**BÀI TẬP LÝ THUYẾT 4**

**1) Trình bày rõ hơn ý nghĩa và chi tiết từng cờ trong lệnh**

***- recvfrom():***

ssize\_t recvfrom(int sockfd, void \*buf, size\_t len, int flags, struct sockaddr \*from, socklen\_t \*fromlen );

Một tham số có thể được đặt thành 0, MSG\_CONNTERM, MSG\_PEEK, MSG\_OOB hoặc MSG\_WAITALL. Cờ MSG\_CONNTERM loại trừ lẫn nhau với các cờ khác.

+ MSG\_CONNTERM: Yêu cầu rằng chức năng hoàn thành chỉ khi kết nối TCP bị ngắt. Nó chỉ hợp lệ cho các socket TCP. Các yêu cầu nhận bình thường khác cũng đã hoàn thành. Ứng dụng phải có khả năng đối phó với thực tế là nhận bình thường và nhận kết thúc kết nối đặc biệt này có thể được điều khiển song song.

+ MSG\_OOB: Đọc mọi dữ liệu ngoài băng tần trên socket. Dữ liệu ngoài băng tần được gửi khi cờ MSG\_OOB bật cho send (), sendto () hoặc sendmsg ().

Lệnh fcntl () nên được sử dụng với F\_SETOWN để chỉ định người nhận, là pid hoặc gid, của tín hiệu SIGURG sẽ được gửi khi dữ liệu ngoài dải được gửi. Nếu không có người nhận nào được đặt, sẽ không có tín hiệu nào được gửi đi. Để biết thêm thông tin, hãy xem lệnh fcntl (). Người nhận dữ liệu xác định xem có nhận nội tuyến dữ liệu ngoài băng tần hay không bằng cách cài đặt tùy chọn SO\_OOBINLINE của setsockopt (). Nếu SO\_OOBINLINE được đặt và cờ MSG\_OOB được bật, byte dữ liệu ngoài dải sẽ được đọc ngoài dòng. Cờ MSG\_OOB được đặt khi SO\_OOBINLINE được bật là không hợp lệ. Nếu có sẵn dữ liệu ngoài băng tần và cờ MSG\_OOB không được đặt (có thể bật hoặc tắt SO\_OOBINLINE), thì dữ liệu tối đa, nhưng không bao gồm, dữ liệu ngoài băng tần sẽ được đọc. Khi con trỏ đọc đạt đến byte dữ liệu ngoài dải, thì chỉ những dữ liệu ngoài băng tần mới được đọc trong lần đọc tiếp theo. Tùy chọn SIOCATMARK của ioctl () có thể được sử dụng để xác định xem con trỏ đọc hiện có ở byte dữ liệu ngoài dải hay không. Để biết thêm thông tin, hãy tham khảo lệnh setsockopt () và ioctl ().

+ MSG\_PEEK: Nhìn vào dữ liệu hiện có trên socket; dữ liệu được trả về nhưng không được sử dụng để thao tác nhận tiếp theo thấy cùng một dữ liệu.

+ MSG\_WAITALL: Yêu cầu khối chức năng cho đến khi có thể trả lại toàn bộ lượng dữ liệu được yêu cầu. Hàm có thể trả về lượng dữ liệu nhỏ hơn nếu bắt được tín hiệu, kết nối bị ngắt, lỗi đang chờ xử lý hoặc SO\_RCVTIMEO được đặt và bộ hẹn giờ hết hạn cho socket. Đối với AF\_UNIX, hàm cũng có thể trả về sớm hơn nếu dữ liệu ngoài băng tần (OOB) được nội tuyến và có dữ liệu OOB để đọc. Trong trường hợp này, dữ liệu lên đến dữ liệu OOB được trả về trên recvfrom () đầu tiên. Dữ liệu OOB được trả về khi yêu cầu đọc tiếp theo.

***-sendto()***

ssize\_t sendto(int sockfd, void \*buf, size\_t len, int flags, struct sockaddr \*to, socklen\_t tolen );

Việc đặt các cờ không được hỗ trợ trong miền AF\_UNIX. Các cờ sau có sẵn:

+ MSG\_OOB: Gửi dữ liệu ngoài băng tần trên socket. Chỉ các socket SOCK\_STREAM hỗ trợ dữ liệu ngoài băng tần. Dữ liệu ngoài dải là một byte đơn.

Trước khi dữ liệu ngoài băng tần có thể được gửi giữa hai chương trình, cần phải có một số nỗ lực phối hợp. Nếu dữ liệu dự định không được đọc nội tuyến, người nhận dữ liệu ngoài băng tần phải chỉ định người nhận tín hiệu SIGURG được tạo ra khi dữ liệu ngoài băng tần được gửi đi. Nếu không có người nhận nào được đặt, sẽ không có tín hiệu nào được gửi. Người nhận được thiết lập bằng cách sử dụng toán hạng F\_SETOWN của lệnh fcntl (), chỉ định pid hoặc gid. Để biết thêm thông tin về toán hạng này, hãy tham khảo lệnh fcntl ().

Người nhận dữ liệu xác định xem có nhận nội tuyến dữ liệu ngoài băng tần hay không bằng cách cài đặt tùy chọn SO\_OOBINLINE của setsockopt (). Để biết thêm thông tin về việc nhận dữ liệu ngoài băng tần, hãy tham khảo các lệnh setsockopt (), recv (), recvfrom () và recvmsg ().

+ MSG\_DONTROUTE: Tùy chọn SO\_DONTROUTE được bật trong suốt thời gian hoạt động. Điều này thường chỉ được sử dụng bởi các chương trình chẩn đoán hoặc định tuyến.

**2) Giá trị trả về EMSGSIZE trong câu lệnh sendto() có ý nghĩa gì?**

Khi Message quá lớn để được gửi dưới dạng một sơ đồ dữ liệu thì sẽ trả về EMSGSIZE.

**3) Các trường hợp nào nên sử dụng UDP và tại sao?**

*Các trường hợp nên sử dụng UDP:*

+ Sử dụng cho các phương thức truyền broadcasting và multicasting khi chúng ta muốn truyền tin với nhiều host.

+ Kích thước datagram nhỏ và trình tự đoạn là không có ảnh hưởng

+ Không cần thiết lập quan hệ

+ Phần mềm không gửi các dữ liệu đáng kể

+ Không cần truyền lại các gói tin

+ Băng thông của mạng đóng vai trò đáng kể

*Lý do:*

Do UDP đáp ứng yêu cầu khi cần tốc độ nhanh và không cần thiết sửa lỗi nên nhiều trường hợp như trên nên sử dụng UDP.