Message Queues

Message queues are a method of Inter-Process Communication (IPC) that allow processes to exchange messages in a queue format. The messages are stored in a kernel-maintained queue, and processes can send and receive these messages using specific system calls.

Here are the key message queue IPC system calls and their usage:

1. msgget - Create or Access a Message Queue

This system call is used to create a new message queue or access an existing one.

Syntax:

```
#include <sys/ipc.h>
#include <sys/msg.h>

int msgget(key_t key, int msgflg);
```

Parameters:

- key: Unique identifier for the message queue.
- msgflg: Flags for permissions and behavior (e.g., IPC_CREAT to create a queue if it doesn't exist).

Return Value:

- Returns the message queue identifier (ID) on success, or -1 on failure.

Example:

```
int msgid = msgget(IPC_PRIVATE, IPC_CREAT | 0666);
if (msgid == -1) {
    perror("msgget failed");
}
```

2. msgsnd - Send a Message to a Queue

This system call is used to send a message to the message queue.

Syntax:

```
#include <sys/msg.h>
int msgsnd(int msqid, const void *msgp, size_t msgsz,
           int msgflg);
```

Parameters:

- msqid: Message queue ID.
- msgp: Pointer to the message to be sent.
- msgsz: Size of the message.
- msgflg: Flags to modify behavior (e.g., IPC_NOWAIT).

Return Value:

- Returns 0 on success, or -1 on failure.

Message Structure:

```
struct msgbuf {
    long mtype;       // Message type
    char mtext[1];    // Message text
};
```

Example:

```
struct msgbuf msg;
msg.mtype = 1; // Set message type
strcpy(msg.mtext, "Hello, World!");
if (msgsnd(msgid, &msg, sizeof(msg.mtext), 0) == -1)
{
     perror("msgsnd failed");
}
```

3. msgrcv - Receive a Message from a Queue

This system call is used to receive a message from the queue.

Syntax:

```
#include <sys/msg.h>
ssize_t msgrcv(int msqid, void *msgp, size_t msgsz,
               long msgtyp, int msgflg);
```

Parameters:

- msqid: Message queue ID.
- msgp: Pointer to the buffer to store the received message.
- msgsz: Maximum size of the message to receive.
- msgtyp: Type of message to receive (0 to receive the first message in the queue).
- msgflg: Flags to modify behavior (e.g., IPC_NOWAIT).

Return Value:

- Returns the size of the received message on success, or -1 on failure.

Example:

```
struct msgbuf msg;
if (msgrcv(msgid, &msg, sizeof(msg.mtext), 1, 0)== 1)
{
    perror("msgrcv failed");
} else {
    printf("Received: %s\n", msg.mtext);
}
```

4. msgctl - Control Operations on a Message Queue

This system call is used to perform control operations on the message queue, such as deleting it.

Syntax:

```
#include <sys/msg.h>
int msgctl(int msqid, int cmd, struct msqid_ds *buf);
```

- **Parameters:**
  - msqid: Message queue ID.
  - cmd: Command to perform (e.g., IPC_RMID to remove the queue).
  - buf: Pointer to a structure containing queue information.

Example:

```
if (msgctl(msgid, IPC_RMID, NULL) == -1) {
    perror("msgctl failed");
} else {
    printf("Message queue deleted.\n");
}
```

Summary of Commands

System Call	Description
msgget	Create or access a message queue
msgsnd	Send a message to the queue
msgrcv	Receive a message from the queue
msgctl	Perform control operations (e.g., delete the queue)

Example Workflow

1. Use msgget to create or access a queue.
2. Use msgsnd to send messages to the queue.
3. Use msgrcv to receive messages from the queue.
4. Use msgctl to delete or modify the queue when done.