System call: Send()-sending data through a socket

send() — Sending Data Through a Socket

The send() system call is a core function used in network programming for transmitting data across a socket connection. It's part of the Berkeley sockets API, widely used for network communication in Unix-based operating systems, including FreeBSD.

The send() function is used to transmit data from the local system (client/server) to a connected socket. It sends raw bytes over the network to another host via TCP or UDP, depending on the socket type.

Function Prototype(C++)

*ssize_t send(int sockfd, const void *buf, size_t len, int flags);*

Parameter	**Description**
Sockfd…..	The socket file descriptor returned by socket()
Buf……	Pointer to the data buffer to send
Len……	Length of the data
Flags…….	Optional flags (usually set to 0)
Return	Number of bytes sent, or -1 on error

Prerequisites

Before using send(), you must:

1. Create a socket with socket()
2. Configure the server address using sockaddr_in
3. Connect to the server using connect()

4. Use send() to transmit data

Required Header File(C++)

```
#include <iostream>
#include <cstring>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/socket.h>
```

These provide functions and types for sockets (socket, connect, send), memory operations (memset, strlen), and I/O (read, write, close).

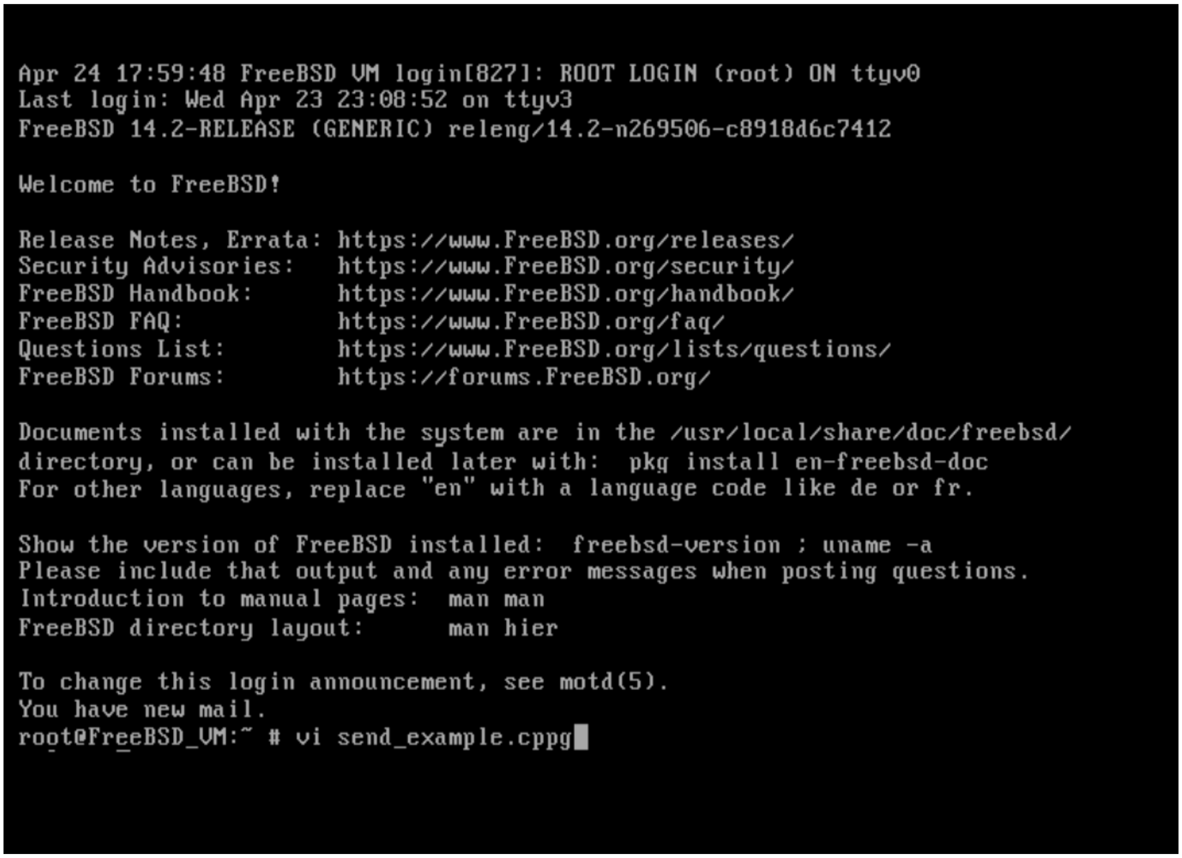
Example Using (C++)

```
#include <iostream>
#include <cstring>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/socket.h>
int main() {
    // 1. Create a socket
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0) {
        std::cerr << "Socket creation failed!\n";
        return 1;
}


    // 2. Define server address
    struct sockaddr_in serverAddr;
    std::memset(&serverAddr, 0, sizeof(serverAddr));
    serverAddr.sin_family = AF_INET;
```

```
    serverAddr.sin_port = htons(8080); // Convert port number to network byte order
    serverAddr.sin_addr.s_addr = inet_addr("127.0.0.1"); // localhost IP

    // 3. Connect to the server
    if (connect(sockfd, (struct sockaddr*)&serverAddr, sizeof(serverAddr)) < 0) {
        std::cerr << "Connection to server failed.\n";
        close(sockfd);
        return 1;
    }
    // 4. Data to send
    const char* message = "Hello from C++ client!";
    // 5. Use send() to send the message
    ssize_t bytesSent = send(sockfd, message, strlen(message), 0);
    if (bytesSent < 0) {
        std::cerr << "Failed to send message.\n";
    } else {
        std::cout << "Sent " << bytesSent << " bytes: " << message << std::endl;
    }
    // 6. Close the socket
    close(sockfd);
    return 0;
}
```


Apr 24 17:59:48 FreeBSD VM login[827]: ROOT LOGIN (root) ON ttyv0
Last login: Wed Apr 23 23:08:52 on ttyv3
FreeBSD 14.2-RELEASE (GENERIC) releng/14.2-n269506-c8918d6c7412
Welcome to FreeBSD!
Release Notes, Errata: https://www.FreeBSD.org/releases/
Security Advisories: https://www.FreeBSD.org/security/
FreeBSD Handbook: https://www.FreeBSD.org/handbook/
FreeBSD FAQ: https://www.FreeBSD.org/faq/
Questions List: https://www.FreeBSD.org/lists/questions/
FreeBSD Forums: https://forums.FreeBSD.org/
Documents installed with the system are in the /usr/local/share/doc/freebsd/
directory, or can be installed later with: pkg install en-freebsd-doc
For other languages, replace "en" with a language code like de or fr.
Show the version of FreeBSD installed: freebsd-version ; uname -a
Please include that output and any error messages when posting questions.
Introduction to manual pages: man man
FreeBSD directory layout: man hier
To change this login announcement, see motd(5).
You have new mail.
root@FreeBSD_VM:~ # vi send_example.cppg


send_example.cpp: new file: line 1

```
1   include <iostream>
2   include <cstring>
3   include <unisdd.h>
4   include <arpa/intet.h>
5
6   int main()
7   {
8       sockfd = socckaddr_in serverAdr
9       if (sockfd <= 0)
10          std:cerr Socket creation failed.';
11      return 1;
12      // Set up ster ddress
13      serverAddr.sin_family=AF_IN
14      serverAddr.sin_port= htons(8080);
15      if (connect(sockfd, (strn.addrr('\"127.0.0.1"))
16          close(sockfd)
17          return 1;
18      scnst char messag<== send(                );
19      el5;cerr std:cerr "Send failed.';
20          ccut.oout = "Message sent: "+ message\n ";
21      close(sockfd)
22  }
23
24  ssize_t bytesSent = "Hello from client!";
36
```

```
ajman - ↔    ./client
Message sent:Hello from client!
ajman - ↔
```

Fg. **Send()-sends data through a socket**