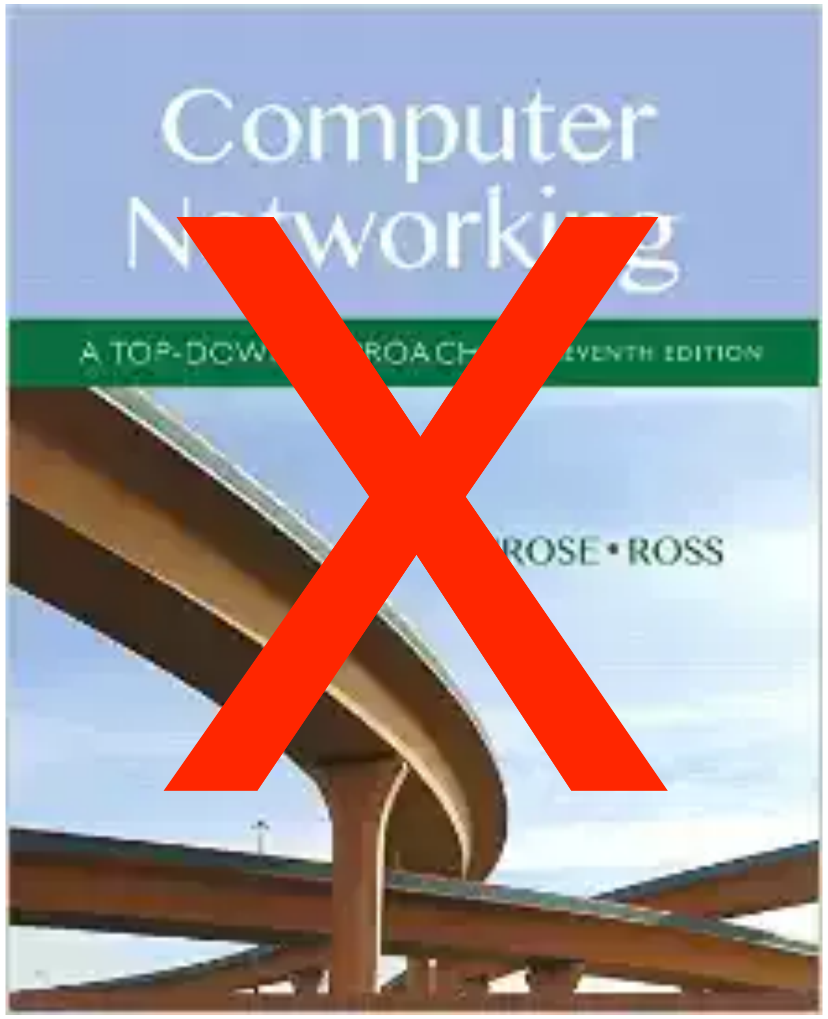
Application Layer, Part 2
Sockets

Mark Allman
Case / ICSI

EECS 325/425
Fall 2018

*"If you walk away, walk away,
I will walk away, walk away …"*

Reading Along ...

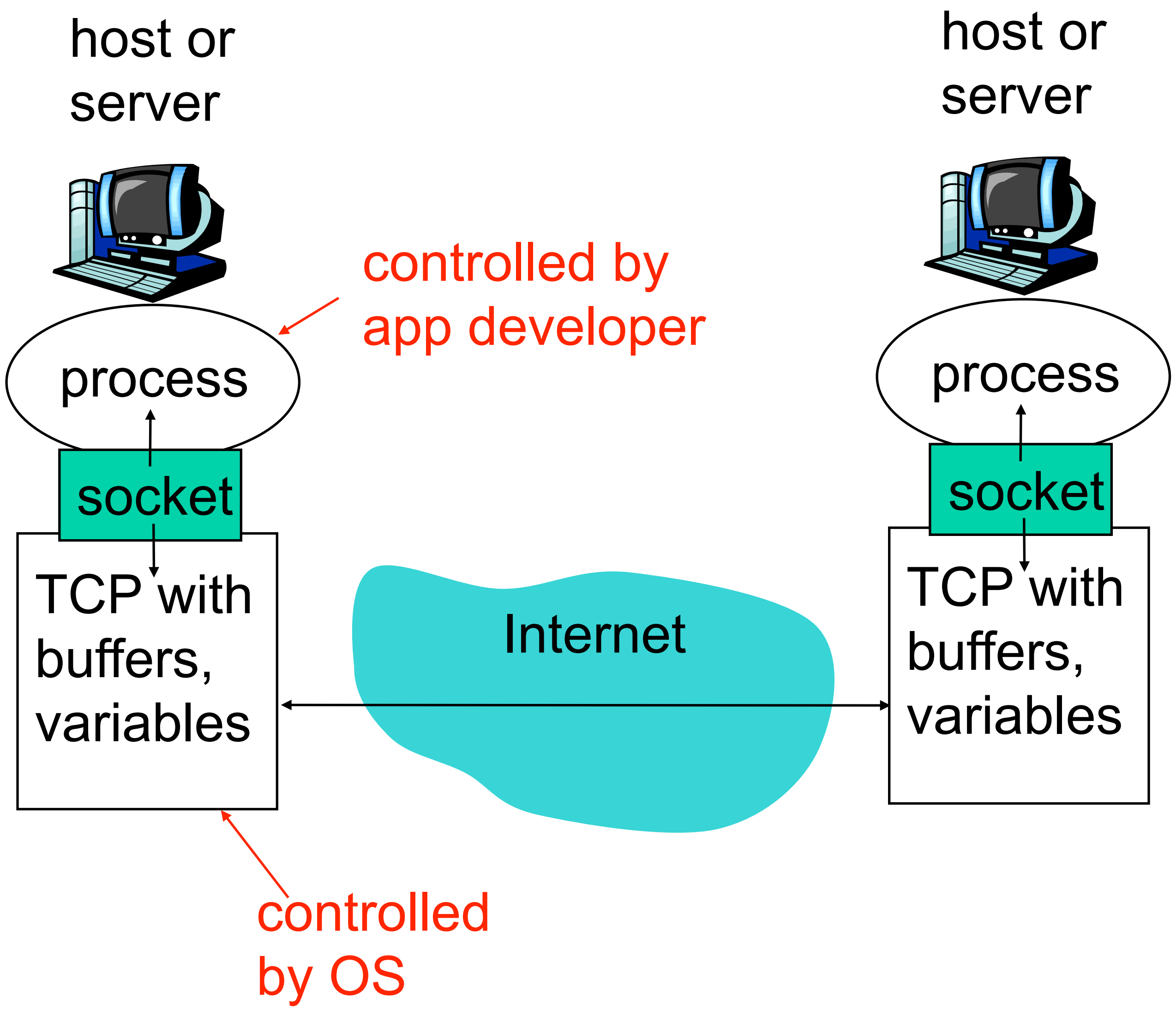


- Sockets programming (client side)

Sockets

Recall that sockets sit between the application process and the transport protocol

Sockets form the glue that allows processes to *interface* with transports (and hence all lower layers)


host or server
host or server
controlled by app developer
process
process
socket
socket
TCP with buffers, variables
TCP with buffers, variables
Internet
controlled by OS

Example TCP Client

```
read (sd,buffer,BUFLEN - 1);
printf (stdout,"%s\n",buffer);
```

Example TCP Client

```
read (sd,buffer,BUFLEN - 1);
printf (stdout,"%s\n",buffer);
```

Example TCP Client

```
sd = socket (PF_INET, SOCK_STREAM, protoinfo->p_proto);
```

```
read (sd,buffer,BUFLEN - 1);
printf (stdout,"%s\n",buffer);
```

Example TCP Client

```
sd = socket (PF_INET, SOCK_STREAM, protoinfo->p_proto);
```

```
read (sd,buffer,BUFLEN - 1);
printf (stdout,"%s\n",buffer);
```

Example TCP Client

```
protoinfo = getprotobyname (“tcp”);

sd = socket (PF_INET, SOCK_STREAM, protoinfo->p_proto);
```

```
read (sd,buffer,BUFLEN - 1);
printf (stdout,"%s\n",buffer);
```

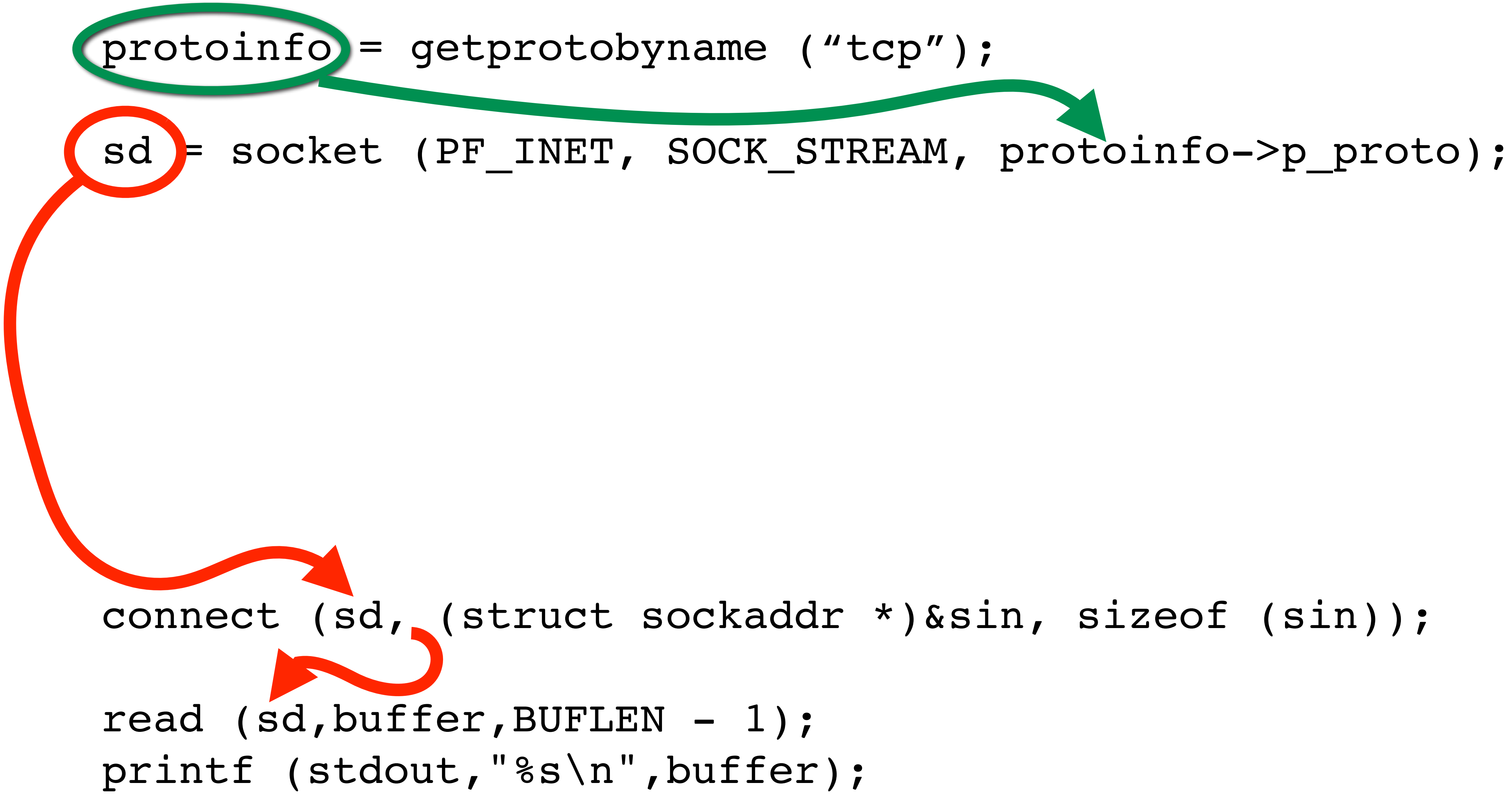
Example TCP Client

```
protoinfo = getprotobyname (“tcp”);

sd = socket (PF_INET, SOCK_STREAM, protoinfo->p_proto);









read (sd,buffer,BUFLEN - 1);
printf (stdout,"%s\n",buffer);
```

Example TCP Client

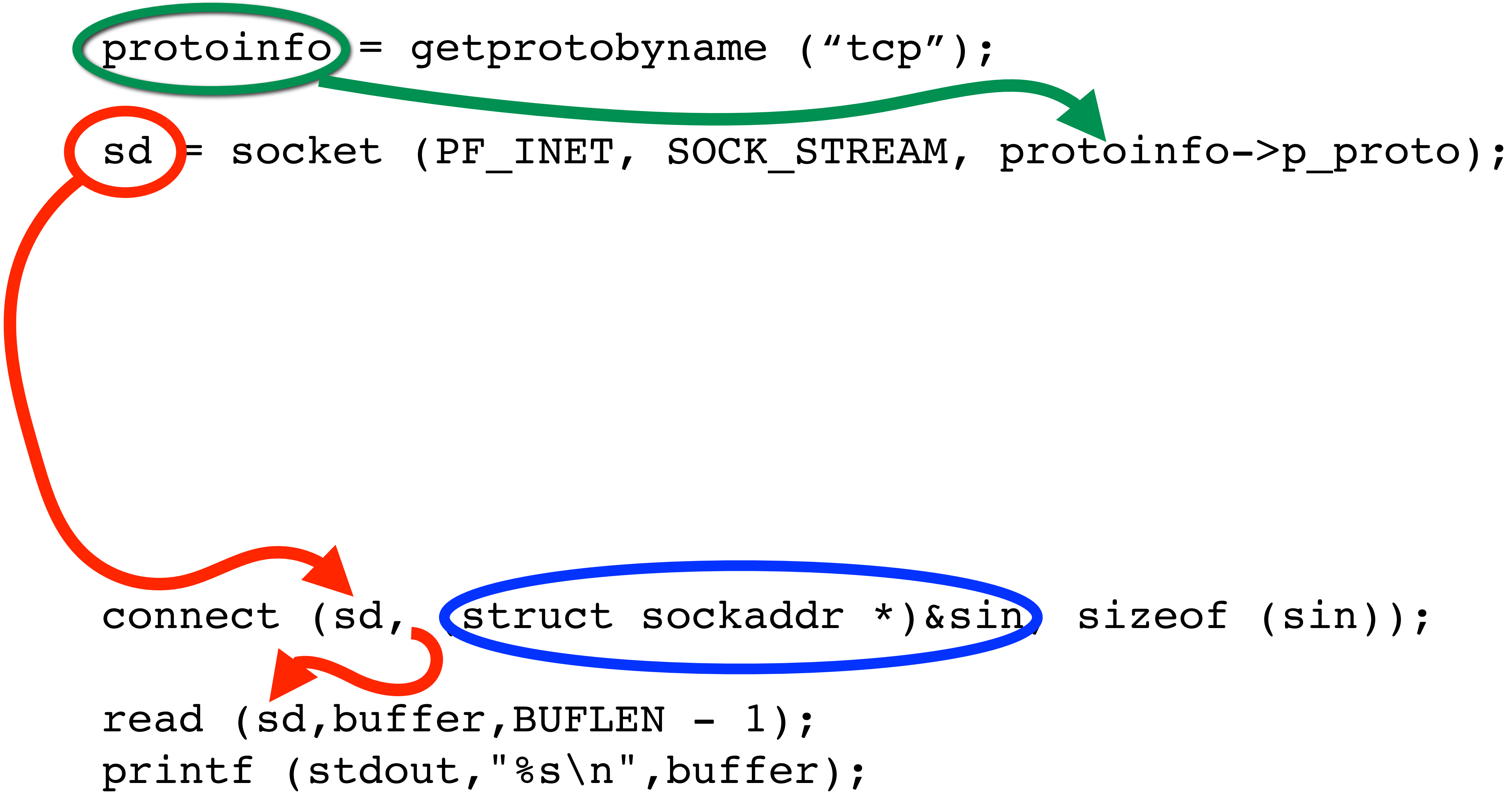
```
protoinfo = getprotobyname (“tcp”);

sd = socket (PF_INET, SOCK_STREAM, protoinfo->p_proto);








connect (sd, (struct sockaddr *)&sin, sizeof (sin));

read (sd,buffer,BUFLEN - 1);
printf (stdout,"%s\n",buffer);
```

Example TCP Client



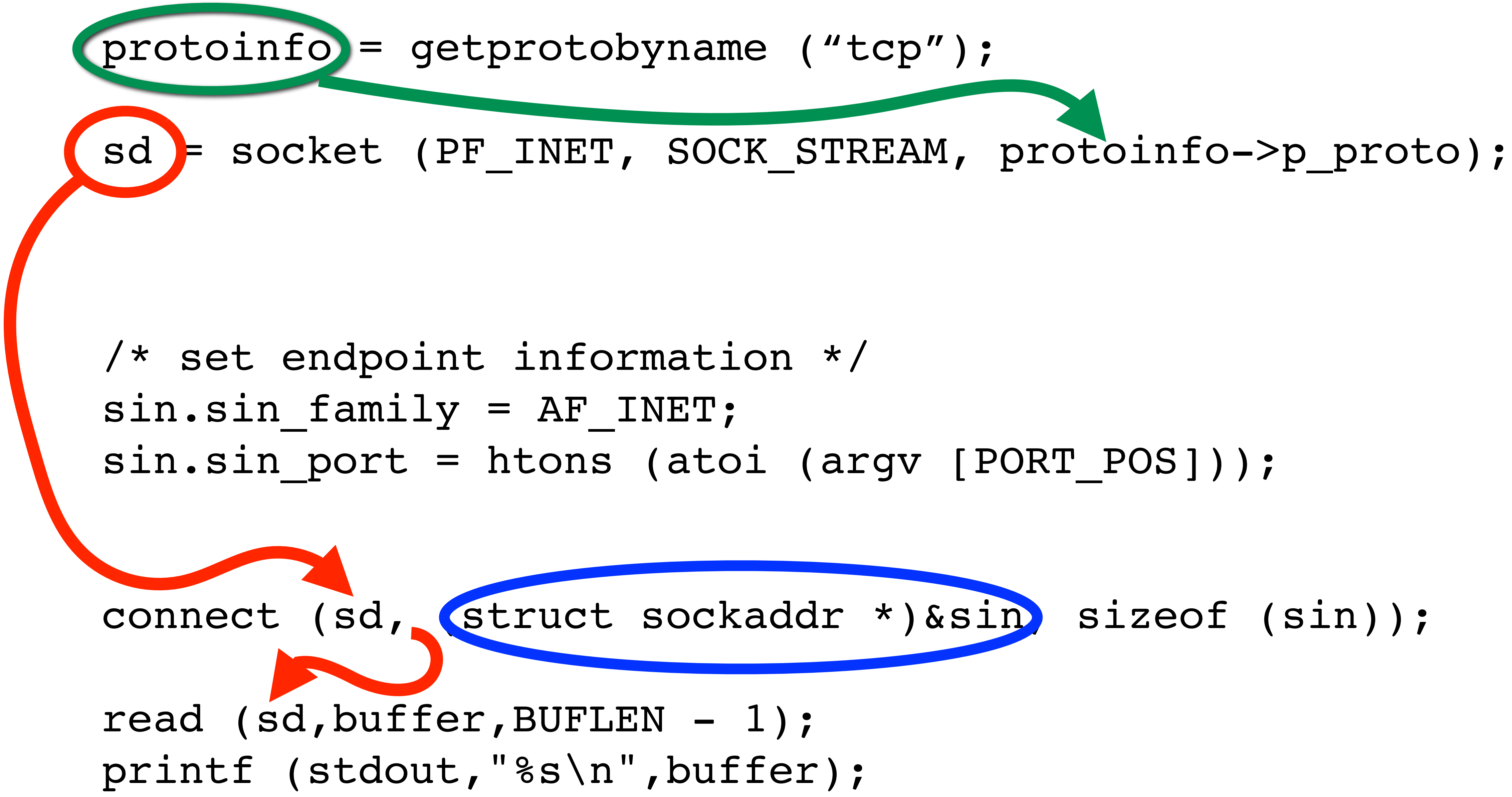
```
protoinfo = getprotobyname (“tcp”);
sd = socket (PF_INET, SOCK_STREAM, protoinfo->p_proto);



connect (sd, (struct sockaddr *)&sin, sizeof (sin));

read (sd,buffer,BUFLEN - 1);
printf (stdout,"%s\n",buffer);
```

Example TCP Client



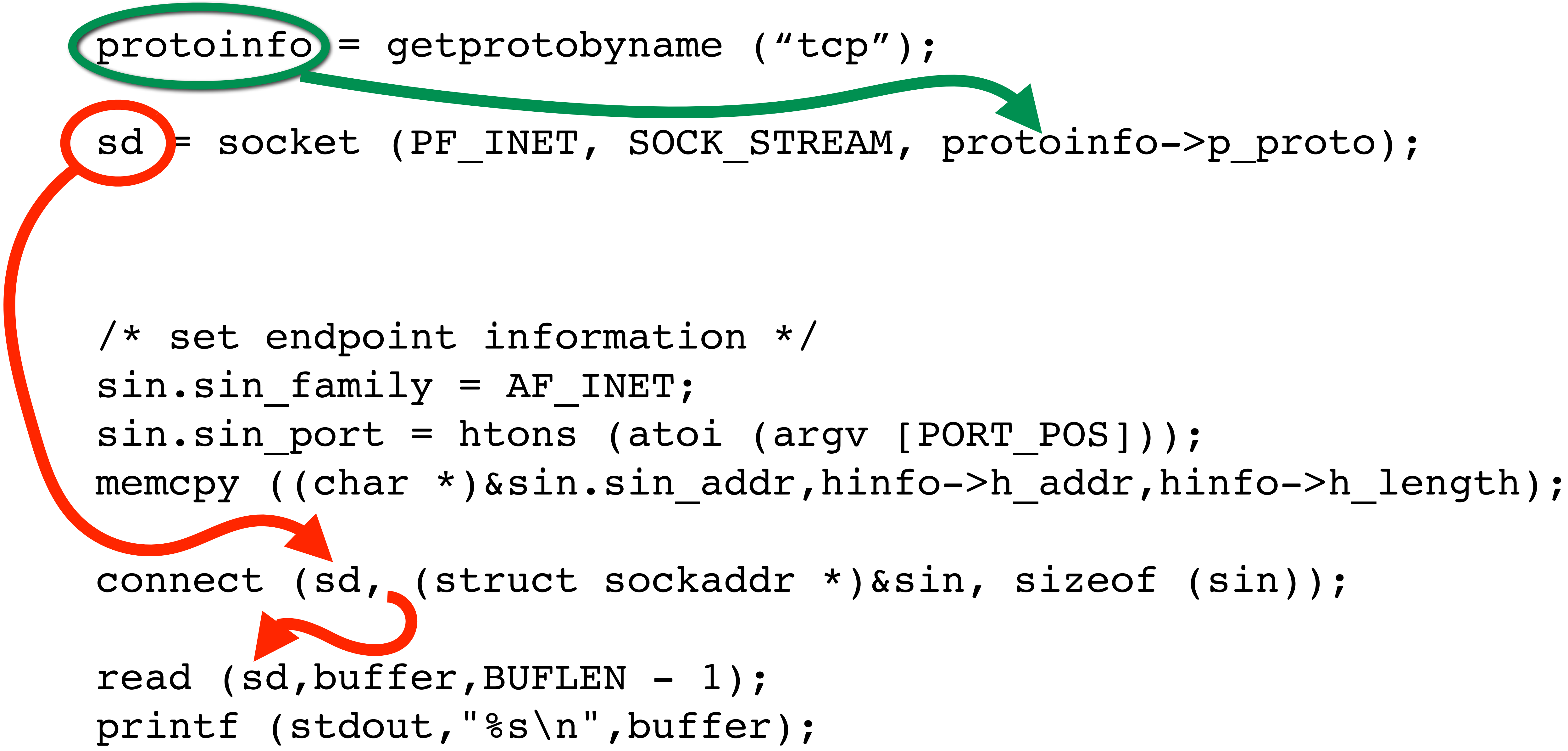
```
protoinfo = getprotobyname (“tcp”);
sd = socket (PF_INET, SOCK_STREAM, protoinfo->p_proto);



connect (sd, (struct sockaddr *)&sin, sizeof (sin));

read (sd,buffer,BUFLEN - 1);
printf (stdout,"%s\n",buffer);
```

Example TCP Client



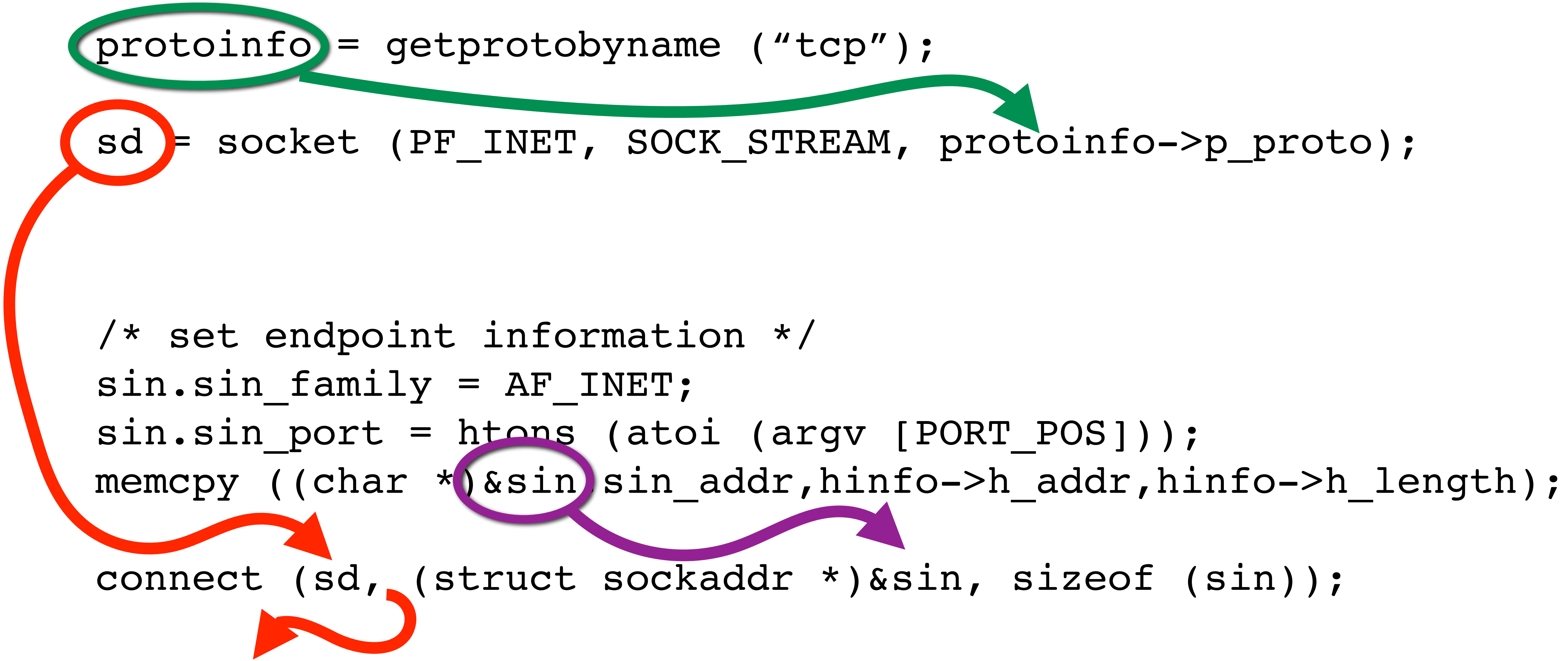
```
protoinfo = getprotobyname (“tcp”);
sd = socket (PF_INET, SOCK_STREAM, protoinfo->p_proto);


/* set endpoint information */
sin.sin_family = AF_INET;
sin.sin_port = htons (atoi (argv [PORT_POS]));


connect (sd, (struct sockaddr *)&sin, sizeof (sin));

read (sd,buffer,BUFLEN - 1);
printf (stdout,"%s\n",buffer);
```

Example TCP Client



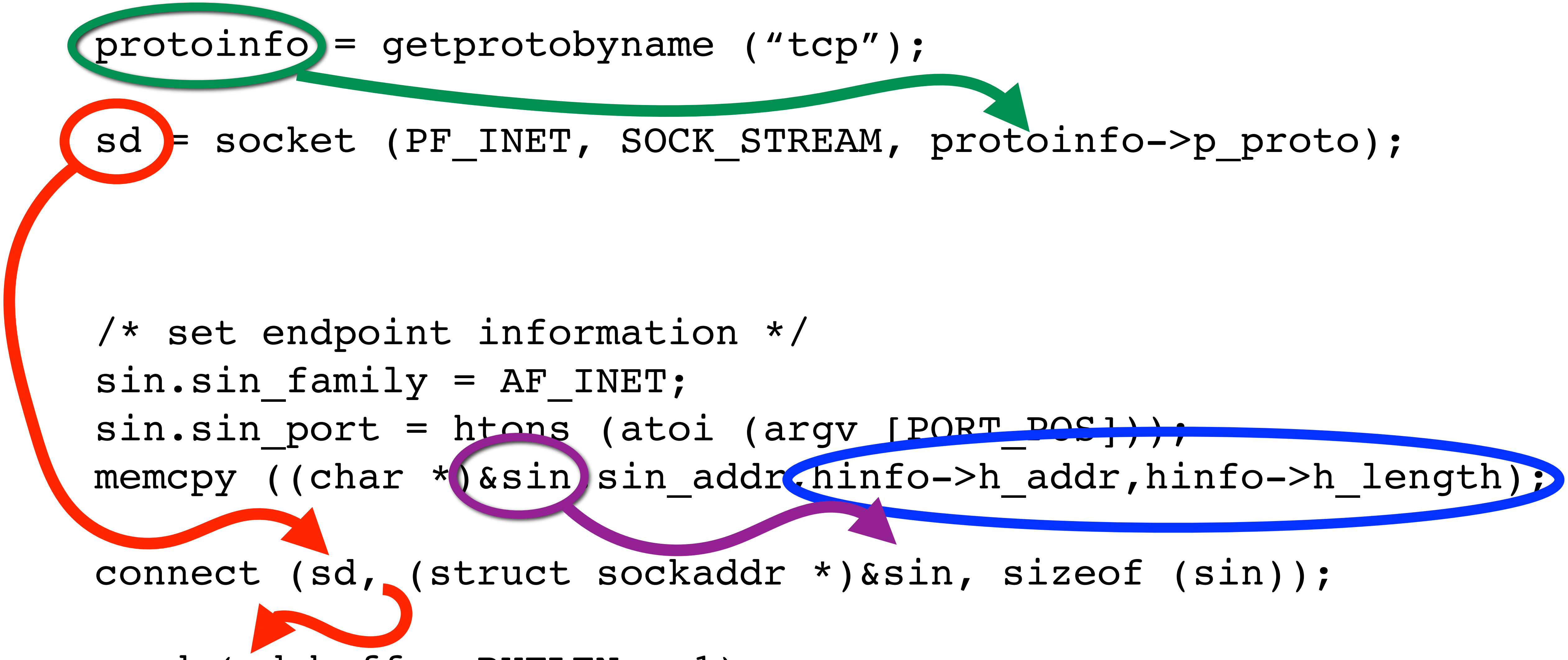
```
protoinfo = getprotobyname (“tcp”);

sd = socket (PF_INET, SOCK_STREAM, protoinfo->p_proto);



/* set endpoint information */
sin.sin_family = AF_INET;
sin.sin_port = htons (atoi (argv [PORT_POS]));
memcpy ((char *)&sin.sin_addr,hinfo->h_addr,hinfo->h_length);

connect (sd, (struct sockaddr *)&sin, sizeof (sin));

read (sd,buffer,BUFLEN - 1);
printf (stdout,"%s\n",buffer);
```

Example TCP Client



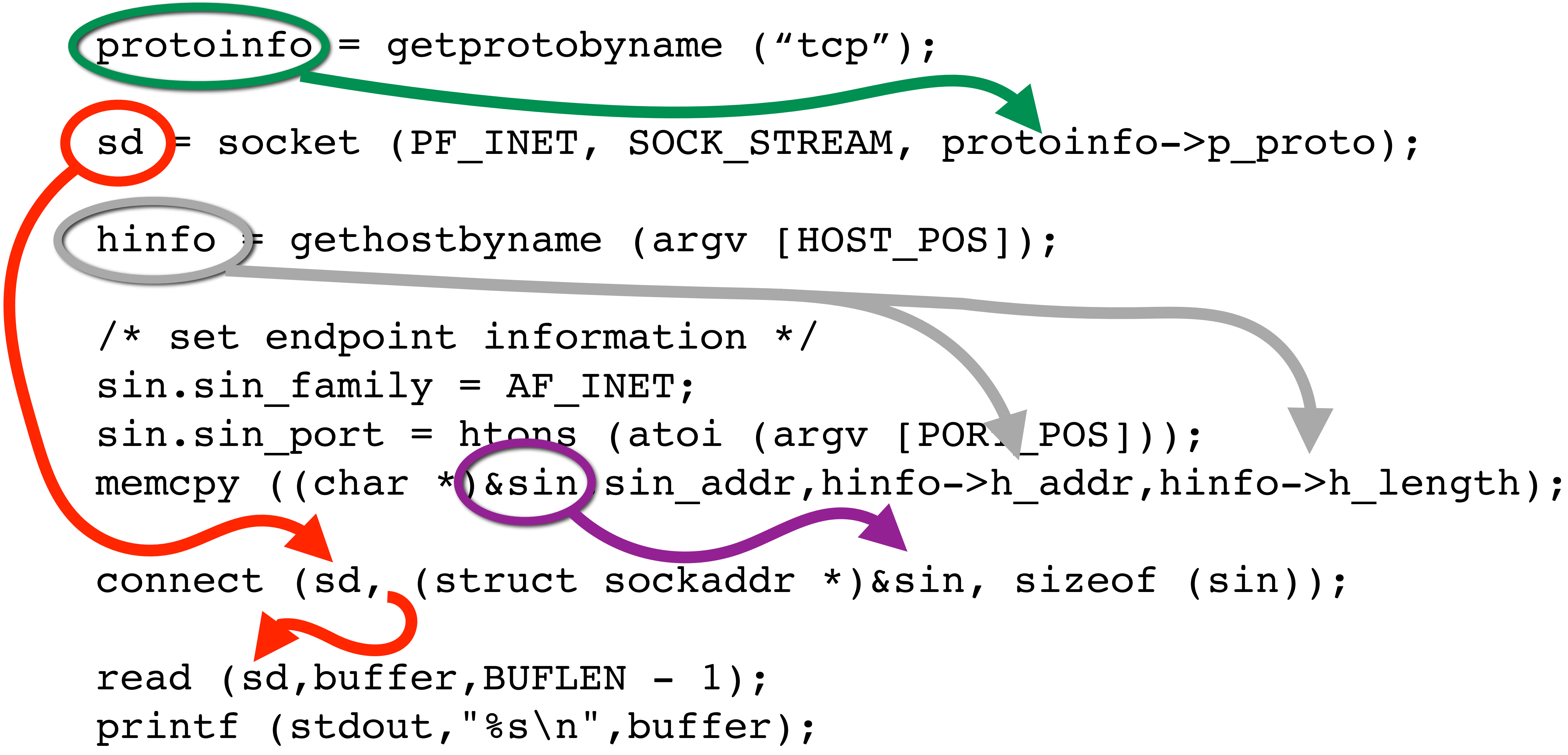
```
protoinfo = getprotobyname (“tcp”);
sd = socket (PF_INET, SOCK_STREAM, protoinfo->p_proto);


/* set endpoint information */
sin.sin_family = AF_INET;
sin.sin_port = htons (atoi (argv [PORT_POS]));
memcpy ((char *)&sin.sin_addr,hinfo->h_addr,hinfo->h_length);

connect (sd, (struct sockaddr *)&sin, sizeof (sin));

read (sd,buffer,BUFLEN - 1);
printf (stdout,"%s\n",buffer);
```

Example TCP Client



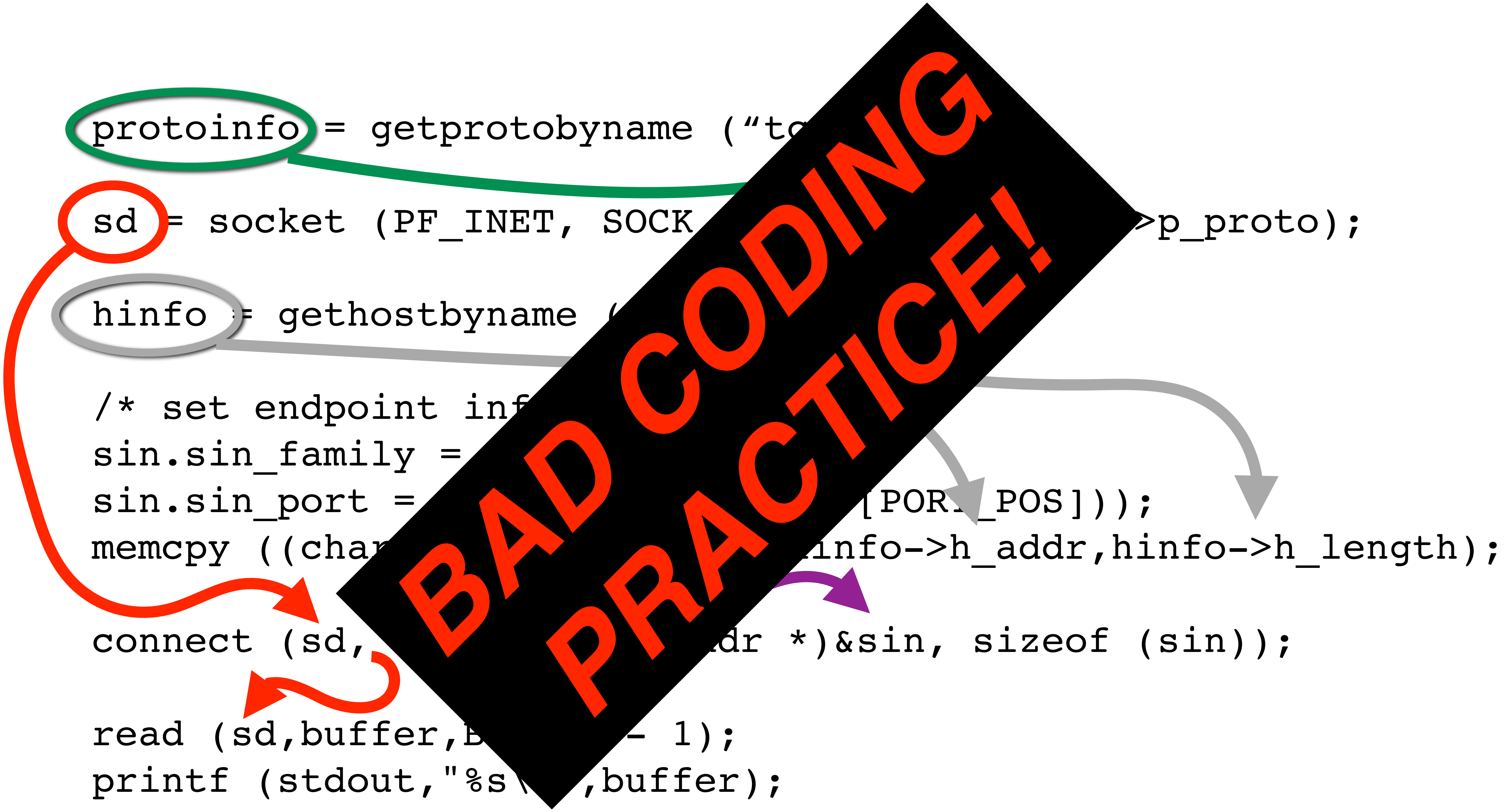
```
protoinfo = getprotobyname (“tcp”);
sd = socket (PF_INET, SOCK_STREAM, protoinfo->p_proto);


/* set endpoint information */
sin.sin_family = AF_INET;
sin.sin_port = htons (atoi (argv [PORT_POS]));
memcpy ((char *)&sin.sin_addr,hinfo->h_addr,hinfo->h_length);

connect (sd, (struct sockaddr *)&sin, sizeof (sin));

read (sd,buffer,BUFLEN - 1);
printf (stdout,"%s\n",buffer);
```

Example TCP Client



```
protoinfo = getprotobyname ("tcp");

sd = socket (PF_INET, SOCK_STREAM, protoinfo->p_proto);

hinfo = gethostbyname (argv [HOST_POS]);

/* set endpoint information */
sin.sin_family = AF_INET;
sin.sin_port = htons (atoi (argv [PORT_POS]));
memcpy ((char *)&sin.sin_addr,hinfo->h_addr,hinfo->h_length);

connect (sd, (struct sockaddr *)&sin, sizeof (sin));

read (sd,buffer,BUFLEN - 1);
printf (stdout,"%s\n",buffer);
```

Example TCP Client



```
protoinfo = getprotobyname ("tc

sd = socket (PF_INET, SOCK                    >p_proto);

hinfo = gethostbyname (

/* set endpoint inf
sin.sin_family =
sin.sin_port =                   [PORT_POS]));
memcpy ((char            hinfo->h_addr,hinfo->h_length);

connect (sd,             dr *)&sin, sizeof (sin));

read (sd,buffer,B      - 1);
printf (stdout,"%s\    ,buffer);
```

BAD CODING PRACTICE!

Example TCP Client

```
protoinfo = getprotobyname (“tcp”);

sd = socket (PF_INET, SOCK_STREAM, protoinfo->p_proto);

hinfo = gethostbyname (argv [HOST_POS]);

/* set endpoint information */
sin.sin_family = AF_INET;
sin.sin_port = htons (atoi (argv [PORT_POS]));
memcpy ((char *)&sin.sin_addr,hinfo->h_addr,hinfo->h_length);

connect (sd, (struct sockaddr *)&sin, sizeof (sin));

read (sd,buffer,BUFLEN - 1);
printf (stdout,"%s\n",buffer);
```

Example TCP Client

```
#define BUFLEN 1024

int sd;
char buffer [BUFLEN];


[...]

      read (sd,buffer,BUFLEN - 1);




fprintf (stdout,"%s\n",buffer);
```

Example TCP Client

```
#define BUFLEN 1024

int sd;
char buffer [BUFLEN];
int ret;

[...]

ret = read (sd,buffer,BUFLEN - 1);




fprintf (stdout,"%s\n",buffer);
```

Example TCP Client

```
#define BUFLEN 1024

int sd;
char buffer [BUFLEN];
int ret;

[...]

ret = read (sd,buffer,BUFLEN - 1);
if (ret < 0)
{
    fprintf (stderr,"reading error”);
    exit (1)
}
fprintf (stdout,"%s\n",buffer);
```

Example TCP Client

```
#define BUFLEN 1024

int sd;
char buffer [BUFLEN];
int ret;

[...]

ret = read (sd,buffer,BUFLEN - 1);
if (ret < 0)
{
    fprintf (stderr,"reading error”);
    exit (1)
}
fprintf (stdout,"%s\n",buffer);
```

Define constants

Example TCP Client

```
#define BUFLEN 1024

int sd;
char buffer [BUFLEN];
int ret;

[...]

ret = read (sd,buffer,BUFLEN - 1);
if (ret < 0)
{
    fprintf (stderr,"reading error”);
    exit (1)
}
fprintf (stdout,"%s\n",buffer);
```

Define constants

Check return values

Example TCP Client

```
sd = socket (PF_INET, SOCK_STREAM, protoinfo->p_proto);
if (sd < 0)
    errexit("cannot create socket",NULL);

/* connect the socket */
if (connect (sd, (struct sockaddr *)&sin, sizeof(sin)) < 0)
    errexit ("cannot connect", NULL);
```

Example TCP Client

- Full code for example client on the class web page

```
% ./sockets eecslab-6.case.edu 22
SSH-2.0-OpenSSH_6.2

% ./sockets envoy.icir.org 22
SSH-2.0-OpenSSH_7.7

% ./sockets mailhost.icsi.berkeley.edu 25
220 fruitcake.ICSI.Berkeley.EDU ESMTP Sendmail 8.12.11.2006[…]
```