

Nama : Vera Febrianti Pakpahan

NPM : 21083010054

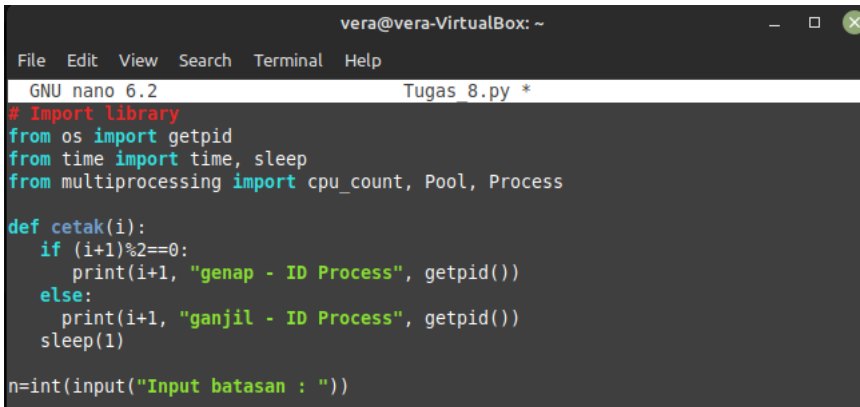
Kelas : Sistem Operasi A

Multiprocessing

- Buat file dengan perintah nano Tugas_8.py

```
vera@vera-VirtualBox:~$ nano Tugas_8.py
vera@vera-VirtualBox:~$
```

- Import semua library yang akan digunakan
 - **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



```
vera@vera-VirtualBox: ~
File Edit View Search Terminal Help
GNU nano 6.2 Tugas_8.py *
# Import library
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("Input batasan : "))
```

- Fungsi cetak digunakan untuk untuk mencetak angka dari variabel i beserta ID proses sejumlah parameter yang diberikan. Untuk fungsi di atas digunakan untuk melihat apakah angka masuk ganjil atau genap. Pada angka yang dimasukan menghasilkan 0 ketika dimodulo 2, maka angka tersebut adalah genap dan jika menghasilkan angka lainnya, maka angka tersebut adalah ganjil. Kita panggil fungsi sleep untuk memberi jeda waktu(detik) sebanyak parameter yang diberikan.

```
vera@vera-VirtualBox: ~  
File Edit View Search Terminal Help  
GNU nano 6.2 Tugas 8.py  
# Sekuensial  
print("sekuensial")  
sekuensial_awal = time()  
for i in range(n):  
    cetak(i)  
sekuensial_akhir = time()  
print()  
  
# Multiprocessing.Process  
print("multiprocessing.Process")  
kumpulan_proses = []  
process_awal = time()  
for i in range(n):  
    p = Process(target=cetak, args=(i,))  
    kumpulan_proses.append(p)  
    p.start()  
for i in kumpulan_proses:  
    p.join()  
process_akhir = time()  
print()  
  
^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute  ^C Location  
^X Exit      ^R Read File ^N Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

- Fungsi sekuensial_awal dan sekuensial_akhir digunakan untuk menyimpan waktu durasi selama proses sekuensial processing berlangsung.
- Fungsi multiprocessing dengan kelas proses
Process_awal dan process_akhir adalah variable untuk menyimpan waktu awal mulai proses dijalankan dan waktu berakhirnya proses. Apabila melakukan looping yang sesuai user pada saat memasukan angka, maka fungsi cetak yang sudah terisi dari awal untuk mencetak setiap angka ganjil atau genap dengan proses idnya masing-masing.
- Fungsi "p.start()" digunakan untuk mengeksekusi fungsi cetak dikelas process.
- Fungsi "p.join()" digunakan agar proses ditunggu hingga proses sebelumnya selesai, sehingga akan menghasilkan proses id yang berbeda-beda pada tiap prosesnya. Kumpulan proses harus ditampung dan digabung menjadi satu(p.join())agar tidak merambah ke proses selanjutnya.

```
# Multiprocessing.Pool  
print("multiprocessing.Pool")  
pool_awal = time()  
pool = Pool()  
pool.map(cetak, range(0,n))  
pool.close()  
pool_akhir = time()  
print()  
  
# Bandingkan waktu eksekusi  
print("Waktu eksekusi sekuensial :", sekuensial_akhir - sekuensial_awal, "detik")  
print("Waktu multiprocessing.Process :", process_akhir - process_awal, "detik")  
print("Waktu multiprocessing.Pool :", pool_akhir - pool_awal, "detik")  
  
^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute  ^C Location  
^X Exit      ^R Read File ^N Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

- Fungsi multiprocessing.Pool
pool_awal dan pool_akhir adalah variable yang bisa menyimpan waktu awal mulainya proses dijalkannya dan waktu berakhirnya proses

- Fungsi “map” digunakan untuk memetakan fungsi cetak kedalam setiap CPU yang tersedia sebanyak 0-n kali. Dan n adalah inputan batasan user
- Dan bandingkan waktu setiap jenis eksekusi dengan waktu akhir - waktu awal untuk melihat berapa lama pemrosesan berlangsung.
- Gunakan perintah python3 diikuti dengan nama file untuk menampilkan output dari kode script yang telah dibuat.

```
vera@vera-VirtualBox:~$ python3 Tugas_8.py
Input batasan : 3
sekuensial
1 ganjil - ID Process 3103
2 genap - ID Process 3103
3 ganjil - ID Process 3103

multiprocessing.Process
1 ganjil - ID Process 3104
2 genap - ID Process 3105
3 ganjil - ID Process 3106

multiprocessing.Pool
1 ganjil - ID Process 3107
2 genap - ID Process 3107
3 ganjil - ID Process 3107

Waktu eksekusi sekuensial : 3.003336191177368 detik
Waktu multiprocessing.Process : 1.016305685043335 detik
Waktu multiprocessing.Pool : 3.0146126747131348 detik
vera@vera-VirtualBox:~$
```