

Nama : Zulfa Febi Afidria

Npm : 21083010096

Kelas : Sistem Operasi B

Tugas 8

Import Library

Import library yang diperlukan

- 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 sebuah class pada library multiprocessing yang digunakan untuk melakukan pemrosesan parallel dengan menggunakan proses sebanyak jumlah CPU pada computer
- Process adalah sebuah class pada library multiprocessing yang digunakan untuk melakukan pemrosesan parallel dengan menggunakan proses secara bruntun pada komputer

```
GNU nano 6.2                                     Tugas8.py
❏ import library
from os import getpid
from time import time, sleep
from multiprocessing import cpu_count, Pool, Process
```

Input

Selanjutnya syntax untuk menginputkan batas perulangan pada variabel angka dan untuk menampilkan output

```
# input
angka = int(input("Input:\n"))

print("\nOutput:")
```

Inisialisasi fungsi

Syntax untuk menentukan bilangan ganjil dan bilangan genap. Jika i habis dibagi 2 maka cetak sesuai perintah syntax dibawah ini, jika i tidak habis membagi 2 maka cetak sesuai perintah syntax dibawah ini. Selanjutnya sleep digunakan untuk memberi jeda selama 1 detik.

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

Pemrosesan Sekuensial

- Syntax perintah untuk memperoleh waktu awal sebelum proses sekuensial
- Proses sekuensial
- Syntax perintah untuk memperoleh waktu akhir setelah proses sekuensial terjadi

```
GNU nano 6.2                                     Tugas8.py
# Pemrosesan Sekuensial
print("\nSekuensial")
sekuensial_awal = time()

for i in range (angka):
    ganjil_genap(i)

sekuensial_akhir = time()
```

Multiprocessing Process

- Membuat variabel kumpulan_proses untuk menyimpan sekumpulan proses.
- Syntax perintah untuk memperoleh waktu awal sebelum proses
- Proses multiprocessing process berlangsung
- Penggabungan sekumpulan proses
- Syntax perintah untuk memperoleh waktu akhir setelah proses terjadi

```
# Multiprocessing Process
print("\nmultiprocessing.Process")
kumpulan_proses = []

process_awal = time()

for i in range (angka):
    p = Process(target=ganjil_genap, args=(i,))
    kumpulan_proses.append(p)
    p.start()

for i in kumpulan_proses:
    p.join()

process_akhir = time()
```

Multiprocessing Pool

- Syntax Perintah untuk memperoleh waktu awal sebelum proses terjadi
- Proses multiprocessing pool
- Syntax Perintah untuk memperoleh waktu akhir setelah proses terjadi

```
# Multiprocessing Pool
print("\nmultiprocessing.Pool")
Pool_awal = time()

Pool = Pool()
Pool.map(ganjil_genap, range(0, 3))
Pool.close

Pool_akhir = time()
```

Perbandingan Waktu

Gambar dibawah ini merupakan syntax untuk menampilkan perbandingan waktu diantara ketiga metode, dengan menghitung selisih antara waktu setelah proses dikurangi dengan waktu sebelum proses

```
# Perbandingan Waktu
print("\nWaktu 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")
```

Output :

```
zulfa@zulfa-VirtualBox:~/sisop8$ python3 Tugas8.py
Input:
3

Output:

Sekuenisial
1 Genap - ID proses 3019
2 Ganjil - ID proses 3019
3 Genap - ID proses 3019

multiprocessing.Process
2 Ganjil - ID proses 3021
1 Genap - ID proses 3020
3 Genap - ID proses 3022

multiprocessing.Pool
1 Genap - ID proses 3023
2 Ganjil - ID proses 3023
3 Genap - ID proses 3023

Waktu eksekusi sekuensial : 3.003624677658081 detik
Waktu eksekusi multiprocessing.process : 1.0928847789764404 detik
Waktu eksekusi multiprocessing.Pool : 3.128530740737915 detik
zulfa@zulfa-VirtualBox:~/sisop8$
```