~~P0~~ $< P_1, P_3, P_4, P_0, P_2 >$

Date

From this if there is deadlock we can say that it is unsafe.

Safe sequence may no deadlock.

Check Need $\leq$ Work

~~P0~~

P_0, $743 \leq 332$ X

P_1 $122 \leq 332$ ✓ $W = W + alloc$ (alloc of P_1)

$= 332 + 200$

New work $= 532$

P_2 $600 \leq 532$ X

P_3 $011 \leq 532$ ✓ $W = W + alloc$ (Allo of P_3)

$532 + 211$

$= 743$

P_4 $431 \leq 532$ ✓ 743

$W = W + alloc$

$= 743 + 002$

$= 745$

~~P5~~

P_0 $743 \leq 745$ ✓ $W = W + alloc$

$745 + 010$

755

P_2 $600 \leq 755$ ✓

$W = W + allo$

$755 + 302$

$= 1057$

final work.

Safe sequence $< P_1, P_3, P_4, P_0, P_2 >$

free 3 & 10.

Date

Here 9 tapes are allocated and 3 are free.

If P_0 is currently holding 5, then it can request 5 more in worst case, similarly P_1 can request 2 & P_2 can request 7.

When P_1 is over. ie it will release the 2 tape drives now free becomes $3+2 = 5$ available

$\langle P_1$

ie with this 5, we can satisfy P_0, and once P_0 is over it will releases 5 tape drives and available becomes 10. and this will satisfy P_2's request

thus $\langle P_0, P_1, P_2 \rangle$ will be a safe sequence

Example

Consider a system with 5 process P_0 through P_4 → A B C are 3 resource types

Example

Resource type A has 10 instances
B " 5 "
C " 7 "

allocation means already allocated

at time T_0,

	Allocation (CPU only process)	Max need (Work)	Available (Current)
	A B C	A B C	A B C
P_0	0 1 0	7 5 3	~~3 3 2~~
P_1	2 0 0	3 2 2	~~5 3 2~~
P_2	3 0 2	9 0 2	~~7 4 3~~
P_3	2 1 1	2 2 2	~~7 4 5~~
P_4	0 0 2	4 3 3	~~7 5 5~~ 10 5 7
Total allocated	7 2 5		

Total 10 5 7 − 7 2 5 = 3 3 2

The contents of need matrix is Max need − Allocation

	Need	available
	A B C	A B C
P_0	7 4 3	3 3 2
P_1	1 2 2 ①	
P_2	6 0 0	$\langle P_1, P_3, P_4, P_0, P_2 \rangle$
P_3	0 1 1 ②	
P_4	4 3 1	

3 3 2 → 5 3 2 → 7 4 3 → 7 4 5 → 7 5 5 → 10 5 7

3 3 2 → 5 3 2 → 7 4 3 → 7 4 5 → 7 5 5 → 10 5 7

Work = available

P_0 request $\leq$ available $\quad$ $< P_0, P_2, P_3, P_4$

$000 \leq 000$

$W = W + \text{allocation}$

$= 000 + 010$

$= 010$

P_1 $202 \leq 010$ X

P_2 $000 \leq 010$ ✓

$W = 010 + 303$

$= 313$

P_3 $100 \leq 313$ ✓

$W = 313 + 211$

$= 524$

P_4 $002 \leq 524$ ✓

$W = 524 + 002$

$= 526$

P_1 $202 \leq 526$ ✓

$W = 526 + 200$

$= 726$

726 = equals, the actual available

So no deadlock.

Deadlock avoidance

Construct An algorithm that ensures that the system will never enter a deadlock state. is the deadlock avoidance algorithm.

* A deadlock avoidance alg dynamically examines the resource allocation state to ensure that a circular wait condition can never exist.

Safe state

A state is safe if the system can allocate resources to each process in some order and still avoid a deadlock.

A system is in a safe state only if there exists a safe sequence.

fig 8.4

Safe, unsafe & and deadlock ~~state spaces~~

* A safe state is not a deadlock state

A deadlock state is an unsafe state.

Not all unsafe states are deadlocks.

An unsafe state may lead to a deadlock. — may or may not

Consider a system with 12 tape drives & 3 processes, P_0, P_1, P_2

P_0 requires 10 tape drives

P_1 " 4 "

P_2 " 9 "

Suppose At time t_0, P_0 is holding 5 tape drives

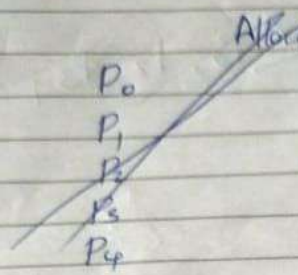
P_1 " 2 "

& P_2 " 2 "

	Maximum needs	Current needs
P_0	10	5
P_1	4	2
P_2	9	2

Date

Deadlock Detection



Total resources A B C
7 2 6

Available A B C		Allocation A B C	Request A B C
0 0 0	P_0 ✓	0 1 0	0 0 0
0 1 0	P_1 ×	2 0 0	2 0 2
3 1 3	P_2	3 0 3	0 0 0
5 2 4	P_3	2 1 1	1 0 0
5 2 6	P_4	0 0 2	0 0 2

Available
A B C
0 0 0
+ 0 1 0
0 1 0
+ 3 0 3
3 1 3
+ 2 1 1
5 2 4
+ 0 0 2
5 2 6
7 2 6

We have allocation matrix, request matrix & available vector.

At the initial state any of the allocation values are all 0 0 0. So any of the finish variables to be true at this time..

Work = allocation Available
= 0 0 0

Check request ≤ available

Look at a process whose request is 0 0 0
mark P_0 as done
add 0 1 0 to available

P_0 ✓

P_1 ✓

P_2 ✓

P_3 ✓

P_4 ✓

$\langle P_0, P_2, P_3, P_4, P_1 \rangle$

No deadlock is detected

Date

deadlock detection

* Allow the system to enter deadlock state

detection

Single instance (wait for graph) → (Detect cycle)

Multiple instances (Variant of Banker's alg)

* Detection of cycle is necessary & sufficient condition for deadlock

Safety alg

size of work m
" " Finish n

work → available

ith process

P1	P2	P3
T	T	T

safe

one process false means unsafe