

# Networking: Sockets (continued); Addressing

---

COSC 208, Introduction to Computer Systems, 2021-12-06

## Announcements

- Project 3 due tomorrow
- Attend research talks tomorrow

## Outline

- Warm-up
- Server program using sockets

## Warm-up

Q1: *What is wrong with this client application?*

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>

#define HTTP_PORT 80
#define WWW_EXAMPLE_COM 0x5DB8D822 // 93.184.216.34

int main() {
    struct sockaddr_in server_addr;
    server_addr.sin_family = AF_INET;
    server_addr.sin_port = HTTP_PORT;
    server_addr.sin_addr.s_addr = WWW_EXAMPLE_COM;
    int sock = connect((const struct sockaddr *) &server_addr, sizeof(server_addr));

    char buf[1024] = "GET / HTTP/1.1\r\nHost: www.example.com\r\n\r\n";
    send(sock, buf, strlen(buf), 0);

    int len = recv(sock, buf, 1024, 0);
    buf[len] = '\0';
    printf("%s\n", buf);
}
```

## Server application

```
1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/socket.h>
4  #include <netinet/in.h>
5  #include <string.h>
6  #include <unistd.h>
7
8  int main() {
9      int server_sock = socket(AF_INET, SOCK_STREAM, 0);
10     if (server_sock < 0) {
11         perror("socket failed");
12         return 1;
13     }
14
15     struct sockaddr_in server_addr;
16     server_addr.sin_family = AF_INET;
17     server_addr.sin_port = htons(13346);
18     server_addr.sin_addr.s_addr = htonl(INADDR_ANY);
19
20     int result = bind(server_sock,
21                       &server_addr, sizeof(server_addr));
22     if (result < 0) {
23         perror("bind failed");
24         return 1;
25     }
26
27     result = listen(server_sock, 0);
28     if (result < 0) {
29         perror("listen failed");
30         return 1;
31     }
32
33     struct sockaddr_in client_addr;
34     unsigned int client_addr_len = sizeof(client_addr);
35     int client_sock = accept(server_sock,
36                              &client_addr, &client_addr_len);
37
38     char buf[2048];
39     int len = recv(client_sock, buf, 2048, 0);
40     if (len < 0) {
41         perror("recv failed");
42         return 1;
43     }
44     buf[len] = '\0';
45     printf("%s\n", buf);
46
47     strcpy(buf, "Acknowledged\n");
48     len = send(client_sock, buf, strlen(buf), 0);
49     if (len < 0) {
50         perror("send failed");
51         return 1;
52     }
53
54     close(client_sock);
55     close(server_sock);
56 }
```