|  |  |
| --- | --- |
| **Exp. No: 1** | **Name of the Exercise**  Study of necessary header files with respect to socket programming. |
| **Date: 17/8/22** |

**Aim:** To study the necessary header files with respect to socket programming.

**Description:**

The SAS/C Socket Library provides header files to enable you to program with socket functions. A list of the header files, accompanied by a brief description of each one and an explanation of its structures, follows:

1. **<sys/types.h> :** This header file contains definitions to allow for the porting of BSD programs. **<sys/types.h>** must usually be included before other socket-related header files; refer to the individual header file descriptions that follow for the specific dependency. Its format is #include <sys/types.h>.
2. **<sys/uio.h>:** The structure in the <sys/uio.h> header file is described in the following section. <sys/types.h> must be included before this header file. The iovec structure is used by the readv , writev , sendmsg , and recvmsg calls. An array of iovec structures describes the pieces of a noncontiguous buffer. Its format is #include <sys/uio.h>.
3. **<errno.h>:** This header file contains definitions for the macro identifiers that name system error status conditions. When a SAS/C Library function sets an error status by assigning a nonzero value to errno , the calling program can check for a particular value by using the name defined in <errno.h>. Its format is #include <errno.h>.
4. **<sys/ioctl.h>:** This header file contains definitions for the symbols required by the ioctl function, as well as the declaration for ioctl. Its format is #include <sys.ioctl.h>.
5. **<sys/socket.h>:** This header file contains macro definitions related to the creation of sockets, for example, the type of socket (stream, datagram, or raw), the options supported, and the address family. (AF\_UNIX is supported if integrated sockets are used.) The SAS/C Compiler only supports the TCP/IP and the AF\_INET Internet address family. The <sys/socket.h> header file contains declarations for most of the functions that operate on sockets. Its format is #include <sys/socket.h>.
6. **<netdb.h>:** This header file contains structures returned by the network database library. Internet addresses and port numbers are stored in network byte order, identical to IBM 370 byte order. Other quantities, including network numbers, are stored in host byte order. Despite the fact that network byte order and host byte order are identical on the IBM System/370, a portable program must distinguish between the two. Its format is #include <netdb.h>.
7. **<netinet/in.h>:** This header file contains constants and structures defined by the Internet system. Several macros are defined for manipulating Internet addresses. Among these are INADDR\_ANY , which indicates that no specific local address is required, and INADDR\_NONE , which generally indicates an error in address manipulation functions. Refer to bind for more information on binding a local address to the socket. Its format is #include <netinet/in.h>.
8. **<netinet/in\_systm.h>:** This header file contains definitions to facilitate the porting of low-level network control and query Internetwork Control and Message Protocol (ICMP), and Internetwork Protocol (IP) raw socket type applications. The Internet ping client utility is an example of such a program. <netinet/in\_systm.h> must usually be included before other ICMP or IP socket related header files such as <netinet/ip.h> and <netinet/ip\_icmp.h>. Its format is #include <netinet/in\_systm.h>.
9. **<netinet/udp.h>:** This header file contains definitions of the User Datagram Protocol (UDP) header for UDP datagrams. UDP datagrams consist of a fixed header section immediately followed by the data section. The length of entire datagram, including the header and data, is maintained in UDP length field as a count of the number of octets (an octet is 8 bits; on IBM S370/390 systems this is 1 byte) in the datagram. Thus, the mininum value is 8 (64 bits), which is the length of the header alone. Its format is #include <netinet/udp.h>.
10. **<arpa/inet.h>:** This header file contains declarations for the network address resolution functions. You must include the <netinet/in.h> header file before this header file. Its format is #include <arpa/inet.h>.
11. **<arpa/nameser.h>:** This header file contains definitions that enable applications to communicate with Internet name servers. The contents of this header file are not of interest to most applications; however, this header file must be included before the <resolv.h> header file. Applications that manipulate resolver options must include this header file. Its format is #include <arpa/nameser.h>.
12. **<resolv.h>:** This header file contains global definitions for the resolver. Definitions and structures in the <resolv.h> header file are discussed in the following sections. You must include the <sys/types.h> , <netinet/in.h> , and <arpa/nameser.h> header files before this header file. Its format is #include <resolv.h>.
13. **<net/if.h>:** This header file contains structures that define the network interface and provide a packet transport mechanism. <net/if.h> is useful only for low-level programming of the network interface. You must include the <sys/types.h> and <sys/socket.h> header files before this header file. Its format is #include <net/if.h>.
14. **<strings.h>:** This header file provides compatibility with the BSD UNIX <strings.h> header file and the index , rindex , bzero , ffs , and bcmp functions.
15. **<netinet/ip\_icmp.h>:**This header file contains definitions of constants and structures required for using the ICMP protocol as described in IBM's RFC 792.

**Result:** Thus, study of necessary header files with respect to socket programming concluded successfully.