NAMA : EDWARD **Mata Kuliah : Operating System**
NIM : 2201741971 **Kode Mata Kuliah : COMP6153**
KELAS : LB-08 **Fakultas / Departemen : School of Computer Science**

I. ESSAY(100%)

1. Process and Interrupts (10%)

Misalkan terdapat suatu proses yang sedang dieksekusi, meminta untuk membaca data dari disk. Jelaskan apa yang terjadi tahap demi tahap. Kaitkan jawaban anda dengan register-register pada sistem komputer dan process state yang bersangkutan. (Catatan: minimal ada 4 tahap).

Jawaban:

a) Fetch Phase

- Pertama, muat Memory Address Register (MAR) dengan isi register dari Program Counter (PC) dan tambah counter register PC sehingga akan mengarah ke instruksi berikutnya.
- Selanjutnya, ambil memori instruksi aktual dari alamat yang diberikan dalam MAR dan tempatkan itu ke dalam Memory Data Register (MDR).
- Pada akhirnya, muat Instruction Register (IR) dengan isi MDR.

b) Decode Phase

- Periksa instruksi untuk mengetahui apa yang prosesor diminta untuk lakukan.
- Memutuskan opcode apa yang perlu diproses di antara opcode-nya.

c) Evaluate Address Phase

- Hitung alamat lokasi memori yang diperlukan untuk memproses instruksi ini.
- Untuk beberapa instruksi, alamat dapat diambil secara langsung.

d) Fetch Operand Phase

- Dapatkan operand sumber yang akan diperlukan untuk memproses instruksi ini.
- Operand dapat berupa memori atau file register.
- Untuk beberapa instruksi seperti LDR, fase ini hanya melibatkan dua langkah dan untuk beberapa bisa lebih.

e) Execute Phase

- Pada fase eksekusi, eksekusi sebenarnya dari instruksi tersebut dilakukan di Arithmetic Logic Unit (ALU).

f) Store Result Phase

- Menyimpan hasil pelaksanaan instruksi di tujuan yang ditentukan.
- Tujuan ini dapat berupa lokasi memori atau register.

NAMA : EDWARD **Mata Kuliah : Operating System**
NIM : 2201741971 **Kode Mata Kuliah : COMP6153**
KELAS : LB-08 **Fakultas / Departemen : School of Computer Science**

2. Multiprocessors and Embedded System (15%)

a. **(10%)** Anda membuat 2 program yang harus memproses data yang cukup besar, sekitar 1 juta record. Anda melakukan pengetesan program menggunakan data yang sedikit dan mengambil kesimpulan sbb:

Proses 1: membutuhkan 1 ms untuk baca, 5 ms untuk proses dan 1 ms untuk tulis.

Proses 2: membutuhkan 3 ms untuk baca, 2 ms untuk proses dan 2 ms untuk tulis.

Karena anda memiliki sedikit informasi mengenai multiprosesor, maka anda memikirkan untuk menggunakan 2 prosesor, agar dapat jalan secara parallel. Hitunglah penghematan waktu yang anda dapatkan jika menggunakan 2 prosesor. Bandingkan ke dua hasil tersebut. Apa kesimpulan anda tentang multi prosesor?

Jawaban:

Jika Menggunakan 1 Prosesor:

a) Proses 1:

totalWaktu = WaktuBaca + WaktuProses + WaktuTulis = 1 + 5 + 1 = 7 ms

b) Proses 2:

totalWaktu = WaktuBaca + WaktuProses + WaktuTulis = 3 + 2 + 2 = 7 ms

Jika Menggunakan 2 Prosesor:

a) Proses 1:

totalWaktu=WaktuBaca+(WaktuProses/2)+WaktuTulis=1+(5/2)+1=4.5 ms

b) Proses 2:

totalWaktu=WaktuBaca+(WaktuProses/2)+WaktuTulis=3+(2/2)+1=5 ms

Kesimpulan:

Jika menggunakan single processor, total waktu eksekusi akan menjadi 7 ms untuk kedua proses, tetapi **jika 2 prosesor digunakan, proses pertama akan berjalan lebih cepat, yaitu lebih cepat 2.5 ms**. **Begitu pula dengan Proses kedua, jika menggunakan 2 Prosesor, akan berjalan lebih cepat 2 ms**.

b. **(5%)** Embedded system biasanya membutuhkan satu atau beberapa sensor. Jelaskan mengapa sistem tersebut membutuhkan sensor. Berikan contoh aplikasinya untuk menguatkan penjelasan anda.

Jawaban:

- Semua Embedded Systems adalah tugas khusus. **Pemutar mp3 hanya akan berfungsi sebagai pemutar mp3**.
- Embedded Systems dibuat untuk melakukan tugas dalam jangka waktu tertentu. **Sistem rem mobil jika melebihi batas waktu dapat menyebabkan kecelakaan**.
- Beberapa Embedded Systems dirancang untuk bereaksi terhadap rangsangan eksternal. **Contohnya Termometer, alat pelacak GPS**.

3. Threads (15%)

Walaupun dengan menggunakan 1 prosesor, penggunaan thread tetap lebih efisien dibandingkan penggunaan multi proses. Namun tentunya menggunakan 2 prosesor akan lebih efisien lagi. Jelaskan secara detail dan jelas, yang dimaksud dengan pernyataan itu beserta alasan dan pembuktiannya.

Jawaban:

Alasan mengapa menggunakan 2 prosesor akan lebih efisien:

- **Pada multiprocessing proses dijalankan secara bersamaan sedangkan pada multithreading hanya satu proses yang dijalankan**. Misal ada 2 Proses di dalam Thread, dia hanya membagi 1 proses menjadi proses yang lebih kecil, sedangkan MultiProcessing, membagi hal tersebut menjadi banyak proses. Jika menggunakan keduanya, tentu akan lebih efisien karena dibagi tugas kecil dan dikerjakan bersamaan.
- **Klasifikasi multiprocessing bersifat sistematis sedangkan multithreading tidak diklasifikasikan**. Buktinya adalah Ketika threading berjalan, maka ada proses yang berat sebelah, ada yang lebih berat, dan ada yang sangat ringan, sedangkan multi prosesor lebih jelas pembagiannya. Jadi, jika keduanya digunakan, maka setelah jelas dibagi oleh multi proses menjadi lebih kecil, maka thread akan lebih cepat juga dan cendurung akan sama, karena di awal sudah dibagi sama rata.

NAMA : EDWARD **Mata Kuliah : Operating System**
NIM : 2201741971 **Kode Mata Kuliah : COMP6153**
KELAS : LB-08 **Fakultas / Departemen : School of Computer Science**

4. **Concurrency (20%)**

a. **(5%)** Bandingkan Semaphore dan Binary Semaphore.

Jawaban:

Binary Semaphore

- Ini hanya dapat memiliki dua nilai, baik 0 atau 1.
- Awalnya, nilainya diinisialisasi ke 1.
- Untuk menyelesaikan masalah bagian kritis dengan beberapa proses.

Semaphore

Digunakan untuk mengontrol dan mengakses sumber daya yang memiliki banyak contoh.

b. **(15%)** Bagaimana Semaphore dan/atau Binary Semaphore dapat digunakan untuk sinkronisasi proses. Berikancontoh aplikasinya.

Jawaban:

Semaphore dalam hal sinkronisasi salah satu contohnya adalah digunakan **untuk memecahkan Producer-Consumer Problem**.

Contoh Penggunaannya:

Untuk mengatasi masalah ini, diperlukan DUA Semaphore:

- Full
  ➔ Semaphore ini untuk melacak jumlah item dalam buffer.
- Empty
  ➔ Semaphore ini untuk melacak jumlah slot kosong di buffer.

Dan juga diperlukan 2 Operasi Berikut:

- wait()
  ➔ Untuk **MENGURANGI value Semaphore Sebanyak 1**.
- signal()
  ➔ Untuk **MENAMBAHKAN value Semaphore Sebanyak 1**.

NAMA : EDWARD **Mata Kuliah : Operating System**
NIM : 2201741971 **Kode Mata Kuliah : COMP6153**
KELAS : LB-08 **Fakultas / Departemen : School of Computer Science**

Langkah 1: Inisialisasi Semaphores:

mutex = 1

Full = 0 // Semua slot di buffer tidak diisi. Jadi, slot penuh sama dengan 0

Empty = n // Awalnya, Semua slot kosong.

Langkah 2: Solusi untuk Producer:

```
do{
        // menghasilkan sebuah item
        wait (Empty);
        wait (mutex);
        // tempatkan di buffer
        signal(mutex);
        signal (Full);
} while (true);
```

Penjelasan:

- Ketika produsen memproduksi satu item maka satu slot akan terisi sekarang, sehingga nilai “kosong” dikurangi 1. Untuk mencegah konsumen mengakses buffer, nilai mutex juga dikurangi.
- Sekarang, produsen telah menempatkan item di buffer.
- Jadi, nilai "penuh" dinaikkan 1.
- Begitu juga nilai mutex juga bertambah 1, hal ini dikarenakan produsen telah menyelesaikan tugasnya dan sekarang konsumen dapat mengakses buffer tersebut.

NAMA : EDWARD **Mata Kuliah : Operating System**
NIM : 2201741971 **Kode Mata Kuliah : COMP6153**
KELAS : LB-08 **Fakultas / Departemen : School of Computer Science**

Langkah 3: Solusi untuk Consumer:

```
do{
        wait (Full);
        wait (mutex);
        // hapus item dari buffer
        signal (mutex);
        signal (Empty);
        // mengkonsumsi item
} while (true);
```

Penjelasan:

- Saat konsumen menghapus item dari buffer, maka nilai "full" dikurangi 1. Demikian pula, nilai mutex juga berkurang sehingga produsen tidak dapat mengakses buffer saat ini.
- Sekarang konsumen sudah mengkonsumsi barang tersebut.
- Jadi, nilai "kosong" dinaikkan 1.
- Demikian pula, Nilai mutex juga dinaikkan sehingga produsen dapat mengakses buffer sekarang.

Pernyataan Producer-Consumer Problem:

Buffer memiliki ukuran tetap. **Produsen dapat menghasilkan suatu barang dan dapat menempatkannya di penyangga**. Demikian pula, **konsumen dapat memilih item dari buffer dan dapat mengkonsumsinya**. Kita **harus mengatur kondisi bahwa ketika produsen menempatkan item di buffer**, maka **konsumen TIDAK boleh mengonsumsi item apa pun pada waktu yang sama**.

NAMA : EDWARD **Mata Kuliah : Operating System**
NIM : 2201741971 **Kode Mata Kuliah : COMP6153**
KELAS : LB-08 **Fakultas / Departemen : School of Computer Science**

5. Process Scheduling (20%)

Anda sedang membangun sebuah Sistem Operasi dan berencana menggunakan algoritma multilevel feedback untuk penjadwalan proses. Anda akan menggunakan 3 antrian. Antrian pertama menggunakan Round Robin dengan quantum= 4 dan antrian ke 2 menggunakan round robin dengan quantum = 8 sedangkan antrian 3 menggunakan FCFC. Menggunakan data dibawah ini, hitunglah ratarata waktu proses dan rata-rata waktu tunggu. Apa yang dapat anda simpulkan dengan menggunakan penjadwalan multilevel feedback?

Process	Waktu Datang	Waktu CPU
A	0	25
B	2	6
C	5	35
D	6	8

Jawaban:

a) Round Robin Dengan Quantum = 4

Process	Burst Time	Arrival Time	Completion Time	TAT	Waiting Time	Response Time
A	~~25~~ ~~21~~ ~~17~~ 13	0	59	59	34	0
B	~~6~~ ~~2~~ 0	2	22	20	14	2
C	~~35~~ ~~31~~ 27	5	74	69	34	7
D	~~8~~ ~~4~~ 0	6	34	28	20	10

TAT → Turn Around Time (CT - AT)

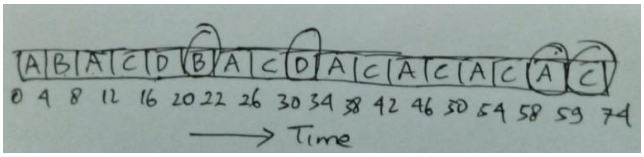
Ready Queue

~~A~~	~~B~~	~~A~~	~~C~~	~~D~~	~~B~~	~~A~~	~~C~~	~~D~~	~~A~~	~~C~~	~~A~~	~~C~~	~~A~~	C

A → ~~25~~ ~~21~~ ~~17~~ ~~13~~ ~~9~~ ~~5~~ ~~1~~ 0

C → ~~35~~ ~~31~~ ~~27~~ ~~23~~ ~~19~~ 15

Running Queue (Grantt Chart)



Rata-rata Waktu Proses = $\frac{59+22+74+34}{4} = \frac{189}{4} = 47.25$

Rata-rata Waktu Tunggu = $\frac{34+14+34+20}{4} = \frac{102}{4} = 25.5$

NAMA : EDWARD **Mata Kuliah : Operating System**
NIM : 2201741971 **Kode Mata Kuliah : COMP6153**
KELAS : LB-08 **Fakultas / Departemen : School of Computer Science**

b) Round Robin Dengan Quantum = 8

Process	Burst Time	Arrival Time	Completion Time	TAT	Waiting Time	Response Time
A	25	0	63	63	38	0
B	~~6~~ 0	2	14	12	6	6
C	35	5	74	69	34	9
D	~~8~~ 0	6	30	24	16	16

TAT → Turn Around Time (CT - AT)

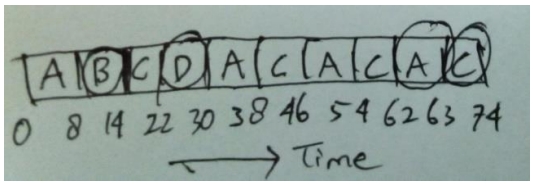
Ready Queue

~~A~~	~~B~~	~~C~~	~~D~~	~~A~~	~~C~~	~~A~~	~~C~~	A	C

A → ~~25~~ ~~17~~ ~~13~~ ~~9~~ ~~5~~ ~~1~~ 0

C → ~~35~~ ~~27~~ ~~19~~ 11

Running Queue (Grantt Chart)



Rata-rata Waktu Proses = $\frac{63+14+74+30}{4} = \frac{181}{4} = 45.25$

Rata-rata Waktu Tunggu = $\frac{38+6+34+16}{4} = \frac{94}{4} = 23.5$

c) FCFS (First Come First Serve)

Process	Burst Time	Arrival Time	Completion Time	TAT	Waiting Time	Response Time
A	25	0	25	25	0	0
B	6	2	31	29	23	23
C	35	5	66	61	26	26
D	8	6	74	68	60	60

Rata-rata Waktu Proses = $\frac{25+31+66+74}{4} = \frac{196}{4} = 49$

Rata-rata Waktu Tunggu = $\frac{0+23+26+60}{4} = \frac{109}{4} = 27.25$

NAMA : EDWARD **Mata Kuliah : Operating System**
NIM : 2201741971 **Kode Mata Kuliah : COMP6153**
KELAS : LB-08 **Fakultas / Departemen : School of Computer Science**

6. Deadlock (20%)

a. **(12%)** Pada suatu saat tertentu pada sebuah sistem komputer, kondisi penggunaan sumber daya untuk proses A, B, C dan D adalah sbb:

Proses	Sumber Daya			
	P	Q	R	S
A	1	0	1	1
B	1	2	0	1
C	1	2	0	2
D	2	2	2	2

Pada komputer tersebut, terdapat 5 unit sumber daya P, 6 unit sumber daya Q, 4 unit sumber daya R, dan 6 unit sumber daya S. Di awal, setiap proses mendeklarasikan kebutuhan sumber dayanya agar proses dapat diselesaikan sbb:

Proses	Sumber Daya			
	P	Q	R	S
A	2	1	3	1
B	2	3	1	1
C	1	2	0	3
D	4	3	3	2

Dengan menggunakan algoritma Banker, tentukan apakah terjadi deadlock, jika tidak tentukan urutan jalannya proses.

Jawaban:

Terdapat 4 sumber daya, yaitu: 5P 6Q 4R 6S

Dengan 4 proses sebagai berikut:

- A = menggunakan 1P 1R 1S meminta 2P 1Q 3R 1S
- B = menggunakan 1P 2Q 1S meminta 2P 3Q 1R 1S
- C = menggunakan 1P 2Q 2S meminta 1P 2Q 3S
- D = menggunakan 2P 2Q 2R 2S meminta 4P 3Q 3R 2S

Total Penggunaan = 5P 6Q 3R 6S

Sisa Sumber Daya = 0P 0Q 1R 0S

Dengan **sisa sumber daya 1R**, maka **tidak dapat menyelesaikan proses manapun** dan **AKAN TERJADI DEADLOCK**.

NAMA : EDWARD **Mata Kuliah : Operating System**
NIM : 2201741971 **Kode Mata Kuliah : COMP6153**
KELAS : LB-08 **Fakultas / Departemen : School of Computer Science**

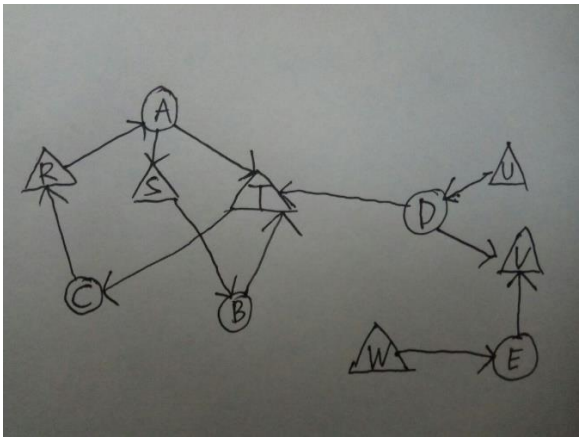
b. (8%) Lima proses berjalan pada sebuah sistem dan melakukan permintaan ataupun sedang memakai sumber daya dengan urutan permintaan / pemakaian sbb:

i. Proses A sedang menggunakan sumber daya R dan dan meminta sumber daya S dan T.
ii. Proses B sedang menggunakan sumber daya S dan dan meminta sumber daya T.
iii. Proses C sedang menggunakan sumber daya T dan dan meminta sumber daya R.
iv. Proses D sedang menggunakan sumber daya U dan dan meminta sumber daya V dan T.
v. Proses E sedang menggunakan sumber daya W dan dan meminta sumber daya V.

Gambarkan Resource Allocation graph dan tentukan apakah terjadi deadlock. Jika ya, sebutkan proses dan sumber daya yang terlibat.

Jawaban:

Resource Allocation Graph:



- Lingkaran menunjukkan proses, Segitiga menunjukkan sumber daya.
- Proses 'C' meminta sumber daya 'R' yang dipegang oleh proses 'A'.
- Pada saat yang sama, proses 'A' meminta resource 'T' yang ditahan oleh proses 'C'.
- **Proses yang terlibat dalam DEADLOCK adalah A dan C.**
- **Sumber daya yang terlibat dalam DEADLOCK adalah R dan T.**