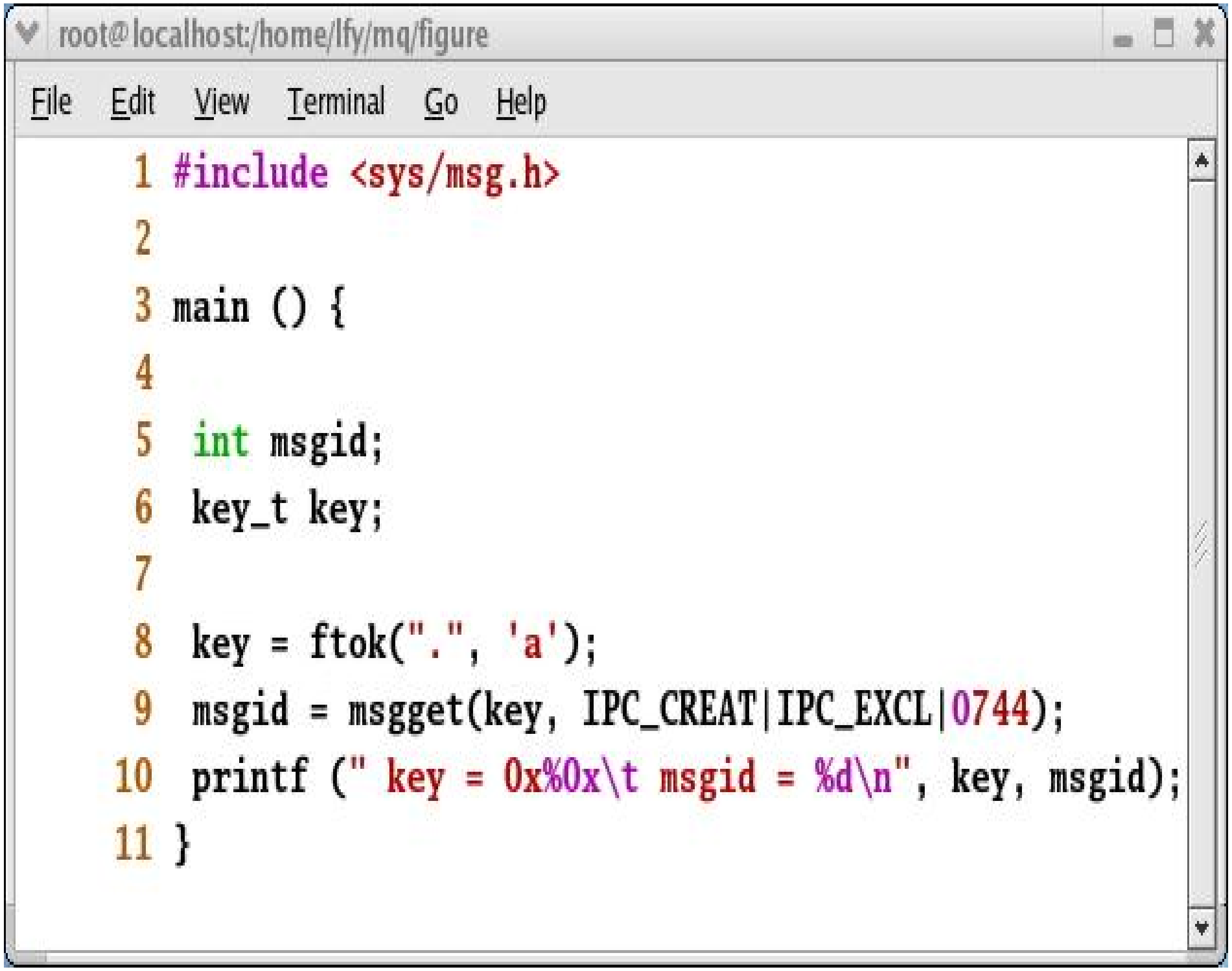
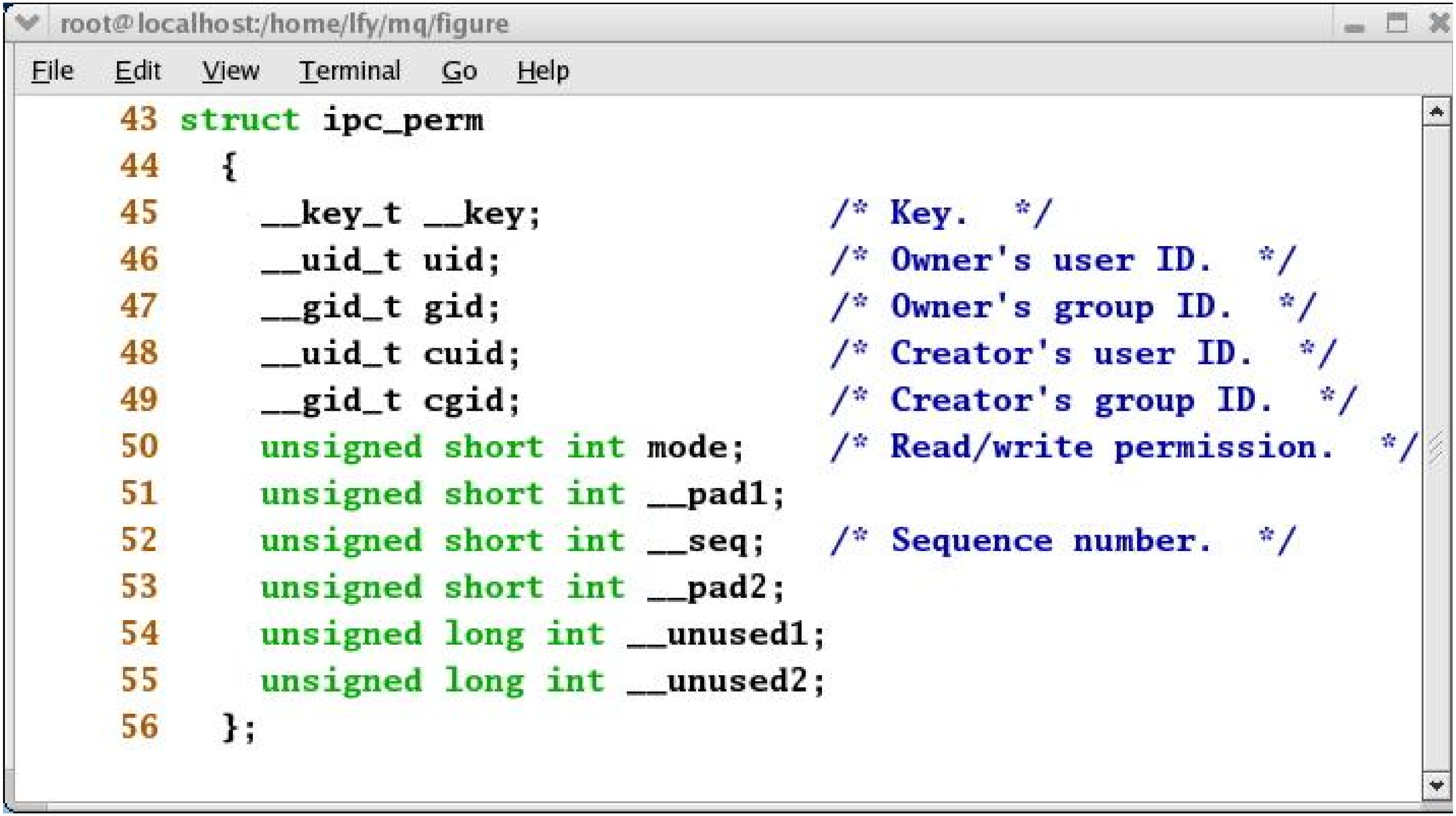


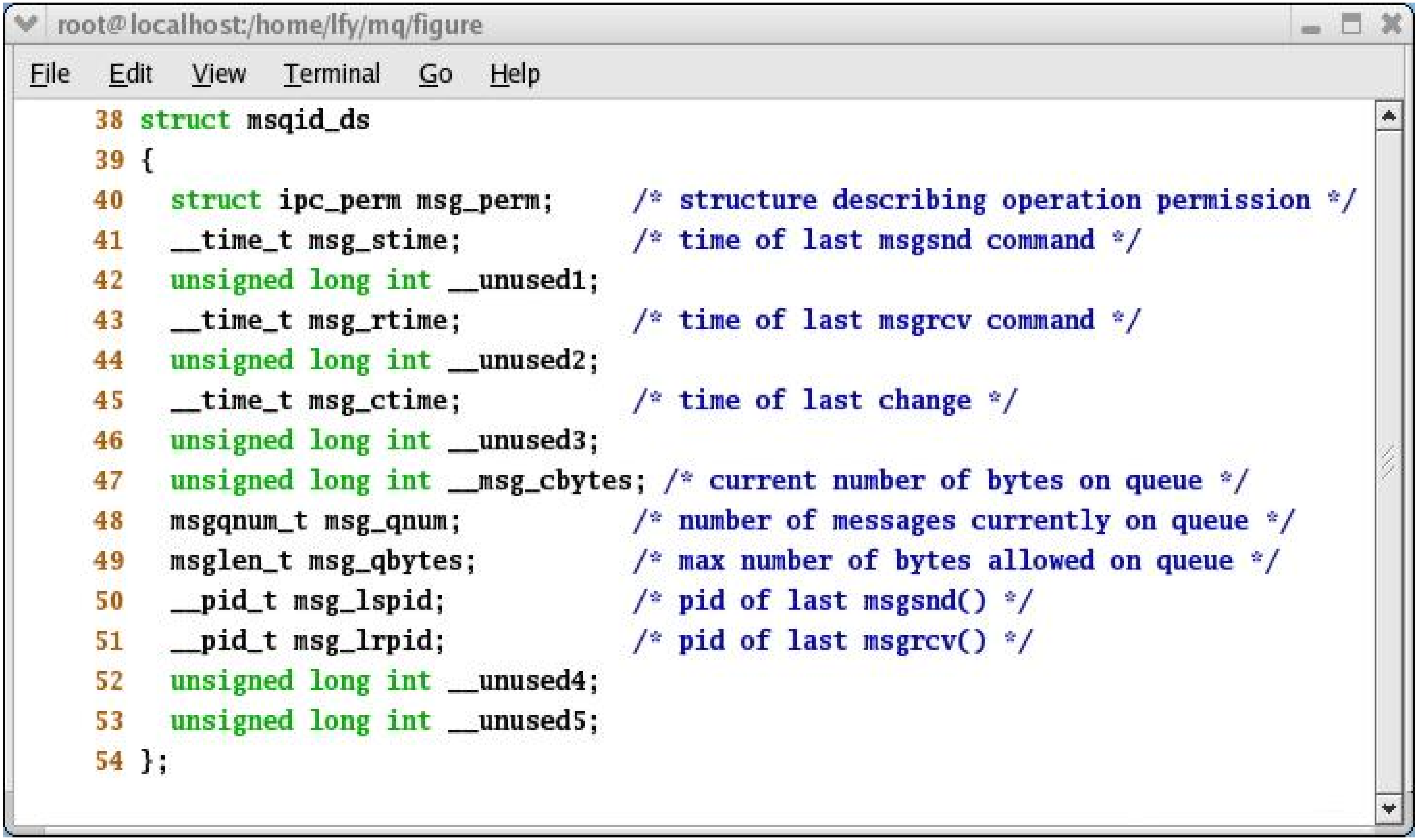
```c
#include <sys/msg.h>

main () {

 int msgid;
 key_t key;

 key = ftok(".", 'a');
 msgid = msgget(key, IPC_CREAT|IPC_EXCL|0744);
 printf (" key = 0x%0x\t msgid = %d\n", key, msgid);
}
```

Figure 1



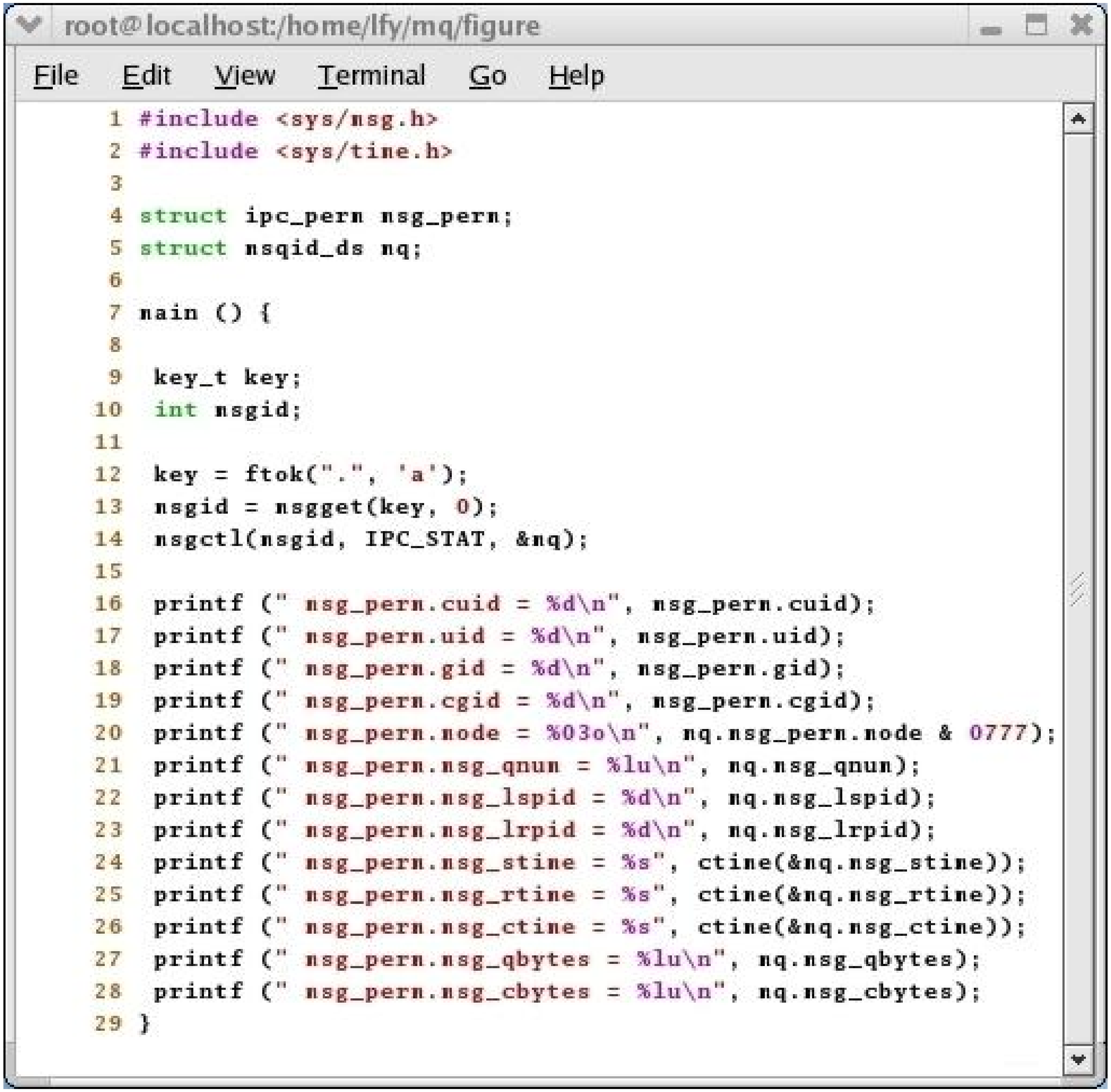
```
struct ipc_perm
  {
    __key_t __key;              /* Key.  */
    __uid_t uid;                /* Owner's user ID.  */
    __gid_t gid;                /* Owner's group ID.  */
    __uid_t cuid;               /* Creator's user ID.  */
    __gid_t cgid;               /* Creator's group ID.  */
    unsigned short int mode;    /* Read/write permission.  */
    unsigned short int __pad1;
    unsigned short int __seq;   /* Sequence number.  */
    unsigned short int __pad2;
    unsigned long int __unused1;
    unsigned long int __unused2;
  };
```

Figure 2



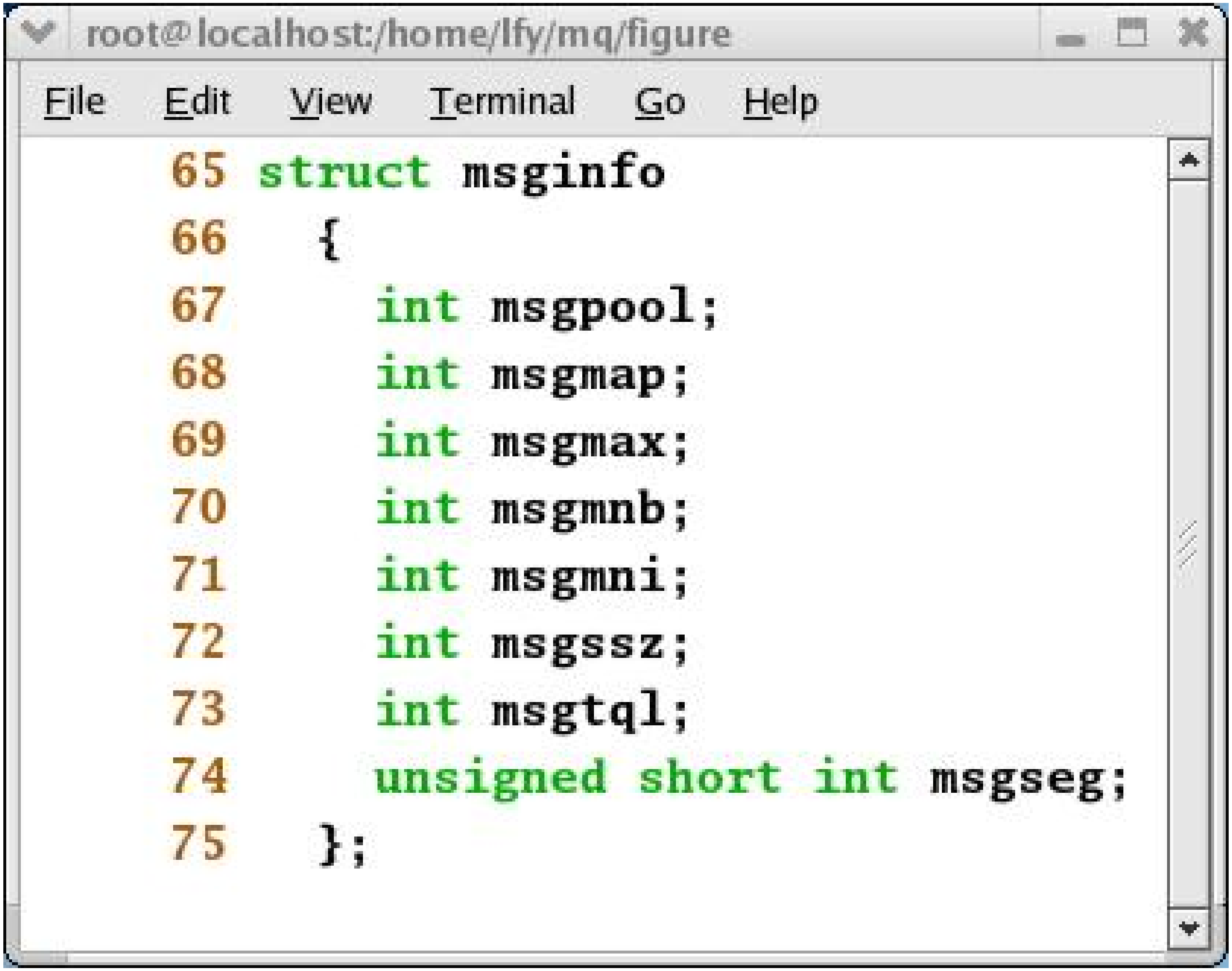
```c
struct msqid_ds
{
  struct ipc_perm msg_perm;     /* structure describing operation permission */
  __time_t msg_stime;           /* time of last msgsnd command */
  unsigned long int __unused1;
  __time_t msg_rtime;           /* time of last msgrcv command */
  unsigned long int __unused2;
  __time_t msg_ctime;           /* time of last change */
  unsigned long int __unused3;
  unsigned long int __msg_cbytes; /* current number of bytes on queue */
  msgqnum_t msg_qnum;           /* number of messages currently on queue */
  msglen_t msg_qbytes;          /* max number of bytes allowed on queue */
  __pid_t msg_lspid;            /* pid of last msgsnd() */
  __pid_t msg_lrpid;            /* pid of last msgrcv() */
  unsigned long int __unused4;
  unsigned long int __unused5;
};
```

Figure 3



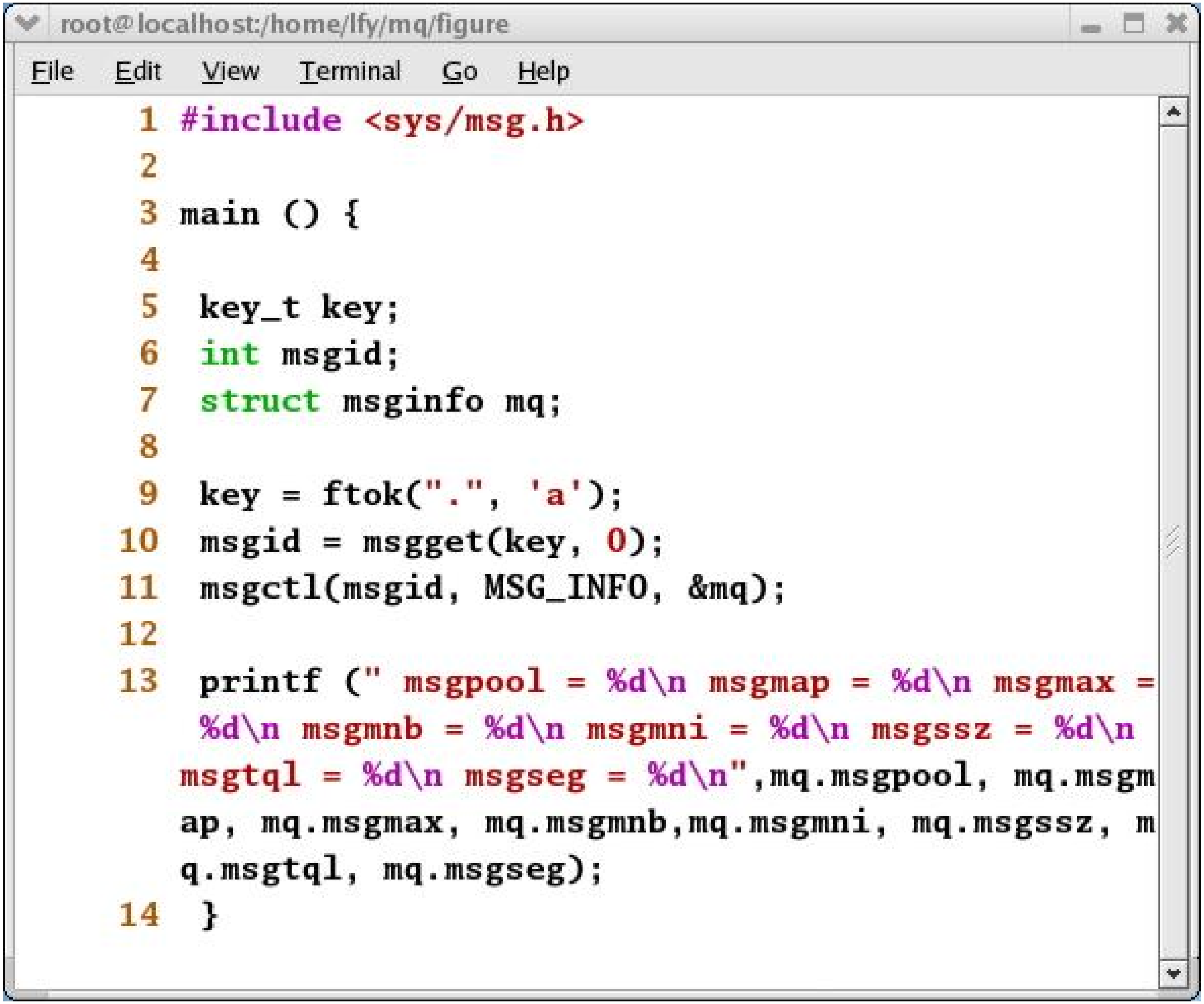
```c
#include <sys/msg.h>
#include <sys/time.h>

struct ipc_perm msg_perm;
struct msqid_ds mq;

main () {

 key_t key;
 int msgid;

 key = ftok(".", 'a');
 msgid = msgget(key, 0);
 msgctl(msgid, IPC_STAT, &mq);

 printf (" msg_perm.cuid = %d\n", msg_perm.cuid);
 printf (" msg_perm.uid = %d\n", msg_perm.uid);
 printf (" msg_perm.gid = %d\n", msg_perm.gid);
 printf (" msg_perm.cgid = %d\n", msg_perm.cgid);
 printf (" msg_perm.mode = %03o\n", mq.msg_perm.mode & 0777);
 printf (" msg_perm.msg_qnum = %lu\n", mq.msg_qnum);
 printf (" msg_perm.msg_lspid = %d\n", mq.msg_lspid);
 printf (" msg_perm.msg_lrpid = %d\n", mq.msg_lrpid);
 printf (" msg_perm.msg_stime = %s", ctime(&mq.msg_stime));
 printf (" msg_perm.msg_rtime = %s", ctime(&mq.msg_rtime));
 printf (" msg_perm.msg_ctime = %s", ctime(&mq.msg_ctime));
 printf (" msg_perm.msg_qbytes = %lu\n", mq.msg_qbytes);
 printf (" msg_perm.msg_cbytes = %lu\n", mq.msg_cbytes);
}
```

Figure 4



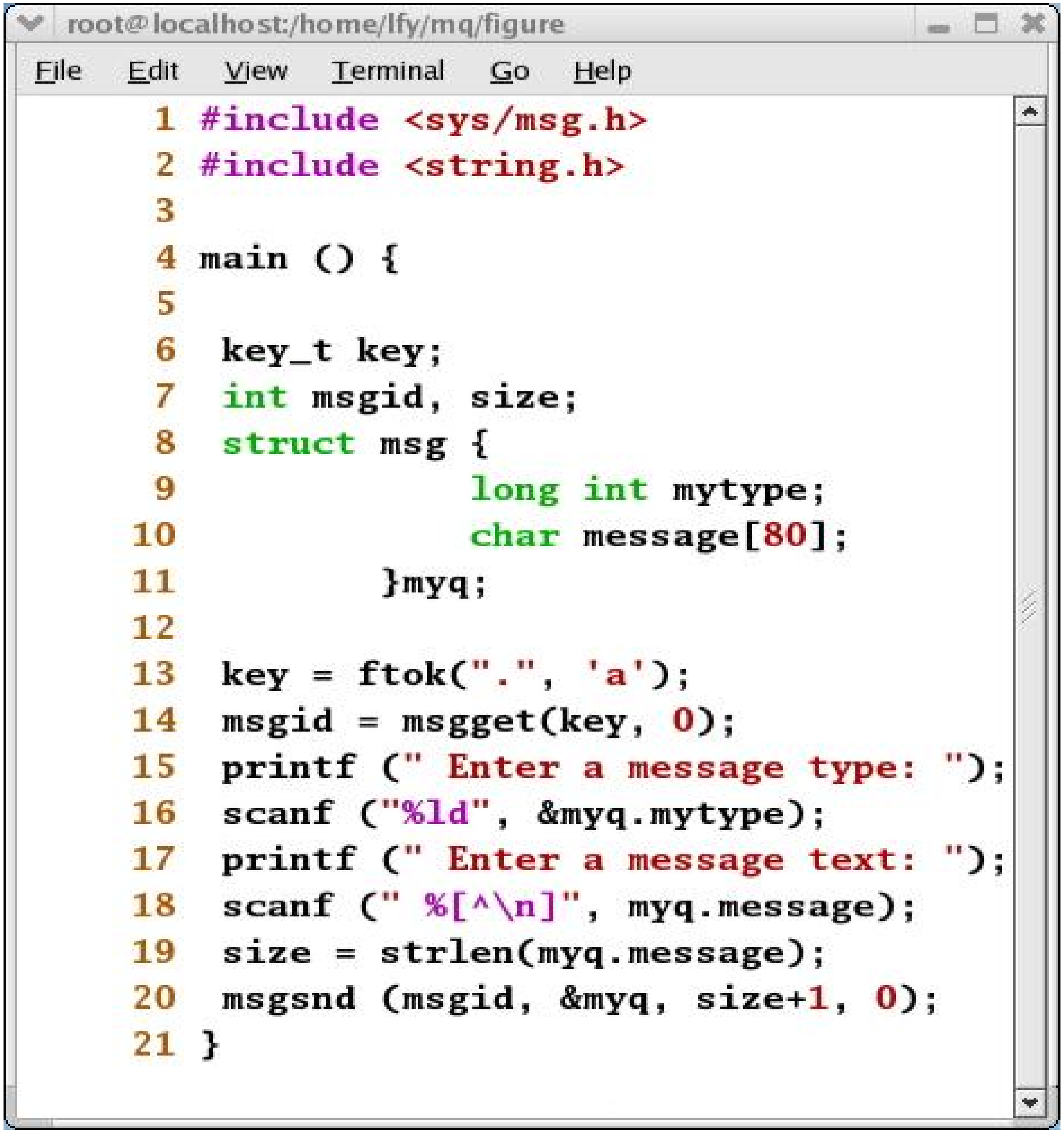
```
struct msginfo
  {
    int msgpool;
    int msgmap;
    int msgmax;
    int msgmnb;
    int msgmni;
    int msgssz;
    int msgtql;
    unsigned short int msgseg;
  };
```

Figure 5



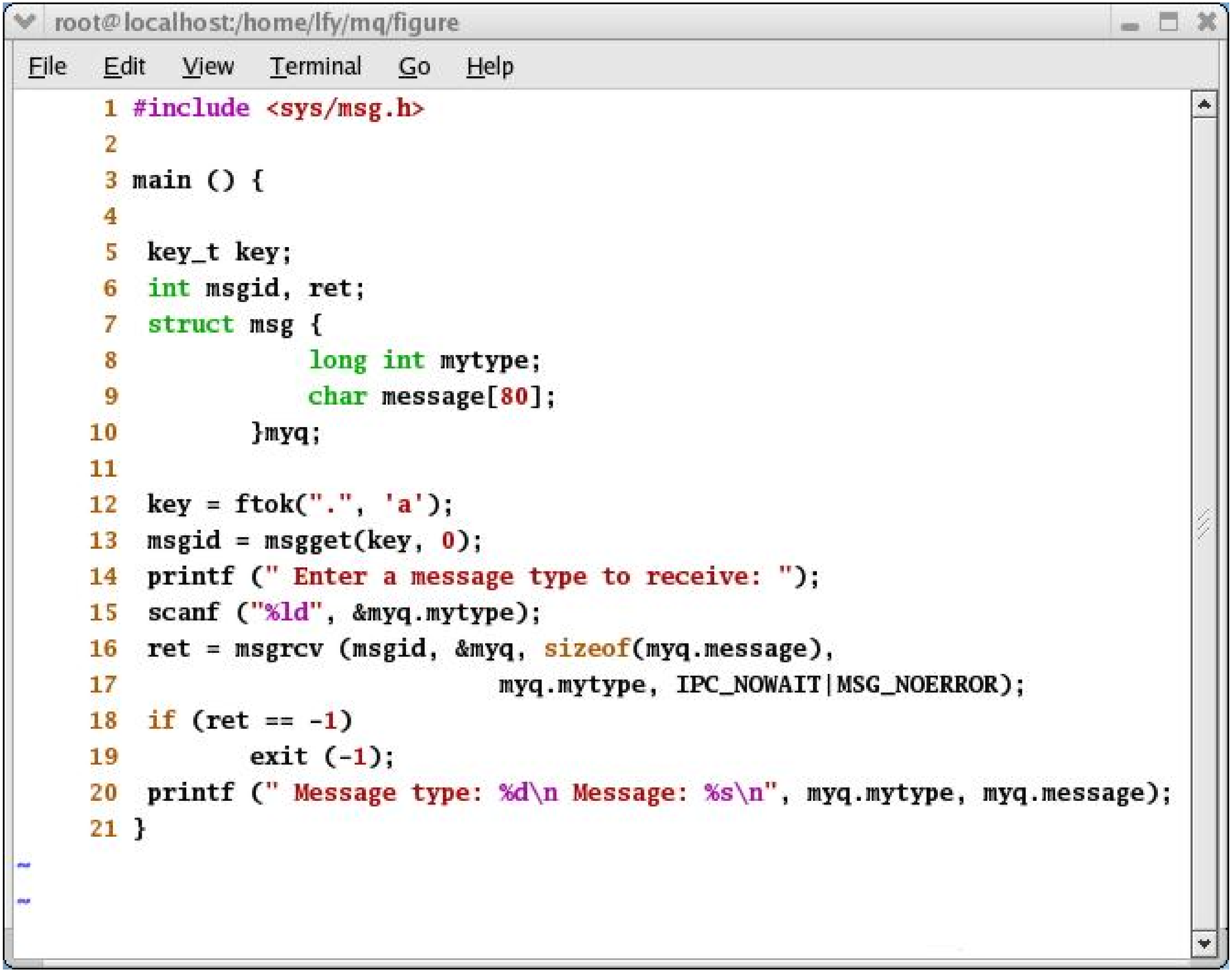
```
#include <sys/msg.h>

main () {

 key_t key;
 int msgid;
 struct msginfo mq;

 key = ftok(".", 'a');
 msgid = msgget(key, 0);
 msgctl(msgid, MSG_INFO, &mq);

 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.msgmap, mq.msgmax, mq.msgmnb,mq.msgmni, mq.msgssz, mq.msgtql, mq.msgseg);
 }
```

Figure 6



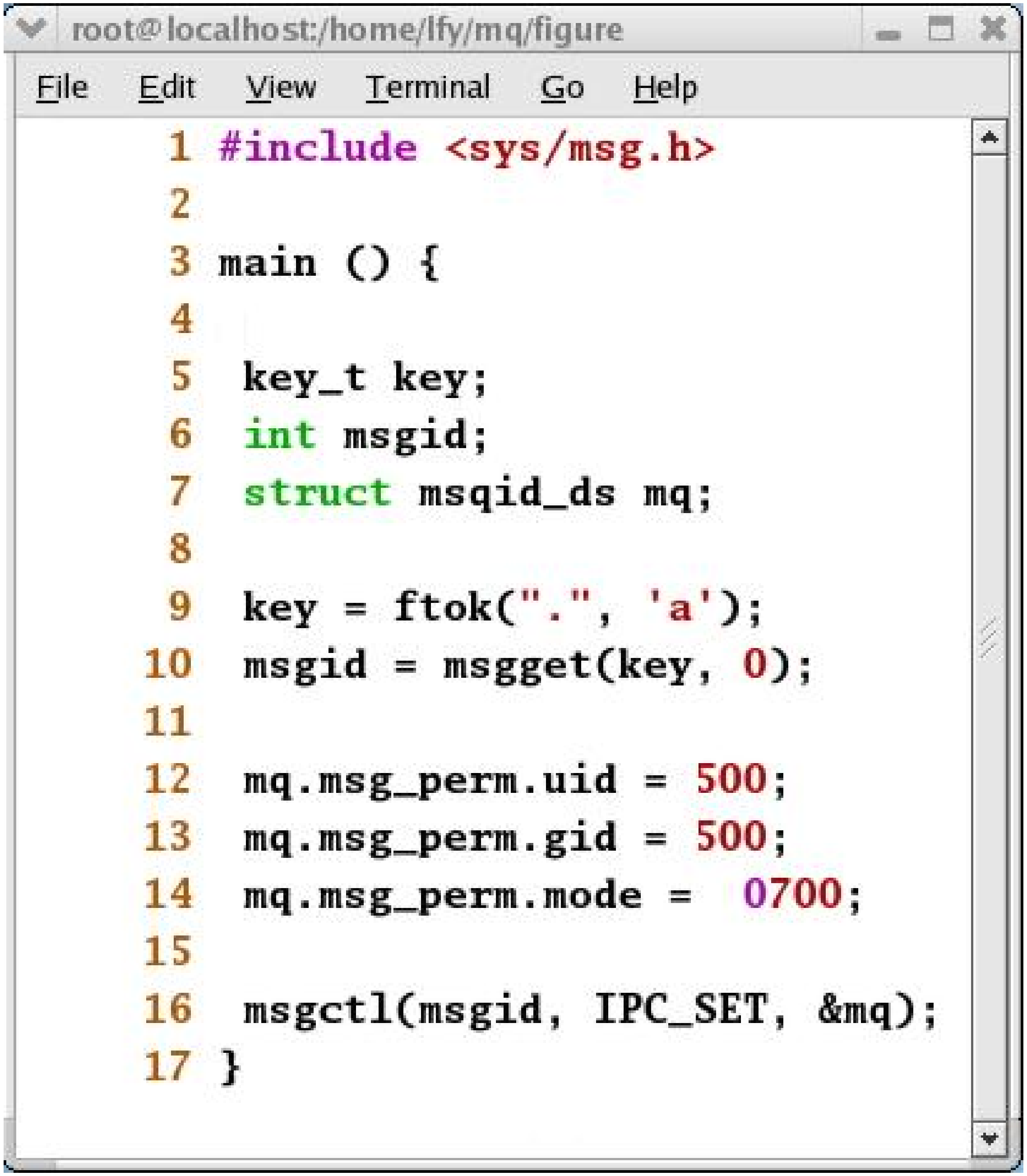
```
#include <sys/msg.h>
#include <string.h>

main () {

 key_t key;
 int msgid, size;
 struct msg {
            long int mytype;
            char message[80];
        }myq;

 key = ftok(".", 'a');
 msgid = msgget(key, 0);
 printf (" Enter a message type: ");
 scanf ("%ld", &myq.mytype);
 printf (" Enter a message text: ");
 scanf (" %[^\n]", myq.message);
 size = strlen(myq.message);
 msgsnd (msgid, &myq, size+1, 0);
}
```

Figure 7



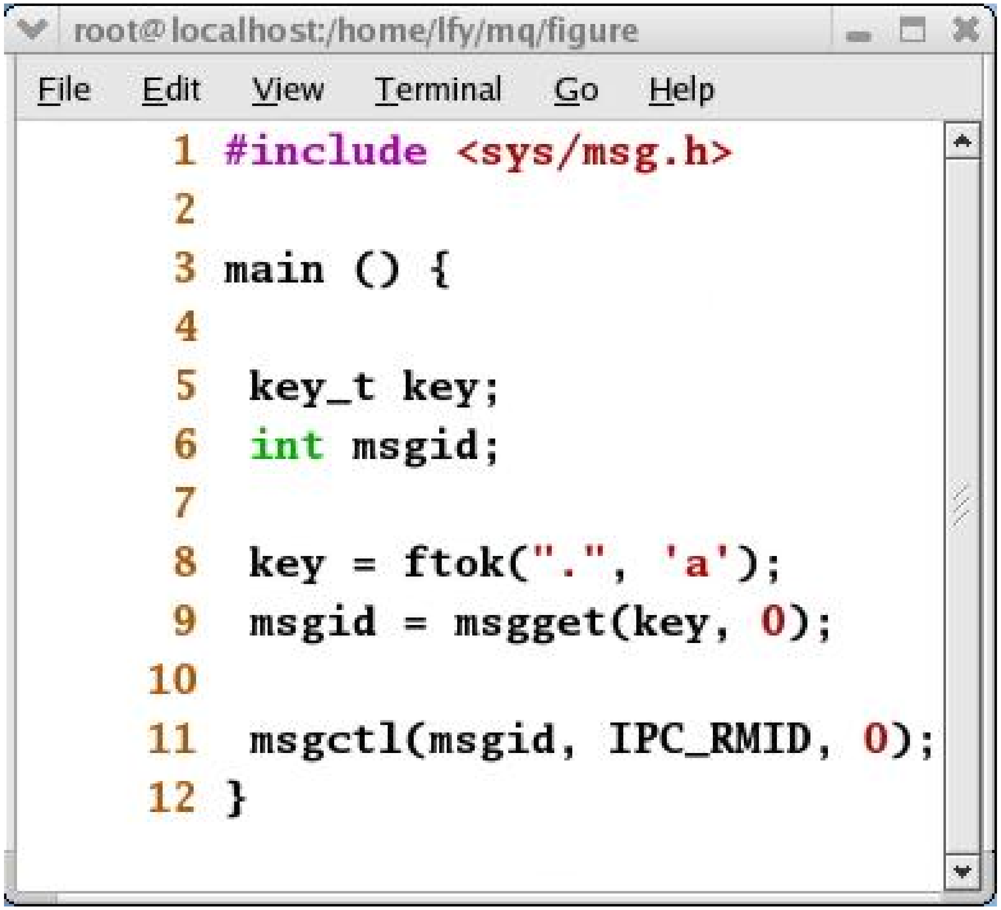
```
#include <sys/msg.h>

main () {

 key_t key;
 int msgid, ret;
 struct msg {
            long int mytype;
            char message[80];
        }myq;

 key = ftok(".", 'a');
 msgid = msgget(key, 0);
 printf (" Enter a message type to receive: ");
 scanf ("%ld", &myq.mytype);
 ret = msgrcv (msgid, &myq, sizeof(myq.message),
                         myq.mytype, IPC_NOWAIT|MSG_NOERROR);
 if (ret == -1)
        exit (-1);
 printf (" Message type: %d\n Message: %s\n", myq.mytype, myq.message);
}
~
~
```

Figure 8



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

main () {

 key_t key;
 int msgid;
 struct msqid_ds mq;

 key = ftok(".", 'a');
 msgid = msgget(key, 0);

 mq.msg_perm.uid = 500;
 mq.msg_perm.gid = 500;
 mq.msg_perm.mode =  0700;

 msgctl(msgid, IPC_SET, &mq);
}
```

Figure 9



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

main () {

 key_t key;
 int msgid;

 key = ftok(".", 'a');
 msgid = msgget(key, 0);

 msgctl(msgid, IPC_RMID, 0);
}
```

Figure 10