

Laporan Praktikum

Sistem Operasi

Dosen pengampu : (*Iwan Lesmana, S.Kom., M.Kom.*)

Modul 2



Nama : Muhammad Rizal Nurfirdaus

NIM : 20230810088

Kelas : TINFC – 2023 – 04

Teknik Informatika

Fakultas Ilmu Komputer

Universitas Kuningan

PRE-TEST

Berikut jawaban dari pertanyaan-pertanyaan tersebut:

1. Apa itu thread dalam konteks sistem operasi?

Thread adalah unit terkecil dari eksekusi dalam sebuah proses yang memungkinkan beberapa bagian dari program dijalankan secara paralel atau bersamaan. Dalam konteks sistem operasi, thread berbagi sumber daya proses induknya, seperti memori dan ruang alamat, tetapi memiliki register dan program counter sendiri. Dengan thread, sistem operasi dapat menjalankan beberapa tugas dalam satu proses secara efisien.

2. Sebutkan dua karakteristik utama dari sebuah proses.

- **Isolasi Sumber Daya:** Proses memiliki ruang alamat yang terpisah dari proses lain, sehingga data atau kode di dalam satu proses tidak bisa langsung diakses oleh proses lain tanpa mekanisme komunikasi antar-proses.
- **Manajemen Sumber Daya:** Setiap proses memiliki alokasi sumber daya seperti memori, waktu CPU, dan I/O yang dikelola oleh sistem operasi. Proses bertanggung jawab atas penggunaan sumber daya yang diberikan.

3. Apa perbedaan antara proses dan thread?

- **Proses:** Unit eksekusi yang berdiri sendiri dengan ruang alamat dan sumber daya yang terpisah. Setiap proses berjalan secara independen dan membutuhkan sumber daya lebih banyak dibandingkan thread.
- **Thread:** Bagian dari proses yang dapat berjalan secara paralel di dalam proses yang sama. Thread berbagi sumber daya dengan proses induknya, sehingga lebih efisien dalam penggunaan memori dan waktu eksekusi dibandingkan proses.

4. Jelaskan fungsi dari program counter dalam sebuah thread.

Program counter (PC) dalam sebuah thread berfungsi untuk menyimpan alamat instruksi berikutnya yang akan dieksekusi oleh thread tersebut. Dengan adanya program counter, setiap thread dapat melacak posisinya dalam alur eksekusi, memungkinkan eksekusi instruksi yang teratur dan pengelolaan alur program yang lebih efisien.

PRAKTIKUM

Menampilkan daftar proses/thread dan pohon proses yang menggambarkan hubungan antar proses induk/anak.

1. Jalankan Simulator. Tuliskan kode berikut ini ke compiler.

```

program ThreadTest1
  sub thread1 as thread
    writeln("In thread1")
    while true
    wend
  end sub

  sub thread2 as thread
    call thread1
    writeln("In thread2")
    while true
    wend
  end sub

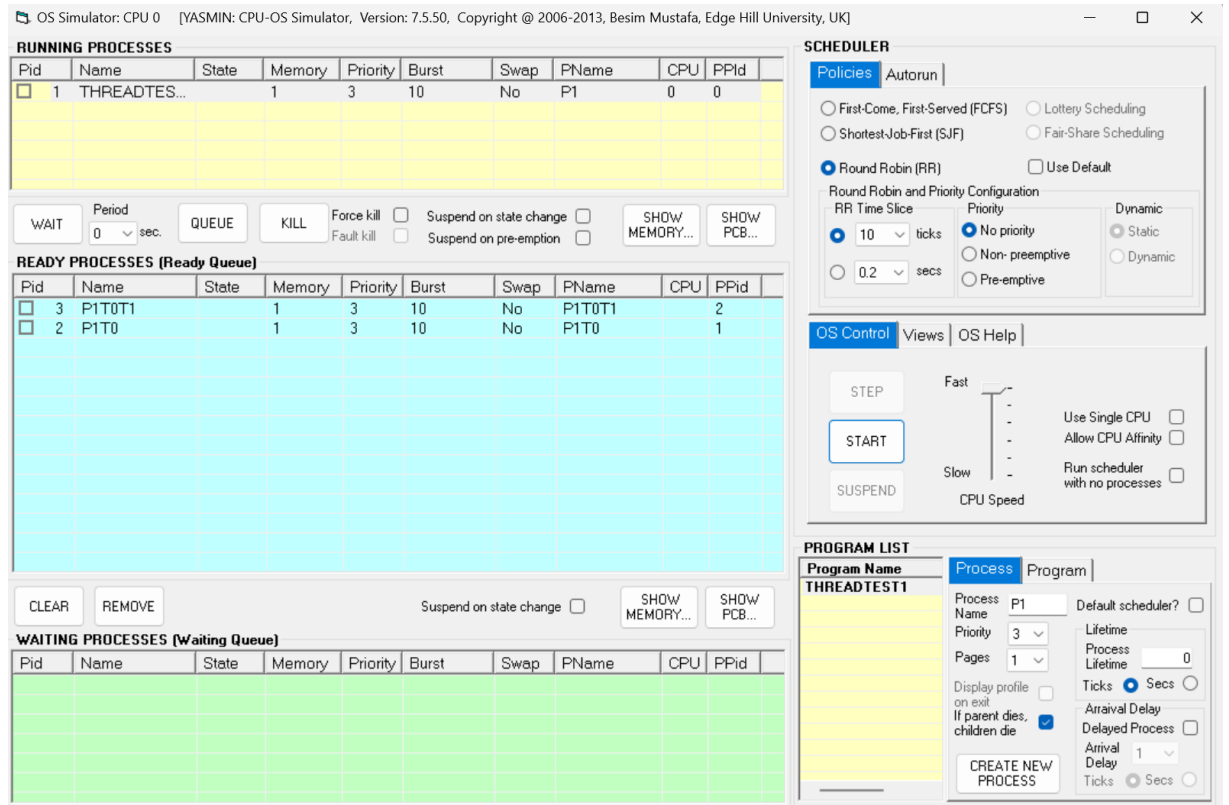
  call thread2
  writeln("In main")

do
loop
end
'''

```

2. Compile dan Run kode menggunakan CPU-OS Simulator

- a. Compile dengan menekan tombol COMPILE pada compiler.
- b. Melalui CPU Simulator, tampilkan jendela konsol dengan klik tombol INPUT/OUTPUT... aktifkan STAY ON TOP.
- c. Beralihlah ke jendela OS Simulator, buat satu proses dengan CREATE NEW PROCESS.
- d. Pastikan jenis penjadwalan adalah ROUND ROBIN, 10 ticks dengan kecepatan simulasi maksimum.
- e. Tekan tombol START dan pada waktu yang bersamaan amati tampilan pada jendela console.



3. a. Berapakah jumlah proses yang telah dibuat?

Terdapat total **3 proses** yang telah dibuat:

- Proses THREADTEST1 dengan PID 1 di bagian **Running Processes**.
- Proses P1TOT1 dengan PID 3 di bagian **Ready Processes**.
- Proses P1T0 dengan PID 2 di bagian **Ready Processes**.

b. Sebutkan mana yang disebut proses dan mana yang disebut thread!

- **Proses:** THREADTEST1, P1TOT1, dan P1T0 semuanya adalah proses karena mereka terdaftar dalam daftar proses dan memiliki PID unik di dalam simulator. Dalam konteks CPU-OS Simulator, semua entri dalam daftar proses ini dianggap sebagai proses utama atau proses turunan.
- **Thread:** Dalam CPU-OS Simulator ini, istilah "thread" merujuk pada sub-proses dalam satu proses utama, namun dalam implementasi ini tidak ada perbedaan eksplisit antara proses dan thread yang terlihat langsung di antarmuka. Namun, secara konsep, setiap subroutine yang dibuat dengan kata kunci `as thread` (seperti `thread1` dan `thread2` dalam kode yang di-compile) akan berjalan sebagai thread dalam konteks proses THREADTEST1.

4. Klik tombol VIEW PROCESS LIST. Kemudian klik PROCESS TREE.. Identifikasi proses induk dan proses turunan!

5. Gunakan tombol KILL untuk menghentikan proses!

Memodifikasi kode sumber untuk membuat versi tanpa thread dan membandingkannya dengan versi yang menggunakan thread.

1. Modifikasi source code ThreadTest1 dengan menghapus instance "as thread" pada deklarasi subroutine. Ganti nama program menjadi ThreadTest3.

2. Compile code hasil modifikasi tersebut.

3. Buat proses baru lagi, kemudian jalankan di OS simulator.

Memahami bagaimana thread-thread berbagi sumber daya dari proses induk.

1. Buatlah program baru dengan memodifikasi source code ThreadTest1. Penambahan kode ditandai dengan cetak tebal dan underline.

program ThreadTest2

var s1 string(6), var s2 string(6)

sub thread1 as thread

s1 = "hello1"

writeln("In thread1")

while true

wend

end sub

sub thread2 as thread

call thread1

s2 = "hello2"

writeln("In Thread2")

while true

wend

end sub

call thread2

```
writeln("In main")
wait
writeln(s1)
writeln(s2)
end
'''
```

2. Compile program ThreadTest2. Klik tombol SYMBOL TABLE pada jendela compiler. Amati informasi tentang variabel s1 dan s2.
3. Melalui CPU Simulator, tampilkan jendela konsol dengan klik tombol INPUT/OUTPUT... aktifkan STAY ON TOP.
4. Beralihlah ke jendela OS Simulator, buat satu proses dengan CREATE NEW PROCESS.
5. Pastikan jenis penjadwalan adalah ROUND ROBIN, kecepatan simulasi maksimum.
6. Tekan tombol START dan pada waktu yang bersamaan amati tampilan pada jendela console. Gunakan tombol SUSPEND untuk menghentikan proses sementara.
7. Tekan tombol SHOW MEMORY..., amati alamat memory yang digunakan oleh variable s1 dan s2. Simpulkan apa yang terjadi.

THREADTEST2: Pid 1

DATA MEMORY

PAdd	LAdd	B0	B1	B2	B3	B4	B5	B6	B7	Data
<input type="checkbox"/> 8224	0000	02	00	00	00	00	00	03	00
<input checked="" type="checkbox"/> 8232	0008	00	00	00	00	00	00	03	68h
<input type="checkbox"/> 8240	0016	00	00	00	00	00	00	00	00
<input type="checkbox"/> 8248	0024	00	00	00	00	03	68	65	6Chel
<input type="checkbox"/> 8256	0032	6C	6F	31	00	00	00	00	00	lol.....
<input type="checkbox"/> 8264	0040	03	49	6E	20	74	68	72	65	.In thre
<input type="checkbox"/> 8272	0048	61	64	31	00	00	00	00	00	adl.....
<input type="checkbox"/> 8280	0056	03	68	65	6C	6C	6F	32	00	.hello2.
<input type="checkbox"/> 8288	0064	00	00	00	00	03	49	6E	20In
<input type="checkbox"/> 8296	0072	74	68	72	65	61	64	32	00	thread2.
<input type="checkbox"/> 8304	0080	00	00	00	00	03	49	6E	20In
<input type="checkbox"/> 8312	0088	6D	61	69	6E	00	00	00	00	main....
<input type="checkbox"/> 8320	0096	00	00	00	00	00	00	00	00
<input type="checkbox"/> 8328	0104	00	00	00	00	00	00	00	00
<input type="checkbox"/> 8336	0112	00	00	00	00	00	00	00	00
<input type="checkbox"/> 8344	0120	00	00	00	00	00	00	00	00

Initialise Data

☐ Integer Value:
☒ Boolean Value:
☐ String Value:
 Address location:

UPDATE

Debug control

Check boxes to suspend when corresponding data byte addresses are modified by code.

☐ B0 ☐ B1 ☐ B2 ☐ B3 ☐ B4 ☐ B5 ☐ B6 ☐ B7

RESET UPDATE

Stay on top ☐ Status: Swapped In

Pages: Size:

Memori yang dialokasikan untuk variabel string s1 dan s2 bersifat persisten dalam proses THREADTEST2, dan variabel ini dibagikan di antara proses utama dan thread-thread di dalamnya. Hal ini memperlihatkan bagaimana thread-thread dalam proses yang sama dapat berbagi data dari memori induk, memungkinkan interaksi antara thread-thread dan proses utama.

POST-TEST

1. Jelaskan bagaimana thread-thread berbagi sumber daya dari proses induk.

Threads dalam satu proses berbagi ruang memori yang sama. Ini berarti mereka dapat mengakses dan mengubah variabel dan struktur data yang sama. Mereka berbagi:

Heap: Alokasi memori dinamis.

Variabel global: Variabel yang dideklarasikan di luar fungsi.

File yang terbuka: Deskriptor file dan handle file.

Selain itu, mereka juga berbagi sumber daya seperti kode, data, dan file terbuka tetapi memiliki register, stack, dan penunjuk instruksi masing-masing.

LATIHAN/TUGAS

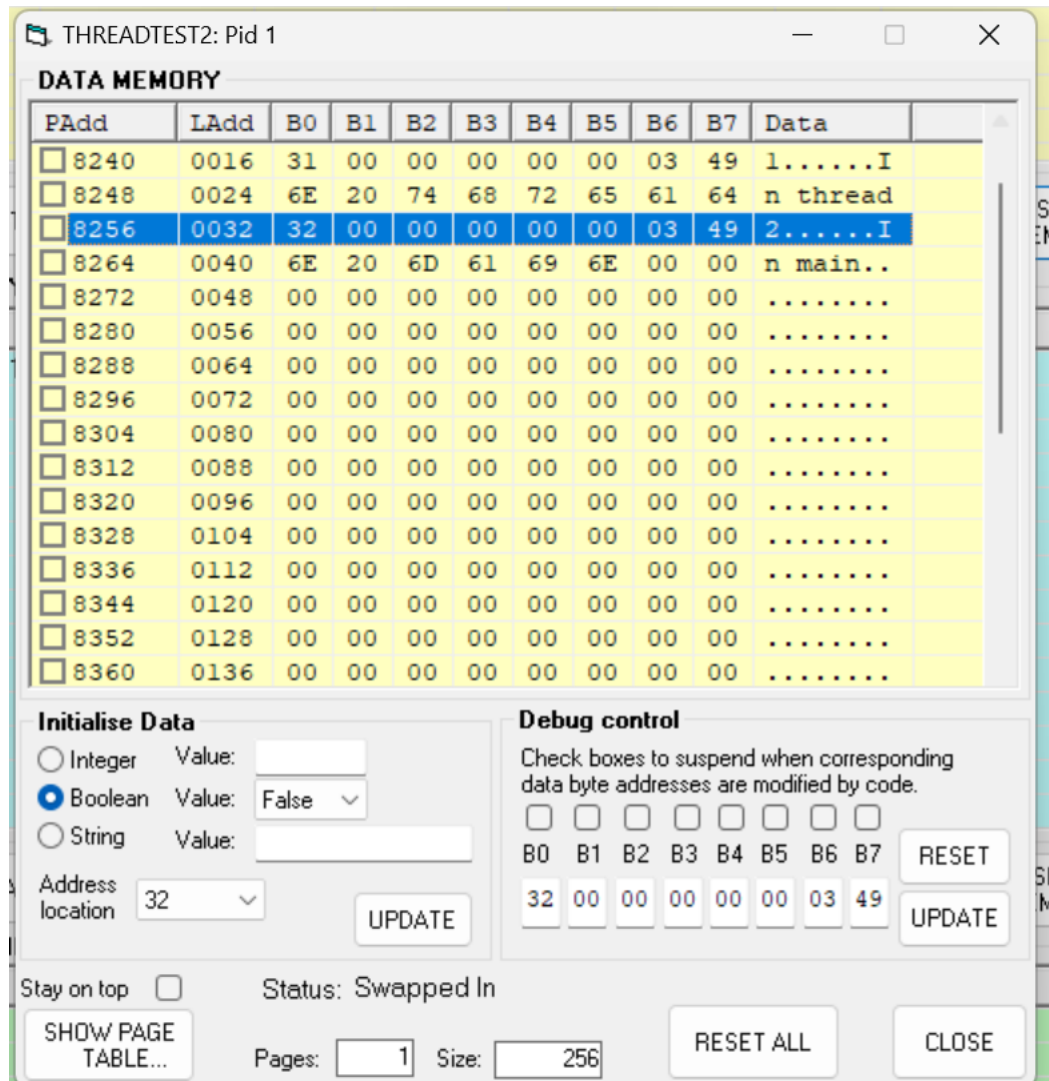
1. Buatlah program baru dengan memodifikasi source code ThreadTest2 (pada contoh program sebelumnya) dengan tahap-tahap yang sesuai dan buat kesimpulan.
program ThreadTest2

```
sub thread1 as thread
    writeln("In thread1")
    while true
    wend
end sub

sub thread2 as thread
    writeln("In thread2")
    thread1
    while true
    wend
end sub

thread2
writeln("In main")

while true
wend
end program
```

Kesimpulan: Program ini berjalan sebagaimana mestinya dengan data string yang dicetak ke dalam memori sesuai dengan output dari masing-masing bagian program (thread1, thread2, dan main). Data dalam memori tampak sesuai dengan harapan, tanpa indikasi kesalahan fatal yang terlihat dari tampilan ini.