20/May/2021.

Bankers algo.

deadlock avoidance.

→ Safety algo. (To find. Safe. Sequence. / or to check a system is in Safe ~~state~~ or not.

+

Resource allocation algo.

Problem with deadlock.

P_{-1} P_0 P_1 (waiting) P_2.

Higher priority

Medium

→ lower.

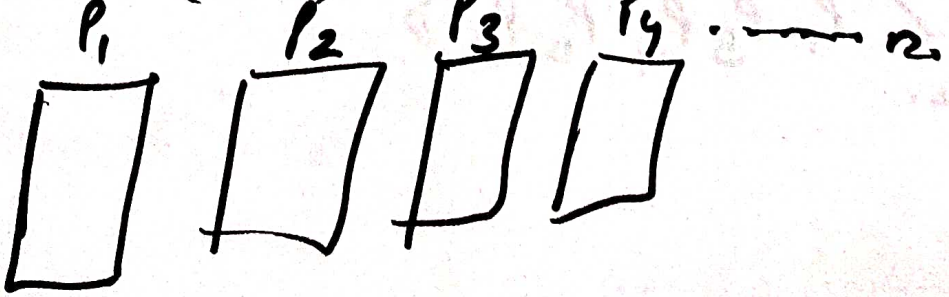
→ Starvation. (problem).

→ Ageing (solution).

(increase priority by 1).

Process identification Number.

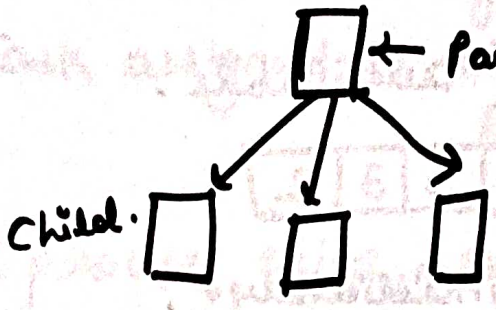
(Pid.)



UNIT 2

Process Synchronization..

Process creation



fork() / Createprocess()
{System call}

⇒ Execution

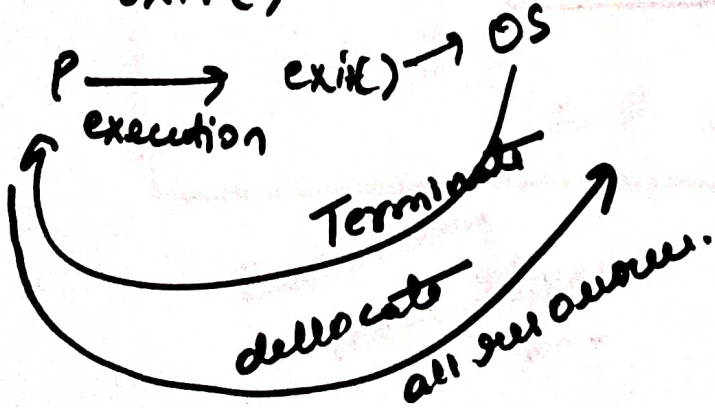
1. Execute concurrently (P. process & C. process).
2. P. process waits until some or all children have been terminated.

⇒ Address space.

1. C. process Duplicate of parent process.
2. C. process has new program loaded into it.

Process Termination

exit()



P. Process May Terminate C. process.

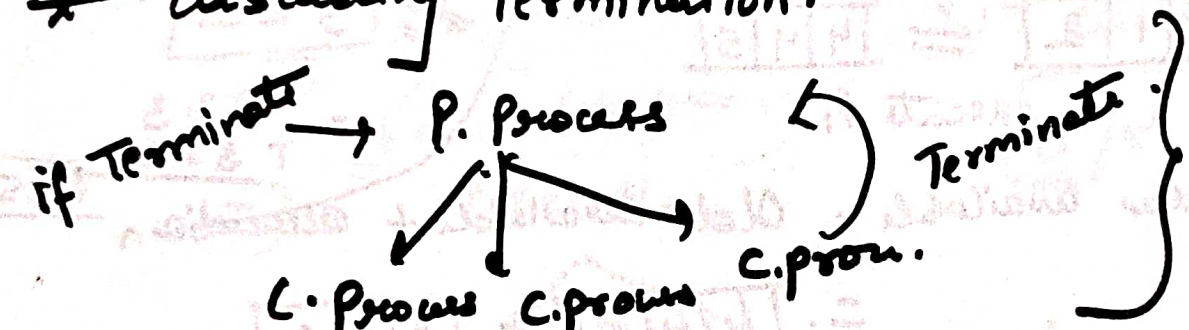
=) Reasons

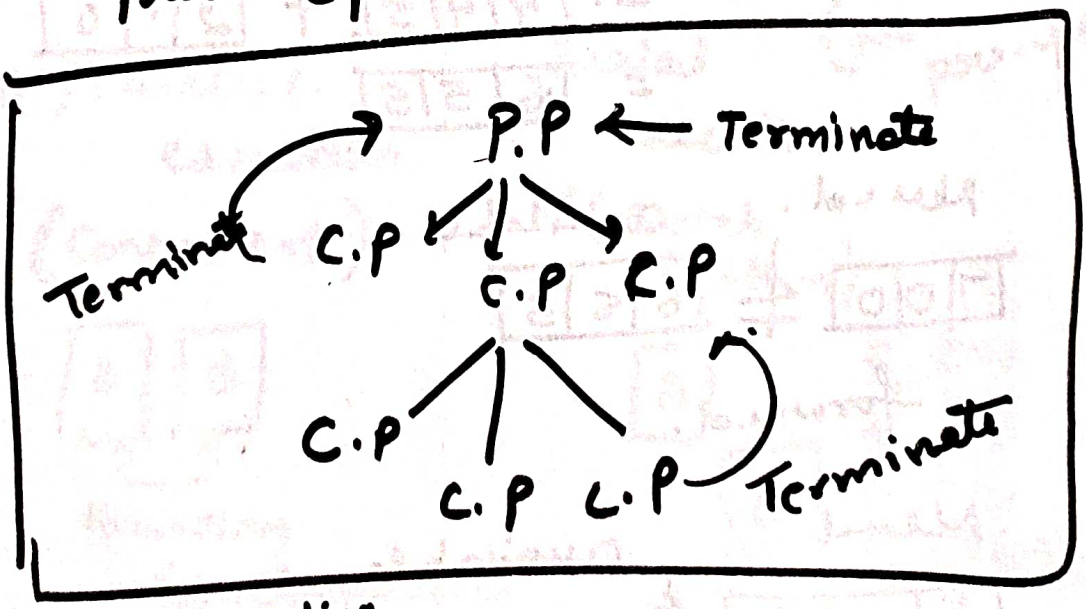
(1) child exceeded usage of given resources.

(2) Task assigned to child is not longer Required.

(3) P. process exits, OS not allows C. process to continue.

* Lascading Termination.



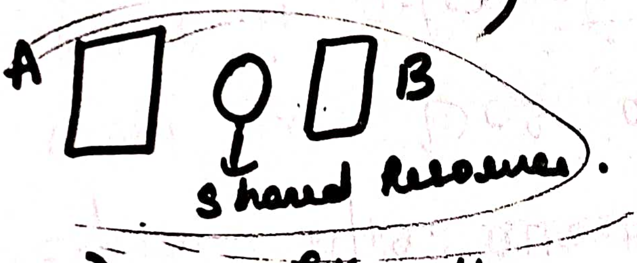


Cascading Termination.

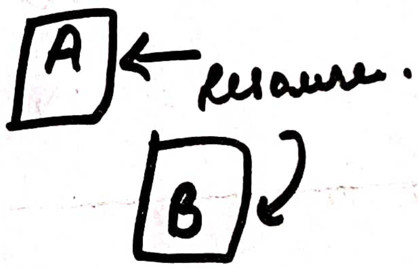
Process Synchronization.

⇒ Process

- (Parallel execution) Cooperative process
  (execution of process affects Other's execution)

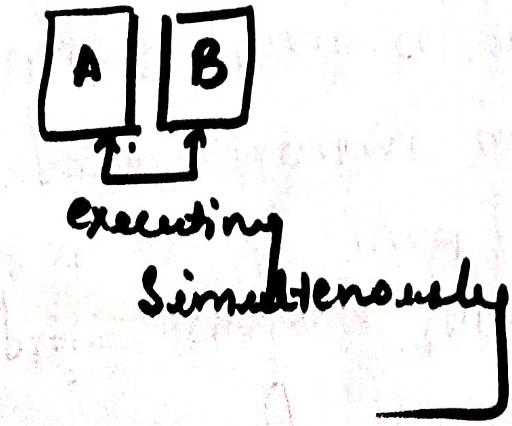


- (Serial execution) Independent process. (Non preemptive)
  (does not affect Other).

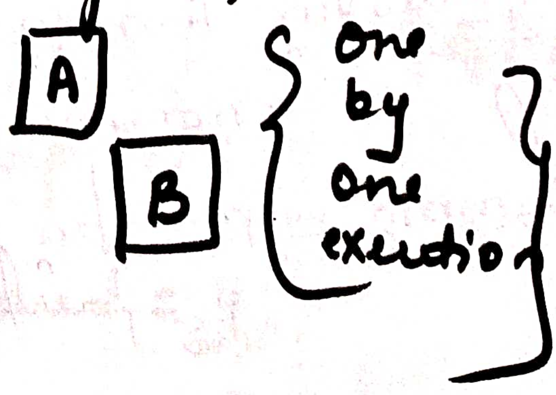


⇒ Execution.

- (preemption) Parallel. execution
  (Concurrent)



- Serial execution. (Non preemption).
  ex. (Batch System)



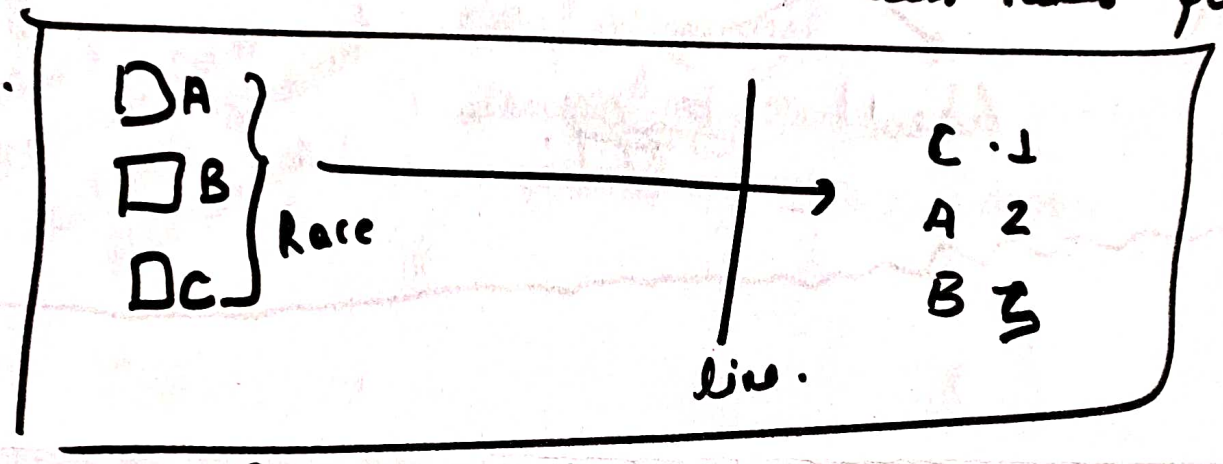
Race Condition.

⇒ Output is depends on the serial of input.

=) Several processes, access and Manipulate the same data Concurrently & Outcome of execution depends on particular. Order in which the access takes place.

$P_1 P_2 P_3 \rightarrow$ □

ex.



ex ≡

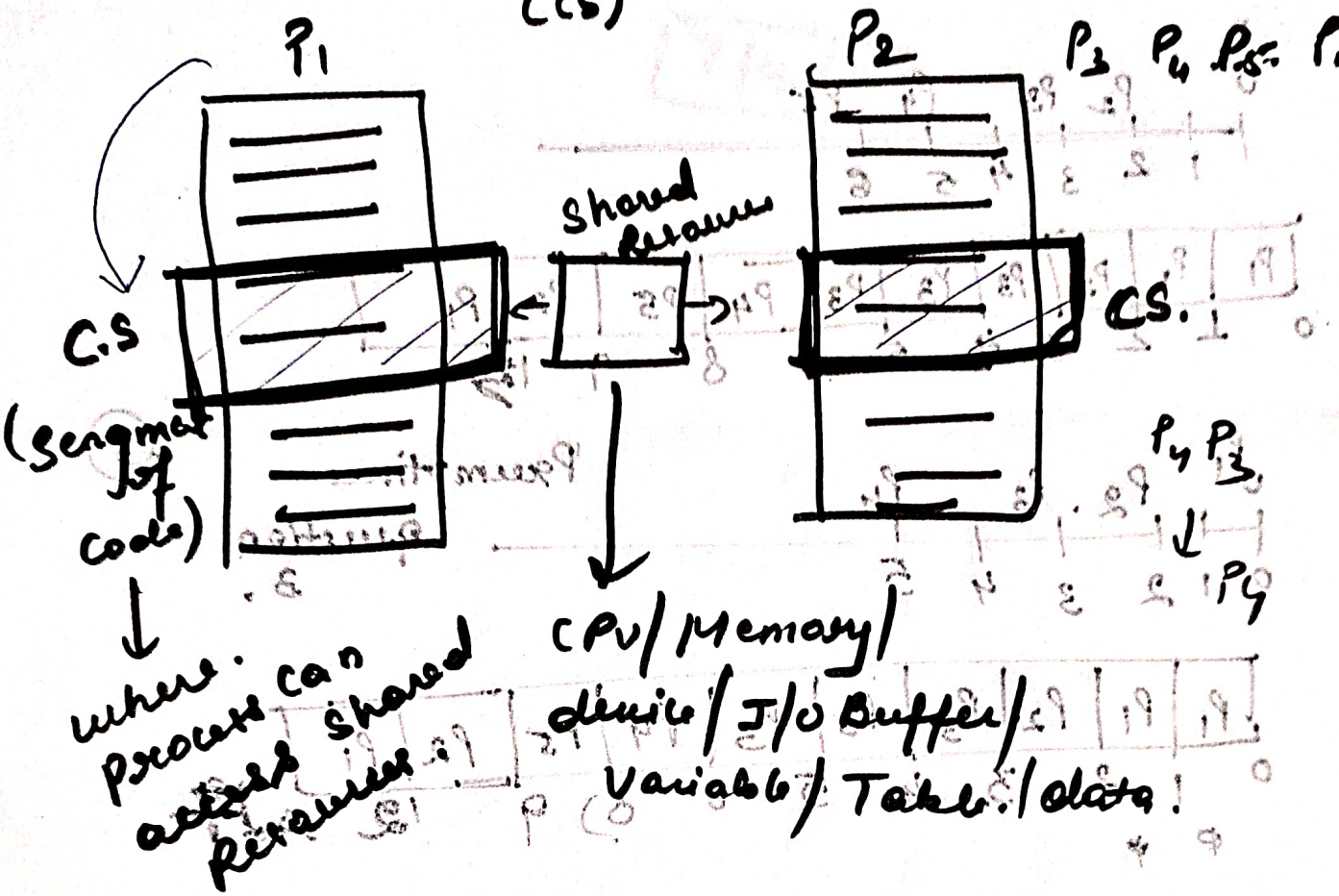
Global var = (shared = ~~5~~ 6 4)

P_1	P_2
1 int x = shared; (5, 5)	int y = shared; (5, 5)
2 x++ 6;	y++ 4;
3. sleep(1)	sleep(1);
4 Shared = x (6, 6)	Shared = y; (4, 4)

$P_1 P_2 P_1 P_2$

Race condition

Critical Section. problem. (CS)



=) Critical Section problem. Solution.

1. Mutual Exclusion.
2. Progress.
3. Bounded wait.