```
int createChan(int msgSize) {
  get an empty channel descriptor and initialize it;
  set return value to the index or address of the descriptor;
  dispatcher();
}

proc sendChan(int chan; byte msg[*]) {
  find descriptor of channel chan;
  if (blocked list empty) {  # save message
    acquire buffer and copy msg into it;
    insert buffer at end of message list;
  }
  else {  # give message to a receiver
    remove process from blocked list;
    copy msg into the process's address space;
    insert the process at end of ready list;
  }
  dispatcher();
}

proc receiveChan(int chan; result byte msg[*]) {
  find descriptor of channel chan;
  if (message list empty) {  # block receiver
    insert executing at end of blocked list;
    store address of msg in descriptor of executing;
    executing = 0;
  }
  else {  # give receiver a stored message
    remove buffer from message list;
    copy contents of buffer into msg;
  }
  dispatcher();
}

bool emptyChan(int chan) {
  bool r = false;
  find descriptor of channel chan;
  if (message list empty)
    r = true;
  save r as the return value;
  dispatcher();
}
```

**Figure 10.1** Asynchronous message passing in a single-processor kernel.

Copyright © 2000 by Addison Wesley Longman, Inc.