Step demo:

1. concurrent.c dengan parameter sedikit proses, dan banyak (misal 1000). amati hasil yang didapatkan. Semestinya akan ada beberapa hasil yang tidak konsisten, misal seperti di gambar berikut. Seharusnya hasil thread 514 akan menambahkan dari hasil thread 481. Namun karena tidak ada sinkronisasai, maka hasilnya me-overwrite dari thread yang lain.

Text

Description automatically generated Text

Description automatically generated



1. concurrentSemaphore.c mengatasi masalah ini dengan memberi tanda semaphore pada kalkulasi yang dilakukan sebuah thread. Sehingga pada critical section, tidak dapat di-overwrite oleh thread yang lain.

Text

Description automatically generated Text

Description automatically generated

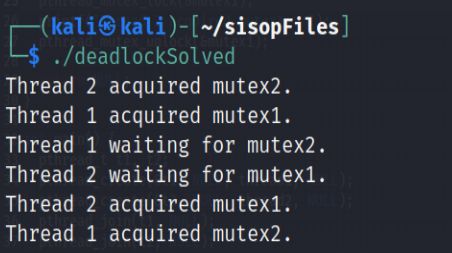


1. deadlock.c akan menunjukkan adanya deadlock dari thread yang saling mengunci resource yang dibutuhkan oleh proses lain. Sehingga kedua thread tidak berhenti bekerja.

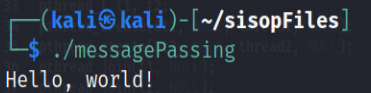
Text

Description automatically generated

1. deadlockSolved.c terdapat penyesuaian pada urutan proses yang mengunci resource. Sehingga deadlock dapat dihindarkan



1. messagePassing.c berisi tentang bagaimana thread saling bertukar pesan. Boleh juga diambilkan dari modul praktikum jika ada.



Terima kasih