OSSP INDIDUAL ASSIGMENET

SYLABLLE OS

NAME	ID
ROBEL AGMAS	BDU1602334

Submission date 16/08/17

Submitted to wendmu baye

System call
recv() is a system call in the Berkeley sockets API used to receive messages from a socket. It is typically used with TCP (stream-oriented) sockets to read incoming data.

The recv() system call is a key function used in network socket programming, specifically for receiving data from a connected socket (usually over TCP). It is part of the BSD sockets API, which is widely supported in Unix/Linux and other operating systems.When a program communicates over a network using sockets, recv() is used read incoming data from the socket's receive buffer into the application's memory.

Cpp code for recv()

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

int main() {
    int sock = socket(AF_INET, SOCK_STREAM, 0);
    if (sock < 0) {
        std::cerr << "Socket creation failed\n";
        return 1;
    }

    sockaddr_in server_addr;
    server_addr.sin_family = AF_INET;
```

```
    server_addr.sin_port = htons(8080); // Server port
    server_addr.sin_addr.s_addr = inet_addr("127.0.0.1"); // Localhost

    // Connect to the server
    if (connect(sock, (struct sockaddr*)&server_addr, sizeof(server_addr)) < 0) {
        std::cerr << "Connection failed\n";
        close(sock);
        return 1;
    }

    char buffer[1024];
    memset(buffer, 0, sizeof(buffer));

    // Receive data from server
    int bytes_received = recv(sock, buffer, sizeof(buffer) - 1, 0);
    if (bytes_received < 0) {
        std::cerr << "recv() failed\n";
    } else if (bytes_received == 0) {
        std::cout << "Connection closed by server\n";
    } else {
        buffer[bytes_received] = '\0'; // Null-terminate
        std::cout << "Server says: " << buffer << "\n";
    }

    close(sock);
    return 0;
}
```