|  |  |
| --- | --- |
| A picture of a winding road and trees  SYSTEM PROGRAMING  12/6/2024 | Abstract  A develop a simulator for a client-server model that efficiently provides two distinct services  Hasan Alhwietat |

Contents

[1.What is the system programming how it differs from traditional programming? Describe how OS-controlled resources are used among processes of different purposes. You must give examples form your solution (Report). P1 2](#_Toc167197625)

[2. Discuss multi-process concepts and their practical applications? 6](#_Toc167197626)

[2.1 Explain multi-process concept and how you implement it in your simulator (Report). P2 6](#_Toc167197627)

[3. Discuss the implementation of multitasking in simulations and its practical coding aspects? 8](#_Toc167197628)

[3.1. Explain the concept of multitasking and describe how it can be implemented in a simulator (Report). P5 8](#_Toc167197629)

[3.3 Discuss the results of the thread management component and report other possible parallelization techniques and show how you handle such a situation if exits in your simulator, you may show examples (Report). M5 10](#_Toc167197630)

[4. After implementing multiprogramming, and multitasking, evaluate the effectiveness of the system's design and the developed components in terms of efficiency and execution time? (Report). D1 12](#_Toc167197631)

[5. Develop a client-server system in C to handle UDP/TCP communications and discuss the inter-process communication in your simulator. 14](#_Toc167197632)

[5.1. Discuss how the communication between processes is handled in your simulator, whether on a single machine or across different machines (Report). M2 14](#_Toc167197633)

[7. Describe -with aid of a diagram(s)- all the communication abilities of different processes in your simulator (Report). P3 16](#_Toc167197634)

[8. Implement a system component using shell scripting to manage file permissions, perform advanced regular expression searches, and handle inter-process communication, and then compare this approach with C programming in terms of system services? 18](#_Toc167197635)

[8.1. Discuss with code examples the usage of shell scripting compared to C programming for system services. This comparison should focus exclusively on services that can be accomplished by both methods, highlighting which practice is more effective and under what circumstances (Report). P4 18](#_Toc167197636)

[9. Critically evaluate the functionality, interface design, and ability to handle concurrent tasks in terms of user requirements (Report). D2 20](#_Toc167197637)

# 1.What is the system programming how it differs from traditional programming? Describe how OS-controlled resources are used among processes of different purposes. You must give examples form your solution (Report). P1

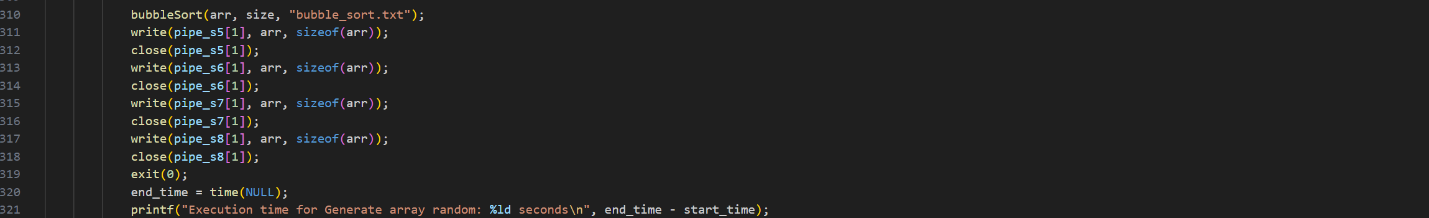
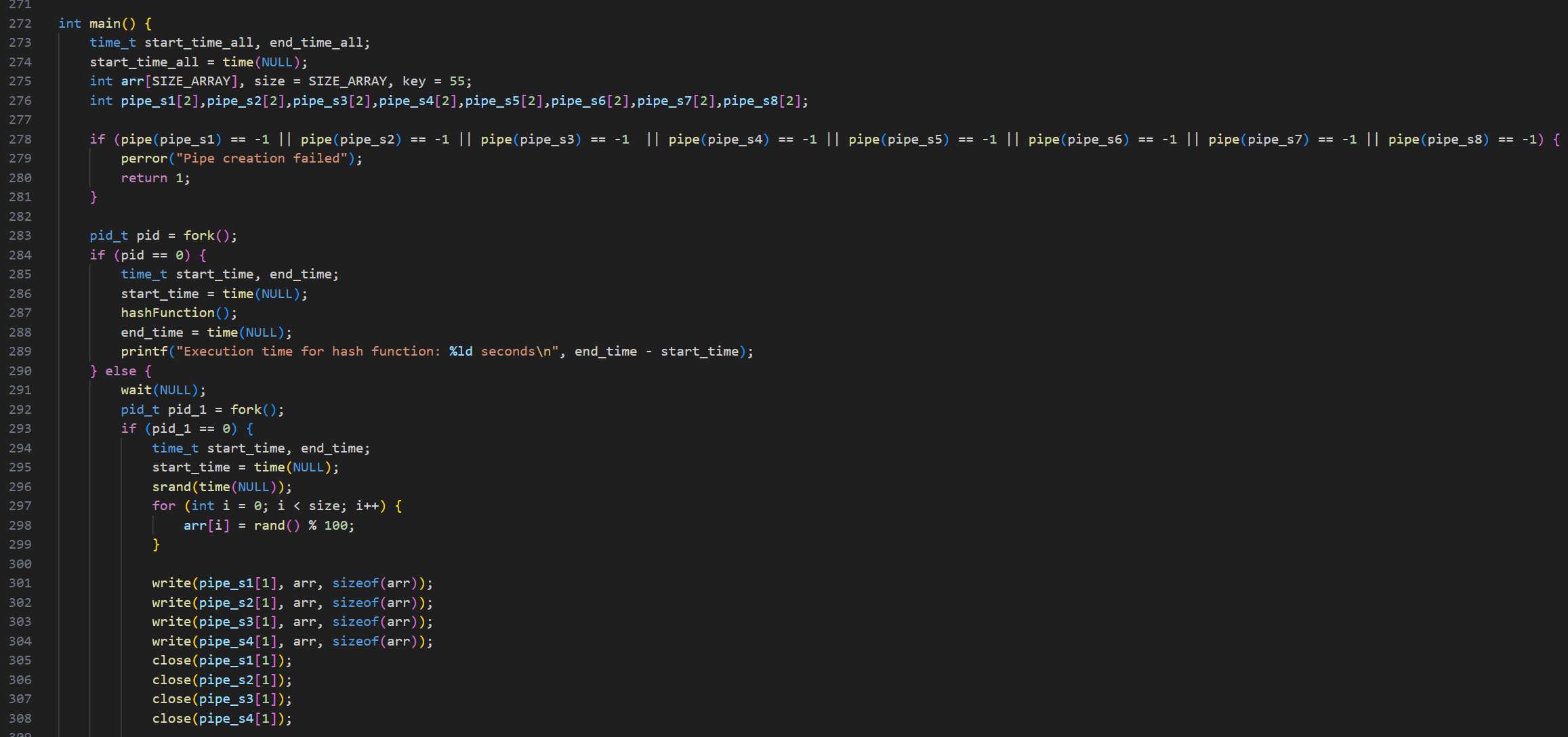
SP: It is part of computer programming that deals with programs for direct interaction with computer components or the operating system. It is the process of building systems using system programming languages. According to the computer hierarchy, the hardware comes first, then the operating system and system programs, and finally application programs so that programs can be developed in system programs and can Managing disks and input and output devices. (GeeksforGeeks, 21/4/2024)  
  
Example: Writing a device driver for a new graphics card to enable it to communicate with the operating system and applications.  
  
Examples SP: - (GeeksforGeeks, 21/4/2024)

1. File Management: A file is a group of data stored in the system’s memory, which includes managing, processing, deleting, and modifying the file. Each file has a different structure for its organization and storage within the systems, in addition to the operations that occur on the file, such as adding, modifying, and deleting data, in addition to file management operations such as copying, moving, and searching, in addition to file permissions such as reading, writing, deleting, etc.
2. Command line interface (CLI): It is a basic tool for the user and provides facilities for writing commands directly to the system to carry out any operation that helps in configuring the system, and processing files. It is one of the operations that take place on the system and allows receiving multiple commands, providing high flexibility and control over the execution of the code you need so that you can manage resources and process commands.
3. Device driver: It works as a simple translator for the operating system, which is an intermediary between the operating system and devices to understand each other’s languages so that they can work efficiently. The operating system allows connection to devices such as graphics, networks, printers, etc., so that it manages the devices and performs operations to ensure interaction and resource utilization, which helps enhance performance and access to the system for each connected device.
4. File Modification: Some files stored on disks or other storage devices are modified so that some of the file contents are searched. Programs and tools called text editing programs are usually used to handle text errors and the possibility of retrieving the old file before it is modified in order to preserve the data so that the size of the file is known and a place in memory is allocated for the file.
5. Programming-Language support: It supports most of the common programming languages that use Compilers, Assemblers, Debuggers, and interpreters. Compilers usually translate the source code into code ready to be implemented within the system, and this is often implemented by using debugging tools such as the IDE that helps find errors and fix them, with the possibility of using libraries and frameworks for rapid construction and to help developers make the code more efficient in terms of effectiveness and performance for each program.
6. Program Loading and Execution: When the program is prepared after completing the compilation, it is loaded into memory for execution. A loader is part of an operating system that is responsible for loading programs and libraries to memory, which allocates a certain space for data and also determines input and output resources and central processing unit time, with the importance of managing these operations that take place on the system largely by executing them simultaneously without overlapping some operations.
7. Communications: Communication between users and computer systems is provided through programs so that users can send messages on the screen, etc. For example, communication over the network is done through the protocols TCP/UDP, HTTP/HTTPs, and FTP to complete the communication process between the user and the server. Communication between processes can also be established through pipes, sockets, shared memory, etc.

OS: It is a program that acts as an intermediary between the computer and applications and provides a user interface on which programs are easy to execute. It handles all computer operations. It controls and monitors all programs on the computer. It is one of the most important system programs in the computer so that it can manage resources and operations. (GeeksforGeeks, Example OS, 1/11/2023)  
  
Examples OS: Windows, Linux, Mac, Android, IOS.

Differences between System Programming and Traditional Programming: (DiffStudy, 5/7/2019)

|  |  |  |
| --- | --- | --- |
| Difference | SP | TP |
| Develop | It focuses on developing computer systems and the internal parts of the operating system. Programming languages are at the lowest level, such as the C language, to control a specific component of the device, such as the graphics card or network interface. | It focuses on developing applications and software solutions for end users, using high-level programming languages such as PHP, JavaScript, etc. To develop websites and application. |
| Interact | Hardware | Software |
| Level | Low level | High level |
| Example | Loader, Linker, Compiler | Notepad, Word Pad, calculator, web browser |
| Level of Abstraction | A lower level of abstraction and closer to internal devices | A higher level of abstraction focuses on user interfaces using libraries and frameworks |
| Scope of Control | It greatly controls hardware resources and system behavior, so that the programmer deals with system settings directly | Limited control over hardware resources within constraints provided by the operating system and the compiler |
| Performance Considerations | It improves performance and efficiency as some operations require real-time constraints | Performance is not so important that the focus is on functionality and user experience |

  
How OS-controlled resources are used among processes of different purposes and give example from my solution?  


In the solution that I attached, the resources controlled by the operating system, such as the CPU and memory, are used by managing the operating system operations so that multiple operations are carried out at the same time, and the system resources are shared. In my solution, I was used in main by creating multiple sub-operations using the "Fork" system. Each process has a specific responsibility. Like in the first Fork in the child process. I implemented the hash function, which converts the sentence that is stored inside the file into encryption and stores it inside a file and then in the parent. I created a second fork. In the child process, a hundred-size array has been established in which different numbers are stored randomly and then stored in PIPE, which is a way to communicate between the various Fork in the parent process is created for a second to be in the child process, the implementation of the first sorting algorithm by reading the PIPE It is the stored array. It is sorted and then the parent process is created for a third fork to be in the child process to implement the other algorithm, and thus the code is carried out through all the sorting and research algorithms. Also calculated time The CPU in the solution. Once each process is completed, such as customer generation data, and options that display searches, screening and hash, the system schedules operations and allocates time and memory. memory is allocated to variables, and files to prevent conflict and ensure effectiveness. The inputs and outputs are also managed so that the data in the file are filled and read so that the inputs and outputs are managed to synchronize and prevent conflicts. For example, in the first process, data is entered into PIPE and then taken by PIPE. The array is sorted and stored in the file for each sorting algorithm and searching algorithms to find keys frequently in the array to reduce synchronization between different processes.

What the relationship between SP and OS?  
It has a direct and close relationship, as the programmer develops programs so that they interact with the operating system or basic resources in the devices. Writing system programs help to visualize and improve the operating systems and distribute operations to benefit from the implementation of all operations in the fastest time and best performance so that operations are scheduled. This allows interaction with the operating system kernel, and effective use of system resources to develop application programs. The kernel is a computer program that lies at the heart of the operating system (OS). It is the part of the operating system that loads first and remains in the main memory. The kernel connects the system hardware to the application software.

# 2. Discuss multi-process concepts and their practical applications?

## 2.1 Explain multi-process concept and how you implement it in your simulator (Report). P2

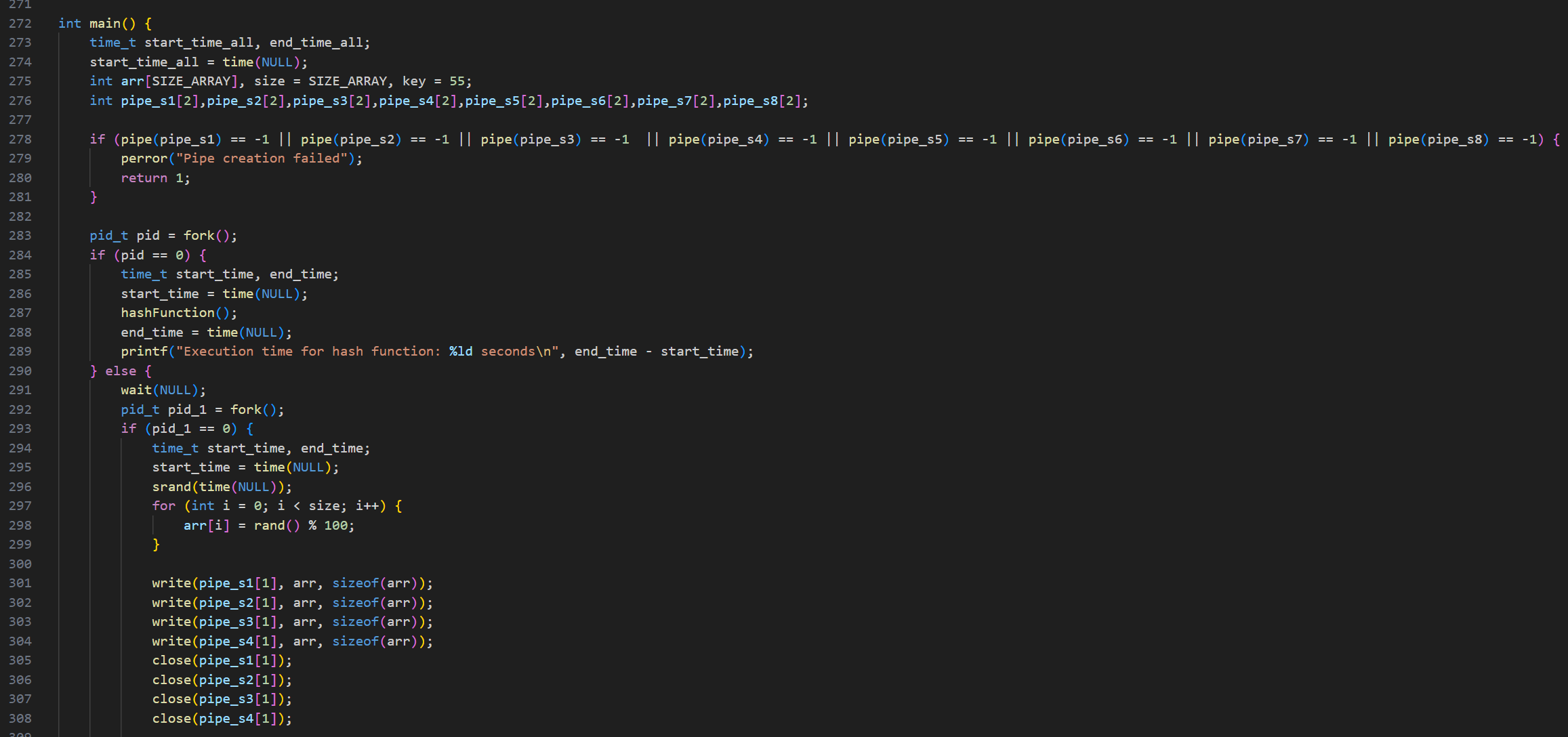
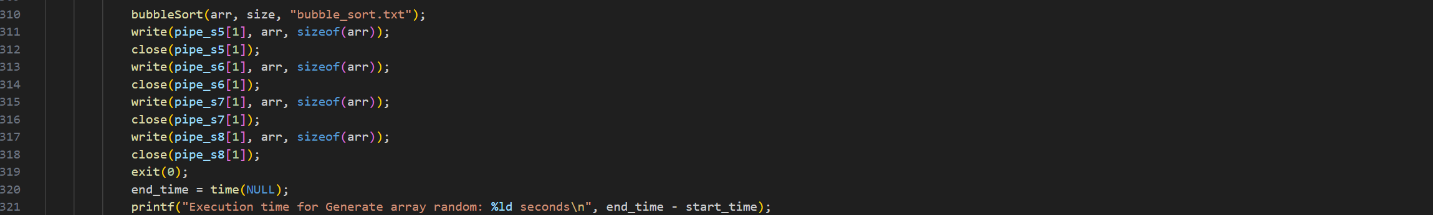
Multi-process: It is multiple programs that are executed simultaneously to achieve the maximum benefit from resources, allowing a single central processing unit to switch between multiple processes to achieve the maximum benefit.   
features of Multiprogramming: Context switching between processes reduces CPU idleness, increases resource usage, and obtains high performance. (Geeksforgeeks, 5/4/2024)

Describe why and how we can apply. (Geeksforgeeks, 5/4/2024)  
Multiple programs are stored in memory, so that the memory is distributed into parts called processes. Before executing the processes, the system chooses the process that should be executed before the other to execute it on the central processing unit. The processes may need input/output and are stored temporarily in the secondary storage unit.

Concepts: - (Geeksforgeeks, 5/4/2024)

1. Concurrency: Multiprogramming allows for the simultaneous execution of multiple processes, so that you quickly switch between programs, allowing the central processing unit to execute the processes in parallel. Through multiple programming, the programs are loaded into memory so that the CPU switches between them quickly, which improves the system’s response, eliminates memory idleness, effectively enhances productivity, and ensures that there are no resources unused to improve performance. Synchronous systems also work to coordinate communication with the possibility of sharing resources and not causing any interference between processes.
2. Process management: It manages processes and allocates time to the processing unit and memory to manage correct and efficient execution. When the system runs a process in response to the user, it allocates the resources it needs to execute the process correctly and with the best result and performance, so it schedules these processes within the operating system so that it arranges the processes on the CPU so that there is common communication between the processes such as passing messages, sharing memory, etc.
3. Memory management: Memory management is important because allocating memory to all processes and preventing automatic or dynamic allocation allows processes to pass and be executed efficiently within the system. So that each process has sufficient allocated memory so that there is no conflict or delay to other processes that affect the progress of the program. Memory management also allows setting up protection chains so that when any data that does not conform to the standards is placed, it is rejected or any way to access system permissions is blocked. Virtual memory can also be used to reduce Missing data and increased productivity

How to implement? (code)

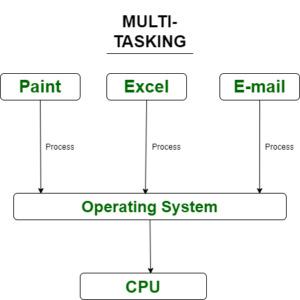


In a multi-process system, each program must be given a piece of memory known as the process, so that the processes are dealt with in the operating system before the execution of the processes occurs, so that the operating system chooses the process that is ready to be executed. In my solution, when we divided the processes using Fork (), In the first process, I implemented the hash function, which converts the sentence that is stored inside the file into encryption and stores it inside a file and then in the parent, and second fork I carried out a random matrix and then we passed it to the rest of the Fork using the PIPE so that the rest of the operations are carried out and then the CPU will confirm that the process and pass it for the other process. Thus, the process of moving between the operations occurs until the implementation is completed. The program benefits from the time and effective resources to increase the system's productivity.

# 3. Discuss the implementation of multitasking in simulations and its practical coding aspects?

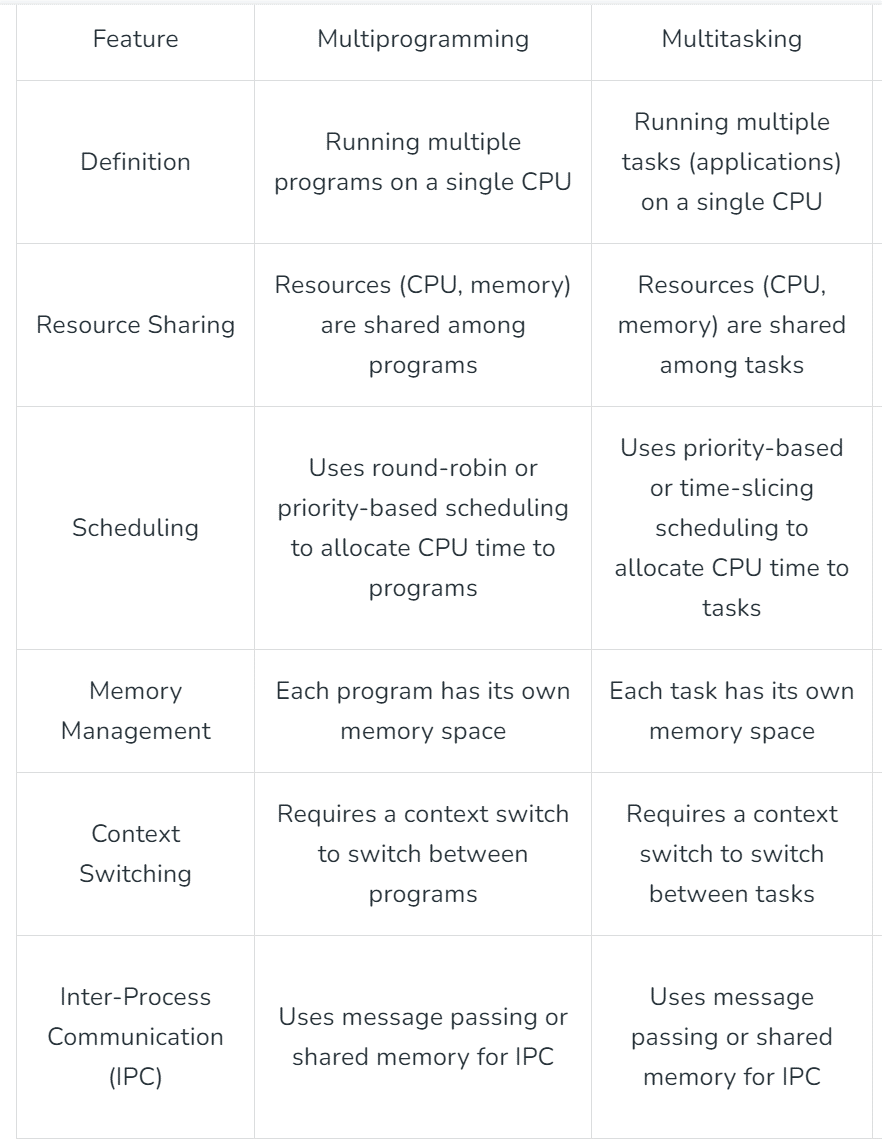
## 3.1. Explain the concept of multitasking and describe how it can be implemented in a simulator (Report). P5

Multitasking: It is the ability of the system to run many tasks or processes at the same time, so that it allows users to run multiple tasks at the same time. For example, it can listen to music, browse the Internet, and download pictures at the same time, and the allocation of system resources such as memory, input and output, and the central processing unit between processes can be easily managed. Using the multitasking system, and this picture shows how it works. (GeeksforGeeks, Multitasking, 16/7/2023)

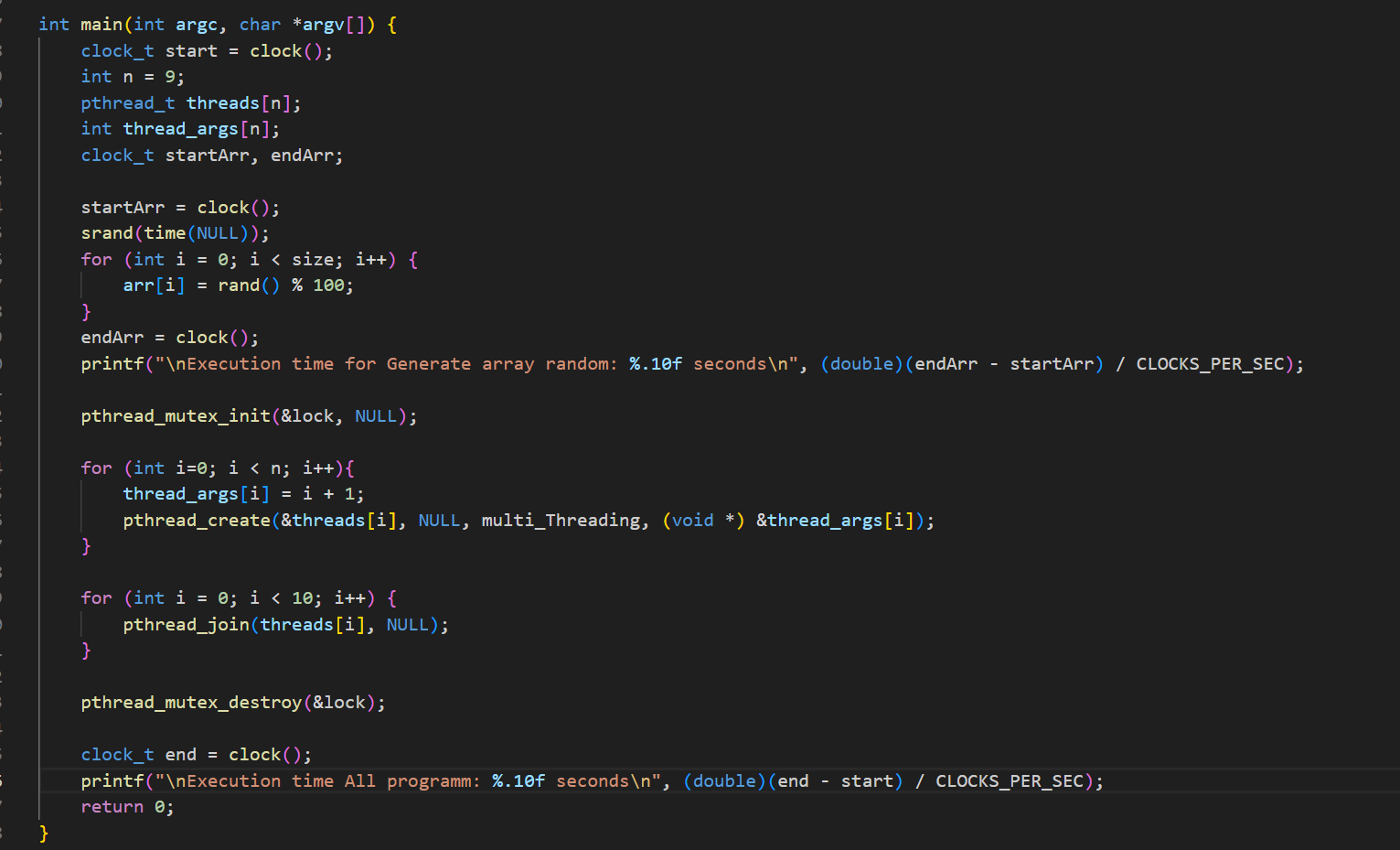
A thread can execute any part of the process code, including parts currently being executed by another thread.  
   
  
Concepts: - (GeeksforGeeks, Multitasking, 16/7/2023)  
1- Concurrency: It runs multiple tasks simultaneously so that they work simultaneously due to rapid switching between them.

2- Time sharing: When the code execution begins, the CPU allocates slices to each task in a circular manner, so that each task gets an allocated time from the CPU for execution, which gives parallel execution to take full advantage of the resources.

3- Context-switching: It is a process in which the state of the task is saved and restored due to the CPU switching from executing one task to another, so that a task is loaded when the other is finished and saved to take advantage of time and not lose data.

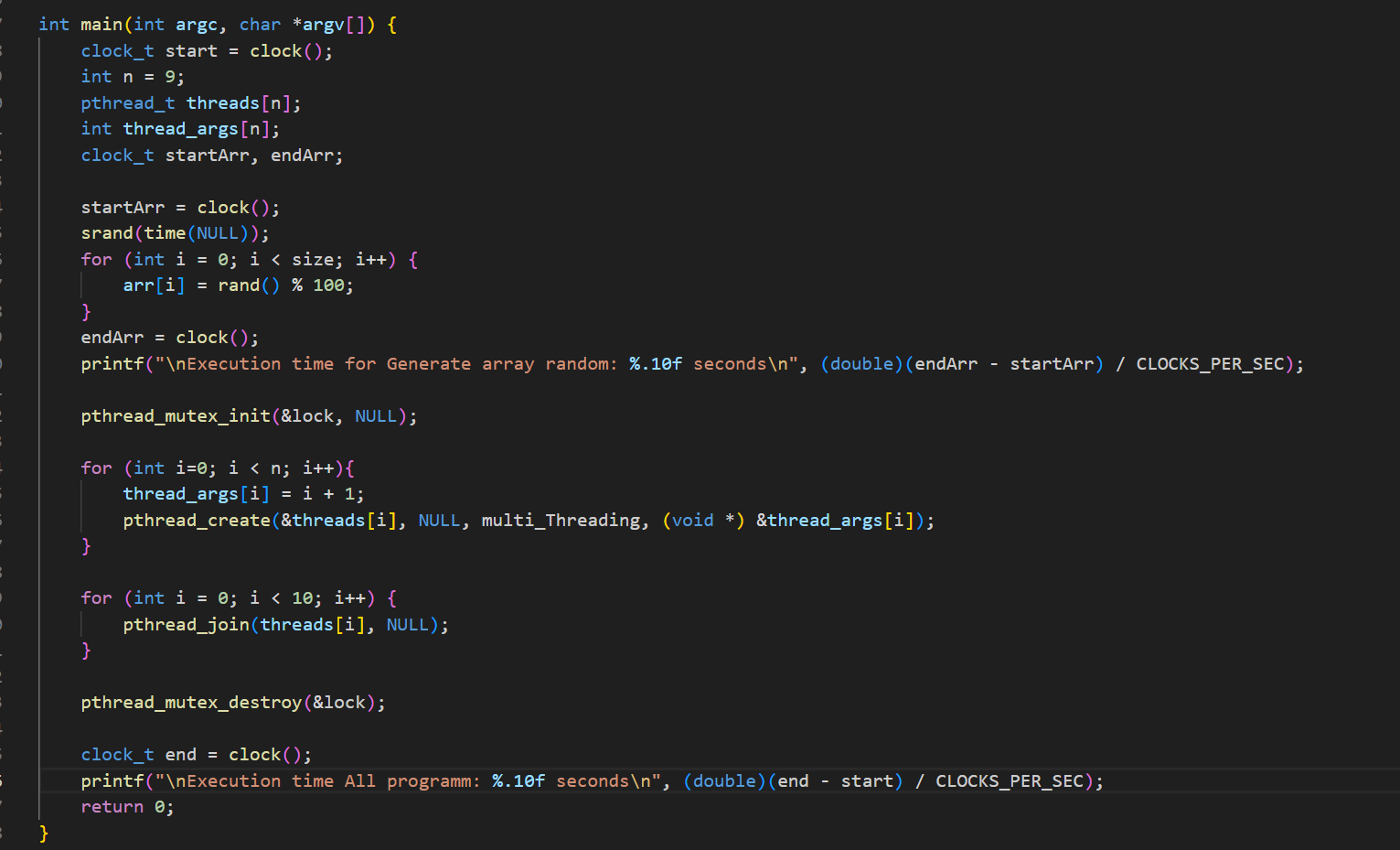
4- Multi-threading: It allows the program to enable more than one user at a time without the need for multiple copies in order to execute the program on the system without slowing it down.

How you implement in my code? (Threads)



In my code, we used many threads to implement the hash, generate random array, search algorithms, sorting algorithms. The principle of threads is greatly benefited from, so that operations are executed without waiting for other operations to finish, so that the CPU is divided between different operations to speed up the program. By sharing system resources and preventing precessions, the pthread\_create () was placed inside the for loop to create more than one thread to perform various operations, and the pthread\_join () was placed inside the for loop to wait for the process to finish at the same time to make the multitasking behavior through threads allow it to execute. Multiple processes within the same single process, so my code executes the hash, generate random array, search algorithms, sorting algorithms as separate threads to achieve balance and execute them simultaneously.

## 3.3 Discuss the results of the thread management component and report other possible parallelization techniques and show how you handle such a situation if exits in your simulator, you may show examples (Report). M5



Based on the results of using multi-threading, it helps to speed up operations without relying on the execution and completion of a single thread, but rather all threads are executed, and each thread has a specific memory and a specific portion of the CPU and other resources. It also reduces the time spent executing the program and reduces the cost as well as the cost. The possibility of a precedence situation occurring, such that if the order of threads depends on the result of a particular thread, then inconsistent and incorrect results will occur. Therefore, I used synchronization to prevent precedence situations. Synchronization refers to the mechanisms and rules that ensure the integrity of shared data objects and coordination between objects executed simultaneously. Among the types of synchronization, I used the lock that allows the entity to continue execution only. The locks are called mutex, which stands for mutual exclusion, so that they are initialized before use. I also used parallel computing, which attempts to use multiple processors that implement parallel algorithms. For example, I used parallel computing in my code. Search and sort algorithms depend on the principle of divide and conquer, like binary search and quicksort in Sequential, the code and algorithms must be executed step by step to finish executing the code, while Parallel executes the tasks in parallel, as I used in my code pthread\_mutex\_destroy(&lock): destroy thread attribute, so that it frees up memory for speed and saves space in the event of creating New threads so that processing operations can occur with higher accuracy The pthread\_join() function waits for a thread to terminate, detaches the thread, then returns the threads exit status. If the status parameter is NULL, the threads exit status is not returned.

The use of multiple threads makes the program faster and the program performs highly, with the ability to share data between threads. In terms of balanced load, when using threads in my example, the program time is reduced and it becomes more scalable. One of the important things is scheduling management, which allows threads to be assigned to CPU cores so that they are not Interrupting the thread. Rather, threads are created that are not affected by other threads to ensure data consistency and non-conflict, in addition to balancing the work and dividing the load between the threads. Mutex was also used to lock the resources and ensure their safety, so that whoever accesses the resources is determined. Also, when the program ends, the mutex will be killed, and in this way, I have ensured the high security of the system and the threading process has been managed successfully.

# 4. After implementing multiprogramming, and multitasking, evaluate the effectiveness of the system's design and the developed components in terms of efficiency and execution time? (Report). D1

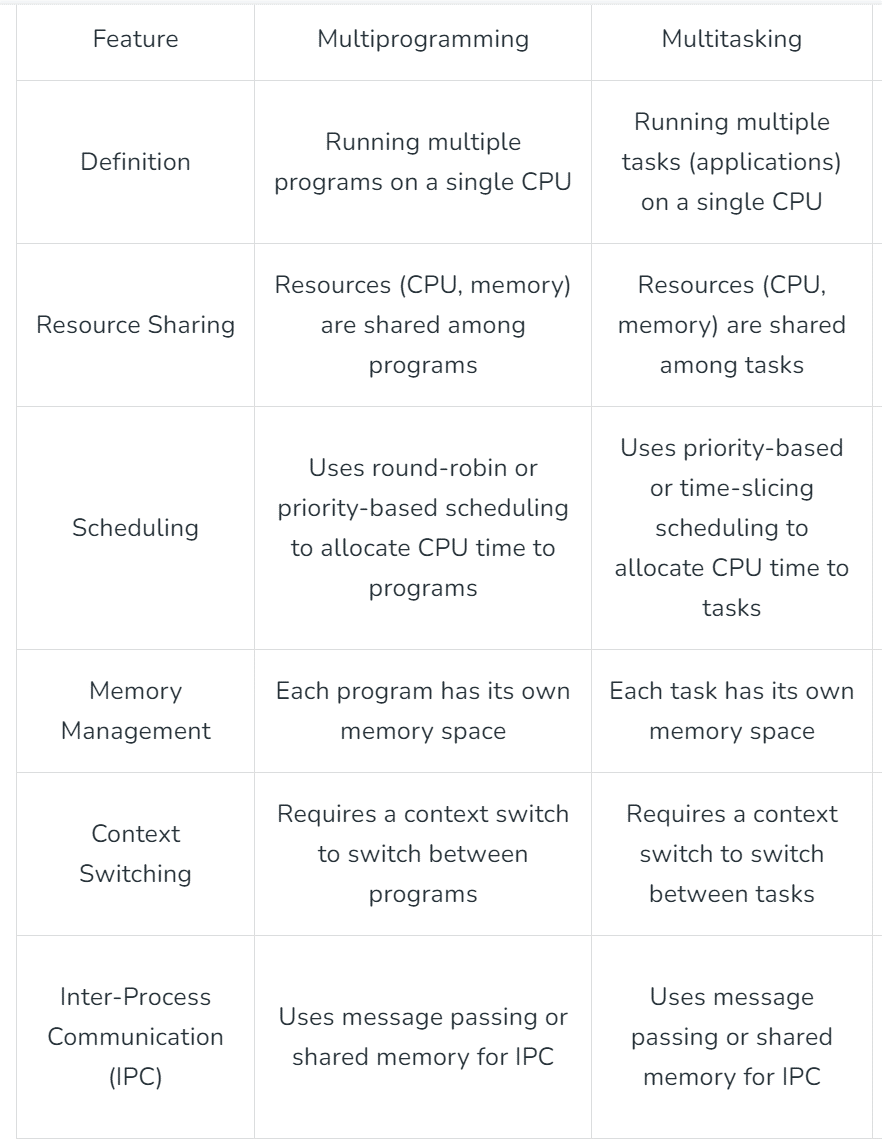
Efficiency is evaluated through the use of resources through the quality of use of these resources such as the central processing unit, memory, input and output, so that in multiprogramming it allows multiple processes to run simultaneously, which leads to the use of resources by keeping the CPU running and not causing any idleness, and greatly benefiting from the capabilities of System resources productivity is achieved by calculating the completed operations within a certain time, so that both multi-programming and multi-tasking seek to increase productivity by making use of all system resources and preventing the occurrence of waiting situations for any of the resources, so that a connection occurs between input and output operations with the CPU. As for time Response is evaluated based on the speed of response that the user gets. For example, multiprogramming is somewhat fast because the programs are executed in a balanced manner, while multitasking is faster because the system resources are divided among the processes, so the speed of productivity and response is faster.

In multiple programming, each program is executed sequentially, so that the first program is executed and then moves on to the next program until all programs are completed. In multitasking, all tasks are executed so that the task that is fastest in execution finishes first. Thus, the task does not wait for any other task to be executed and grows. All tasks are executed without affecting any task. This is the principle of multitasking, which divides system resources to make all tasks work without being affected by other tasks. Multithreading to handle different tasks simultaneously, with the goal of reducing overall execution time and improving efficiency. Different sorting and searching algorithms are implemented in separate threads

As for evaluating the time until the execution processes are completed, when testing the time for multiple programs on my code, it takes more time than executing it on multiple tasks because it needs time to process hash operations and search and sort algorithms. In terms of scalability, multitasking is better in terms of greater productivity. It is fast in performance, also in terms of user experience, the ability to make modifications, and high efficiency.

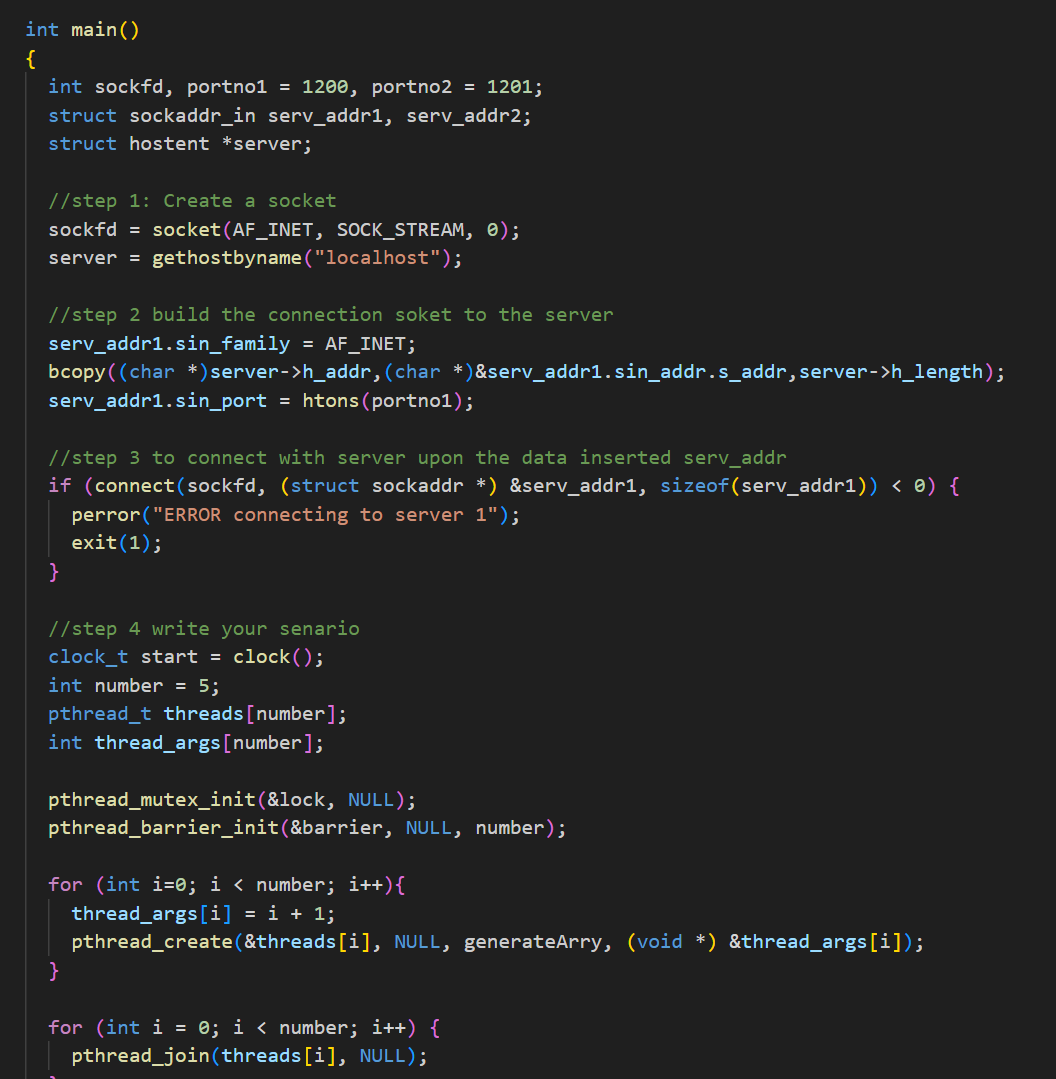
|  |  |  |
| --- | --- | --- |
| Compare | Multi-thread | Multi-processes |
| Time | Execution time All program: 0.0006220 seconds | Execution time All program: 0.0074070 seconds |
| Performance | High speed in execution, data sharing, and synchronization in data execution, so that it does not depend on the completion of the task until the other is executed, but rather it is executed in a balanced manner, and thus the use of resources is better and more effective and efficient in performance. | It is slow when executing large operations and does not perform the rest of the operations until one process after another process is completed, and so on |

Through the results of multi-thread and multi-processes, and by comparing the program execution time, performance, and scalability, it was shown that multi-threading is faster and better when there are complex and large operations that require reducing the time spent. The use of threads helps in this, with the possibility of improving and developing the code in the future.

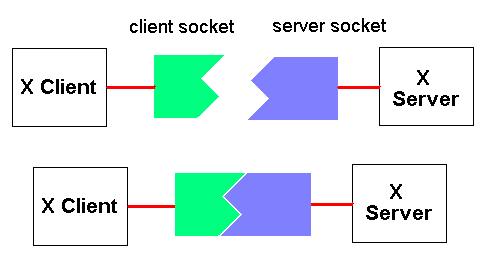


# 5. Develop a client-server system in C to handle UDP/TCP communications and discuss the inter-process communication in your simulator.

## 5.1. Discuss how the communication between processes is handled in your simulator, whether on a single machine or across different machines (Report). M2

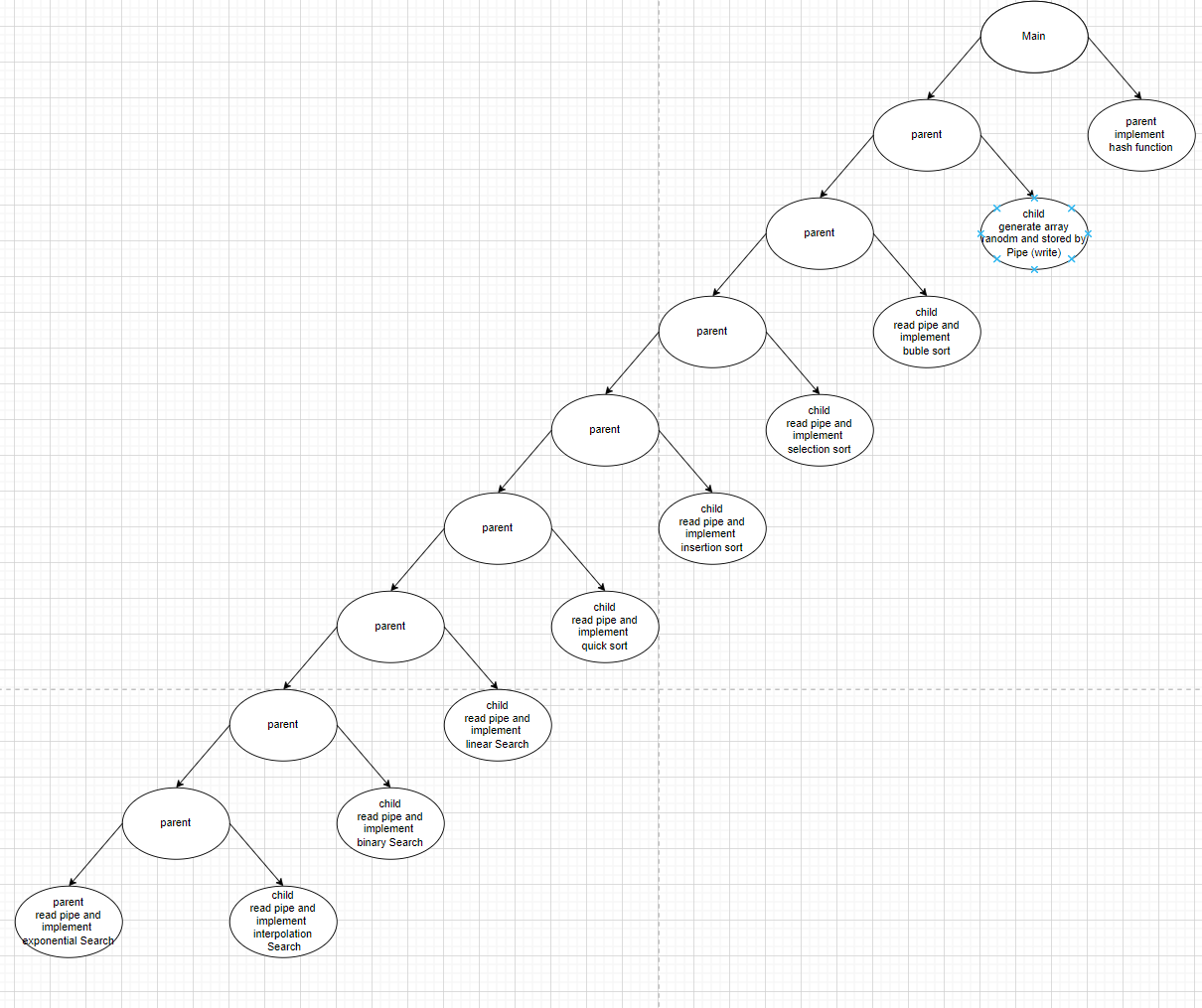


In my simulation device, I relied on running the program on a single device, where I linked the code between the server and the client through the port and IP address, where communication between the two processes is achieved by programming the TCP sockets so that the program processes run on the same device, where in the server code you configured the socket through the system call, and the server is linked to a port through the bind() function and listens for connections through listen() and the client connects to the server through the connect() function, specifying both the server port and its appropriate address. Data is transferred through the send function to send data from the client to the server and receiver to receive the data. Multiple threads are used in the client and server to carry out operations simultaneously. This process also provides many advantages such as reducing response time due to running the program on the same device. It makes security high and less vulnerable to risk and makes the use of... The system resources are more efficient due to the use of the TCP protocol, which manages basic network communications and ensures reliable delivery and transfer of data between the client and the server. It also allows for the possibility of expansion so that you can distribute the program to more than one device so that loads are distributed, which leads to a high response speed. This feature allows us to run TCP/IP applications on the same computer without actually being connected to the Internet. Before connecting, the client needs to know the server's existence and address. The server does not need to know the client's address before establishing the connection. Once the connection is established, both sides can send and receive information.

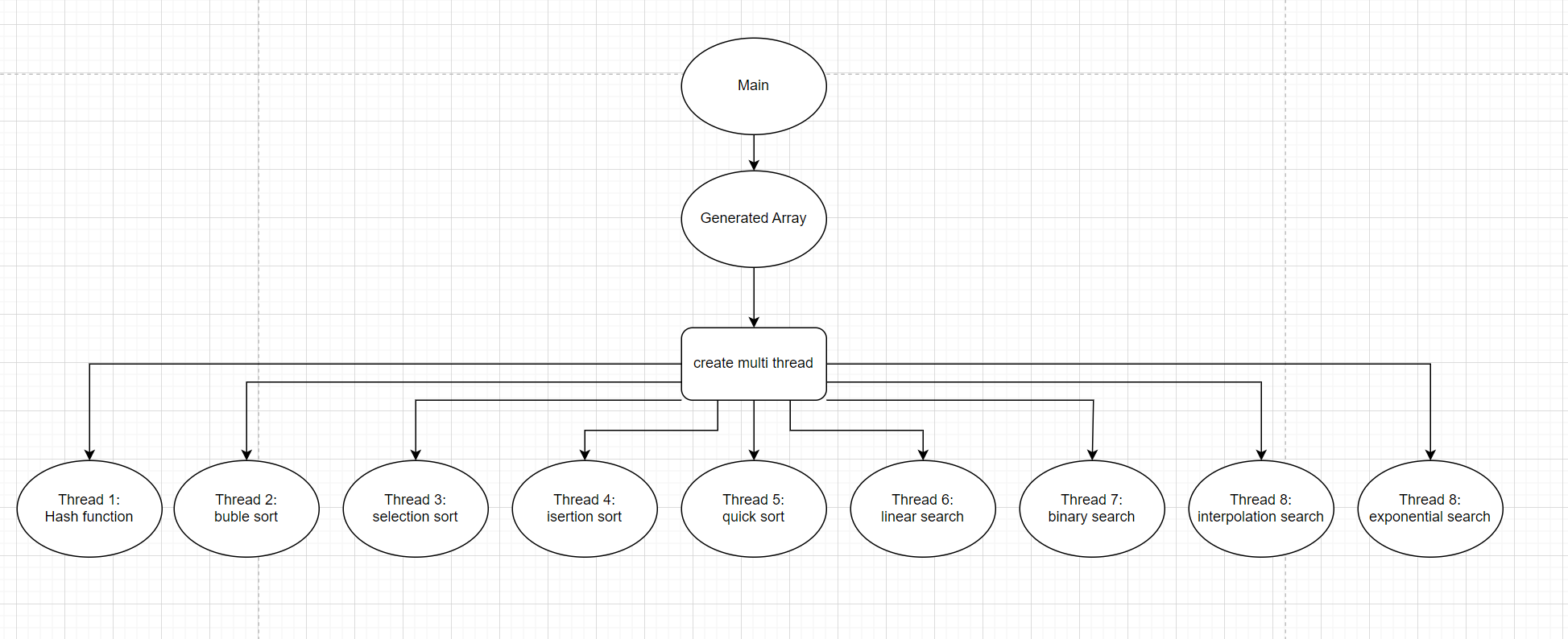


In order to communicate between the client and the server, you need sockets in order to communicate between the two applications and to send and receive data. There are several ways to do this, but the simplest is using the system calls read () and write ().

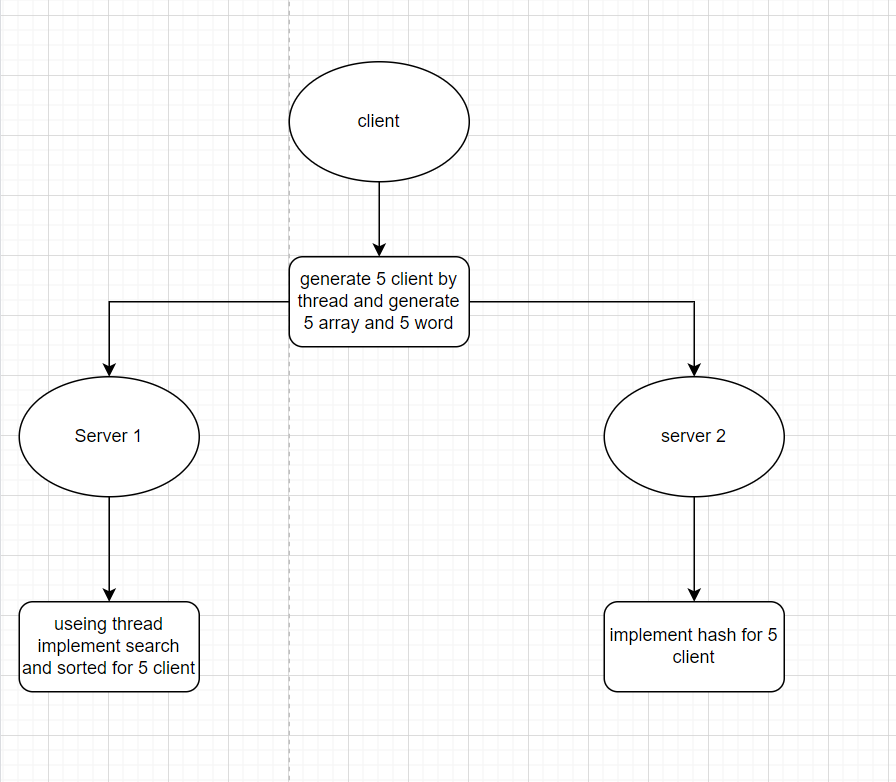
# 7. Describe -with aid of a diagram(s)- all the communication abilities of different processes in your simulator (Report). P3



In my code, operations communicate with each other by exchanging data and syncing. All operations communicate with each other by PIPE. Initially, the first process I implemented was the hash function, which converts the sentence that is stored inside the file into encryption and stores it inside a file and then in the parent process creates a second fork to create a random matrix and stores it inside the PIPE so that it sends these stored data to the various processes that will be carried out, such as search and sorting algorithms and to complete the process of sharing memory and communication between all operations. Then he reads the matrix passed from the first process to other operations through the PIPE and the code is carried out within each process. For example, the hash function and all sorting algorithms are performed, and then the search algorithms are performed. for interface show results will appear for the user so that the sediment that was sorted, as well as all the results related to searching the number and the frequent number with the time taken for all operations. This program displays communication between operations by allowing communication between operations, to complete the communication process, which allows the exchange of data through the PIPE, which makes it flexible and synchronized between different functions and benefits from the efficiency of the system, customized memory, and the rapid change between operations through the CPU.



Multi thread: These pictures show my scenario using multiple threads, in which each thread is separate from the other threads, which does not affect the progress of the program and will run at high speed as a result of using threads because they divide resources, memory, and CPU so that the program is executed in the fastest possible way and is more scalable and capable of carrying high data, for example. In the code, at the beginning, in the main, we created the array that takes random numbers, and then we created the multiple threads, and we sent the array with it so that it could perform the search and sorted algorithms and hash function, each of which works separately, unrelated to the other. In this way, the program runs faster, as the figure shows. Partitioning occurs so that synchronization occurs and the load is shared among other threads to reduce it



Socket TCP: Through the figure that shows how the program works by running it in one or more devices that occurs in the client, the IP and port number for each server is determined and an array is generated randomly and then sent to the first server using Write, which sends the array to the server and from Then it sends to the second server the word for which the client wants to hash it. When it is transmitted to server1 through the use of the IP and port number for server1, and using multiple threads, it receives more than one client, so that the server is not busy executing one client, but rather more than one client is searched. and sorted algorithms for each client. As for server2, it creates a hash for the word sent by the user and checks if it had previously encrypted the word through the file in which the words are stored. Through the use of multiple threads, the server has become more capable of being used by more than one client, which makes it more suitable than Driver programs that distribute loads, share data, and balance resources.

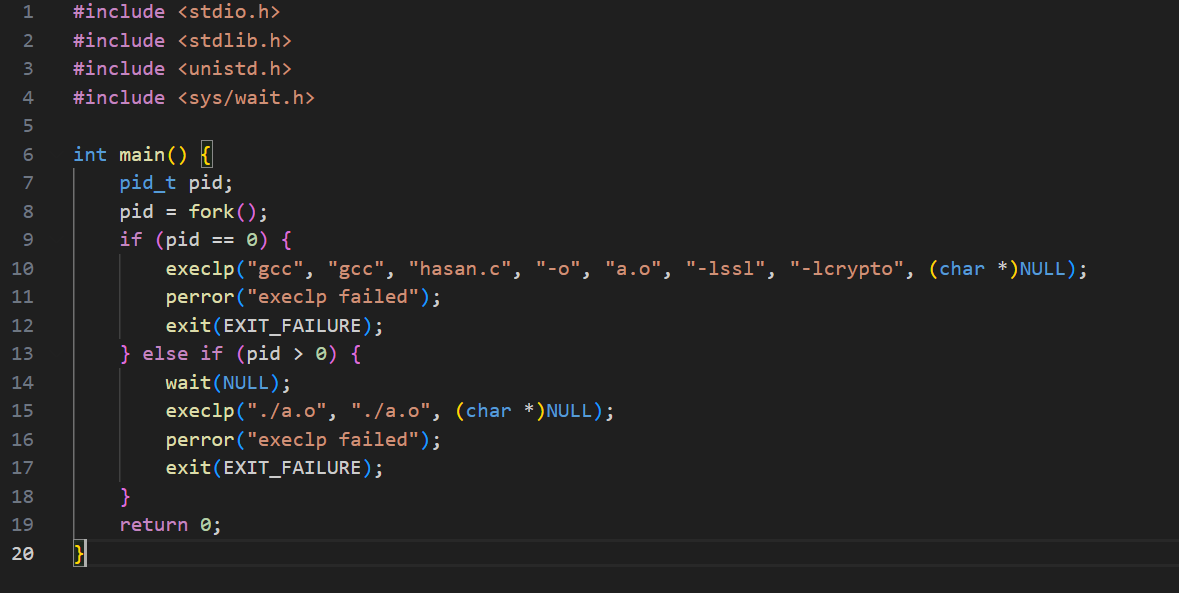
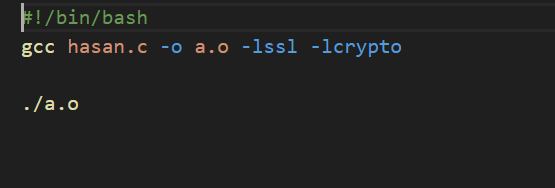
# 8. Implement a system component using shell scripting to manage file permissions, perform advanced regular expression searches, and handle inter-process communication, and then compare this approach with C programming in terms of system services?

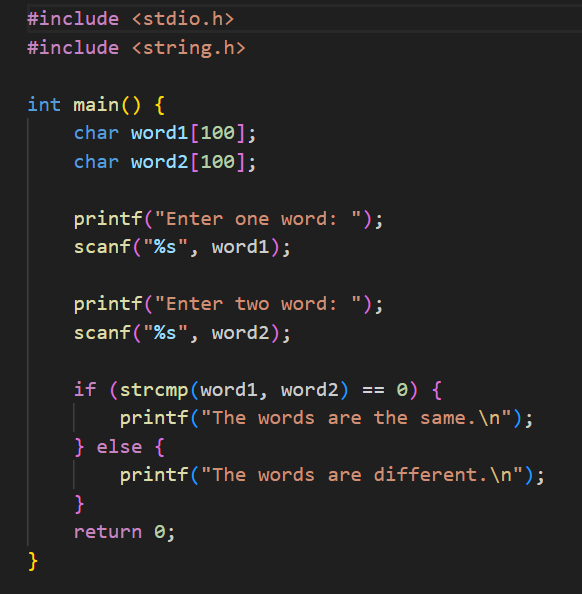
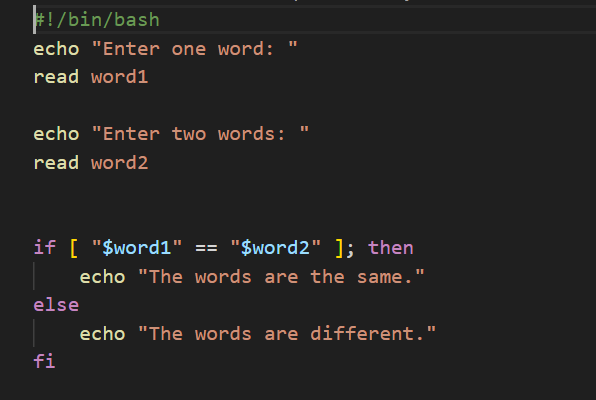
## 8.1. Discuss with code examples the usage of shell scripting compared to C programming for system services. This comparison should focus exclusively on services that can be accomplished by both methods, highlighting which practice is more effective and under what circumstances (Report). P4

The shell language is considered an easy language to write and understand. It is considered one of the simple languages ​​and the least complicated in writing code. The shell is a command interpreter that converts the text you write on the command line into simple procedures. It is an interface that takes commands from the user and is specific to operating system processes. Bash is not just a command interface. But it is a scripting language that allows us to use the capabilities of the shell and automate many tasks that may require many commands. For example, when we want to write a program for file operations, through shell script it is easier to understand and write compared to C, which requires building a complex program for file operations, but the C language provides better performance for tasks that require CPU-intensive use. However, shell scripts are more portable because they rely on compilers such as Bash, but C provides more features for controlling system resources and better error handling for complex tasks.

* What is the difference between programming and scripting languages:

Programming languages ​​are much stronger and faster than scripting languages ​​because programming languages ​​are compiled into an executable file that is run by the computer, while scripting is not compiled into an executable file, but it reads the code written in the source file and is run by a compiled program and executes all instructions without the need for compilation. Programming languages ​​are used to develop applications and operating programs that require high performance and direct access to system resources. They are written dynamically, which allows for flexibility and simplicity, but there are usually errors during operation. In terms of speed, programming languages ​​are better and faster because of the compiler and direct access to systems.





The four pictures show two examples of using shell script and the C language. The code shows the complexity of writing the code using the C language, but it is faster and better in performance and in using system resources. As for the shell script, which simplifies writing the code and makes it easier for the user to understand the code, but its performance is weak in executing complex programs.

Advantages shell script:

Shell script is important and simple when used to avoid repetitive tasks. You can run a series of codes with one command when using it. It is suitable for automating tasks that are performed repeatedly. It is considered easy to use. You can also use it for file operations such as backups and performing routine tasks. It helps save time and effort when executing tasks. Repetitive applications, such as file analysis and backups, therefore help improve productivity and efficiency, especially in implementing multiple tasks. It provides a high level of flexibility for system management, program development, and interaction with system resources. It also allows you to create customized solutions that suit your business and needs, integrate with services, provide a better experience for users, and makes the program more efficient.

Disadvantages shell script:

Miswriting a shell script can be costly, as the speed of program execution is very slow and it has major compatibility problems with different platforms.

It is possible to combine C and shell script, which will provide an integrated combination and a better way to take advantage of the strengths that will provide a balanced and more effective solution when combined. For example, you can write the code using C and run the program using shell script.

# 9. Critically evaluate the functionality, interface design, and ability to handle concurrent tasks in terms of user requirements (Report). D2

In the beginning, when we created the code using multiple forks and communicating between them through Pipe, after finishing writing the code and reviewing the results and the time taken for implementation, it became clear that the program suffers from problems, including the long time consuming and the multiple operations are not executed separately, but rather a process must be executed and then finished. You move to the other process for execution, and so on until the end of the program. Therefore, it had the disadvantages of a long time and slow performance. It becomes more complex to develop, not scalable, and less efficient in sharing resources and distributing burdens. It is difficult for anyone to understand and organize the code, but by using multitasking, things have become somewhat better. In using system resources and dividing them among various tasks and distributing burdens, there has become a balance in tasks because the nature of the work of multiple threads in sharing data and the CPU in a balanced manner guarantees in the future development and expansion and also meeting user requirements and is also more readable and understandable and more organized than multiple processes and also takes much less time. Among them, but when using socket and TCP, there is greater ability to expand and develop the program, so your program can be run on more than one device or device, which makes there greater data sharing, reduced burden, high flexibility and efficiency of the program, and better performance. This program can also be used by More than one client simultaneously and processing requests more effectively, but when using Socket and TCP and multi process and multi-tasking and shell script together, there becomes a program that has great features and provides high and distinguished performance in distributing burdens and loads and is more scalable and development in the future and more efficient and flexible. High guarantee in data processing, consistency and possibility of automating repetitive tasks and integration with external systems

Are there any potential issues, vulnerabilities, or limitations in the implementation?

1. Communication failure between the client and the server due to errors in data transfer.
2. Communication failure between the client and the server due to an increase in the number of client requests to enter the server, which leads to a major failure in achieving a large number of requests.
3. It is possible that race conditions and system resources are managed incorrectly.
4. Problems with scalability for multiple processes due to the lack of burden sharing between the various processes.
5. Failed to connect to the server due to security vulnerabilities exploited by the Attacker.

Improvement:

1. Using data structure can improve performance, synchronization, and better interaction with multiple clients while using multitasking.
2. It is possible to combine Socket and TCP and multi process and multi-tasking and shell script, so that when combining them in one program, it helps to use the features of each of them and use them in implementing the program, which makes there a benefit in sharing system resources, integration, synchronization, sharing burdens, and data processing. And its consistency is better
3. Verifying the correctness of data entry into the files and ensuring that they are exited without any change or modification to them by any third party.
4. In the event of errors such as a communication error, an error in sending data, or an exceeding of the amount of storage space, messages will be returned to the client explaining the problem the system is suffering from.

Recommendations:

Establishing greater security measures for both the client and the server, in addition to placing a firewall and a demilitarized zone to ensure the integrity of the data and its transfer in an encrypted form between the client and the servers, which ensures the continued operation of the program without interruption from security measures such as TLS/SSL, which is used to encrypt the connection.

It is also possible to perform maintenance on the program every period to ensure that the functions work well and develop the program and make it more capable of accommodating large requests from clients to the server.

Data and operations can be backed up in the event of a system crash or attack to protect data from loss