CS-362L Operating System Lab 04

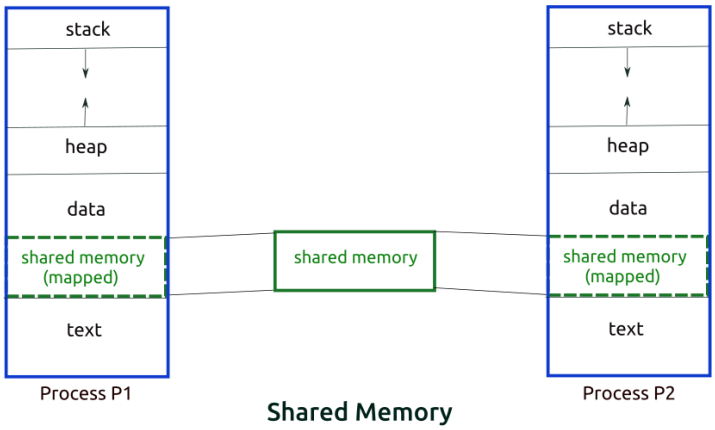
Assignment 4

Inter-process Communication Using Shared Memory.

Objectives: To understand what is inter-process communication and implement it using shared memory.

What is inter-process communication?
IPC (inter-process communication) provides a way for the processes to communicate with each other. Processes executing concurrently in the operating system might be either independent processes or cooperating processes. A process is independent if it cannot be affected by the other processes executing in the system.

Shared Memory
In the shared-memory model, a region of memory which is shared by cooperating processes gets established. Processes can be then able to exchange information by reading and writing all the data to the shared region.Shared memory permit processes to communicate by simply reading and writing to a specified memory location.



Step 1: Creation/Initialization of Shared Memory
To create a new shared memory, the function **shmget()** is used. The syntax is as follow:
shmget(key_t key, size_t size, int shmflg);

- key_t key: a unique key to shared memory.
- size_t size: size of shared memory.
- shmflg: IPC_CREAT—0666 to create new shard memory.

- It retuns the value of identifier associated with shared memory. On failure, it returns -1.

Step 2: Attach To Shared Memory

To attach the process to shared memory, **shmat()** function is used. The syntax is given below:

void *shmat(int shmid, const void *shmaddr,int shmflg);

shmat() attaches the shared memory segment identified by shmid to the address space of the calling process. The attaching address is specified by shmaddr. If shmaddr is NULL, the system chooses a suitable (unused) address at which to attach the segment.

The value of flag is set to 0.

Step 3: Write Into Shared Memory

Write into the shared memory using **memcpy()** function. The syntax is as follow:

memcpy(pointer (returned by shmat), string, size)

Step 4: Detach process.

shmdt() is used to detach the process from shared memory. The syntax is:

shmdt(pointer)

Header Files Used:

```
#include <sys/types.h>
#include <sys/shm.h>
#include <string.h>
#include <sys/ipc.h>
#include <stdio.h>
```

How it works?

- Create two processes. One for writing into shared memory (write.c), other for reading from shared memory (read.c).
- Write process creates a shared memory of size defined by you and attaches the shared memory.
- Write process writes data (string) into shared memory.
- After writing, write process detach itself.
- Repeat the same steps for read process. Instead of writing, read from shared memory.

Conclusion: At the end of this lab, student will be able to implement inter-process communication using shared memory.

Class Activity

1. Write a C program to implement inter-process communication between two processes using shared memory.

Exercise

1. Write a C/C++ program to demonstrate inter-process communication using shared memory. Implement the following:
   Process 1 creates the shared memory and writes some string into shared memory then waits for other process to read and modify the contents of shared memory.
   Process 2 gets the shared memory created by process 1, read the string, display the actual string and reverse of the string and modifies the content of shared memory.

References:

- ”Shared Memory Create/Initialize”
  https://linux.die.net/man/2/shmget
- ”Shared Memory”
  https://www.tutorialspoint.com/inter_process_communication/inter_process_communication_shared_memory.htm