

10. Komputeran Berprestasi Tinggi

10.0 Perkakasan Pengiraan

10.1 Teras dan kopemproses

10.2 Senibina selari

10.3 Sistem teragih

- Tugas 17

Tugasan 17

1. Aturcara Python berikut mencipta suatu subproses, yang mencetak nombor pengenalan yang diberikan:

```
import multiprocessing
import time

class Process(multiprocessing.Process):
    def __init__(self, id):
        super(Process, self).__init__()
        self.id=id
    def run(self):
        time.sleep(1)
        print("Saya proses dgn id: {}".format(self.id))
```

Kod berikut mengawalkan objek `Process` dan menyeru `Process.start()` untuk menelurkan subproses ini.

```
if __name__=='__main__':
    p = Process(0)
    p.start()
    p.join()
```

`Process.start()` memulakan proses baharu dan menyeru `Process.run()`.
`Process.join()` mengarahkan kawalan proses utama '`__main__`' menunggu subproses berkenaan tamat sebelum menjalankan arahan seterusnya.

Uji aturcara ini dengan menelurkan 2 subproses.

Tulis semula aturcara supaya subproses mengira nombor Fibonacci:

```
def fib(n):
    return n if n < 2 else fib(n-2) + fib(n-1)
```

Gunakan `multiprocessing.cpu_count()` untuk dapatkan bilangan teras dalam sistem anda, dan telurkan subproses mengira sebanyak mana ada teras. Masukkan arahan untuk mengira masa daripada mula pelakuan aturcara hingga akhirnya, dan bandingkan masa pelaksanaan bagi kalau menggunakan satu proses sahaja dan kalau multiproses.

Cuba penggunaan `multiprocessing.Pool` untuk pelaksanaan selari suatu fungsi dengan berlainan data masukan.

Kini tulis dan uji aturcara untuk mengira nombor Fibonacci menggunakan satu proses, tetapi di dalamnya multibebeban, menggunakan `threading.Thread`.

2. Uji sambungan soket antara pelayan dan pelanggan seperti berikut. Anda boleh gunakan computer anda sebagai kedua-dua pelayan ('localhost', atau '127.0.0.1') dan pelanggan (guna tettingkap lain).

Sebelah pelayan:

```
import socket

s = socket.socket()
print ("Soket berjaya diwujudkan")

# rizab suatu liang, katakan 40044
port = 40044

# ikat kpd liang ini
# rentetan kosong di medan ip bermakna dengar permintaan
# drp computer lain dalam rangkaian
s.bind(('', port))
print ("soket diikat kpd %s" %(port))

# biarkan soket dlm mod mendengar (sehingga 5 permintaan)
s.listen(5)
print ("soket mendengar")

# ulang selamanya
while True:

    # bina sambungan dengan pelanggan
    c, addr = s.accept()
    print ('dapat sambungan drp ', addr )

    # hantar mesej kpd pelanggan
    c.send(b'Terima kasih menyambung')

    # tutup sambungan
    c.close()
```

Sebelah pelanggan:

```
import socket

s = socket.socket()

# liang yg mahu disambung
port = 40044

# sambung ke pelayan
s.connect(('127.0.0.1', port))

# terima data drp pelayan
print(s.recv(1024))

# tutup
s.close()
```

