

"Jawaban belum tentu benar & hanya sebagai referensi!!!"

Pembahasan UTS Sistem Operasi 2020

~ Nomor 1 ~

① Penggunaan semaphore dapat mengatasi algoritma sleep dan wake up serta memenuhi semua persyaratan solusi untuk mengatasi race condition karena:

* Pada semaphore terdapat 2 operasi atomik

a. Down : jika 0, maka sleep (blocked)

jika > 0, maka decrement value

b. Up : increment value, sleeping process mengesksekusi down.

* Mengatasi permasalahan busy waiting dengan menempatkan proses ke antrian blocked

* Up & down diimplementasikan menggunakan system call dengan menisipkan interrupt.

* Menjamin mutual exclusion & sinkronisasi agar dapat mengatasi race condition

→ Mutex, awal = 1 : menjamin 1 proses dapat mengakses buffer

jika 0 : blocked hingga ada yang up

→ Empty, awal = 1 : memastikan prosedur berhenti bila penuh, menaruh item empty

jika 0 : blocked hingga up setelah remove item

→ Full, awal = 0 : memastikan konsumen berhenti bila kosong, menunggu bil item & full

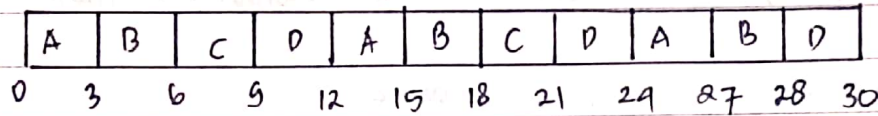
jika 0 : blocked hingga up setelah remove item.

* Count terjaga & menghindari penblock ketika konsumen & produsen sleep

② Semaphore full = 100, mutex = 1 → konsumen masuk ke critical section maka down (& mutex) di tengah critical section (item = remove_item()), Produser juga ingin masuk ke critical section tetapi tidak bisa (mutex = 0) dan belum di up sehingga blocked.

Nomor 2

a) GanttChart



b.) Tabel untuk menentukan rerata waktu tunggu dan rerata turn around time.

Round Robin Ckuanthum = 3)

Nama Proses	Waktu Tiba	Lama Eksekusi	Waktu Tunggu	Turn. Around Time
A	0	9	18	27
B	0	7	21	28
C	0	6	15	21
D	0	8	22	30
Total			76	106
Rata - Rata			19	26.5

~ by Diklat HMIF 2021 ~