

```
psc@psc-virtual-machine: ~  
psc@psc-virtual-machine:~$ ls  
MyMsg.h      cleanup.c    ex.out       moss.pl      server.c     string.out   다운로드  
array-1.cpp  cleanup.out  fstat        mutex.c      server.out   thread.c     문서  
array-1.out  client.c     fstat.c      mutex.out    signal       thread.out   바탕화면  
array-2.cpp  client.out   getpwuid     original.txt signal.c     tic.c        비디오회  
array-2.out  con.c       getpwuid.c   ox           stat         vector.cpp   오  
chdir        copy.txt    mkdir        readdir      stat.c       vector.out   음  
chdir.c     ex.c        mkdir.c      readdir.c    string.cpp   공개         악  
psc@psc-virtual-machine:~$ ./server.out  
Wait ...  
Receive: 1 + 1  
Send: 2  
Wait ...  
Receive: 2 - 1  
Send: 1  
Wait ...  
Receive: 3 * 2  
Send: 6  
Wait ...  
Receive: 5 / 2  
Send: 2  
Wait ...  
^Cpsc@psc-virtual-machine:~$  
psc@psc-virtual-machine:~$ ls  
MyMsg.h      cleanup.c    ex.out       moss.pl      server.c     string.out   다운로드  
array-1.cpp  cleanup.out  fstat        mutex.c      server.out   thread.c     문서  
array-1.out  client.c     fstat.c      mutex.out    signal       thread.out   바탕화면  
array-2.cpp  client.out   getpwuid     original.txt signal.c     tic.c        비디오회  
array-2.out  con.c       getpwuid.c   ox           stat         vector.cpp   오  
chdir        copy.txt    mkdir        readdir      stat.c       vector.out   음  
chdir.c     ex.c        mkdir.c      readdir.c    string.cpp   공개         악  
psc@psc-virtual-machine:~$ ./client.out  
Input: [x] [op] [y]  
<<< 1 + 1  
>>> 2  
<<< 2 - 1  
>>> 1  
<<< 3 * 2  
>>> 6  
<<< 5 /2  
>>> 2  
<<< ^C  
psc@psc-virtual-machine:~$
```

```
psc@psc-virtual-machine: ~  
psc@psc-virtual-machine:~$ ls  
MyMsg.h      cleanup.c    ex.out       moss.pl      server.c     string.out   다운로드  
array-1.cpp  cleanup.out  fstat        mutex.c      server.out   thread.c     문서  
array-1.out  client.c    fstat.c      mutex.out    signal       thread.out   바탕화면  
array-2.cpp  client.out  getpwuid     original.txt signal.c     tic.c        비디오  
array-2.out  con.c       getpwuid.c   ox           stat         vector.cpp   사진  
chdir        copy.txt    mkdir        readdir      stat.c       vector.out   음악  
chdir.c      ex.c        mkdir.c      readdir.c    string.cpp   공개         템플릿  
psc@psc-virtual-machine:~$ cat server.c  
#include "MyMsg.h"  
#include <signal.h>  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <sys/ipc.h>  
#include <sys/msg.h>  
#include <sys/types.h>  
  
void signalHandler(int signum);  
int calculate(Calc calc);  
  
key_t mykey = 0;  
int msqid = 0;  
  
int main(int argc, char const *argv[]) {  
    MsgCalc msgCalc;  
    MsgRslt msgRslt;  
  
    mykey = ftok("mymsgkey", 1);  
    msqid = msgget(mykey, IPC_CREAT | 0600);  
  
    signal(SIGINT, signalHandler);  
  
    while (1) {  
        puts("Wait ...");  
        memset(&msgCalc, 0x00, sizeof(MsgCalc));  
        msgrcv(msqid, &msgCalc, MSG_SIZE_CALC, MSG_TYPE_CALC, 0);  
        printf("Receive: %d %c %d\n", msgCalc.calc.x, msgCalc.calc.op,  
            msgCalc.calc.y);  
  
        memset(&msgRslt, 0x00, sizeof(MsgRslt));  
        msgRslt.mtype = MSG_TYPE_RSLT;  
        msgRslt.rslt = calculate(msgCalc.calc);  
        msgsnd(msqid, &msgRslt, MSG_SIZE_RSLT, 0);  
        printf("Send: %d\n", msgRslt.rslt);  
  
        fflush(stdout);  
    }  
  
    return 0;  
}
```

```
psc@psc-virtual-machine: ~  
int main(int argc, char const *argv[]) {  
    MsgCalc msgCalc;  
    MsgRslt msgRslt;  
  
    mykey = ftok("mymsgkey", 1);  
    msqid = msgget(mykey, IPC_CREAT | 0600);  
  
    signal(SIGINT, signalHandler);  
  
    while (1) {  
        puts("Wait ...");  
        memset(&msgCalc, 0x00, sizeof(MsgCalc));  
        msgrcv(msqid, &msgCalc, MSG_SIZE_CALC, MSG_TYPE_CALC, 0);  
        printf("Receive: %d %c %d\n", msgCalc.calc.x, msgCalc.calc.op,  
              msgCalc.calc.y);  
  
        memset(&msgRslt, 0x00, sizeof(MsgRslt));  
        msgRslt.mtype = MSG_TYPE_RSLT;  
        msgRslt.rslt = calculate(msgCalc.calc);  
        msgsnd(msqid, &msgRslt, MSG_SIZE_RSLT, 0);  
        printf("Send: %d\n", msgRslt.rslt);  
  
        fflush(stdout);  
    }  
  
    return 0;  
}  
  
void signalHandler(int signum) {  
    if (signum == SIGINT) {  
        msgctl(msqid, IPC_RMID, NULL);  
        exit(0);  
    }  
}  
  
int calculate(Calc calc) {  
    switch (calc.op) {  
        case '+':  
            return calc.x + calc.y;  
        case '-':  
            return calc.x - calc.y;  
        case '*':  
            return calc.x * calc.y;  
        case '/':  
            return (int)(calc.x / calc.y);  
        default:  
            return 0;  
    }  
}  
psc@psc-virtual-machine:~$
```

```
psc@psc-virtual-machine: ~  
psc@psc-virtual-machine:~$ ls  
MyMsg.h      cleanup.c    ex.out       moss.pl      server.c     string.out   다운로드  
array-1.cpp  cleanup.out  fstat        mutex.c      server.out   thread.c     문서  
array-1.out  client.c     fstat.c      mutex.out    signal       thread.out   바탕화면  
array-2.cpp  client.out   getpwuid     original.txt signal.c     tic.c        사진  
array-2.out  con.c       getpwuid.c   ox           stat         vector.cpp   음악  
chdir        copy.txt    mkdir        readdir      stat.c       vector.out   템플릿  
chdir.c      ex.c        mkdir.c      readdir.c    string.cpp   공개  
psc@psc-virtual-machine:~$ cat client.c  
#include "MyMsg.h"  
#include <signal.h>  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <sys/ipc.h>  
#include <sys/msg.h>  
#include <sys/types.h>  
  
int main(int argc, char const *argv[]) {  
    key_t mykey = ftok("mymsgkey", 1);  
    int msqid = msgget(mykey, IPC_CREAT);  
  
    int x = 0;  
    int y = 0;  
    char op = 0;  
  
    MsgCalc msgCalc;  
    MsgRslt msgRslt;  
  
    puts("Input: [x] [op] [y]");  
    while (1) {  
        printf("<<< ");  
        scanf("%d %c %d", &x, &op, &y);  
        fflush(stdout);  
        fflush(stdin);  
        memset(&msgCalc, 0x00, sizeof(MsgCalc));  
        msgCalc.mtype = MSG_TYPE_CALC;  
        msgCalc.calc.x = x;  
        msgCalc.calc.y = y;  
        msgCalc.calc.op = op;  
        msgsnd(msqid, &msgCalc, MSG_SIZE_CALC, 0);  
  
        memset(&msgRslt, 0x00, sizeof(MsgRslt));  
        msgrcv(msqid, &msgRslt, MSG_SIZE_RSLT, MSG_TYPE_RSLT, 0);  
        printf(">>> %d\n", msgRslt.rslt);  
    }  
  
    return 0;  
}  
psc@psc-virtual-machine:~$
```