# **SYST 44288 – Operating Systems & Systems Programming**

# **Assignment 2 – Processes and Errors**

## Due Date: See SLATE calendar.

For this assignment, work in groups of two. Your group **MUST** use version control software (ex: git, mercurial, subversion, etc) to manage your code. You may use a public repository like GitLabs, GitHub, BitBucket, etc to host your code. Here is a great tutorial from GitHub on how to use git (<https://try.github.io/levels/1/challenges/1>).

**Question 1.**

An echo server echoes back whatever it receives from a client. For example, if a client sends the server the string Hello there!, the server will respond with Hello there!

Write an echo server using the Java networking API described in Section 3.6.1. This server will wait for a client connection using the accept() method. When a client connection is received, the server will loop, performing the following steps:

• Read data from the socket into a buffer.

• Write the contents of the buffer back to the client.

The server will break out of the loop only when it has determined that the client has closed the connection.

The date server shown in Figure 3.21 uses the java.io.BufferedReader class. BufferedReader extends the java.io.Reader class, which is used for reading character streams. However, the echo server cannot guarantee that it will read characters from clients; it may receive binary data as well. The class java.io.InputStream deals with data at the byte level rather than the character level. Thus, your echo server must use an object that extends java.io.InputStream. The read() method in the java.io.InputStream class returns −1 when the client has closed its end of the socket connection.

Apply appropriate error handling, see the General Notes Section below.

Input sanitizing - While it is always recommended to sanitize input, the fact that this program will compile and execute any code it is sent over the Internet basically makes it **impossible to secure** properly. Just do basic sanitizing and don’t worry too much.

**Question 2.**

Design a file-copying program, in the C programming language, named filecopy using ordinary pipes. This program will be passed two parameters: the name of the file to be copied and the name of the copied file. The program will then create an ordinary pipe and write the contents of the file to be copied to the pipe. The child process will read this file from the pipe and write it to the destination file. For example, if we invoke the program as follows:

filecopy input.txt copy.txt

the file input.txt will be written to the pipe. The child process will read the contents of this file and write it to the destination file copy.txt.

Write this program using UNIX pipes.

Apply appropriate error handling, see the General Notes Section below.

**General Notes:**

Error handling - Handle errors using appropriate error handling techniques.

* A bad or missing command line argument should cause the program to exit after it prints usage instructions in the same way many Linux command line utilities do. Ex: “usage: myprog arg1 arg2”
* Regular errors should be printed to stderr by means of fprintf(). The choice of when to terminate the program on specific errors is left to you.
* Failed system calls should use the perror() function for reporting error messages.

## Grading Rubric

* 5 points for the successful implementation for each question (total 10).
* 1 point for input sanitizing and error handling for each question (total 2).
* 1 point for appropriate commenting for each question (total 2).

Total: 14 points.

## Deliverables

1. The programs should be commented where necessary.
2. Submit only source code and other necessary documentation.
3. Please name your files appropriately to make it clear which question you are answering.
4. Include the names of your group in a comment at the top of each program file.
5. Your submission should be in a single zip file in the following format, where lastname1 is the lastname of student1 and lastname2 is the lastname of student2:

lastname1-lastname2-a2.zip

1. Upload your zip file to the SLATE dropbox.