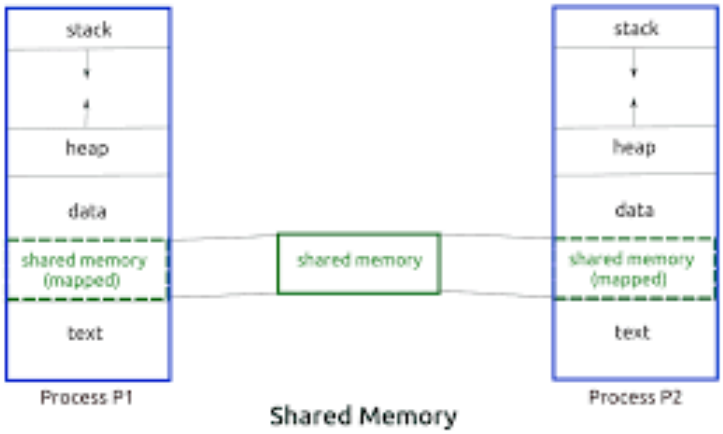
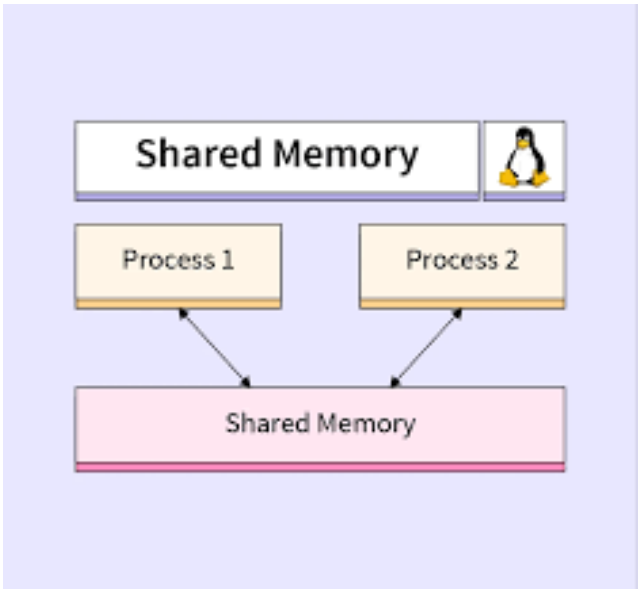
Bhartiya
Vidya Bhavan's
Sardar Patel Institute of Technology
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India
(Autonomous College Affiliated to University of Mumbai)

NAME:	Isha Bamel
UID:	2022300007
SUBJECT	Operating system
EXPERIMENT NO :	10
DATE OF PERFORMANCE	17/4/24
DATE OF SUBMISSION	17/4/24
AIM:	**Create a Chat Bot using Shared Memory in Linux**
THEORY:	Inter Process Communication through shared memory is a concept where two or more processes can access the common memory and communication is done via this shared memory where changes made by one process can be viewed by another process. The problem with pipes, fifo and message queue – is that for two processes to exchange information. The information has to go through the kernel. ● Server reads from the input file. ● The server writes this data in a message using either a pipe, fifo or message queue. ● The client reads the data from the IPC channel,again requiring the data to be copied from the kernel's IPC buffer to the client's buffer. ● Finally, the data is copied from the client's buffer.



A total of four copies of data are required (2 read and 2 write). So, shared memory provides a way by letting two or more processes share a memory segment. With Shared Memory the data is only copied twice – from input file into shared memory and from shared memory to the output file.



System calls used:

shmget() : Upon successful completion, shmget() returns an identifier for the shared memory segment.

shmat() : Before you can use a shared memory segment, you have to attach yourself to it using shmat(). Here, shmid is a shared memory ID and shmaddr specifies the specific address to use but we should set it to zero and the OS will automatically choose the address.

shmdt() : When you're done with the shared memory segment, your program should detach itself from it using shmdt().

Code:

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

#define SHARED_MEMORY_KEY 1234
#define SHARED_MEMORY_SIZE 1024
#define RESPONSE_PREFIX "Bot: "

int main() {
        int shmid;
        char *shm_ptr;
        char user_input[256];

        // Create a shared memory segment
        shmid = shmget(SHARED_MEMORY_KEY,
SHARED_MEMORY_SIZE, IPC_CREAT | 0666);
        if (shmid == -1) {
        perror("shmget");
        exit(1);
        }

        // Attach to the shared memory segment
        shm_ptr = shmat(shmid, NULL, 0);
        if (shm_ptr == (char *) -1) {
        perror("shmat");
        exit(1);
        }

        // Main loop
        while (1) {
        // Wait for user input
        printf("User: ");
        fgets(user_input, sizeof(user_input), stdin);

        // Respond to user input
        if (strncmp(user_input, "Hi", 2) == 0) {
        strcpy(shm_ptr, RESPONSE_PREFIX);
        strcat(shm_ptr, "Hello, how can I help you today?");
```

```
        } else if (strncmp(user_input, "What's the weather today?", 25)
== 0) {
        strcpy(shm_ptr, RESPONSE_PREFIX);
        strcat(shm_ptr, "The current temperature is 34 degrees
Celsius.");
        } else {
        strcpy(shm_ptr, RESPONSE_PREFIX);
        strcat(shm_ptr, "Sorry, I didn't understand that.");
        }

        // Wait for a moment to simulate processing time
        sleep(1);

        // Print bot's response
        printf("%s\n", shm_ptr);
        }

        // Detach from the shared memory segment
        shmdt(shm_ptr);

        // Clean up the shared memory segment
        shmctl(shmid, IPC_RMID, NULL);

        return 0;
}
```

Output:

```
students@students-OptiPlex-3020:~/Desktop$ gcc -o chatbot chatbot.c
students@students-OptiPlex-3020:~/Desktop$ ./chatbot
User: Hi
Bot: Hello, how can I help you today?
User: What's the weather today?
Bot: The current temperature is 34 degrees Celsius.
User: What's the date today?
Bot: Sorry, I didn't understand that.
```

Explanation:

shmget(): It creates a shared memory segment using the specified key and size. If the creation fails, an error message is displayed, and the program exits.

Shared Memory Attachment:

shmat(): It attaches the program to the shared memory segment. If the attachment fails, an error message is displayed, and the program exits.

The program enters a while loop that continues indefinitely.

- It prompts the user to input a message ("User: ") and reads the input using fgets().

Response:

	• Based on the user's input, the program generates a response. • If the user input starts with "Hi", the bot responds with a greeting. • If the user input exactly matches "What's the weather today?", the bot responds with the current temperature. • Otherwise, the bot responds with a generic apology for not understanding the input. Shared Memory Write: • The generated response is written to the shared memory segment. It begins with the response prefix. • If the user's input matches the weather query, the response includes the current temperature. • The program waits for a moment using sleep(1) to simulate processing time. • Then, it prints the bot's response, which is retrieved from the shared memory segment. • shmdt(): It detaches the program from the shared memory segment. • shmctl(): It cleans up the shared memory segment when the program exits.

CONCLUSION:	In conclusion, the provided C code demonstrates a simple chat bot implementation using shared memory in Linux. It listens for user input, generates responses based on predefined rules, and communicates with the user through the shared memory segment. This approach allows for inter-process communication between the bot and the user, facilitating seamless interaction within a Linux environment. By utilizing shared memory, the bot efficiently exchanges messages without the need for complex socket programming or file I/O operations.

Vidya Bhavan's
Sardar Patel Institute of Technology

Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India
(Autonomous College Affiliated to University of Mumbai)

Vidya Bhavan's
Sardar Patel Institute of Technology

Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India
(Autonomous College Affiliated to University of Mumbai)