ADVANCED SYSTEMS PROGRAMMING
ASSIGNMENT – 3

DETAILS:

Name : Sai Vishnu Teja Vempali
UFID : 16141381
email : vishnu24@ufl.edu

INTRODUCTION :

The assignment consists of three files. Two cpp files and one header file.
* host.cpp
* philosopher.cpp
* sem.h

The assignment implements the solution to the Dining Philosopher's problem using shared memory for the philosophers. Here, each philosohper is a separate process and the inter-process communication between them is achieved using the PTHREAD_PROCESS_SHARED feature for mutex and condition variables and mma/munmap system calls.

IMPLEMENTATION:

Host:

The host.cpp file initializes the shared data structures for the N philosophers. It initializes four shared data structures.
1. Shared semaphore data structure for the forks(N)
2. Shared semaphore data structure for the barrier(1)
3. Shared semaphore data structure for the state of each philosopher
4. Shared semaphore data structure for all the philosophers together.

It then creates the N philosophers using "execv" system call and sends necessary arguments for the philosopher.

Philosopher:

The philosopher.cpp implements a philosopher that sits a loop which executes M times(M is passed as an argument) and each time the philosopher prints its current state to the terminal.

Sem:

The sem.h header file consists of the shared data structures and their definitions. It also declares and defines all the functions that are used by these shared data structures.

INPUT:

The input to the host program is two integer values N(= No of philosophers) amd M(= No of times each philosopher should execute). These are passed as command line arguments.

OUTPUT:

Each philosopher prints his current state on the terminal.

HOW TO RUN:

1. Download the zip file containing all the required files.

2. Extract the files.

3. Copy the Assignment3 directory to a convenient location.

4. Change directory to Assignment3 In the command prompt:

> cd Assignment3

> make

> ./host No_of_philosophers No_of_times_philosopher_runs

Sample : ./host 10 10