**SOCKET TRONG LINUX**

1. **Socket là gì?**
   1. **Khái niệm socket**

**Socket là một phương thức giao trong hệ thông. Nó cho phép nhiều tiến trình trong hệ thông kết nối với nhau. Ta cũng có thể dùng socket để giao tiếp và chuyển dữ liệu giữa các hệ thống.**

* 1. **Các thành phần cấu thành socket**

**Socket có 3 thuộc tính đó là :domain, type, và protocol. Chúng cũng có một địa chỉ được dùng như tên. Các format của địa chỉ phụ thuộc vào tên miền. nó còn có tên khác là protocol family.**

* 1. **Vị trí socket trong mô hình OSI**

**Nó nằm trong layer 4 (Transport layer- điều khiển luông, phân đoạn gói tin, hợp nhất gói tin, sửa lỗi, báo nhận lại)**

1. **Các cấu trúc địa chỉ socket**

**Hầu hết cấu trúc địa chỉ của một socket sẽ được một con trỏ tới giống như một tham số. Từng giao thức sẽ được hỗ trợ cấu trúc địa chỉ riêng. Tên của cấu trúc sẽ bắt đầu sockaddr\_ và kết thúc với hậu tố đặc trưng của nó.**

* 1. **Cấu trúc địa chỉ socket IPv4**

**Địa chỉ IPv4 thường hay được gọi là Internet Socket address structure.**

struct in\_addr {

in\_addr\_t s\_addr; /\* 32-bit IPv4 address \*/

/\* network byte ordered \*/

};

struct sockaddr\_in {

uint8\_t sin\_len; /\* length of structure (16) \*/

sa\_family\_t sin\_family; /\* AF\_INET \*/

in\_port\_t sin\_port; /\* 16-bit TCP or UDP port number \*/

/\* network byte ordered \*/

struct in\_addr sin\_addr; /\* 32-bit IPv4 address \*/

/\* network byte ordered \*/

char sin\_zero[8]; /\* unused \*/

};

* 1. **Cấu trúc địa chỉ socket chung**

**Cấu trúc địa chỉ của socket được gọi đến giống như một tham số để tham triếu tới các hàm. Nhưng các hàm đó phải phù hợp với cấu trúc địa chỉ của socket và hỗ trợ họ giao thức đó. Vấn đề là làm như thế nào để làm được điều đấy, thì hầu như chugns ta thường dùng con trỏ void \* Nhưng cũng có cách khác là sử dụng địa chỉ socket chung được định nghĩa trong <sys/socket.h> header.**struct sockaddr {

uint8\_t sa\_len;

sa\_family\_t sa\_family; /\* address family: AF\_xxx value \*/

char sa\_data[14]; /\* protocol-specific address \*/

};

* 1. **Cấu trúc địa chỉ socket Ipv6**

truct in6\_addr {

uint8\_t s6\_addr[16]; /\* 128-bit IPv6 address \*/

/\* network byte ordered \*/

};

#define SIN6\_LEN /\* required for compile-time tests \*/

struct sockaddr\_in6 {

uint8\_t sin6\_len; /\* length of this struct (28) \*/

sa\_family\_t sin6\_family; /\* AF\_INET6 \*/

in\_port\_t sin6\_port; /\* transport layer port# \*/

/\* network byte ordered \*/

uint32\_t sin6\_flowinfo; /\* flow information, undefined \*/

struct in6\_addr sin6\_addr; /\* IPv6 address \*/

/\* network byte ordered \*/

uint32\_t sin6\_scope\_id; /\* set of interfaces for a scope \*/

};

* 1. **Cấu trúc địa chỉ socket chung mới**

**Cấu trúc địa chỉ socket chung mới được định nghĩa là một phần của IPv6 sockets API. Nó bù đắp một số thiếu xót của struct sockaddr. Không giống như sockaddr, cấu trúc mới này lớn hơn đủ để lưu trữ bất kì địa chỉ nào. Nó định nghĩa một cấu trúc gọi là sockaddr\_storage trong file header <netinet/in.h>**

struct sockaddr\_storage {

uint8\_t ss\_len; /\* length of this struct (implementation dependent) \*/

sa\_family\_t ss\_family; /\* address family: AF\_xxx value \*/

/\* *implementation-dependent elements to provide:*

\* *a) alignment sufficient to fulfill the alignment requirements of*

\* *all socket address types that the system supports*.

\* *b) enough storage to hold any type of socket address that the*

\* *system supports*.

\*/

};

1. **Các hàm chuyển đổi địa chỉ**
   1. **inet\_aton, inet\_addr và inet\_ntoa**
   2. **inet\_pton và inet\_ntop**
   3. **sock\_ntop**
2. **Hàm tạo và hủy socket**
   1. **Tạo socket**

**Để tạo được một socket ta có thể sử dụng hàm socket(int domain, int type, int protocol)**

**tham số domain được định nghĩa :AF\_UNIX socket trong local system. AF\_INET socket cho mạng…. Nhưng thường ta chỉ sử dụng hai trường này.**

**Trường type: thường SOCK\_STREAM phương thức truyền TCP và SOCK\_DGRAM phương thức truyền UDP.**

**Trường protocol: thường đặt là 0 là giá trị mặc định.**

**Địa chỉ socket: ta có hai cấu trúc là struct sockadd\_un cho local system và sockaddr\_in cho socket Internet.**

**Hàm bind(int socket, const struct sockaddr\* address, size\_t address\_len)**

**int socket là kết quả tra ra từ hàm socket()**

**const struct sockaddr\* address: thường là (struct sockaddr\*)&dia chi struct address,**

**size\_t address\_len: sizeof(address)**

**Hàm int accept(int socket, struct sockaddr \* address, size\_t \*address\_len);**

**int socket đã được đề cập ở trên.**

**struct sockaddr \* address :đề cập ở trên.**

**size\_t \*address\_len= địa chỉ của sizeof(address).**

**int connect(int socket, const struct sockaddr \* address, size\_t address\_len);**

**Nếu thành công return 0, -1 errors, các error của nó được miêu tả trong sách.**

* 1. **Hủy socket**

**hàm close(int socket);**

* 1. **Các hàm đọc ghi dữ liệu: readn, writen và readline**

1. **UDP socket**
   1. **UDP socket là gì?**

**Trong một số trường hợp việc thiết lập trước và duy trì một kết nối thì là không cần thiết. Nếu có một dịch vụ cả ngày sử dụng một lần và ta lại luôn kết nối thì đó là lãng phí và không phù hợp vì thế socket UDP có thể giải quyết điều đó.**

* 1. **Đặc điểm của UDP socket**

**Nó hiệu quả về chi phí và thời gian nhưng không đảm bảo về độ tin cậy**

* 1. **Các hàm chức năng của UDP socket**
     1. **Hàm recvfrom và hàm sendto**