Post-lab Questions of Lab 7

CMPE 324

April 15, 2015

1 Question 1

Consider the scenario where you are about to implement a TCP server (e.g. a HTTP server). From the perspective of socket programming, your code should call the following functions (sorted randomly):

- `accept`.
- `read` and `write`.
- `socket`.
- `close`.
- `bind`.
- `listen`.

Your task in this question is to sort the functions above in the *right order*, that allows the program to (ultimately) read/write from/to connected peers.

2 Question 2

From the POSIX functions above, list those that block[1].

[1]Sockets can be in blocking, or nonblocking modes. In nonblocking mode, none of the functions above block. In blocking mode, some of them block. In lab 7, we worked on blocking sockets only, and this question is specific to blocking sockets.

A Function synopsis

This appenedix lists relevant function synopsis for your own reference. The order by which the functions are listed here is also random (intentional).

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

int conn_sockfd = accept(int sockfd, struct sockaddr *addr2,
    socklen_t *addrlen2);
```

Figure 1: The synopsis of the `accept` function.

```
include <unistd.h>

ssize_t read(int conn_sockfd, void *buf, size_t count);
```

Figure 2: The synopsis of the `read` function.

```
#include <unistd.h>

ssize_t write(int conn_sockfd, const void *buf, size_t count);
```

Figure 3: The synopsis of the `write` function.

```c
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/tcp.h>

int sockfd = socket(AF_INET, SOCK_STREAM, 0);
```

Figure 4: The synopsis of the `socket` function.

```c
#include <unistd.h>

/* fd can be any socket, sockfd or conn_sockfd */
int close(int fd);
```

Figure 5: The synopsis of the `close` function.

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

int bind(int sockfd, const struct sockaddr *addr,
         socklen_t addrlen);
```

Figure 6: The synopsis of the `bind` function.

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

int listen(int sockfd, int backlog);
```

Figure 7: The synopsis of the `listen` function.