



**FATİH
SULTAN
MEHMET**
VAKIF ÜNİVERSİTESİ

Student:

Name: Emirhan

Surname: Soylu

ID Number: 2121251011

Department: Software Engineering

Project:

Topic: Distributed Calculator

Course:

Name: Operating System

Instructor: Ali Yılmaz Çamurcu

Content

1- Project Topic.....	3
2- Tasks Completed During the Project.....	3
3- Additional Notes.....	4
4- References.....	5



**FATİH
SULTAN
MEHMET
VAKIF ÜNİVERSİTESİ**

1-Project Topic

The **Distributed Calculator Project** implements a system where mathematical operations (addition, subtraction, multiplication, division) are distributed across separate subprocesses. The main program ('Calculator') communicates with these subprocesses using pipes. Additionally, a dedicated saver program records the results into a file, ensuring a robust and distributed system design. This project was developed as part of the operating systems coursework.

2-Tasks Completed During the Project

Main Program Implementation

- Developed the main program calculator.c to:
 - Create four subprocesses for handling mathematical operations.
 - Display a menu interface for users to select operations.
 - Send user inputs (two operands and operation) to the appropriate subprocess via pipes.
 - Receive computed results from the subprocesses through pipes.
 - Display the computed result to the user.
 - Exit subprocesses gracefully when the main program terminates.

Subprocess Implementation

- Created separate C programs for each operation:
 - addition.c: Performs addition of two integers.
 - subtraction.c: Performs subtraction of two integers.
 - multiplication.c: Performs multiplication of two integers.
 - division.c: Performs division of two integers, with error handling for division by zero.

Saver Functionality

- Designed the saver.c program to:
 - Accept calculation results as command-line arguments.
 - Save the results in a structured format: Result: <operand1> <operation> <operand2> = <result>.
 - Append these results to results.txt for persistent storage.

Inter-process Communication

- Implemented communication between the main program and subprocesses using dedicated pipes:
 - Pipes handle data transfer for inputs and results.
 - The subprocesses loop continuously to process multiple requests until the main program exits.

Graceful Termination

- Ensured all subprocesses terminate gracefully when the main program exits:
 - Used wait system call to synchronize the parent process with its children.
 - Closed unused pipe ends in both parent and child processes to prevent resource leaks.

Makefile Creation

- Developed a Makefile for automated compilation of all project components.
 - Included targets for compiling calculator, addition, subtraction, multiplication, division, and saver.
 - Simplified project building for seamless execution.

3-Additional Notes

Design Decisions

- Each subprocess uses a dedicated pipe for input and output to simplify communication.
- The saver program operates independently, receiving results as command-line arguments rather than via pipes, aligning with project requirements.
- The implementation uses modularity to enhance maintainability and scalability.

Error Handling

- Handled division by zero in division.c with appropriate error messages.
- Included error handling for failed system calls (e.g., pipe, fork, execlp) with descriptive error messages.

User Experience

- Provided a clear and interactive menu-driven interface in the calculator program.

- Saved results in a human-readable format in results.txt for easy verification.

Test Coverage

- Conducted extensive tests to validate functionality for all operations.
 - Verified multiple sequential operations to ensure pipes and subprocesses work as expected.
 - Confirmed correct error handling for invalid inputs and edge cases.

4-References

- 5- Operating Systems Course Material, [Instructor Notes and Lectures].
- 6- Official Linux Programmer's Manual (man pages for `fork`, `pipe`, `exec`, `wait`).
- 7- Online Resources:
 - a. Stack Overflow discussions for specific pipe and fork-related questions.
 - b. GNU Make documentation for creating `Makefile`.
 - c. GeeksForGeeks



FATİH
SULTAN
MEHMET
VAKIF ÜNİVERSİTESİ