

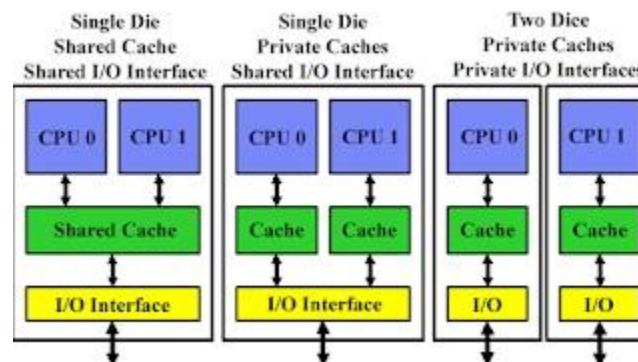
Nama : Lydia Almira Rahma Novangga

NPM : 21083010119

## Pertemuan 8

### Multiprocessing

Multiprocessing adalah istilah teknologi informasi dalam bahasa Inggris yang merujuk kepada kemampuan pemrosesan computer yang dilakukan secara serentak. Hal ini dimungkinkan dengan menggunakan dua CPU atau lebih dalam sebuah system computer.



#### Kelebihan Multiprocessor

1. Peningkatan throughput, karena lebih banyak proses/thread yang berjalan dalam satu waktu sekaligus (jika proses yang antri di ready queue sedikit). Perlu diingat hal ini tidak berarti daya komputasinya menjadi meningkat sejumlah prosesor. Yang meningkat adalah jumlah pekerjaan yang bisa dilakukannya dalam waktu tertentu.
2. Economy of sale (ekonomis), ekonomis dalam devices yang dibagi bersama-sama. Prosesor-prosesor terdapat dalam satu komputer dan dapat membagi peripheral (ekonomis) seperti disk dan catu daya listrik.
3. Peningkatan kehandalan (reliabilitas), jika satu prosesor mengalami suatu gangguan, maka proses yang terjadi masih dapat berjalan dengan baik karena tugas prosesor yang terganggu diambil alih oleh prosesor lain. Hal ini dikenal dengan istilah Graceful Degradation. Sistemnya sendiri dikenal bersifat fault tolerant atau failloft system.

Muat built-in libraries yang akan digunakan :

**getpid** digunakan untuk mengambil ID proses

**time** digunakan untuk mengambil waktu(detik)

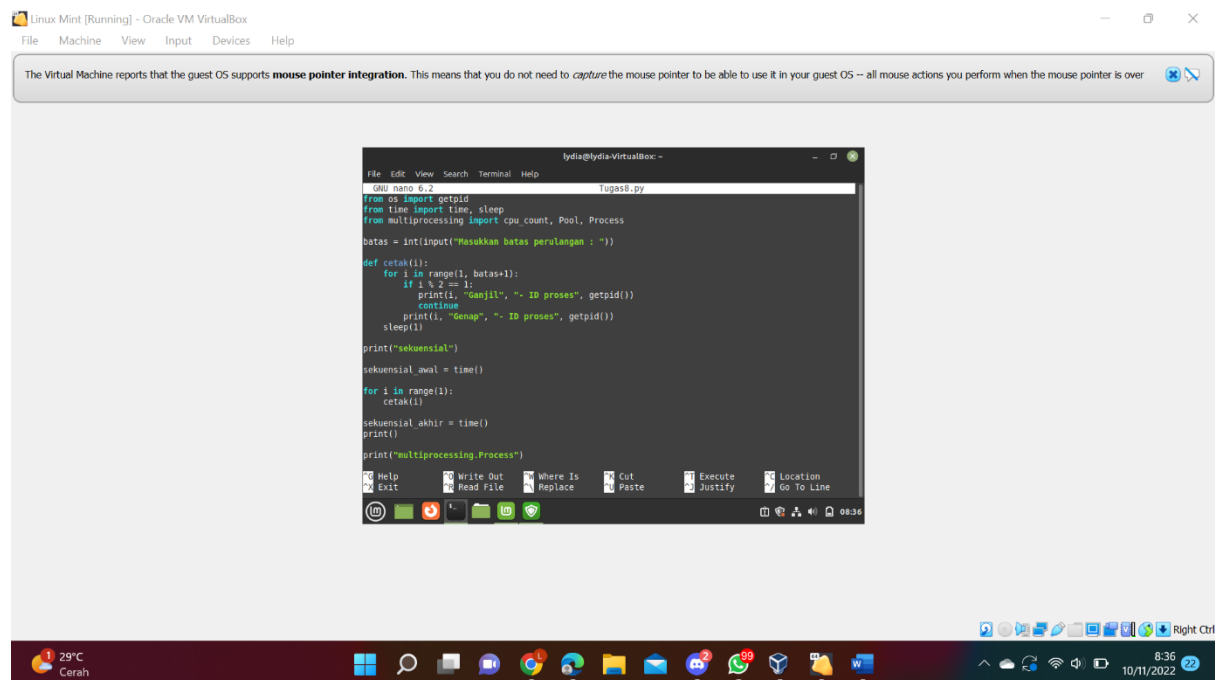
**sleep** digunakan untuk memberi jeda waktu(detik)

**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 computer.

Input :



```
kumpulan_proses = []
process_awal = time()

for i in range(1):
    p = Process(target=cetak, args=(i,))
    kumpulan_proses.append(p)
    p.start()

for i in kumpulan_proses:
    p.join()

process_akhir = time()
print()

print("multiprocessing.Pool")

pool_awal = time()

pool = Pool()
pool.map(cetak, range(0,1))
pool.close()
```

## Bandingkan Waktu Eksekusi :

```
pool_akhir = time()
print()

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")
```

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 hasilnya akan seperti ini.

## Output :

