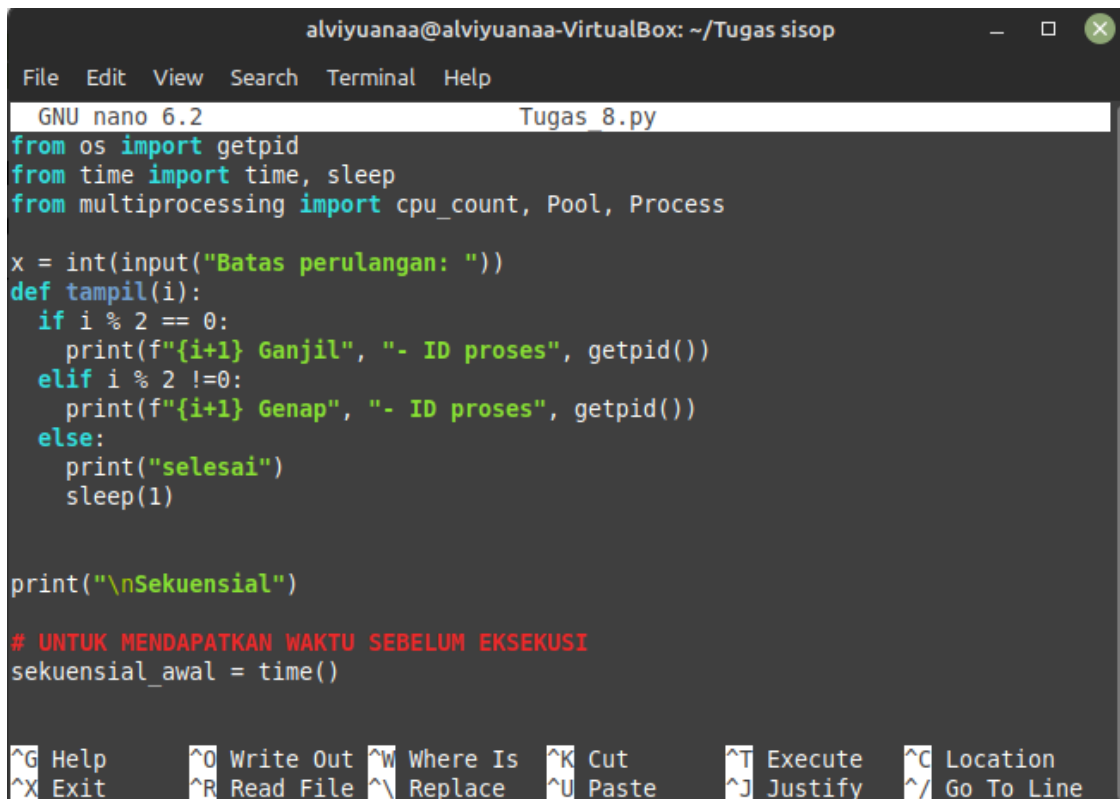


Nama : Alvi Yuana Febriyanti
NPM : 21083010030
Mata Kuliah : Sistem Operasi (A)

LAPORAN PRAKTIKUM LINUX

“praktikum 8”

➤ Script dari soal Latihan Mutiprocessing



```
alviyuanaa@alviyuanaa-VirtualBox: ~/Tugas sisop
File Edit View Search Terminal Help
GNU nano 6.2 Tugas 8.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()

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

```
alviyuanaa@alviyuanaa-VirtualBox: ~/Tugas sisop
File Edit View Search Terminal Help
GNU nano 6.2 Tugas_8.py

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

# 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,))

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

```
alviyuanaa@alviyuanaa-VirtualBox: ~/Tugas sisop
File Edit View Search Terminal Help
GNU nano 6.2 Tugas_8.py

# 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")

# UNTUK MENDAPATKAN WAKTU SEBELUM EKSEKUSI
pool_awal = time()

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

```
alviyuanaa@alviyuanaa-VirtualBox: ~/Tugas sisop
File Edit View Search Terminal Help
GNU nano 6.2 Tugas_8.py

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")

```

➤ Penjelasan Script

- 1) Kita import dulu library yang diperlukan

```
alviyuanaa@alviyuanaa-VirtualBox: ~/Tugas sisop
File Edit View Search Terminal Help
GNU nano 6.2 Tugas_8.py
from os import getpid
from time import time, sleep
from multiprocessing import cpu_count, Pool, Process

```

- 2) Fungsi ini digunakan untuk mencetak angka dari variabel i beserta ID proses sejumlah parameter yang diberikan. Kita panggil fungsi sleep untuk memberi jeda waktu(detik) sebanyak parameter yang diberikan.

```
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)

```

3) Pemrosesan Sekuensial

```
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()
```

4) Multiprocessing dengan kelas Process

```
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()
```

Dapat diperhatikan dengan seksama bahwa ID proses tiap memanggil fungsi cetak adalah berbeda-beda. Ini menandakan bahwa tiap pemanggilan fungsi cetak ditangani oleh satu proses saja. Kemudian untuk pemanggilan selanjut'nya ditangani oleh proses yang lain. Kumpulan proses harus ditampung dan digabung menjadi satu(`p.join()`) agar tidak merambah ke proses selanjutnya. Silahkan eksekusi file berikut pada terminal anda, maka anda akan paham apa yang saya maksudkan.

5) Multiprocess dengan kelas 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 Setelah Eksekusi
pool_akhir = time()
```

Jumlah ID proses terbatas pada empat saja karena jumlah CPU pada komputer saya hanyalah 6. Jangan risaukan urutan angka yang dicetak jika tidak berurutan, kan memang ini pemrosesan paralel. Fungsi map() itu memetakan pemanggilan fungsi cetak ke dalam 6 CPU sebanyak 10 kali.

6) Bandingkan Waktu Eksekusi

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

Sudah sewajarnya proses sekuensial lebih lambat dibanding multiprocessing namun bukan berarti kita harus melakukan multiprocessing terus menerus, gunakan metode sesuai kebutuhan. Nah apabila barisan kode di atas dikumpulkan jadi satu maka jadinya akan seperti ini.

➤ Output

```
alviyuanaa@alviyuanaa-VirtualBox:~/Tugas sisop$ python3 Tugas_8.py
Batas perulangan: 3

Sekuensial
1 Ganjil - ID proses 2167
2 Genap - ID proses 2167
3 Ganjil - ID proses 2167

Kelas Process
2 Genap - ID proses 2169
3 Ganjil - ID proses 2170
1 Ganjil - ID proses 2168

Kelas Pool
1 Ganjil - ID proses 2171
2 Genap - ID proses 2171
3 Ganjil - ID proses 2171

Sekuensial : 3.7670135498046875e-05 detik
Kelas Process : 0.01887679100036621 detik
Kelas Pool : 0.05294156074523926 detik
alviyuanaa@alviyuanaa-VirtualBox:~/Tugas sisop$
```