



main.c



Share

Run

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/ipc.h>
4 #include <sys/shm.h>
5 #include <unistd.h>
6
7 #define SHM_SIZE 1024
8
9 int main() {
10     int shmid;
11     key_t key = 1234;
12     char *str;
13
14     shmid = shmget(key, SHM_SIZE, 0666 | IPC_CREAT);
15     str = (char*) shmat(shmid, NULL, 0);
16     printf("Write Data: ");
17     fgets(str, SHM_SIZE, stdin);
18     shmdt(str);
19     printf("Press Enter to remove shared memory...\n");
20     getchar();
21     shmctl(shmid, IPC_RMID, NULL);
22
23     return 0;
24 }
25
```

Output

Write Data: psyche erose
Press Enter to remove shared memory...

=== Session Ended. Please Run the code again ===

main.c

Share

Run

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <sys/ipc.h>
5 #include <sys/msg.h>
6 #include <unistd.h>
7 #define MAX_TEXT 512
8 struct message {
9     long msg_type;
10    char text[MAX_TEXT];
11 };
12 int main() {
13     struct message msg;
14     int msgid;
15     key_t key;
16     key = ftok("progfile", 65);
17     msgid = msgget(key, 0666 | IPC_CREAT);
18     msg.msg_type = 1;
19     strcpy(msg.text, "Hello, this is a message!");
20     msgsnd(msgid, &msg, sizeof(msg), 0);
21     printf("Message sent: %s\n", msg.text);
22     msgrcv(msgid, &msg, sizeof(msg), 1, 0);
23     printf("Message received: %s\n", msg.text);
24     msgctl(msgid, IPC_RMID, NULL);
25     return 0;
26 }
27
```

Output

```
Message sent: Hello, this is a message!
Message received: Hello, this is a message!

=== Code Execution Successful ===
```