal dari setiap proses untuk mengetahui berapa lama pemrosesan berlangsung.

### c. Periksa output dari script dengan mengetik python3 namafile.py

```
fitri@fitri-VirtualBox:~/tugas8$ python3 Tugas_8.py
Masukkan batas perulangan: 3
Sekuenial
1 Ganjil - ID proses 2357
2 Genap - ID proses 2357
3 Ganjil - ID proses 2357
multiprocessing.Process
1 Ganjil - ID proses 2358
2 Genap - ID proses 2359
3 Ganjil - ID proses 2360
multiprocessing.Pool
1 Ganjil - ID proses 2361
2 Genap - ID proses 2361
3 Ganjil - ID proses 2361
Waktu eksekusi sekuensial : 3.004176378250122 detik
Waktu eksekusi multiprocessing.Process : 3.01225209236145 detik
Waktu eksekusi multiprocessing.Pool : 3.0205190181732178 detik
```

- Di sini user menginputkan angka 3 sebagai batas perulangan.
- Pada sekuensial terlihat bahwa ID prosesnya sama, karena sekuensial akan melakukan eksekusi pada pemroses yang sama.
- Pada multiprocessing dengan kelas process terlihat bahwa ID prosesnya berbeda dan beruntun. Hal ini menunjukkan bahwa setiap pemanggilan fungsi cetak ditangani oleh satu proses saja.
- Pada multiprocessing dengan kelas Pool terlihat bahwa ID prosesnya sama karena di laptop saya ada 1 CPU maka akan dihasilkan 1 ID proses saja.