

Learning Objectives

- ☐ To implement programs with system calls.
- ☐ To create clients and servers in C.

Exercises

You should attempt the exercises below by using only the C constructs that you learnt up to teaching week 8, and:

1. Write pseudo code to describe the required algorithm to solve the exercise (or draw up a flowchart), before writing and testing the actual code.
2. Add comments to your code.
3. Make your code neat, by using indentation and parenthesis (where appropriate).
4. Give meaningful names to functions and variables.

Exercise 1

The program in Figure 1 must perform a system call to be complete. This incomplete program writes a timestamp in the end of a log file each time the log file receives a new input. In order to understand the code of this program, search for information about the *time* library in C, especially for the functions `time()`, `asctime()` and the type `time_t`. Add the missing line and test your program three times to see the results in the log file. Also in few words:

1. Explain the contents of the string “cmd” in terms of syntax and semantics when the system call is called.
 2. Explain the function `now()` based on your search on the `time` library in C.
- Hint: Use the system call you have seen in class and have a look at Chapter 9 of Head First C.

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

// This function returns a string containing the current date and time
char* now()
{
    time_t t;
    time (&t);
    return asctime(localtime(&t));
}

// Master Control Program utility. Records guard patrol check-ins.
int main()
{
    char comment[80];
    char cmd[120];
    fgets(comment,80,stdin);
    sprintf(cmd,"echo '%s %s'>>reports.log",comment,now());
    .....;
    return 0;
}

```

Figure 1

Exercise 2

Write a TCP echo server similar to the example given in class, but the server must listen to more than one client through repetition. Test the new server with more than one client. Hint: Use the client code of the lecture example for testing your server.

Exercise 3

Write a TCP client and server that the client can ask the server to return its local time by sending the command `"/Time"`. You will need to create a simple communication protocol for your client and server based on the command above. You also need to use a suitable function from the `time` library in C to complete this exercise.

Exercise 4

Answer the following questions:

- What should be used instead of `inet_ntoa()`?
- What is the major problem with the current server codes? Think about a client that takes a long time to finish its communication with the server. What is the problem?
- Only conceptually (do not worry about coding or how you are going to code your answer), how would you change the code of the server to solve the problem of item b).