

```
root@localhost:/home/lfy/mq/figure
File Edit View Terminal Go Help

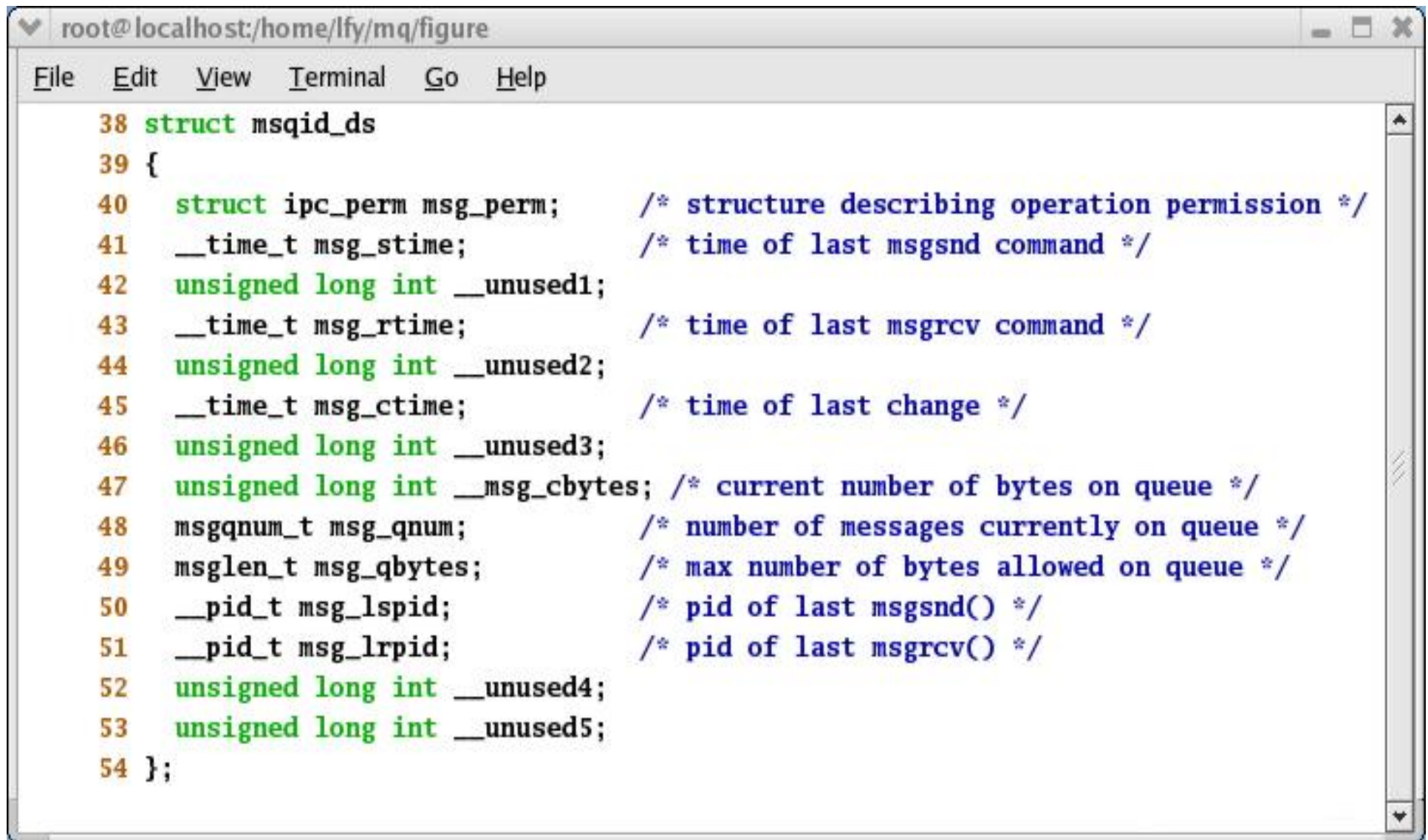
1 #include <sys/msg.h>
2
3 main () {
4
5     int msgid;
6     key_t key;
7
8     key = ftok(".", 'a');
9     msgid = msgget(key, IPC_CREAT|IPC_EXCL|0744);
10    printf (" key = 0x%0x\t msgid = %d\n", key, msgid);
11 }
```

Figure 1

```
root@localhost:/home/lfy/mq/figure
File Edit View Terminal Go Help

43 struct ipc_perm
44 {
45     __key_t __key;           /* Key. */
46     __uid_t uid;             /* Owner's user ID. */
47     __gid_t gid;             /* Owner's group ID. */
48     __uid_t cuid;           /* Creator's user ID. */
49     __gid_t cgid;           /* Creator's group ID. */
50     unsigned short int mode; /* Read/write permission. */
51     unsigned short int __pad1;
52     unsigned short int __seq; /* Sequence number. */
53     unsigned short int __pad2;
54     unsigned long int __unused1;
55     unsigned long int __unused2;
56 };
```

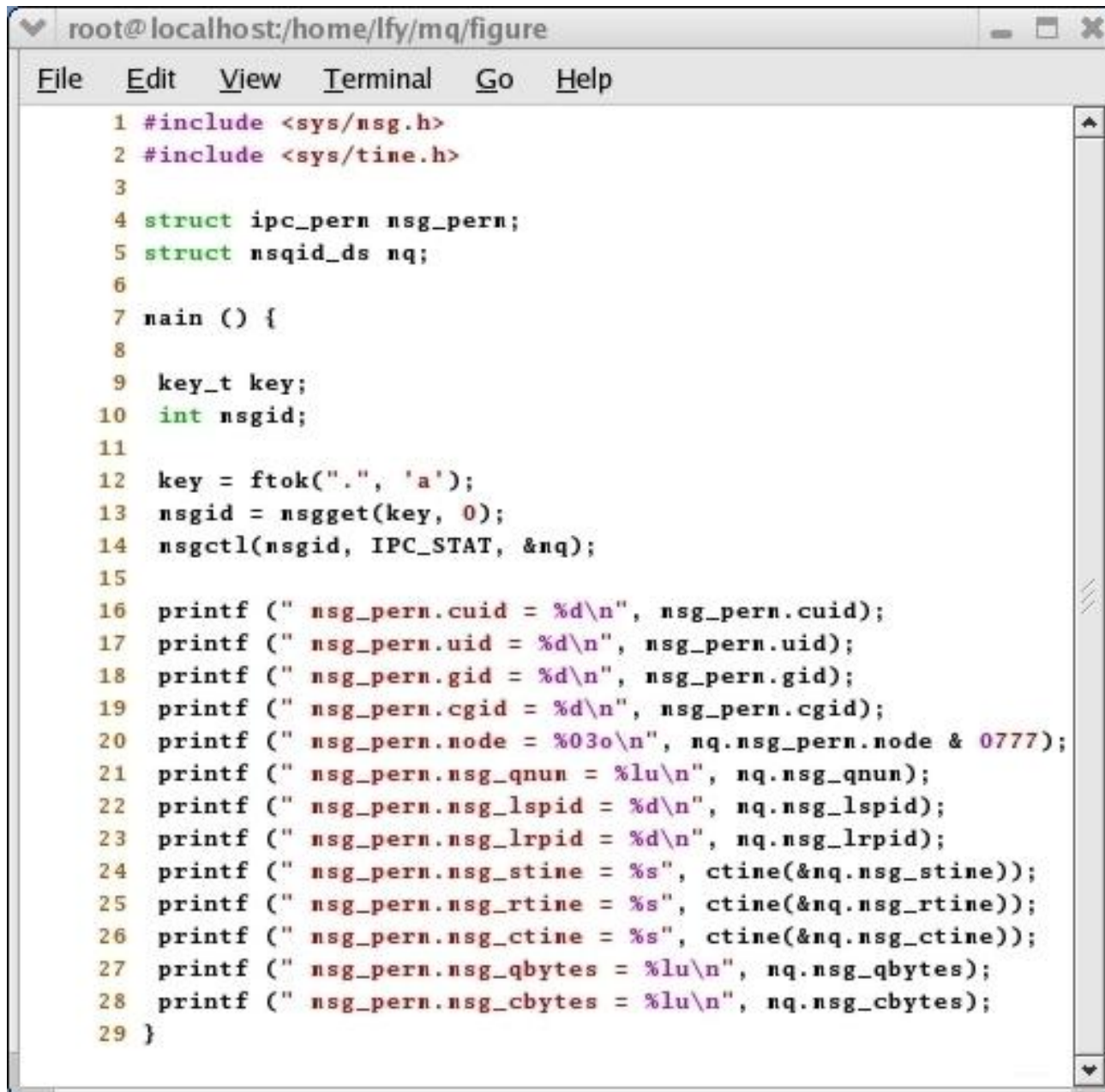
Figure 2



```
root@localhost:/home/lfy/mq/figure
File Edit View Terminal Go Help

38 struct msqid_ds
39 {
40     struct ipc_perm msg_perm;      /* structure describing operation permission */
41     __time_t msg_stime;            /* time of last msgsnd command */
42     unsigned long int __unused1;
43     __time_t msg_rtime;            /* time of last msgrcv command */
44     unsigned long int __unused2;
45     __time_t msg_ctime;            /* time of last change */
46     unsigned long int __unused3;
47     unsigned long int __msg_cbytes; /* current number of bytes on queue */
48     msgqnum_t msg_qnum;            /* number of messages currently on queue */
49     msglen_t msg_qbytes;           /* max number of bytes allowed on queue */
50     __pid_t msg_lspid;             /* pid of last msgsnd() */
51     __pid_t msg_lrpid;            /* pid of last msgrcv() */
52     unsigned long int __unused4;
53     unsigned long int __unused5;
54 };
```

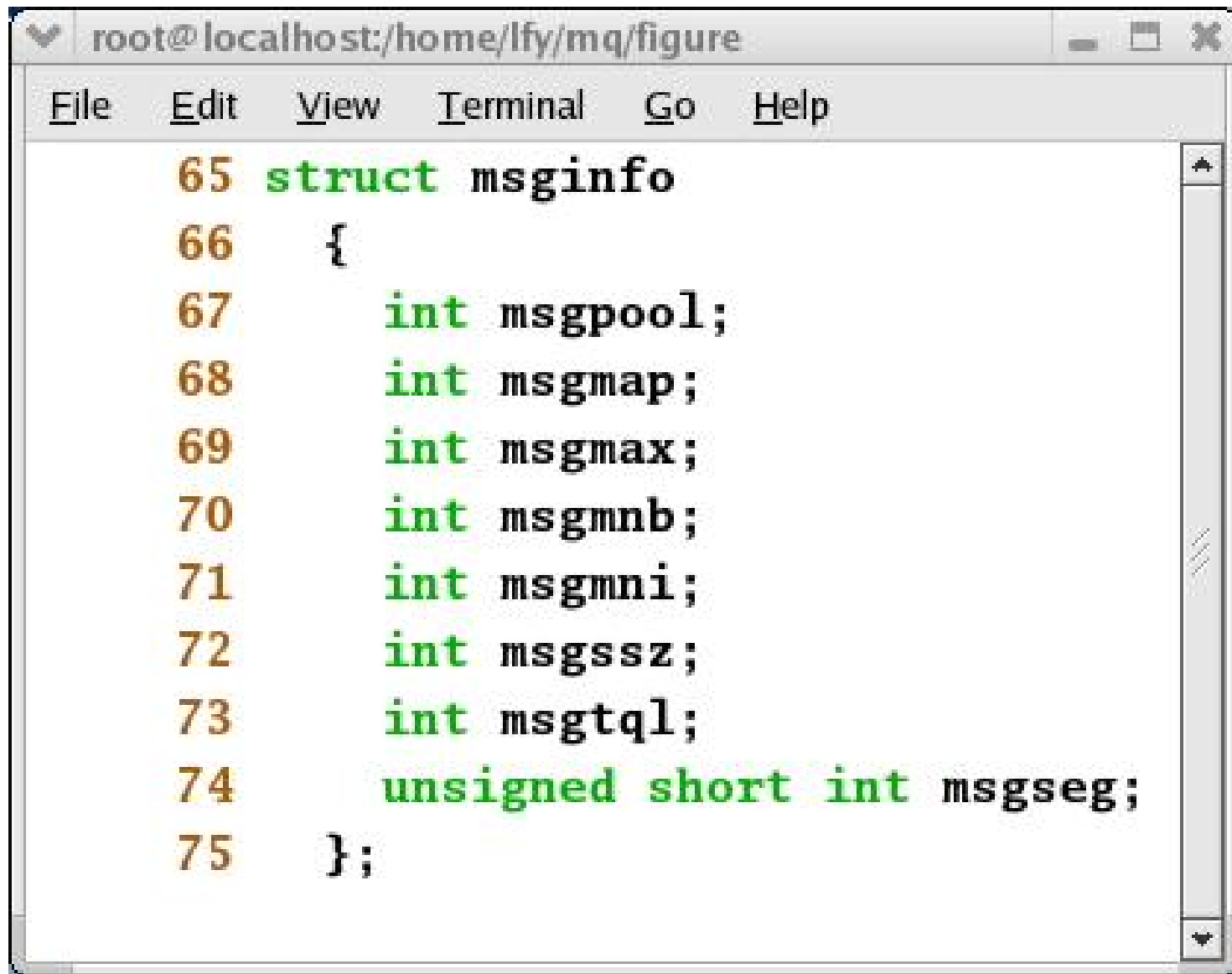
Figure 3



The image shows a terminal window with a title bar that reads "root@localhost:/home/lfy/mq/figure". The window contains a C program that uses the `nsqctl` library to retrieve and print IPC statistics for a specific nsq instance. The code is as follows:

```
1 #include <sys/nsg.h>
2 #include <sys/tine.h>
3
4 struct ipc_pern nsg_pern;
5 struct nsqid_ds nq;
6
7 nain () {
8
9     key_t key;
10    int nsgid;
11
12    key = ftok(".", 'a');
13    nsgid = nsgget(key, 0);
14    nsgctl(nsgid, IPC_STAT, &nq);
15
16    printf (" nsg_pern.cuid = %d\n", nsg_pern.cuid);
17    printf (" nsg_pern.uid = %d\n", nsg_pern.uid);
18    printf (" nsg_pern.gid = %d\n", nsg_pern.gid);
19    printf (" nsg_pern.cgid = %d\n", nsg_pern.cgid);
20    printf (" nsg_pern.node = %03o\n", nq.nsg_pern.node & 0777);
21    printf (" nsg_pern.nsg_qnun = %lu\n", nq.nsg_qnun);
22    printf (" nsg_pern.nsg_lspid = %d\n", nq.nsg_lspid);
23    printf (" nsg_pern.nsg_lrpipd = %d\n", nq.nsg_lrpipd);
24    printf (" nsg_pern.nsg_stine = %s", ctine(&nq.nsg_stine));
25    printf (" nsg_pern.nsg_rtime = %s", ctine(&nq.nsg_rtime));
26    printf (" nsg_pern.nsg_ctine = %s", ctine(&nq.nsg_ctine));
27    printf (" nsg_pern.nsg_qbytes = %lu\n", nq.nsg_qbytes);
28    printf (" nsg_pern.nsg_cbytes = %lu\n", nq.nsg_cbytes);
29 }
```

Figure 4



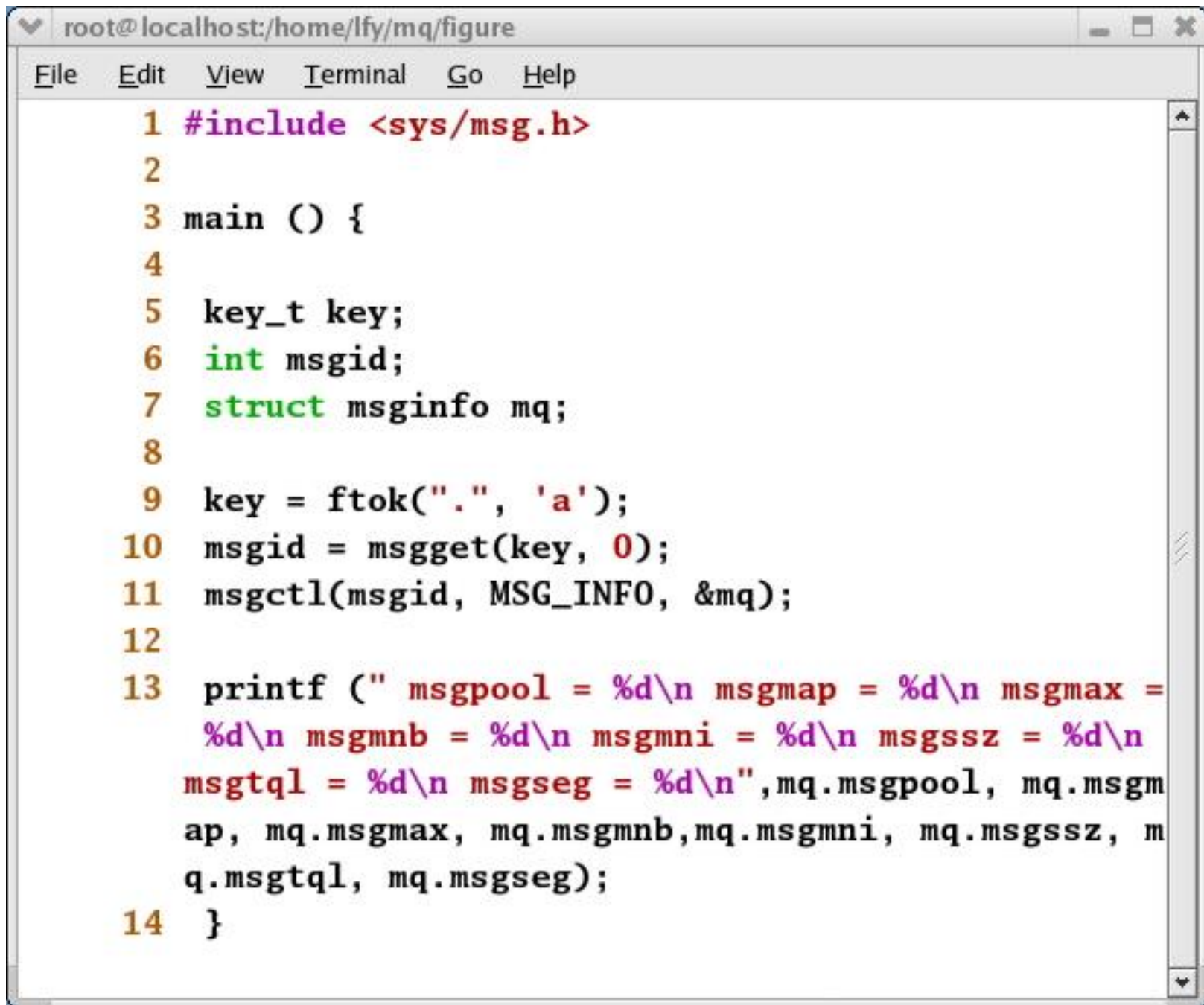
The image shows a screenshot of a code editor window. The title bar at the top reads "root@localhost:/home/lfy/mq/figure". Below the title bar is a menu bar with the following items: "File", "Edit", "View", "Terminal", "Go", and "Help". The main area of the window contains C code defining a struct named "msginfo". The code is as follows:

```
65 struct msginfo
66 {
67     int msgpool;
68     int msgmap;
69     int msgmax;
70     int msgmnb;
71     int msgmni;
72     int msgssz;
73     int msgtql;
74     unsigned short int msgseg;
75 };
```

The code is color-coded: line numbers are in orange, the keyword "struct" is in green, and the variable names and their types are in green. The struct definition is enclosed in curly braces, and the closing brace is followed by a semicolon.

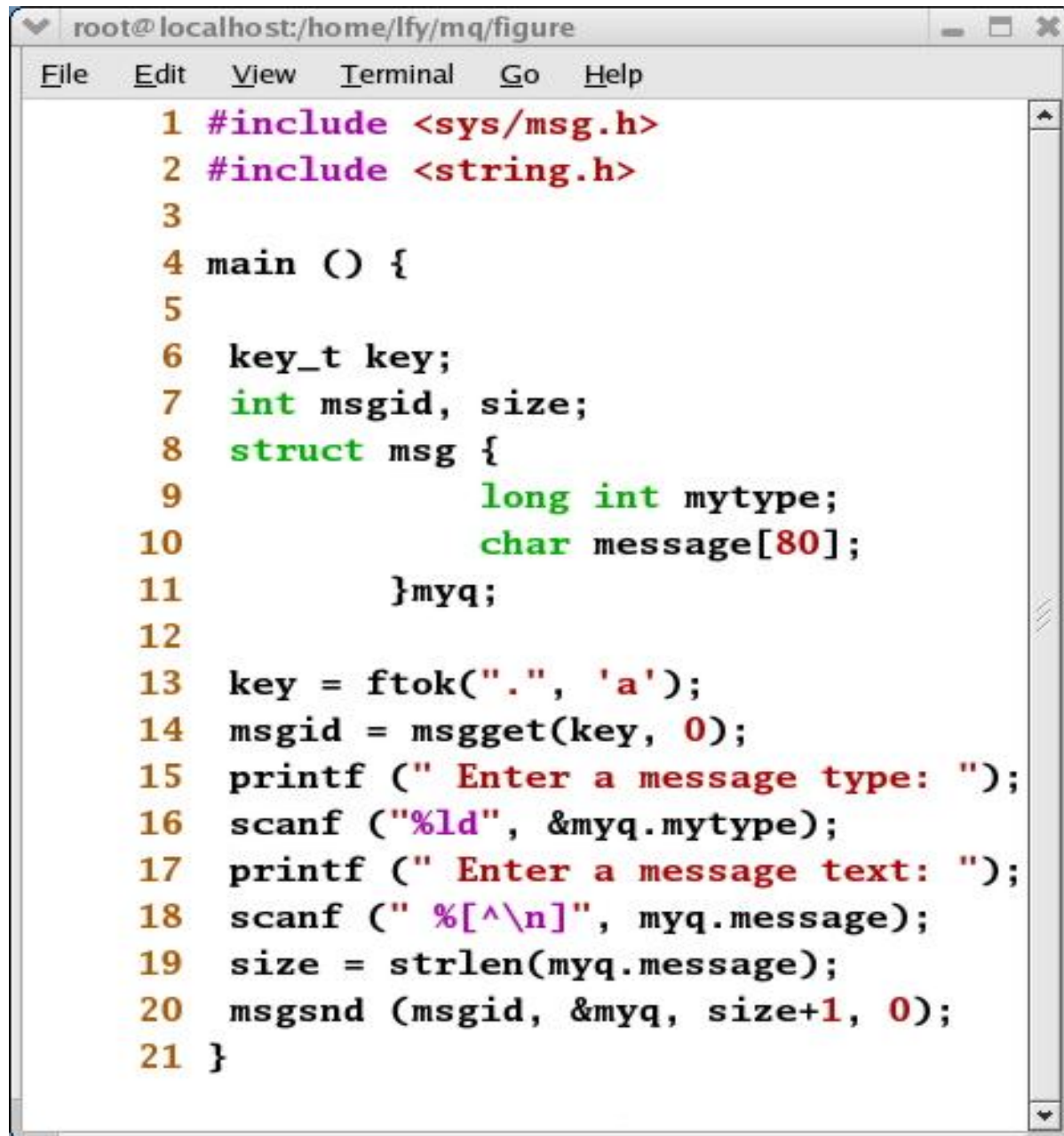
Figure 5





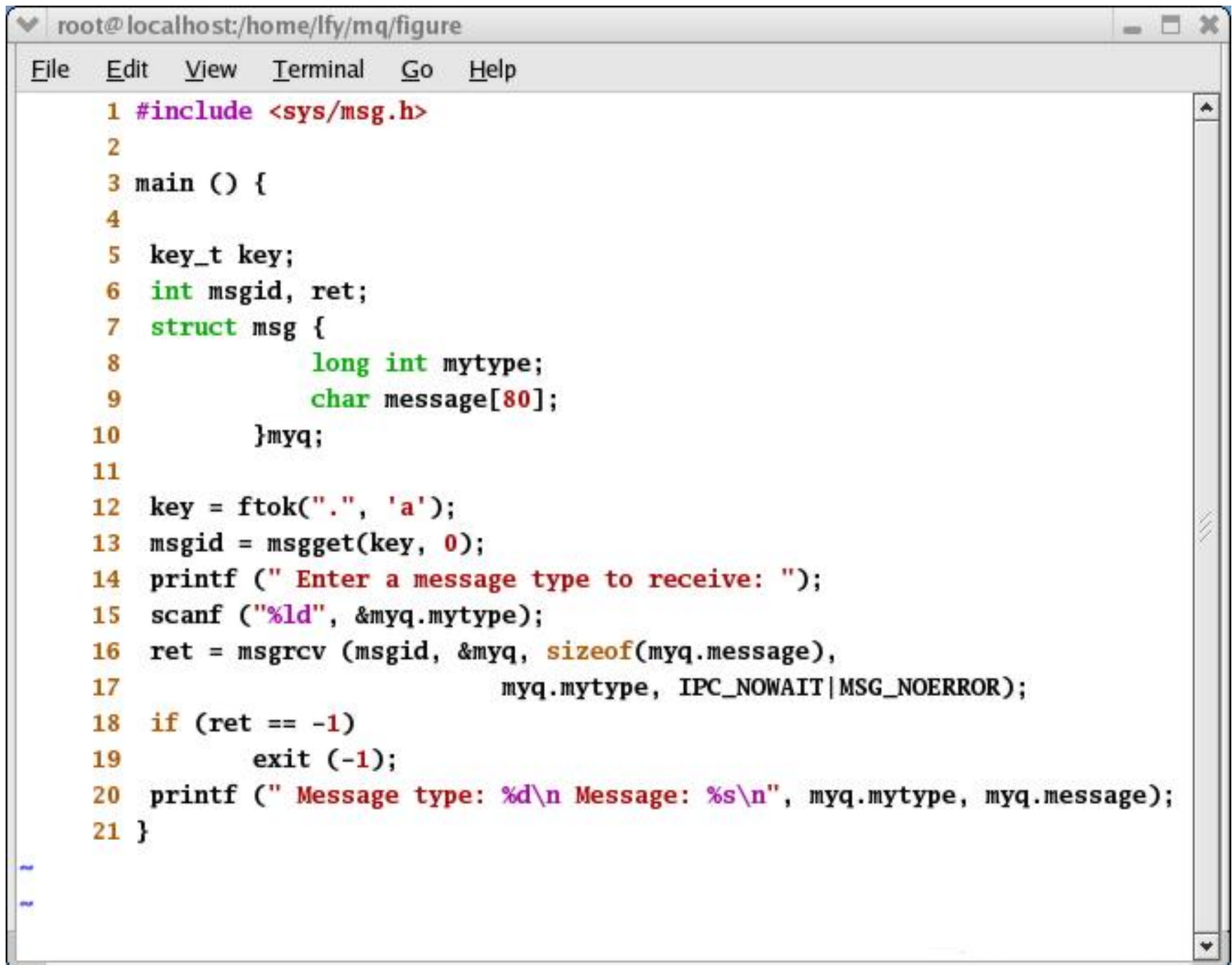
```
root@localhost:/home/lfy/mq/figure
File Edit View Terminal Go Help
1 #include <sys/msg.h>
2
3 main () {
4
5     key_t key;
6     int msgid;
7     struct msginfo mq;
8
9     key = ftok(".", 'a');
10    msgid = msgget(key, 0);
11    msgctl(msgid, MSG_INFO, &mq);
12
13    printf (" msgpool = %d\n msgmap = %d\n msgmax =
        %d\n msgmnb = %d\n msgmni = %d\n msgssz = %d\n
        msgtql = %d\n msgseg = %d\n",mq.msgpool, mq.msgm
        ap, mq.msgmax, mq.msgmnb,mq.msgmni, mq.msgssz, m
        q.msgtql, mq.msgseg);
14 }
```

Figure 6



```
1 #include <sys/msg.h>
2 #include <string.h>
3
4 main () {
5
6     key_t key;
7     int msgid, size;
8     struct msg {
9         long int mytype;
10        char message[80];
11    }myq;
12
13    key = ftok(".", 'a');
14    msgid = msgget(key, 0);
15    printf (" Enter a message type: ");
16    scanf ("%ld", &myq.mytype);
17    printf (" Enter a message text: ");
18    scanf (" %[^\n]", myq.message);
19    size = strlen(myq.message);
20    msgsnd (msgid, &myq, size+1, 0);
21 }
```

Figure 7

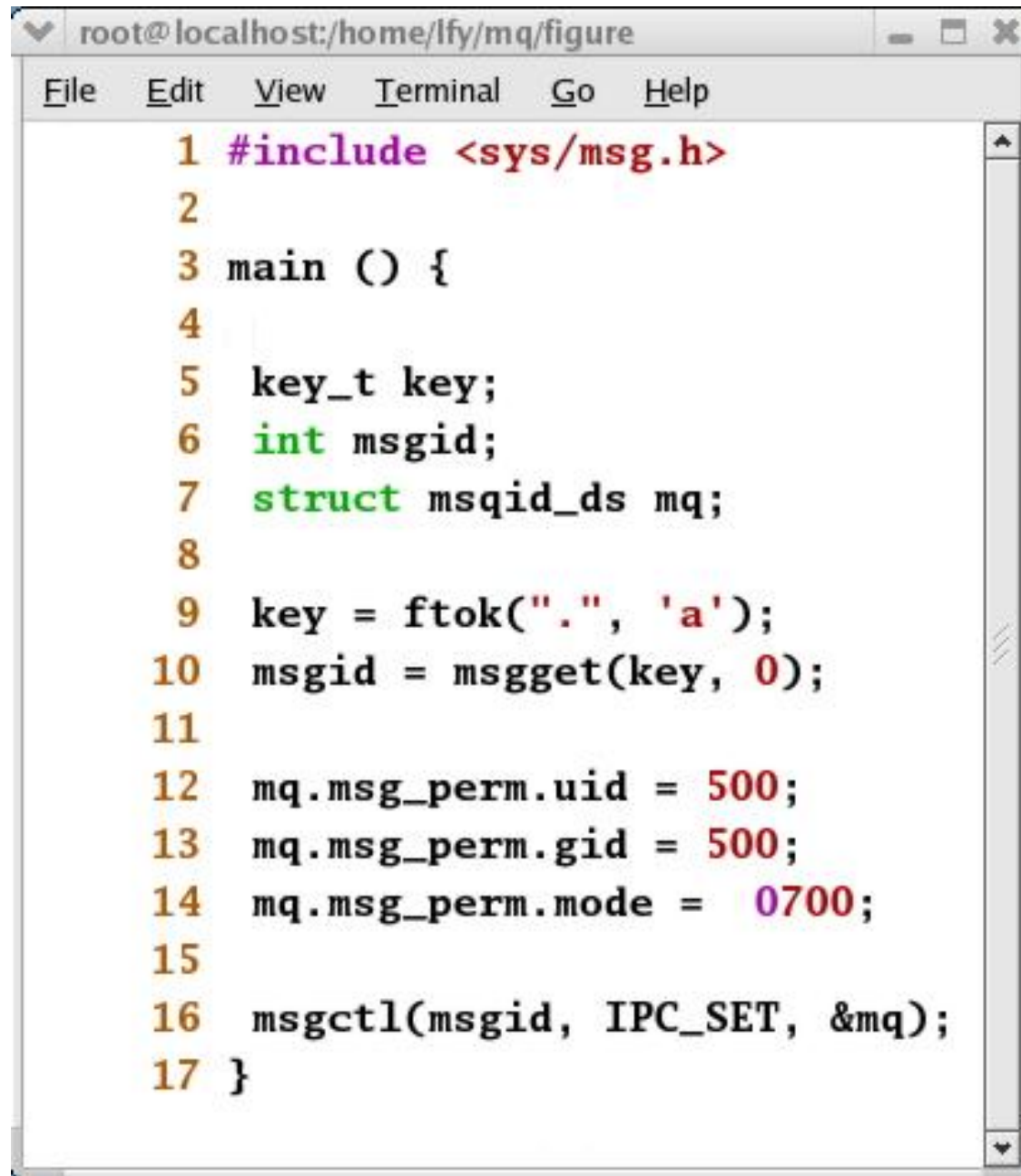


The image shows a code editor window with a title bar that reads "root@localhost:/home/lfy/mq/figure". The editor has a menu bar with "File", "Edit", "View", "Terminal", "Go", and "Help". The code is a C program for message queue communication, with line numbers 1 through 21 on the left. The code includes the header <sys/msg.h>, defines a main function, and uses the ftok, msgget, msgrcv, and printf functions. It prompts the user to enter a message type and then receives a message from a queue. The code is as follows:

```
1 #include <sys/msg.h>
2
3 main () {
4
5     key_t key;
6     int msgid, ret;
7     struct msg {
8         long int mytype;
9         char message[80];
10    }myq;
11
12    key = ftok(".", 'a');
13    msgid = msgget(key, 0);
14    printf (" Enter a message type to receive: ");
15    scanf ("%ld", &myq.mytype);
16    ret = msgrcv (msgid, &myq, sizeof(myq.message),
17                  myq.mytype, IPC_NOWAIT|MSG_NOERROR);
18    if (ret == -1)
19        exit (-1);
20    printf (" Message type: %d\n Message: %s\n", myq.mytype, myq.message);
21 }
```

Figure 8

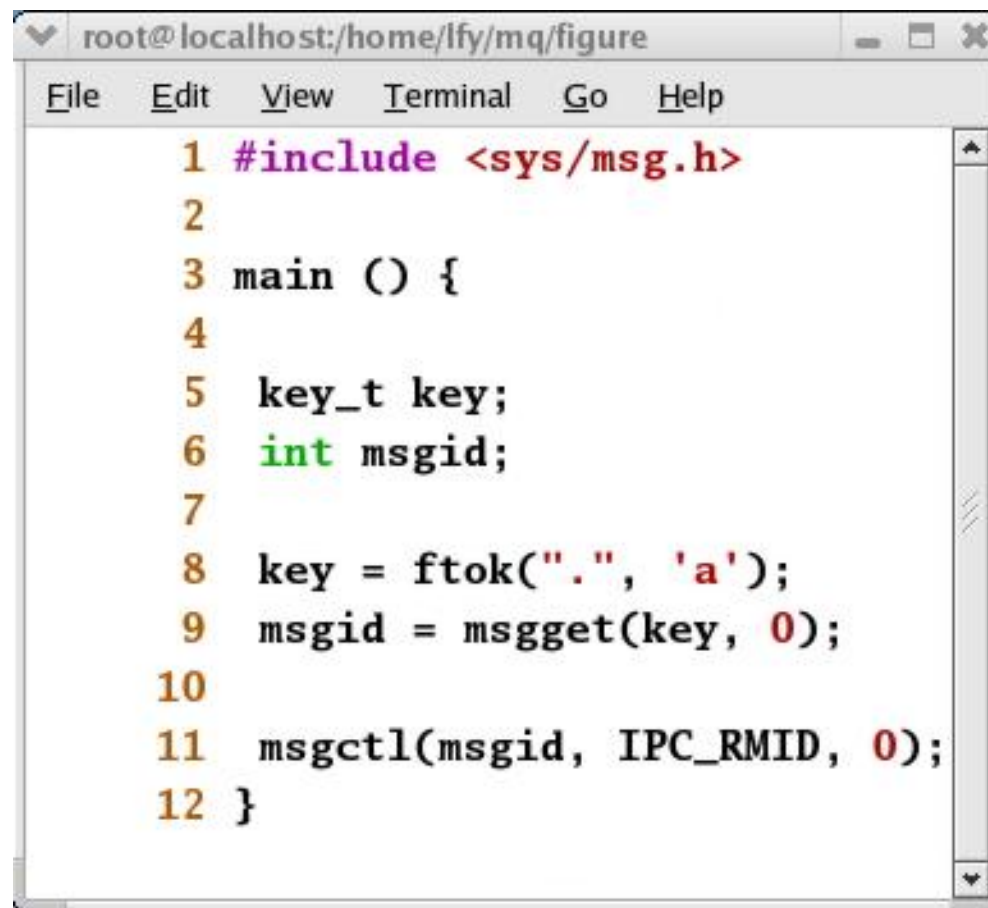




The image shows a code editor window with a title bar that reads "root@localhost:/home/lfy/mq/figure". The window has a menu bar with "File", "Edit", "View", "Terminal", "Go", and "Help". The code is written in C and is as follows:

```
1 #include <sys/msg.h>
2
3 main () {
4
5     key_t key;
6     int msgid;
7     struct msqid_ds mq;
8
9     key = ftok(".", 'a');
10    msgid = msgget(key, 0);
11
12    mq.msg_perm.uid = 500;
13    mq.msg_perm.gid = 500;
14    mq.msg_perm.mode = 0700;
15
16    msgctl(msgid, IPC_SET, &mq);
17 }
```

Figure 9



```
root@localhost:/home/lfy/mq/figure
File Edit View Terminal Go Help
1 #include <sys/msg.h>
2
3 main () {
4
5     key_t key;
6     int msgid;
7
8     key = ftok(".", 'a');
9     msgid = msgget(key, 0);
10
11     msgctl(msgid, IPC_RMID, 0);
12 }
```

Figure 10