

Pembahasan UAS

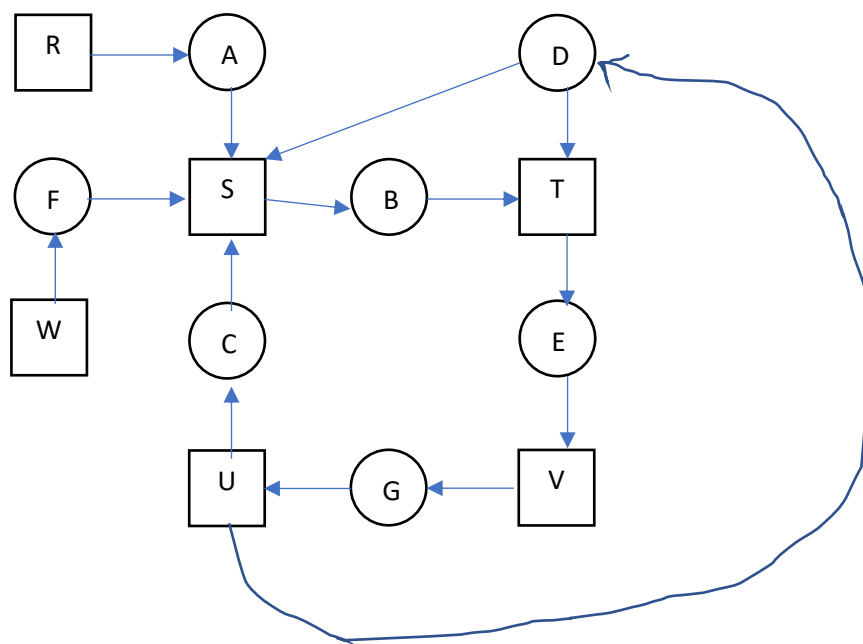
Sistem Operasi 2022

Jawaban hanyalah sebuah referensi, belum tentu benar!

Made with ❤️ by Diklat 2023

SOAL A

1. Resource kotak, proses bulat



Dari graf diatas dapat disimpulkan bahwa terjadi deadlock dikarenakan terjadinya circular wait. Deadlock tersebut terjadi pada proses B, E, G, dan C. Proses B sedang menggunakan resource S dan menginginkan resource T, dimana resource T sedang digunakan oleh proses E dan proses E menginginkan resource V, dimana resource V sedang digunakan oleh proses G dan proses G menginginkan resource U, dimana resource U sedang digunakan oleh proses C dan proses C menginginkan resource S yang dimana resource S sedang digunakan oleh proses B.

2. Banker's algorithm

Proses	Has	Max
A	2	7
B	1	5
C	2	4
D	4	7
Free = 2		

Cari secara berurutan yang jika $has + free = max$, free boleh sisa

A tidak memenuhi

B tidak memenuhi

C memenuhi

Proses	Has	Max
A	2	7
B	1	5
C	4	4
D	4	7
Free = 0		

$Free = free + max (0 + 4 = 4)$

Proses	Has	Max
A	2	7
B	1	5
C	0	-
D	4	7
Free = 4		

D memenuhi

Proses	Has	Max
A	2	7
B	1	5
C	0	-
D	7	7
Free = 1		

$Free = free + max (1 + 7 = 8)$

Proses	Has	Max
A	2	7
B	1	5
C	0	-
D	0	-
Free = 8		

A memenuhi

Proses	Has	Max
A	7	7
B	1	5
C	0	-
D	0	-
Free = 3		

$$\text{Free} = \text{free} + \text{max} (3 + 7 = 10)$$

Proses	Has	Max
A	0	-
B	1	5
C	0	-
D	0	-
Free = 10		

B memenuhi

Proses	Has	Max
A	0	-
B	5	5
C	0	-
D	0	-
Free = 6		

$$\text{Free} = \text{free} + \text{max} (6 + 5 = 11)$$

Proses	Has	Max
A	0	-
B	0	-
C	0	-
D	0	-
Free = 11		

<C, D, A, B> merupakan state safe

Hasil akhir harus free = (total has + free awal); $11 = 2+1+2+4 + 2$; $11 = 9+2$;
 $11=11$;

Kesimpulan:

Dengan menggunakan banker algorithm, state tersebut safe dengan cara proses dijalankan dengan urutan <C, D, A, B>

3. Existing resource (E) = (6 3 4 2)

Processed resource (P) = (5 3 2 2)

Available resource (A) = (1 0 2 0)

Resource Assigned (C) =

3 0 1 1

0 1 0 0

1 1 1 0

1 1 0 1

0 0 0 0

Resource still need (R) =

1 1 0 0

0 1 1 2

3 1 0 0

2 0 1 0

2 1 1 0

Buat table dengan kolom proses, alokasi(C), max, need(R), dan available(A)

Dimana:

max = alokasi + need

need = max - alokasi

(E) adalah hasil akhir dan (P) adalah total alokasi semua proses

Jika need < available maka eksekusi dan available dijumlahkan dengan alokasi dari proses yang dieksekusi.

A: 1100 > 1020 (tidak memenuhi)

B: 0112 > 1020 (tidak memenuhi)

C: 3100 > 1020 (tidak memenuhi)

D: 0010 < 1020 (memenuhi) = A = $A_{n-1} + C = 1020 + 1101 = 2121$

E: 2110 < 2121 (memenuhi) = A = $A_{n-1} + C = 2121 + 0000 = 2121$

A: 1100 < 2121 (memenuhi) = A = $A_{n-1} + C = 2121 + 3011 = 5132$

B: 0112 < 5132 (memenuhi) = A = $A_{n-1} + C = 5132 + 0100 = 5232$

C: 3100 < 5232 (memenuhi) = A = $A_{n-1} + C = 5232 + 1110 = 6342$

Proses	Alokasi	Need	Max	Available	
A	3 0 1 1	1 1 0 0	4 1 1 1	1 0 2 0	
B	0 1 0 0	0 1 1 2	0 2 1 2	2 1 2 1	D
C	1 1 1 0	3 1 0 0	4 2 1 0	2 1 2 1	E
D	1 1 0 1	0 0 1 0	1 1 1 1	5 1 3 2	A
E	0 0 0 0	2 1 1 0	2 1 1 0	5 2 3 2	B
				6 3 4 2	C

Hasil akhir (6 3 4 2) = E (6 3 4 2)

Kesimpulan:

Dengan menggunakan banker algorithm, state tersebut safe dengan cara proses dijalankan dengan urutan <D, E, A, B, C>

4. Empat kondisi penyebab deadlock:

- Mutual exclusion : Adalah suatu keadaan dimana Hanya ada satu proses yang boleh memakai sumber daya, dan proses lain yang ingin memakai sumber daya tersebut harus menunggu hingga sumber daya tadi dilepaskan atau tidak ada proses yang memakai sumber daya tersebut.
- Hold and wait : yaitu dimana proses yang sedang memakai sumber daya boleh meminta sumber daya lagi maksudnya menunggu hingga benar-benar sumber daya yang diminta tidak dipakai oleh proses lain.
- No preemption : yaitu ketika sumber daya yang ada pada sebuah proses tidak boleh diambil begitu saja oleh proses lainnya.

- d. Circular wait : yaitu kondisi yang menyatakan bahwa adanya rantai saling meminta sumber daya yang dimiliki oleh suatu proses oleh proses lainnya

Bagaimana deadlock prevention terjadi yaitu dengan meyakinkan bahwa paling sedikit satu dari kondisi deadlock tidak terjadi yaitu dengan:

- a. Meniadakan mutual exclusion
Dengan melakukan spooling perangkat-perangkat yang didedikasikan ke suatu proses. Dengan spooling, permintaan-permintaan diantrikan di hardisk. Setiap job diantrian spooler akan dialayani satu persatu.
- b. Meniadakan hold and wait
Dengan mengalokasikan semua sumber daya atau tidak sama sekali dan hold and release.
- c. Meniadakan non pre-emption
- d. Meniadakan circular wait
Yaitu dengan proses hanya diperbolehkan menggenggam satu sumber daya. Penomoran global semua sumber daya.