

Operating System Lab 02

Task

Write the reader (or receiver) program, named as: **receiver.c** and print the received messages by the reader

Idea

1. In receiver.c, create a message queue with the **same identifier**(namely, the key) as sender.c
2. Use the `msgrcv()` function to continuously read the information transmitted in the sender through a loop, and when the queue is empty, the `msgrcv()` function will block and wait
3. Handle the behavior of the receiver when the message queue is closed: the updated `errno` when the `msgrcv()` function returns `-1` determines the next behavior of the receiver

Code Structure

```
#include <errno.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <sys/types.h>

struct my_msgbuf {
    long mtype;
    char mtext[200];
};

int main(int argc, char *argv[]) {
    struct my_msgbuf buf;
    int msqid;
    key_t key;

    /**
     * Keep the key of message queue being the same as the sender
     */
    if ((key = ftok("sender.c", 'B')) == -1) {
        perror("ftok");
        exit(1);
    }

    /**
     * Receiver just needs the readable permission
     */
    if ((msqid = msgget(key, 0444 | IPC_CREAT)) == -1) {
        perror("msgget");
        exit(1);
    }

    /**
     * Specify msgflg: MSG_NOERROR,
     * if the length of the message obtained by the function is greater than
     * msgsz, it will only return the information of the length of msgsz.
     */
    while (msgrcv(msqid, &buf, sizeof(buf.mtext), 1, MSG_NOERROR) != -1) {
        printf("Data Received is : %s \n", buf.mtext);
    }

    /**
```

```

* If the sender was closed, the message queue has been removed,
* the upper msgrcv() will return -1, then it will update the errno to EIDRM,
* when errno == EIDRM, the receiver doesn't need to destroy the message
* queue.
*/
if (errno == EIDRM) {
    printf("Message queue has been close\n");
} else {
    if (msgctl(msqid, IPC_RMID, NULL) == -1) {
        perror("msgctl");
        exit(1);
    }
}

return 0;
}

```

Execution Results

```

lab2 — h3art@H3ArtdeMacBook-Air — ..em/OSLAB/lab2 — -zsh — 120x36
[13:46:30] [cost 0.037s] clear
[13:47:42] [~/Documents/文稿/JNU_Course/Sophomore/OperatingSystem/OSLAB/lab2] >>> clang sender.c -g -o sender
[13:48:01] [cost 0.360s] clang sender.c -g -o sender
[13:48:13] [~/Documents/文稿/JNU_Course/Sophomore/OperatingSystem/OSLAB/lab2] >>> ./sender
Enter lines of text, ^D to quit:
Hello, OS world!
hzb564@jnu.edu.cn
H3Art
[13:49:47] [cost 90.950s] ./sender
[13:50:03] [~/Documents/文稿/JNU_Course/Sophomore/OperatingSystem/OSLAB/lab2] >>>

lab2 — h3art@H3ArtdeMacBook-Air — ..em/OSLAB/lab2 — -zsh — 120x36
[13:46:32] [cost 0.038s] clear
[13:48:03] [~/Documents/文稿/JNU_Course/Sophomore/OperatingSystem/OSLAB/lab2] >>> clang receiver.c -g -o receiver
[13:48:12] [cost 0.119s] clang receiver.c -g -o receiver
[13:48:18] [~/Documents/文稿/JNU_Course/Sophomore/OperatingSystem/OSLAB/lab2] >>> ./receiver
Data Received is : Hello, OS world!
Data Received is : hzb564@jnu.edu.cn
Data Received is : H3Art
Data Received is : 
Data Received is : 
Message queue has been close
[13:49:47] [cost 83.768s] ./receiver

```