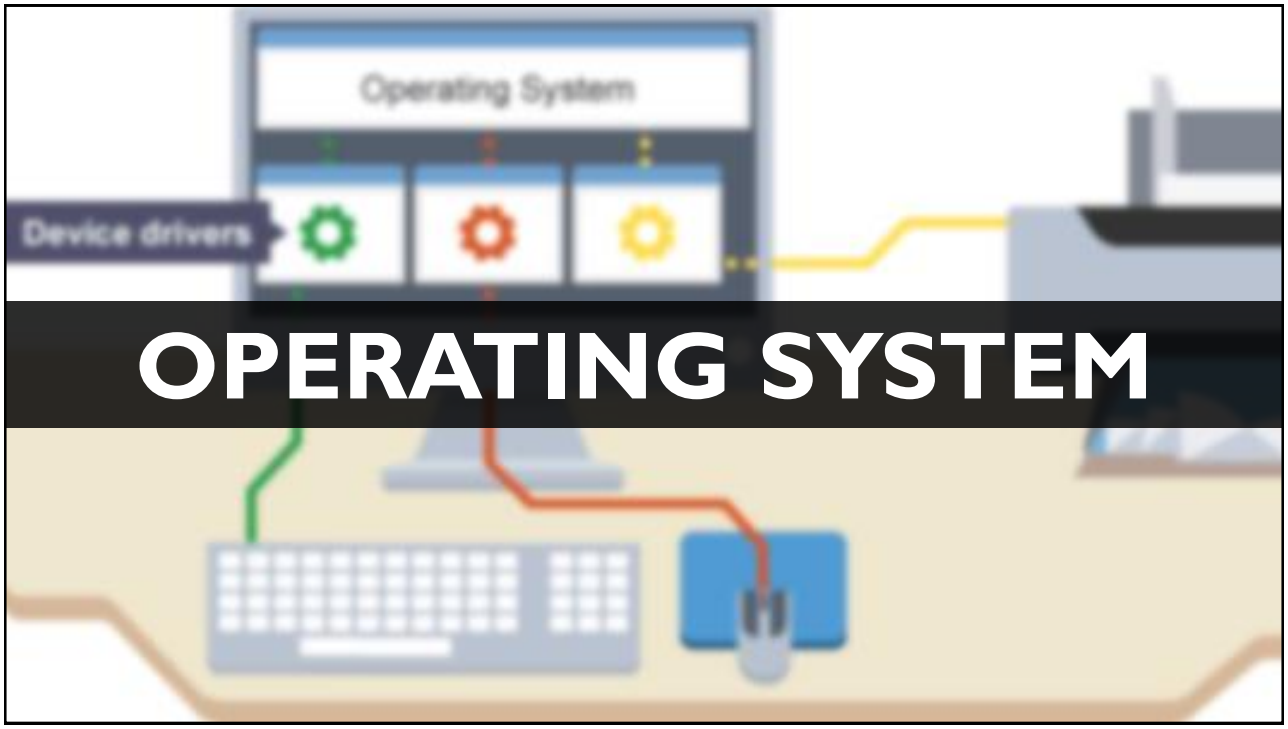


OPERATING SYSTEM

KONKURENSI
OPERATING SYSTEM
BAGIAN 2

CAPAIAN PEMBELAJARAN

- Mahasiswa memahami konsep konkurensi sistem operasi
- Mahasiswa memahami kondisi-kondisi sebagai imbas dari konkurensi
- Mahasiswa memahami penyebab dan penanggulangan konkurensi

Agenda.

- Deadlock
- Startvasion
- Interaksi Antar Proses
- Solusi Konkurensi

4

KONKURENSI | PENGHINDARAN DEADLOCK

Safe State.

Proses	Resources yang Digunakan (MB)	Maksimum resources yang dibutuhkan (MB)
JOB1	4	8
JOB2	3	10
JOB3	5	9
Jumlah resoures tersedia		7

5

KONKURENSI | PENGHINDARAN DEADLOCK

1: Misalkan *job1* mendapatkan alokasi *resources* sebanyak 4 MB

Proses	Resources yang Digunakan (MB)	Maksimum resources yang dibutuhkan (MB)
JOB1	8	8
JOB2	3	10
JOB3	5	9
Jumlah resources tersedia		3

6

KONKURENSI | PENGHINDARAN DEADLOCK

Maka setelah *job1* selesai, kondisi *resources* menjadi:

Proses	Resources yang Digunakan (MB)	Maksimum resources yang dibutuhkan (MB)
JOB1	0	0
JOB2	3	10
JOB3	5	9
Jumlah resources tersedia		11

7

KONKURENSI | PENGHINDARAN DEADLOCK

2: Kemudian *job3* mendapatkan *resources* sebanyak 4 MB

Proses	Resources yang Digunakan (MB)	Maksimum resources yang dibutuhkan (MB)
JOB1	o	0
JOB2	3	10
JOB3	9	9
Jumlah resources tersedia		7

8

KONKURENSI | PENGHINDARAN DEADLOCK

Maka setelah *job3* selesai, kondisi *resources* menjadi:

Proses	Resources yang Digunakan (MB)	Maksimum resources yang dibutuhkan (MB)
JOB1	0	0
JOB2	3	10
JOB3	0	0
Jumlah resources tersedia		16

9

KONKURENSI | PENGHINDARAN DEADLOCK

3: Berikutnya giliran *job2 yang* mendapatkan *resources* sebanyak 7 MB

Proses	Resources yang Digunakan (MB)	Maksimum resources yang dibutuhkan (MB)
JOB1	0	0
JOB2	10	10
JOB3	0	0
Jumlah resources tersedia		9

10

KONKURENSI | PENGHINDARAN DEADLOCK

Maka setelah *job2* selesai, kondisi *resources* menjadi:

Proses	Resources yang Digunakan (MB)	Maksimum resources yang dibutuhkan (MB)
JOB1	o	0
JOB2	0	0
JOB3	0	0
Jumlah resoures tersedia		19

Dengan demikian ketiga proses dapat menyelesaikan prosesnya dengan sempurna.

11

KONKURENSI | PENGHINDARAN DEADLOCK

Usafe State.

Proses	Resources yang Digunakan (MB)	Maksimum resources yang dibutuhkan (MB)
JOB1	4	8
JOB2	3	10
JOB3	4	9
Jumlah resoures tersedia		7

12

KONKURENSI | PENGHINDARAN DEADLOCK

1: Misalkan *job1* dan *job2* masing-masing mendapatkan alokasi *resources* sebanyak 4 MB dan 3 MB

Proses	Resources yang Digunakan (MB)	Maksimum resources yang dibutuhkan (MB)
JOB1	8	8
JOB2	6	10
JOB3	4	9
Jumlah resources tersedia		0

13

KONKURENSI | PENGHINDARAN DEADLOCK

Maka setelah diproses, kondisi *resource* menjadi:

Proses	Resources yang Digunakan (MB)	Maksimum resources yang dibutuhkan (MB)
JOB1	0	0
JOB2	6	10
JOB3	4	9
Jumlah resources tersedia		8

14

KONKURENSI | PENGHINDARAN DEADLOCK

2: Berikutnya giliran *job2* dan *job3 yang* mendapatkan *resources* masing-masing sebanyak 4 MB dan 5 MB

Proses	Resources yang Digunakan (MB)	Maksimum resources yang dibutuhkan (MB)
JOB1	0	0
JOB2	10	10
JOB3	8 (-1)	9
Jumlah resources tersedia		0 (-1)

Karena terdapat sebuah proses yang **kekurangan *resource* disaat *resource* telah habis** maka besar kemungkinan proses pada *job3* tidak akan selesai.

15

KONKURENSI | PENGHINDARAN DEADLOCK

Kasus #1.

Perhatikan tabel di samping ini kemudian gambarkan proses pendistribusian suber daya untuk semua proses agar terhindar dari ***deadlock*** jika pendistribusian sumber daya yang diberikan untuk setiap proses setiap saat hanya 2 MB dan dapat dilakukan oleh lebih dari 1 proses dalam waktuyang bersamaan.

Job	Sumber daya yang dimiliki (MB)	Kebutuhan total sumber daya (MB)
1	2	10
2	4	15
3	3	9
4	8	16
5	1	5
Jumlah sumber daya yang tersisa (MB)		10

16

KONKURENSI | DETEKSI & PEMULIHAN DEADLOCK

- **Deteksi adanya *deadlock***
  Teknik yang digunakan untuk mendeteksi apakah *deadlock* terjadi serta mengidentifikasi proses-proses dan *resource* yang terlibat *deadlock*.

- **Pemulihan dari *deadlock***
  Faktor yang merumitkan pemulihan *deadlock*:
  1. *Deadlock* tidak dapat dideteksi secara cepat
  2. Kebanyakan sistem tidak memiliki fasilitas *suspend* dan *resume* proses.

17

KONKURENSI | DETEKSI & PEMULIHAN DEADLOCK

- **Pendekatan pemulihan *deadlock***
  1. Abaikan semua proses yang terlibat *deadlock*
  2. Lakukan *backup* semua proses yang terlibat *deadlock* ke suatu *checkpoint* yang telah didefinisikan sebelumnya.
  3. Secara berturutan abaikan proses-proses hingga tidak terdapat lagi *deadlock*.
  4. Secara berturutan *preempt resources* hingga tidak terdapat lagi *deadlock*.

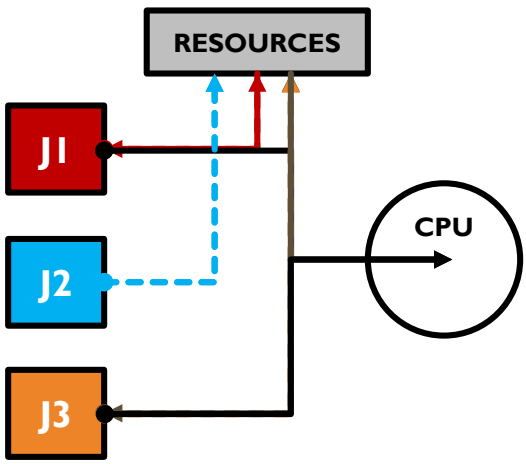
18

KONKURENSI | PENANGANAN DEADLOCK

- **Pendekatan penanganan *deadlock*:**
  1. Kelompokkan *resources* kedalam sejumlah kelas *resources*
  2. Gunakan strategi pengurutan sirkular
  3. Dalam satu kelas *resources*, gunakan algoritma yang paling cocok untuk kelas-kelas *resources* tersebut

19

KONKURENSI | STARTVASION



Suatu kondisi dimana setiap terdapat sebuah *job* yang tidak pernah mendapatkan layanan OS dalam menggunakan *resource*

20

SOLUSI KONKURENSI

1. Mengasumsikan adanya *resource* yang digunakan bersama
2. Tidak mengasumsikan adanya *resource* yang digunakan bersama

Kategori Interaksi.

1. Proses independent
2. Proses saling peduli secara tidak langsung
3. Proses saling peduli secara langsung

21

KONKURENSI

OPERATING SYSTEM

BAGIAN I