16.7) Out-of-Band Data

* Allows higher-priority delivery of data than normal
* Sent ahead of any data that is already queued for transmission
* Supported by TCP, not UDP
* Referred as “urgent” data by TCP
* TCP supports only a single byte of urgent data
* Specify MSG\_OOB flag to any of the 3 send functions
* If multiple bytes are sent, last byte will be the urgent data
* SIGURG signal sent when data is received, if signal generation by the socket was arranged
* Fcntl(sockfd, F\_SETOWN, pid; //2nd arg used to set ownership of socket; if pid > 0 then pid = process ID, otherwise if pid < -1, pid = process group ID
* Replace F\_SETOWN with F\_GETOWN and pid = 0, to retrieve the current socket ownership
* Owner = fcntl(sockfd, F\_GETOWN, 0); // owner > 0 then owner = process ID and owner < 0 then owner = absolute value of process ID group
* Urgent mark = the point in the normal data stream that the urgent data will go
* Int sockatmark(int sockfd); // returns 1 if at mark, 0 if not at mark, -1 on error

16.8) Nonblocking and Asynchronous I/O

* Send functions will block when there is not enough room in the socket’s output queue to send the message
* But if function is in nonblocking mode, function will fail and return error instead
* Then use poll/select to determine when we can receive or transmit data
* Enabling asynchronous I/O
  + Establish socket ownership; signals can be delivered to proper processes
    - Use F\_SETOWN with fcntl
    - Use FIOSETOWN with ioctl
    - Use SIOCSPGRP with ioctl
  + Inform socket that signal must be sent when I/O operations won’t block
    - Use F\_SETFL with fcntl and enable O\_ASYNC
    - Use FIOASYNC with ioctl