CSE231 – Operating Systems

Assignment – 3

Question 1 – Dining Philosophers

- Semaphores and Locks are used to ensure mutual exclusion and atomicity of the critical section between the threads (philosophers).
- Binary semaphores are used between the threads which represent each fork and counting semaphore is used for implementation of two sauce bowls as required by the problem.
- **Possible Deadlock Situation:** Deadlock can occur if each philosopher sitting on the table holds the fork on their left or each philosopher sitting on the table holds the fork on their right, this will create a case of circular weight.
- **Solution:** We can change the order in which each philosopher grabs their respective fork, if even numbered philosopher grab the fork on their left first and then grab the fork on their right, and odd numbered philosophers grab the fork on their right first and the fork on their left is held latter.
  This will deter the scenario in which the philosophers have their hand on only one fork.
- **Possible Deadlock situation with sauce bowl:** Deadlock will occur when a Philosopher X has two forks, Philosopher Y has one fork, and one sauce bowl and Philosopher Z only has a sauce bowl.
- **Solution:** A philosopher can only proceed to have a fork only if he has a sauce bowl.

Question 2 – Interprocess Communication (IPC)

- Program P1 generates an array of 50 randomly generated string and sends 5 strings along with their index to process P2 through the desired gateway using a structure (custom made) which holds the string and its index.
- Program P2 receives the strings and selects the highest index of string from the received packet and returns the index to the waiting process P1 which performs next iteration after receiving the same.
- This is implement using FIFO, shared memory, and UNIX domain sockets
- Sender and receiver functions are made in both the programs which do their respective tasks of sending string and indexes to each other
- `mkfifo` and knowledge of file descriptors is used for IPC by FIFO
- `shmget()` to get key of shared memory, `shmat()` to get pointer to memory and `shmdt()` to detach the process from memory [IPC through Shared Memory]
- `socket()` to create socket, `bind()` – binds socket to address in UNIX domain, `listen()` – listen to incoming connections from client program, `accept()` – accept connection, `recv()` – receive data, `send()` – send data and `close()` to close the socket connection between server and client [IPC through UNIX domain socket]
- Reference: https://beej.us/guide/bgipc/

Question 3 – Kernel Module

- It is required to be run by root user and with kernel having build (6.0.8 in my case)
- We use `task_struct` (which is maintained by the kernel) to get details about all the current working processes in the system.
- `for_each_process()` function is used to iterate over the list wherein we use `strstr()` function to find the desired process and print the required detail about the process.
- `module_init()` is called when we load the module which class a function to do the desired ask and `module_exit()` to unload it as required by the question.