DINING PHILOSOPHER PROBLEM.

AIM:

TO WRITE A PROGRAM FOR DINNING PHILOSOPHER PROBLEM USING C LANGUAGE.

The dining philosophers problem is another classic synchronization problem which is used to evaluate situations where there is a need of allocating multiple resources to multiple processes.

the **dining philosophers problem** is an example problem often used in [concurrent](https://en.wikipedia.org/wiki/Concurrency_(computer_science)) algorithm design to illustrate [synchronization](https://en.wikipedia.org/wiki/Synchronization_(computer_science)) issues and techniques for resolving them.

ALGORITHM:

->we use the semaphore to solve the dining hilosopher problem which represents the fork.

->A fork can be picked by executing wait operation on semaphore and released by executing a single semaphore.

->Initially the element of fork are initialised to 1 as the forks are on the table and not picked by philisophers.

->the first wait operation is performed on fork[i] and fork[(i+1)%5].

->Which means that philosopher i has picked up the fork on his sides.Then eating action is performed.

->After that signal operation is performed on fork[i] and fork[(i+1)%5].That the philosopher i has ate and put down the fork on his sides.And he goes thinkging and this process will be continued.

Graphical user interface, application, Word

Description automatically generated

PROGRAM:

Graphical user interface, text, application

Description automatically generated

OUTPUT:

Table

Description automatically generated

RESULT:

DINING PHILOSOPHER WAS EXCUTED SUCCESSFULLY USING C LANGAUGE.