function first. As described previously, we don't consider message queues connectionless, since some technique is required to obtain the identifier for a queue. Since all these forms of IPC are restricted to a single host, all are reliable. When the messages are sent across a network, the possibility of messages being lost becomes a concern. Flow control means that the sender is put to sleep if there is a shortage of system resources (buffers) or if the receiver can't accept any more messages. When the flow control condition subsides, the sender should automatically be awakened.

One feature that we don't show in Figure 14.15 is whether the IPC facility can automatically create a unique connection to a server for each client. We'll see in Chapter 15 that streams and Unix stream sockets provide this capability.

The next three sections describe each of the three forms of System V IPC in detail.

14.7 Message Queues

Message queues are a linked list of messages stored within the kernel and identified by a message queue identifier. We'll call the message queue just a "queue" and its identifier just a "queue ID." A new queue is created, or an existing queue is opened by `msgget`. New messages are added to the end of a queue by `msgsnd`. Every message has a positive long integer type field, a nonnegative length, and the actual data bytes (corresponding to the length), all of which are specified to `msgsnd` when the message is added to a queue. Messages are fetched from a queue by `msgrcv`. We don't have to fetch the messages in a first-in, first-out order. Instead, we can fetch messages based on their type field.

Each queue has the following `msqid_ds` structure associated with it. This structure defines the current status of the queue.

```
struct msqid_ds {
  struct ipc_perm  msg_perm; /* see Section 14.6.2 */
  struct msg  *msg_first;  /* ptr to first message on queue */
  struct msg  *msg_last;   /* ptr to last message on queue */
  ulong        msg_cbytes; /* current # bytes on queue */
  ulong        msg_qnum;   /* # of messages on queue */
  ulong        msg_qbytes; /* max # of bytes on queue */
  pid_t        msg_lspid;  /* pid of last msgsnd() */
  pid_t        msg_lrpid;  /* pid of last msgrcv() */
  time_t       msg_stime;  /* last-msgsnd() time */
  time_t       msg_rtime;  /* last-msgrcv() time */
  time_t       msg_ctime;  /* last-change time */
};
```

The two pointers, `msg_first` and `msg_last` are worthless to a user process, as these point to where the corresponding messages are stored within the kernel. The remaining members of the structure are self-defining.

Figure 14.16 lists the system limits (Section 14.6.3) that affect message queues.