ADVANCED SYSTEMS PROGRAMMING

ASSIGNMENT – 3

NAME: CHARAN TEJA GANDHAM

UFID: 19343642

Email Id: gcteja@ufl.edu

PROBLEM STATEMENT:

In this Assignment , we are required to implement dining philosopher's problem where each philosopher is run as a separate process and synchronization between these processes should be achieved using Pthread library's PTHREAD_PROCESS_SHARED features and mmap/munmap system calls

IMPLEMENTATION:

This Assignment is implemented by this program

Host: This program consist of functionalities of both Host and N Philosophers mentioned in the problem .From the command line number of philosophers N and number of iterations M per Philosopher is given then Host program creates N semaphores indicating N forks, a semaphore of count l indicating number of philosophers allowed per critical section and a barrier semaphore of count N to make sure every philosopher passes through the barrier. Now N philosopher processes are initiated using fork and are iterated M times .In the critical section each takes both the adjacent forks and state changes proceeds from THINKING, HUNGRY TO EATING.

sem.h: This header file consist of shared data structures used in host.c inorder to achieve synchronization among processes .this file consist of semaphore structures with resemblance to the online code provided . Additional to that function such as barrier creation, initiation and wait are also implemented in this header file.

How To Run:

1. Download the compressed file named Assignment3_ID19343642
2.Extract the file and go to that folder
3.Enter make command
4.Execute the program the following command
./Assignment3_ID19343642 make
Complilation done
./Assignment3_ID19343642 #Noofphilospher #Noofiterations
E.g.:/ Assignment3_ID19343642 10 10

Output: Output is displayed as following printf statements

" I th philosopher is in iteration J and he is THINKING"

" I th philosopher is in iteration J and he is HUNGRY"

" I th philosopher is in iteration J and he is EATING"