**EXPERIMENT – 10**

10. Illustrate the concept of inter-process communication using message queue with a C

program.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/ipc.h>

#include <sys/msg.h>

#include <unistd.h>

#define MAX 100

struct msg\_buffer {

long msg\_type;

char msg\_text[MAX];

};

int main() {

key\_t key;

int msgid;

key = ftok("progfile", 65); // Ensure "progfile" exists or use another method

msgid = msgget(key, 0666 | IPC\_CREAT);

if (msgid == -1) {

perror("msgget");

exit(1);

}

if (fork() == 0) {

struct msg\_buffer message;

message.msg\_type = 1;

strcpy(message.msg\_text, "Hello from child process!");

if (msgsnd(msgid, &message, sizeof(message.msg\_text), 0) == -1) {

perror("msgsnd");

} else {

printf("Child: Message sent.\n");

}

} else {

sleep(1); // Ensure child sends first

struct msg\_buffer message;

if (msgrcv(msgid, &message, sizeof(message.msg\_text), 1, 0) == -1) {

perror("msgrcv");

} else {

printf("Parent: Received message: %s\n", message.msg\_text);

}

msgctl(msgid, IPC\_RMID, NULL);

}

return 0;

}

**SAMPLE OUTPUT:**

Child: Message sent.

Parent: Received message: Hello from child process!