CHAPTER 16

Socket Interface

16.1 MULTIPLE-CHOICE QUESTIONS

1. d	3. c	5. d	7. b	9. a
11. d	13. d	15. a	17. c	19. d
21. c	23. b	25. b	27. a	29. d
31. b	33. b	35. b		

16.2 EXERCISES

37. The bind function associates a socket with a local socket address. The listen function tells the operating system that a server process is ready to accept connections through an existing socket.

39.
```
activeSocket = socket ( AF_INET, SOCK_STREAM, 0 ) ;
memset ( &remoteAddr, 0, sizeof ( remoteAddr ) ) ;
remoteAddr.sin_family = AF_INET ;
remoteAddr.sin_port = htons ( 23 ) ;
hptr = gethostbyname ( "xxx.yyy.edu" ) ;
memcpy ( ( char * ) &remoteAddr.sinaddr.s_addr,
            hptr -> h_addr_list [ 0 ], hptr -> h_length ) ;
if ( connect ( activeSocket, &remoteAddr, sizeof ( remoteAddr ) ) == -1 )
{
 printf ( "\n\a ==== Bad Connection ====\n\n" ) ;
 exit ( 101 ) ;
}
```

41.
```
passiveSocket = socket ( AF_INET, SOCK_STREAM, 0 ) ;
memset ( &localAddr, 0, sizeof ( localAddr ) ) ;
localAddr.sin_family = AF_INET ;
```

```
localAddr.sin_port = htons ( port_being_used ) ;
localAddr.sin_addr.s_addr = htonl ( INADDR_ANY ) ;
bind ( passiveSocket, &localAddr, sizeof ( localAddr ) ) ;
```

43.
```
for ( ; ; )
{
 childSocket = accept ( passiveSocket, &clientAddr, &clientAddrLength ) ;
 pid = fork ( ) ;
 if ( pid != 0 )
  {
 close ( childSocket ) ;
 continue ;
  } /* if parent process */
 else
  {
 close ( passiveSocket ) ;
 ...
  } /* else child process */
} /* for ( ever ) */
```

45. The recvfrom function extracts the next message that arrives at a socket. The read function reads data from an open connection to a remote process. No connection is required with the recvfrom function.

47. When either the client or server call the bind function, they are still in the CLOSED state. It is not until the client calls connect or the server calls listen that the state moves from the CLOSED state.

49. When the server calls the accept function, it is still in the LISTEN state. If no clients are waiting for service, the server remains in the LISTEN state until a connection request actually arrives from a client. At this point, the server moves to the SYN-RCVD state and stays there until the client moves to the ESTABLISHED state and sends an acknowledgment. When the server receives the acknowledgment, it moves to the ESTABLISHED state as well.