```
type chanName = rec(int machine, index);

chanName createChan(int machine) {
  chanName chan;
  if (machine is local) {
    get an empty channel descriptor and initialize it;
    chan = chanName(local machine number, address of descriptor);
  } else {
    netWrite(machine, CREATE_CHAN, executing);
    insert descriptor of executing on delay list;
    executing = 0;
  }
  dispatcher();
}

proc remoteCreate(int creator) {
  chanName chan;
  get an empty channel descriptor and initialize it;
  chan = chanName(local machine number, address of descriptor);
  netWrite(creator, CHAN_DONE, chan);
  dispatcher();
}

proc chanDone(int creator; chanName chan) {
  remove descriptor of process creator from the delay list;
  save chan as return value for creator;
  insert the descriptor of creator at the end of the ready list;
  dispatcher();
}

proc sendChan(chanName chan; byte msg[*]) {
  if (chan.machine is local)
    same actions as sendChan in Figure 10.1;
  else
    netWrite(chan.machine, SEND, msg);
  dispatcher();
}
```

```
proc remoteSend(chanName chan; int buffer) {
  find descriptor of channel chan;
  if (blocked list empty)
    insert buffer on message list;
  else {
    remove process from blocked list;
    copy message from buffer to the process's address space;
    insert the process at the end of the ready list;
  }
  dispatcher();
}

proc receiveChan(int chan; result byte msg[*]) {
  same actions as receiveChan in Figure 10.1;
}

bool emptyChan(int chan) {
  same actions as emptyChan in Figure 10.1;
}
```

Figure 10.4 Distributed kernel primitives.