

## Goal:

*Write a C program that copies files by using both system and function calls. The source and destination files are provided by the user. Another option is to repeatedly copy a file to a destination file in order to gauge how quickly the function performs. In this case, the user specifies the maximum file size and step size.*

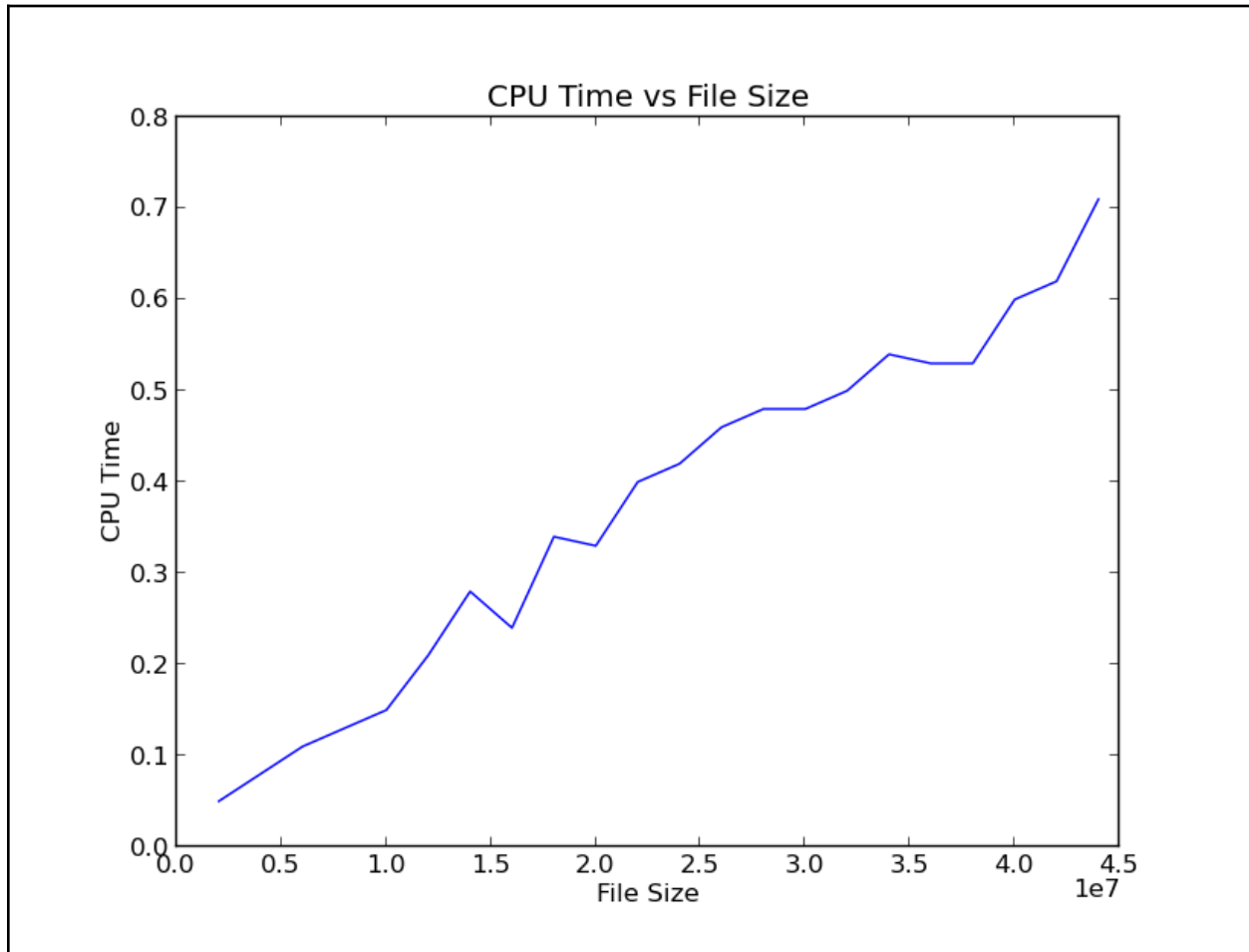
## Procedures:

- The program allows the user to copy a file from one location to another
- It has the option to measure the performance of the file copy
- Uses the C standard library for file operations and the time library for measuring the time taken to copy the file
- First prompts the user to select an option: 1 for file copy, or 2 for file copy with performance measurement
- The application will ask for the names of the source and destination files, open them, and use the `copier()` function to copy the file from the source to the destination if the user chooses option 1.
- The application will also ask for the maximum file size and the step size to increase the file size if the user chooses option 2. By increasing the file size step by step, copying the file, and monitoring the time it takes to duplicate the file, the performance of the file copy is measured using these inputs. This procedure is repeated until max file size is achieved.
- The program will then output the time taken to copy the file at each file size increment
- This can be useful for measuring performance of file copy under different scenarios/file sizes

The procedures outlined in the program are designed to measure the performance of file copying by allowing the user to select between a standard file copy option or a performance measurement option. The program uses the C standard library for file operations and the time library to measure the time taken to copy the file. The user is prompted to input the names of the source and destination files, and the `copier()` function is used to copy the file from the source to the destination if the user chooses option 1. If the user chooses option 2, the program will also ask for the maximum file size and the step size to increase the file

size. By increasing the file size step by step, copying the file, and monitoring the time it takes to duplicate the file, the performance of the file copy is measured. This procedure is repeated until the maximum file size is achieved. The program then outputs the time taken to copy the file at each file size increment, allowing the user to see the relationship between file size and CPU time in Table 1 below.

Table 1: CPU Time vs File Size.<sup>1</sup>



<sup>1</sup>. Final data visualization of the relationship between file size and cpu time

## Conclusion:

Even though there may occasionally be transmission and propagation delays, response time and CPU use for file transfers grow linearly as file size and quantity stay constant. The relationship between CPU time and file size can vary depending on the specific system and method used to copy the file. However, in general, as the file size increases, the CPU time required to copy the file will also increase. This is because larger files require more processing power and memory to transfer, and the system must work harder to copy the larger amount of data. This relationship can be linear, meaning that the CPU time increases proportionally with the file size, or it can be non-linear, meaning that the CPU time increases at a faster rate than the file size.