

# **Laporan Praktikum**

## **Sistem Operasi**

### **Modul 1**



**Nama : Asep haryana saputra**

**NIM : 20230810043**

**Kelas : TINFC-2023-04**

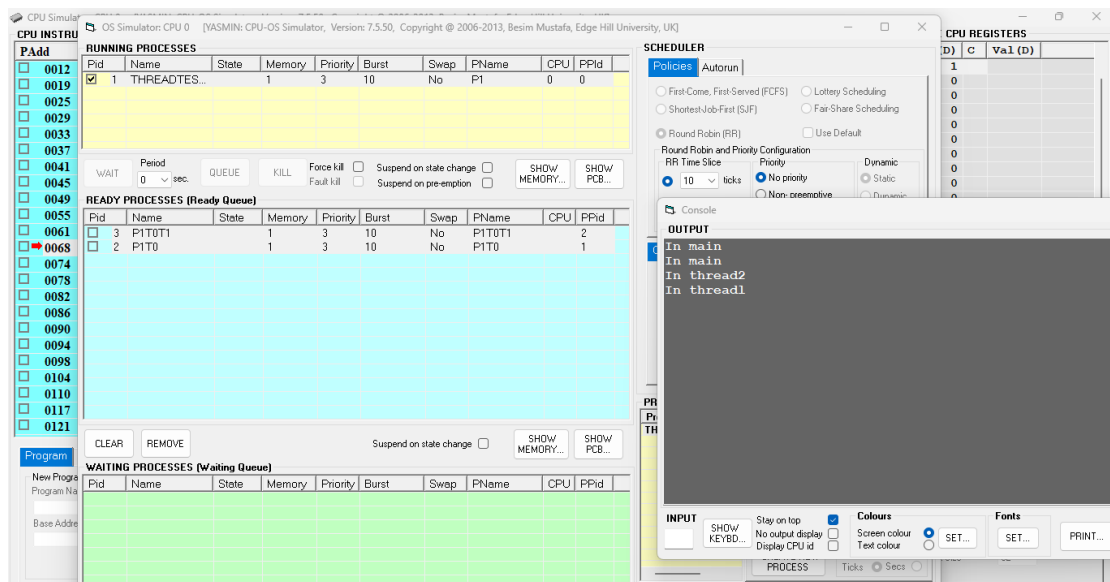
**Teknik Informatika**  
**Fakultas Ilmu Komputer**  
**Universitas Kuningan**

## Pretest

1. **Thread dalam konteks sistem operasi** adalah unit eksekusi terkecil dalam sebuah proses yang memungkinkan beberapa tugas berjalan secara bersamaan dalam satu proses.
2. **Dua karakteristik utama proses:**
  - Memiliki ruang alamat tersendiri yang terpisah.
  - Memiliki kontrol penuh atas sumber daya yang dialokasikan, seperti memori dan file.
3. **Perbedaan antara proses dan thread:**
  - Proses memiliki ruang alamat terpisah, sedangkan thread berbagi ruang alamat dengan thread lain dalam proses yang sama.
  - Proses lebih berat dalam manajemen dan memerlukan lebih banyak waktu untuk dibuat atau dihentikan dibandingkan dengan thread.
4. **Fungsi program counter dalam thread** adalah untuk melacak instruksi berikutnya yang akan dieksekusi oleh thread, memastikan alur eksekusi program berjalan sesuai urutan yang benar.

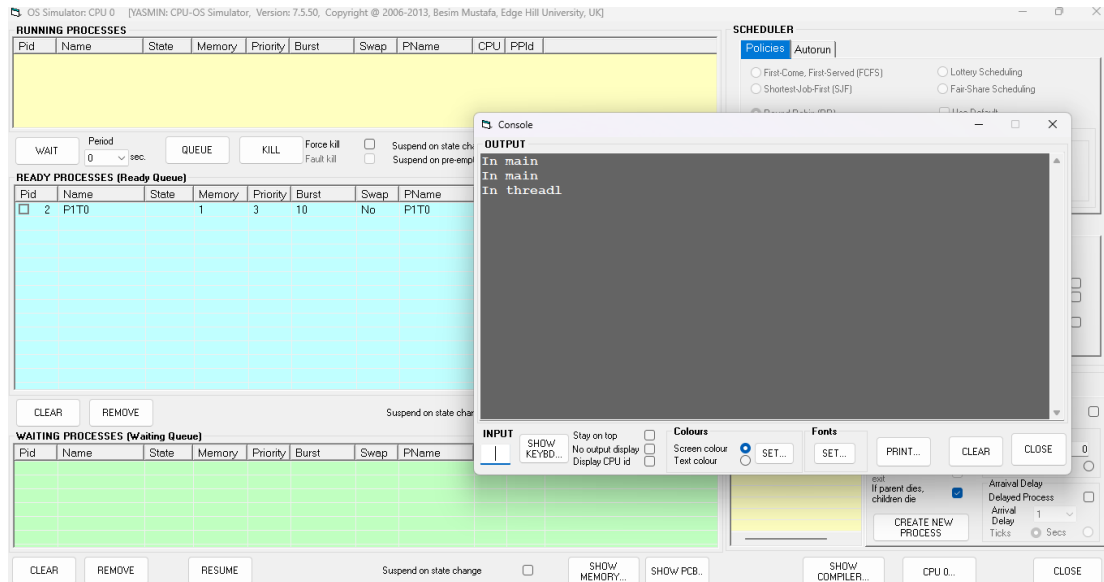
## Praktikum

1.



- a. **Jumlah proses yang telah dibuat:**
  - Program ini hanya memiliki satu proses utama, yaitu main
- b. **Sebutkan mana yang disebut proses dan mana yang disebut thread!**
  - **Proses:** Program utama (main) yang menjalankan seluruh kode program.
  - **Thread:** thread1 dan thread2

2.



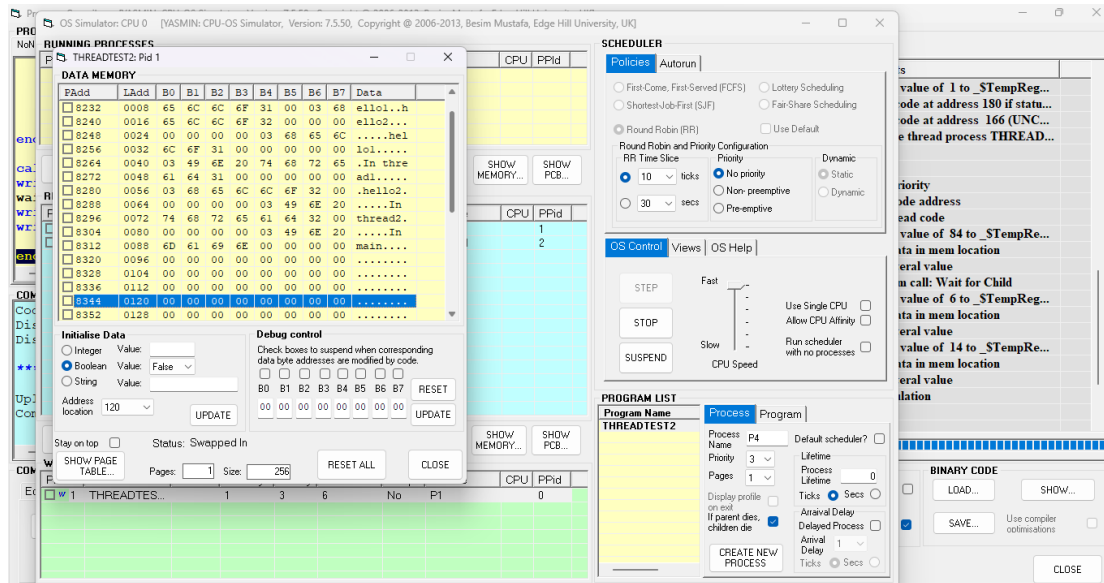
### Proses Tanpa Thread:

- Jika kode dijalankan tanpa menggunakan thread, eksekusi program akan berjalan secara berurutan dan satu tugas harus selesai sebelum tugas lainnya dimulai.
- Console hanya akan menampilkan satu output dalam satu waktu, menunggu hingga setiap bagian selesai.
- Pada program tanpa thread, jika terdapat loop yang berjalan terus-menerus (while true), program akan terjebak pada bagian tersebut dan tidak akan pernah mencapai kode berikutnya.

### Proses Menggunakan Thread:

- Dengan thread, beberapa bagian program dapat berjalan bersamaan (multithreading), sehingga thread1, thread2, dan main bisa menampilkan output di console tanpa harus menunggu bagian lainnya selesai.
- Console akan menampilkan output dari setiap thread secara bergantian atau bersamaan, tergantung pada pengaturan penjadwalan thread.
- Pada kode ini, thread1 dan thread2 akan menghasilkan output bersamaan di console, sehingga kita bisa melihat pesan dari beberapa thread aktif secara paralel.

3.



1. Variabel s1 diinisialisasi di alamat memory 8320 dan s2 diinisialisasi di alamat memory 8312.
2. Saat thread 1 dieksekusi, nilai string "hello1" disimpan ke variabel s1 di alamat memory 8320.
3. Saat thread 2 dieksekusi, nilai string "hello2" disimpan ke variabel s2 di alamat memory 8312.

Hal ini menunjukkan bahwa meskipun variabel s1 dan s2 dideklarasikan di program utama, masing-masing thread memiliki salinannya sendiri dari variabel tersebut yang disimpan di lokasi memory yang berbeda.

## PostTest

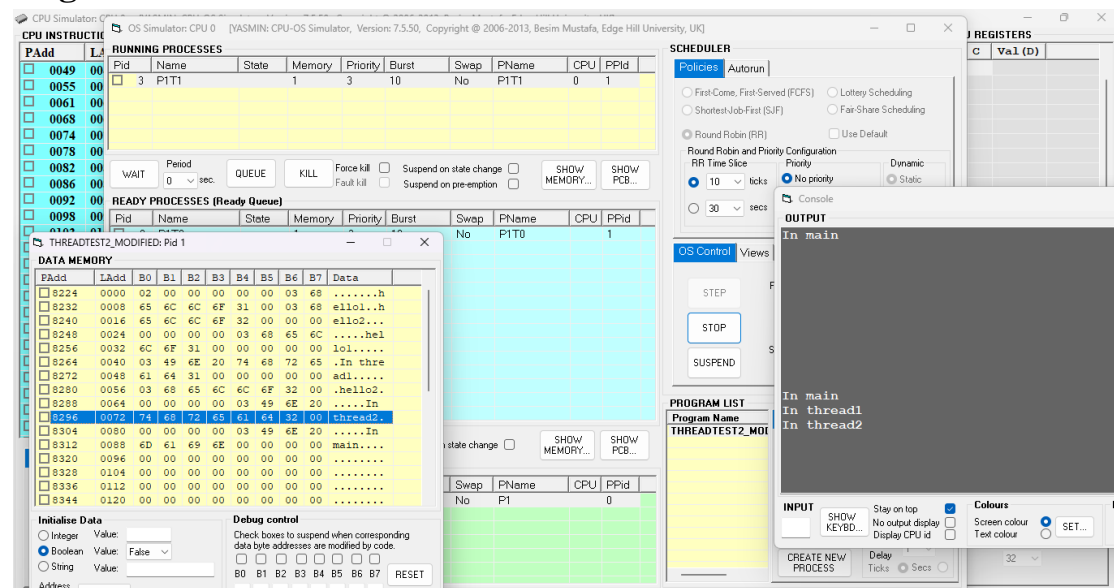
1.

Thread-thread tidak berbagi sumber daya dari proses induk. Masing-masing thread memiliki stack sendiri yang berisi data lokal, sehingga tidak ada akses langsung ke sumber daya yang dimiliki thread lainnya.

Jika thread perlu berbagi data, mereka harus menggunakan mekanisme seperti:

- **Shared memory:** Membagikan area memory tertentu untuk diakses bersama oleh semua thread.
- **Message passing:** Mengirimkan pesan antara thread untuk berbagi informasi.
- **Synchronization objects:** Membuat objek seperti mutex atau semaphore untuk mengatur akses ke sumber daya yang dibagikan.

## Tugas



Save File:

<https://github.com/MythEclipse/Praktikum-Sistem-Operasi/tree/main/Modul%202>

- **Eksekusi Paralel:** Modifikasi ini memungkinkan thread1 dan thread2 berjalan secara paralel tanpa saling menunggu. Thread utama (main) juga dapat terus memantau dan menampilkan nilai s1 dan s2.
- **Penampilan Nilai Variabel:** Program utama (main) secara periodik menampilkan nilai dari s1 dan s2, yang memperlihatkan bahwa kedua thread berhasil mengubah nilai variabel secara paralel.
- **Efektivitas Penggunaan Thread:** kita dapat melihat hasil dari setiap thread secara bersamaan di console, yang menunjukkan bahwa setiap thread berjalan sesuai fungsinya tanpa mengganggu eksekusi thread lainnya.

## Kesimpulan

modul ini menunjukkan cara menggunakan beberapa thread dan proses dalam komputasi paralel