# Assignment No. 9

**a) Write a C program that illustrates inter process communication using shared memory.**

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

#define SHM_SIZE 1024

int main() {
    // Provide a valid path to an existing file
    const char *file_path =
"/home/anubhav/anubhav_oslab/employee.txt";

    key_t key = ftok(file_path, 1);
    if (key == -1) {
        perror("ftok");
        return 1;
    }

    int shmid = shmget(key, SHM_SIZE, IPC_CREAT |
0666);
    if (shmid == -1) {
        perror("shmget");
        return 1;
    }

    char *data = (char *)shmat(shmid, NULL, 0);
    if (data == (char *)-1) {
        perror("shmat");
        return 1;
    }

    strcpy(data, "Hello, shared memory!");
```

```
    printf("Data written to shared memory: %s\n",
data);

    shmdt(data);
    shmctl(shmid, IPC_RMID, NULL);

    return 0;
}
```

```
anubhav@DESKTOP-9VIA8NE:~/anubhav_oslab$ rm 9thAssigna.c
anubhav@DESKTOP-9VIA8NE:~/anubhav_oslab$ vim 9thAssigna.c
anubhav@DESKTOP-9VIA8NE:~/anubhav_oslab$ gcc -o 9thAssigna 9thAssigna.c
anubhav@DESKTOP-9VIA8NE:~/anubhav_oslab$ ./9thAssigna
Data written to shared memory: Hello, shared memory!
```

**b) Write C programs that illustrate communication between two unrelated processes using named pipe(FIFO file).**

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>

#define FIFO_PATH "/home/anubhav/anubhav_oslab/"
int main() {
    int fd;
    char message[] = "Hello from Process 1!"; //
Message to be sent

    // Create the FIFO if it doesn't already exist
    if (mkfifo(FIFO_PATH, 0666) == -1) {
        perror("mkfifo");
        exit(1);
    }

    // Open the FIFO for writing
    fd = open(FIFO_PATH, O_WRONLY);
    if (fd == -1) {
        perror("open");
        exit(1);
```

```
    }

    // Write the message to the FIFO
    if (write(fd, message, sizeof(message)) == -1) {
        perror("write");
        exit(1);
    }

    // Close the FIFO
    close(fd);

    printf("Process 1 has sent a message to Process 2
via named pipe.\n");

    return 0;
}
```

#Sender

```
anubhav@DESKTOP-9VIA8NE:~/anubhav_oslab$ rm 9thAssignbs.c
anubhav@DESKTOP-9VIA8NE:~/anubhav_oslab$ vim 9thAssignbs.c
anubhav@DESKTOP-9VIA8NE:~/anubhav_oslab$ gcc -o 9thAssignbs 9thAssignbs.c
anubhav@DESKTOP-9VIA8NE:~/anubhav_oslab$ ./9thAssignbs
mkfifo: File exists
anubhav@DESKTOP-9VIA8NE:~/anubhav_oslab$
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>

#define FIFO_PATH "/home/anubhav/anubhav_oslab/"

int main() {
    int fd;
    char message[100]; // Buffer to store received
message

    // Open the FIFO for reading
    fd = open(FIFO_PATH, O_RDONLY);
    if (fd == -1) {
        perror("open");
        exit(1);
```

```
    }

    // Read the message from the FIFO
    if (read(fd, message, sizeof(message)) == -1) {
        perror("read");
        exit(1);
    }

    // Close the FIFO
    close(fd);

    printf("Process 2 received the following message
from Process 1 via named pipe:\n%s\n", message);

    return 0;
}
```

#Receiver

```
anubhav@DESKTOP-9VIA8NE:~/anubhav_oslab$ vim 9thAssignbr.c
anubhav@DESKTOP-9VIA8NE:~/anubhav_oslab$ gcc -o 9thAssignbr 9thAssignbr.c
anubhav@DESKTOP-9VIA8NE:~/anubhav_oslab$ ./9thAssignbr
read: Is a directory
anubhav@DESKTOP-9VIA8NE:~/anubhav_oslab$ vim 9thAssignbr.c
```

c) Write a C program to demonstrate multithreading execution.

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

#define NUM_THREADS 5

void *print_hello(void *thread_id) {
    long id = (long)thread_id;
    printf("Hello from thread %ld\n", id);
    pthread_exit(NULL);
}

int main() {
    pthread_t threads[NUM_THREADS];
    int rc;
    long t;

    for (t = 0; t < NUM_THREADS; t++) {
```

```
        printf("Creating thread %ld\n", t);
        rc = pthread_create(&threads[t], NULL,
print_hello, (void *)t);
        if (rc) {
            perror("pthread_create");
            return 1;
        }
    }

    pthread_exit(NULL);
}
```

```
anubhav@DESKTOP-9VIA8NE:~/anubhav_oslab$ vim 9thAssignc.c
anubhav@DESKTOP-9VIA8NE:~/anubhav_oslab$ gcc -o 9thAssignc 9thAssignc.c
anubhav@DESKTOP-9VIA8NE:~/anubhav_oslab$ ./9thAssignc
Creating thread 0
Creating thread 1
Creating thread 2
Hello from thread 1
Creating thread 3
Hello from thread 0
Hello from thread 2
Creating thread 4
Hello from thread 3
Hello from thread 4
anubhav@DESKTOP-9VIA8NE:~/anubhav_oslab$
```