

Multithreading dan Contoh pemrograman yang tidak *thread-safe*

Multithreading

- Salah satu bentuk implementasi dari *concurrency*
- Concurrency:
 - Tugas-tugas berjalan bersamaan dalam waktu yang overlap: program menjadi lebih *usable*
 - Dapat diterapkan pada komputer dengan prosesor tunggal
 - Tidak sama dengan paralelisme

Keunggulan multithreading

- Dalam aplikasi:
 - Dapat menjadikan aplikasi lebih responsif
 - Berguna dalam operasi yang *I/O-bound* (*networking*, file, database, ...)
- Lebih ringan, dibandingkan dengan pembuatan proses baru (*fork*: bentuk implementasi concurrency lainnya)
- *Context switching* yang lebih cepat, dibandingkan dengan antar proses

Contoh multithreading

- *Web browser:*
 - Mengirimkan *request* dan mendapatkan *response* dari HTTP server
 - Sambil menampilkan sebagian konten yang telah didapatkan
 - Sambil merespon input dari pengguna

Pemrograman

- Sejumlah *thread* dapat dibuat
- Seringkali sejumlah thread tersebut harus bekerja dengan sebuah struktur data yang sama (*shared*)
- Padahal, struktur tersebut dan/atau cara pemrograman barangkali tidak thread-safe

Contoh sederhana

- Operasi penambahan berikut mungkin terlihat sederhana (kode Python, perhatikanlah baris terakhir)

`a = 1`

`a = a + 1`

Contoh sederhana (lanjutan)

- Tapi, sesungguhnya tidak! Butuh beberapa instruksi *bytecode* (baris 6 – 13).
- Di tengah operasi, thread switch dapat terjadi!

1	0 LOAD_CONST	0 (1)
	3 STORE_NAME	0 (a)
2	6 LOAD_NAME	0 (a)
	9 LOAD_CONST	0 (1)
	12 BINARY_ADD	
	13 STORE_NAME	0 (a)
	16 LOAD_CONST	1 (None)
	19 RETURN_VALUE	

Contoh kode tidak thread-safe

```
from threading import Thread

a = 1

def test():
    global a
    for i in range(1000):
        a = a + 1

def main():
    threads = [Thread(target=test) for i in range(10)]
    for t in threads:
        t.start()
    for t in threads:
        t.join()
    print(a)

if __name__ == '__main__':
    main()
```


Contoh kode tidak thread-safe (lanjutan)

```
$ python test.py  
6485  
$ python test.py  
8019  
$ python test.py  
4815  
$ python test.py  
9847  
$ python test.py  
8013
```

Dengan 10 thread,
harusnya a akan
bernilai 10001
(1+10x1000)

Diskusi

- Untuk kode tersebut, lock dapat digunakan
- Untuk kebutuhan lebih serius, struktur data/implementasi yang thread-safe dapat digunakan
- Atau, ubah cara pemrograman