Unix Domain Protocols

A way of performing the client server communication on same host using same API that is used for communication of clients and servers on different hosts. (similar to IPC)

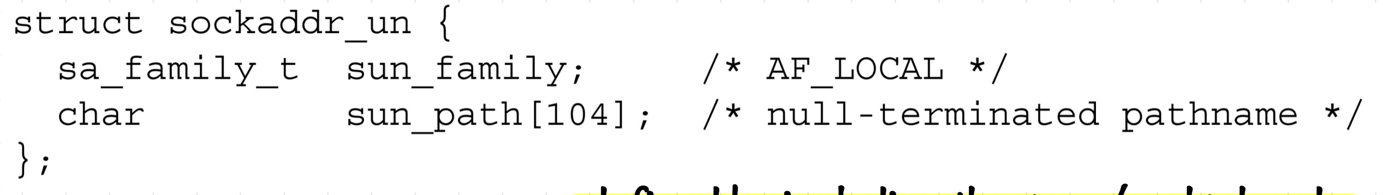
Why use this?

1. Unix domain sockets are often twice as fast as a TCP socket when both peers are on the same host.

2. Unix domain sockets are used when passing descriptors between processes on the same host.

3. Unix domain sockets provide the client's credentials, which provide additional security checking options.

Unix Domain Socket Address Structure

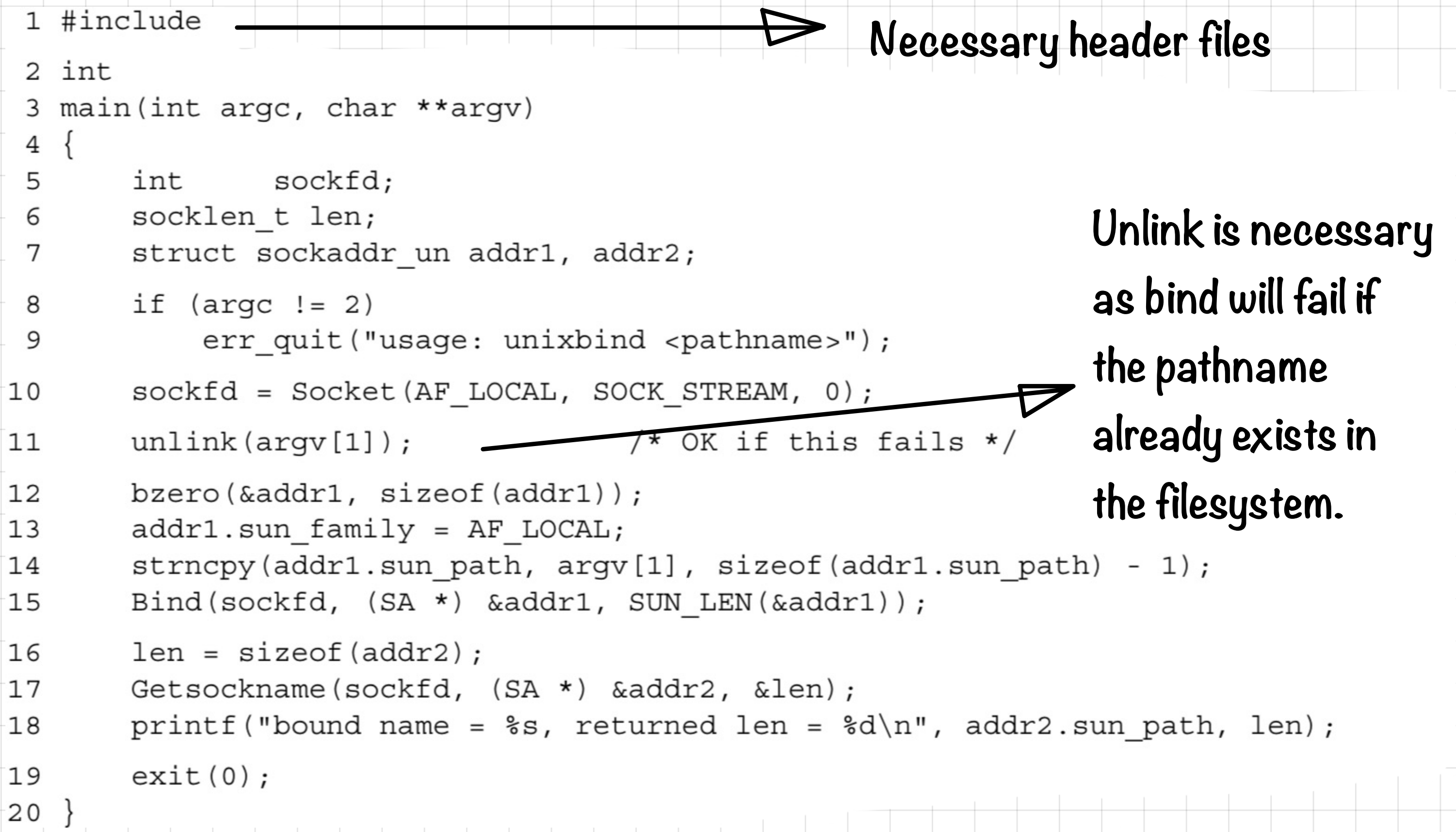


```
struct sockaddr_un {
  sa_family_t  sun_family;     /* AF_LOCAL */
  char         sun_path[104];  /* null-terminated pathname */
};
```

defined by including the <sys/un.h> header.

The pathname stored in the sun_path array must be null-terminated.

bind of Unix Domain Socket



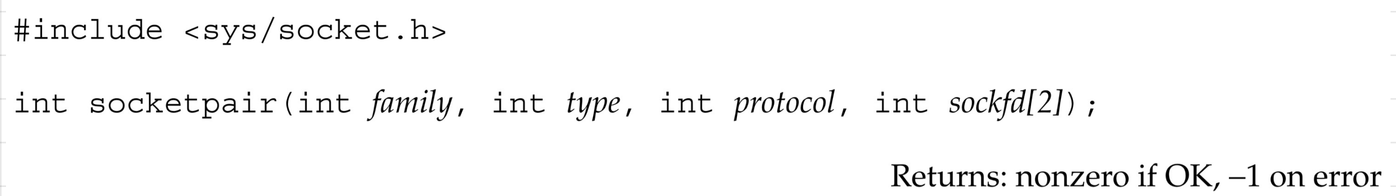
```
#include

int
main(int argc, char **argv)
{
    int     sockfd;
    socklen_t len;
    struct sockaddr_un addr1, addr2;

    if (argc != 2)
        err_quit("usage: unixbind <pathname>");

    sockfd = Socket(AF_LOCAL, SOCK_STREAM, 0);

    unlink(argv[1]);            /* OK if this fails */

    bzero(&addr1, sizeof(addr1));
    addr1.sun_family = AF_LOCAL;
    strncpy(addr1.sun_path, argv[1], sizeof(addr1.sun_path) - 1);
    Bind(sockfd, (SA *) &addr1, SUN_LEN(&addr1));

    len = sizeof(addr2);
    Getsockname(sockfd, (SA *) &addr2, &len);
    printf("bound name = %s, returned len = %d\n", addr2.sun_path, len);

    exit(0);
}
```

Necessary header files

Unlink is necessary as bind will fail if the pathname already exists in the filesystem.

socketpair Function

The socketpair function creates two sockets that are then connected together. This function applies only to Unix domain sockets.



```
#include <sys/socket.h>

int socketpair(int family, int type, int protocol, int sockfd[2]);
```

Returns: nonzero if OK, –1 on error

The family must be AF_LOCAL and the protocol must be 0. The type, however, can be either SOCK_STREAM or SOCK_DGRAM. The two socket descriptors that are created are returned as sockfd[0] and sockfd[1].

This function is similar to the Unix pipe function: Two descriptors are returned, and each descriptor is connected to the other.

What is Unix (domain) Socket?

- Socket: a bidirectional communication end-point
- Berkeley socket: sockets used for data communication over Internet
- Unix socket: sockets used for data communication between processes on the same Unix system.
- It supports transmission of a reliable stream of bytes (SOCK_STREAM) as well as unreliable transmission of datagrams (SOCK_DGRAM)