Homework 9

Problem 1

Assume we want to write a tick program which prints a "BEEP" in console every second. If there is any client connected, the BEEP will be sent to client instead of being printed on server. When client closes the connection, "BEEP" should be printed in console again. Here is parts of the program:

```
1. #include <sys/types.h>
2. #include <sys/socket.h>
3. #include <errno.h>
4. #include <fcntl.h>
5. #include <signal.h>
6. #include <unistd.h>
7. #include <stdlib.h>
8. #include "csapp.h"
9.
10. void handler (int sig) {
11.     Write(1, "BEEP\n", 5);
12.     Alarm(1);
13. }
14.
15. // This function will block and return after the client
16. // close the connection
17. void wait_disconnect (int fd) {
18.     char c;
19.     while (read(fd, &c, 1) > 0 || errno == EINTR) ;
20.}
21.
22.int main(void) {
23.     int listenfd, connfd;
24.     Close(2); // You are not allowed to use stderr
25.     Signal(SIGALRM, handler);
```

```
26.     Alarm(1);
27.     listenfd = Open_listenfd(1234);
28.     while (1) {
29.         connfd = Accept(listenfd, NULL, NULL);
30.         int stdout_backup = dup(1);
31.         Dup2(connfd, 1);
32.         wait_disconnect(connfd);
33.         Close(connfd);
34.         Dup2(stdout_backup, 1);
35.         Close(stdout_backup);
36.     }
37.     exit(0);
38.}
```

1. Complete the program.
2. Please write a command to test your program, without writing any client program.
    1. In terminal-1, compile and run the server program. You should see "BEEP" printed every second.
    2. In terminal-2, type "telnet localhost 1234". You should observe terminal-1 stops printing "BEEP", while terminal-2 starts printing "BEEP" every second.
    3. In terminal-2, quit telnet. Terminal-2 stops printing "BEEP", and terminal-1 starts printing "BEEP" again.

Problem 2

Extend TINY to support the HTTP HEAD method and check the correctness.

In doit() function, allow method string to be "HEAD" (previously only "GET" is allowed by `if(strcasecmp(method, "GET"))`).

In "serving static content" case, call `serve_head()` function if method is "HEAD", while `serve_head()` is almost the same as `serve_static()`, except that `serve_head()` omit the "Send response body to client" part.

Check this extension by telnet.

Problem 3

11.3.2 (textbook) shows examples of **multiple domain names mapped to the same IP address**, and also **a domain name mapped to multiple IP addresses**.

1. Think about the benefits of mapping a domain names to multiple IP addresses.

For example, to obtain good availability of your server, you might want multiple servers, each with a different IP address, to serve the same web page to your clients.

2. In shared web hosting case, many websites can reside on one web server (sharing a single IP address) connected to the Internet. How does this server decide which website to show the user?

HTTP request includes the host name (refer to telnet), the server can use this information to decide which domain to serve.