Operating Systems: Assignment-3
Harshil Mital (2021050)
December 2022

1. Dining Philosophers Problem

a)

1) Deadlock prevention using Strict Ordering

Philosophers are represented using pthread threads and forks are represented using pthread mutex locks.
A deadlock is a situation in which all philosophers wait for the other philosopher to release a fork, and are thus stuck.
To prevent this the following logic has been implemented:
If a philosopher is even numbered they will pick the left fork first and then the right one.
If a philosopher is odd numbered they will pick the right fork first and then the left one.

This makes it so that the situation of deadlock is made impossible.

2) Deadlock prevention using Semaphores

Philosophers are represented using pthread threads and forks are represented using binary semaphores. There is another counting semaphore whose value is initialized to 4.
To prevent a deadlock the counting semaphore `mutex` a maximum of 4 philosophers to try and pick forks at one time.
This makes it so that it is impossible for all 5 philosophers to pick forks at the same time and thus be stuck in a deadlock.

b) Deadlock prevention using Semaphores (modified problem)

The implementation is identical to the previous question. The only difference is the presence of another counting semaphore `bowls` initialized with value 2.

2. Interprocess Communication using Sockets, FIFOs and shared memory

a) Sockets

Unix domain sockets have been used to transfer 50 strings from 2a_p1.c to 2a_p2.c

b) FIFOs

FIFOs have been used to transfer 50 strings from 2b_p1.c to 2b_p2.c

c) Shared memory

Shared Memory has been used to transfer 50 strings from 2c_p1.c to 2c_p2.c

3. Kernel Module to print `task_struct`

taskstruct-1 directory created in /root/new_kernel/linux-5.19.9
Inside it taskstruct-1.c and Makefile created.
Make command is run

To install module run:
insmod taskstruct-1.ko <pid>

To see output:
dmseg

To remove module:
Rmmod taskstruct.ko