Ex.No:2

IMPLEMENTATION OF ADDRESS CONVERSION ROUTINES

AIM

To implement address conversion routines using C program.

DESCRIPTION

Network Byte Ordering

The network byte order is defined always to be Big-Endian, which may differ from host byte order on a particular machine. Using network byte ordering for data exchanged between hosts allows hosts using different architecture to exchange address information without confusion because of byte ordering.

Host Byte Order

Host Byte Order refers that how the bytes are arranged when referring to the computer architecture of a host computing platform. It is generally Little-Endian.

Big-Endian

Big-Endian is an order in which most significant byte is stored at the lowest storage address.

Little-Endian

Little-Endian is an order in which least significant byte is stored at the lowest storage address.

BYTE ORDERING FUNTIONS

htonl()

Syntax:

```
Uint 32_thtonl(uint32_t hostlong);
```

Purpose:

htonl() function converts the unsigned integer hostlong from host byte order to network byte order.

htons()

Syntax:

```
Uint 16_t htons(uint 16_t host_short);
```

purpose:

htons() function converts the usigned 16-bit short integer from host byte order to network byte order.

ntohl()

Syntax:

```
Uint32_t ntohl(uint32_t netlong);
```

Purpose:

The ntohl() function converts the unsigned 32 bit integer netlong from network byte order to host byte order.

ntohs()

Syntax:

uint16_t ntohs(uint16_t netshort);

Purpose:

The ntohs() function converts the unsigned short integer netshort from network byte order to host byte order.

ADDRESS CONVERSION FUCNTIONS

inet_aton()

Syntax:

int inet_aton(const char *strptr, struct in_addr *addrptr);

returns 1 if the string is valid

returns 0 on error.

purpose:

inet_aton converts the c character string pointed to by strptr into its 32-bit binary network byte ordered value, which is stored through the pointer addrptr. If successful 1 is returned else 0 is returned.

inet_addr()

Syntax:

in_addr_t inet_addr(const char *strptr);

Purpose:

inet_addr returns the 32-bit binary network

byte ordered value as the return value.

inet_ntoa()

Syntax:

```
char *inet_ntoa(struct in_addr inaddr);
```

Purpose:

This function converts a 32-bit binary network byte ordered IPv4 address into its corresponding dotted-decimal string. The string pointed to by the return value of the function reside in static memory.

inet_pton()

Syntax:

```
int inet_pton(int family, const char *strptr,
              void *addrptr);
```

Purpose:

The inet_pton() tries to convert the string pointed by strptr, storing the binary result through the pointer addrptr.

inet_ntop()

Syntax:

```
const char *inet_ntop(int family, const void *addrptr,
                      char *strptr, size_t len);
```

purpose:

It converts numeric (addrptr) to presentation (strptr). The len argument is the size of the destination, to prevent the function from overflowing the caller's buffer.