**. Inter-process Communication using Shared Memory using System**

#include <sys/ipc.h>

#include <sys/shm.h>

#define PROJECT\_ID 220

#define READ\_BY\_CLIENT 0

#define WRITTEN\_BY\_SERVER 1 #define ARRAY\_LENGTH 5

typedef struct SharedMemory {

int status;

int array[ARRAY\_LENGTH];

}

SharedMemory; key\_t getKey() { return ftok(".", PROJECT\_ID);

} int shm\_init() { return shmget(getKey(), sizeof(SharedMemory),

IPC\_CREAT | 0666);

}

SharedMemory \* attach(int shm\_id) { return (SharedMemory \* ) shmat(shm\_id, NULL, 0);

}

int detach(SharedMemory \* shm) {

return shmdt((void \* ) shm);

}

**// reader.c**

#include <stdio.h>

#include <stdlib.h> #include "SharedMemory.c" int main() { int shm\_id, i; if ((shm\_id = shm\_init()) == -1) { perror("Error occured while initialising Shared Memory\n"); exit(-1);

}

SharedMemory \* mSharedMemory = attach(shm\_id); if

(mSharedMemory -> status == READ\_BY\_CLIENT) {

printf("Server hasn't written value yet\n"); exit(-1);

}

printf("Printing %d Numbers\n", ARRAY\_LENGTH); for (i = 0; i < ARRAY\_LENGTH; i++) { printf("%d\n", mSharedMemory -> array[i]);

}

mSharedMemory -> status = READ\_BY\_CLIENT; if (detach(mSharedMemory) == -1) { perror("Error occured while detaching Shared memory\n"); exit(-1);

}

#include <stdio.h>

#include <stdlib.h> #include "SharedMemory.c" int main() { int shm\_id, i; if ((shm\_id = shm\_init()) == -1) { perror("Error occured while initialising Shared Memory\n"); exit(-1);

}

SharedMemory \* mSharedMemory = attach(shm\_id); if (mSharedMemory -> status == WRITTEN\_BY\_SERVER) {

printf("Client hasn't read value yet\n");

exit(-1);

}

printf("Enter %d Numbers\n", ARRAY\_LENGTH); for (i = 0; i < ARRAY\_LENGTH; i++) { scanf("%d", & mSharedMemory -> array[i]);

}

mSharedMemory -> status = WRITTEN\_BY\_SERVER; if (detach(mSharedMemory) == -1) { perror("Error occured while detaching Shared memory\n"); exit(-1);

} char c; printf("Press any key to exit\n");

scanf(" %c", & c); }

}