National University of Computer and Emerging Sciences

Operating Systems Lab

Lab Manual 8

Muhammad Hassan Raza
Fall 2024

Department of Software Engineering
FAST-NU, Lahore, Pakistan

Question 1: You are required to implement a program that processes a long text containing several sentences. The goal is to determine which sentence is the longest and find the word with the maximum number of vowels in that sentence. Use named pipes (FIFO) to pass data between processes.

Task Breakdown:
1. The program will create a named pipe `fifo1`.
2. It will fork two child processes:
   - Child Process 1:
     - Reads the text file (provided as the first command-line argument).
     - Breaks the text into individual sentences.
     - Passes each sentence to the second child process via the named pipe `fifo1`.
   - Child Process 2:
     - Reads the sentences from the pipe, computes the length of each sentence, and tracks the longest sentence.
     - Finds the word in the longest sentence with the most vowels and writes the result to another file, say `result.txt`.
3. After both processes complete, the parent process should display the longest sentence and the word with the most vowels.

Input:
- The first command-line argument is the path to a text file.

Example:
Input File (`input.txt`):
"This is a test. Pipes are interesting. Inter-process communication is essential."

Expected Output:
Longest sentence: "Inter-process communication is essential."
Word with the most vowels: "communication"

Steps:
1. Create a named pipe `fifo1`.
2. Fork two child processes.
3. Child Process 1: Reads sentences from the file and sends them to the pipe.
4. Child Process 2: Reads sentences from the pipe, finds the longest sentence and the word with the most vowels, and writes the result to `result.txt`.
5. The parent process will print the results.

Question 2: You are required to write a program that takes a 2D matrix of integers as input through command-line arguments. The matrix's dimensions are passed as the first two arguments (number of rows and columns), followed by the matrix elements. You need to calculate the sum of each row, the sum of each column, and the overall sum using anonymous pipes for inter-process communication.

Task Breakdown:

1. The parent process should fork three child processes:
   - Child Process 1:
     - Reads the 2D array from the command line.
     - Calculates the sum of each row and sends it to the parent through an anonymous pipe.
   - Child Process 2:
     - Reads the 2D array from the command line.
     - Calculates the sum of each column and sends it to the parent through another anonymous pipe.
   - Child Process 3:
     - Calculates the overall sum of all the matrix elements and sends the result to the parent process through a third anonymous pipe.
2. The parent process collects the row sums, column sums, and overall sum from the child processes and prints the results.

Input:

- The first argument is the number of rows.
- The second argument is the number of columns.
- The remaining arguments are the matrix elements.

Example:

Command:

`./program 2 3 1 2 3 4 5 6`

Expected Output:

```
Row sums: 6 15
Column sums: 5 7 9
Overall sum: 21
```

Steps:

1. Fork three child processes to calculate row sums, column sums, and overall sum.
2. Use anonymous pipes to communicate between the parent and the children.
3. The parent process prints the results after receiving the data from the children.