



CSE360: Computer Architecture

Section: 01

Mini-project Presentation



Our project is on-

Implement a Dynamic Branch Predictor Using a Perceptron Learning Algorithm in C

Submitted To:
Dr. Md. Nawab Yousuf Ali Professor, Department of Computer Science & Engineering .

Submitted by: (group – 09)	
Rihan Mahmud	2020-2-60-123
Junnun Mohamed Karim	2022-1-60-108
Md. Yousuf Hozaifa	2022-1-60-162

Introduction

What is Branch Prediction?

Branch prediction is a key component in modern processors that enhances performance by guessing the outcome of conditional branch instructions before they are resolved.

Why does it matter?

- A wrong prediction wastes cycles as instructions from the wrong path are flushed.
- Correct predictions minimize delays, boosting overall throughput.

What is a Perceptron?

A perceptron is a simple machine learning model inspired by biological neurons, designed to classify inputs based on a weighted sum of features and an activation threshold.

Why Perceptrons?

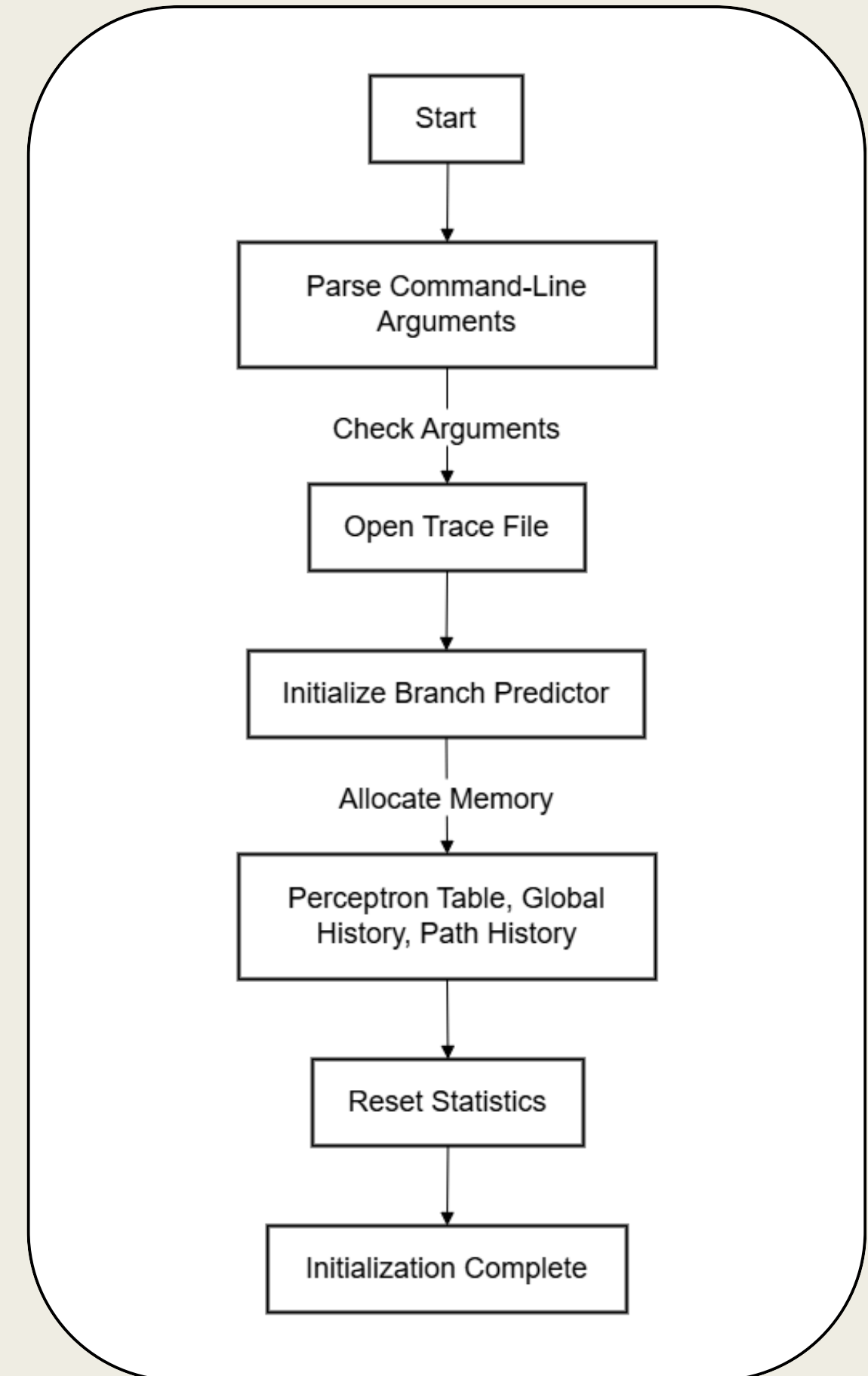
Perceptrons can learn patterns in historical data, making them ideal for identifying correlations between branch outcomes and branch history.

Program Flow

Initialization

This phase sets up the necessary resources, including logging, memory allocation, and global variables.

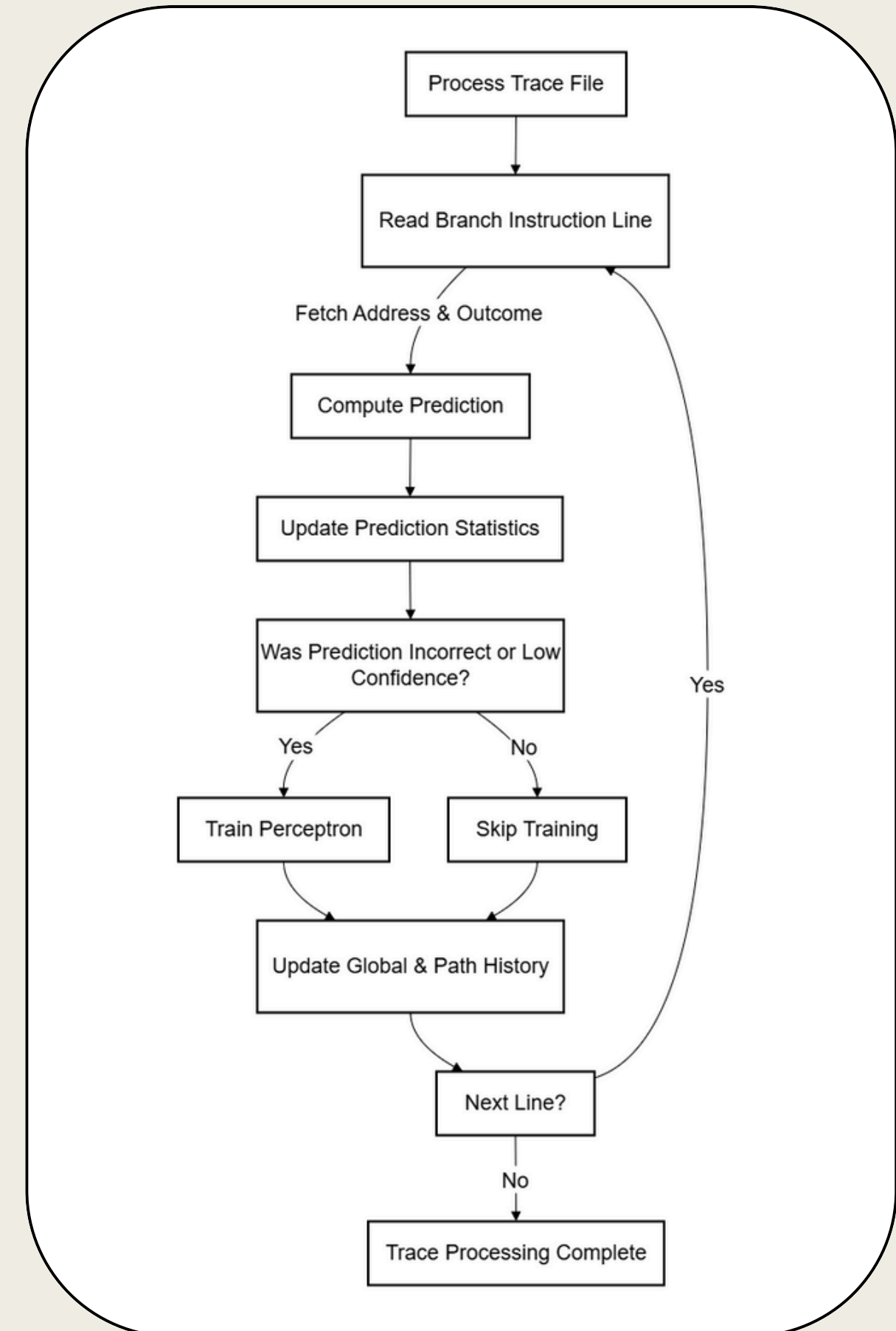
- Parse command-line arguments:
 - Check for *trace-file* and *--debug mode*.
- Open the trace file for reading.
- Initialize the branch predictor:
 - Initialize logging (if debug mode is enabled)
 - Allocate memory for:
 - Perceptron table.
 - Global history.
 - Path history.
 - Reset statistics.



Trace Processing

This phase processes the input trace file and performs branch predictions and training.

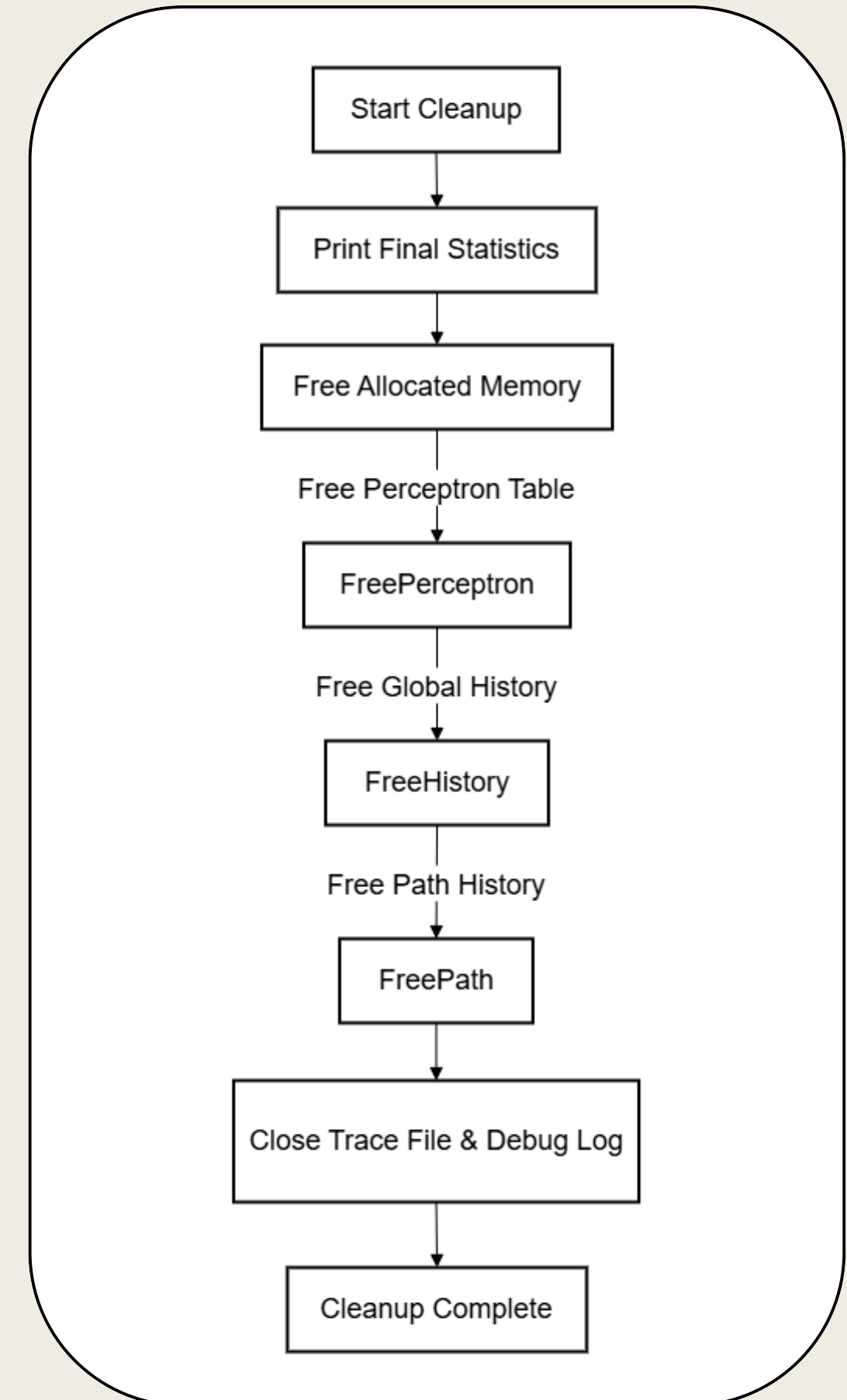
- Start reading the trace file line by line.
- For each branch instruction:
 - Fetch the branch address and actual outcome.
 - Compute the prediction using the perceptron predictor.
 - Update statistics based on prediction accuracy.
 - Train the perceptron if:
 - The prediction was incorrect, OR
 - The confidence is low.
- Update global and path history.



Cleanup and Reporting

This phase handles memory cleanup and final statistics reporting.

- Print final statistics:
 - Total predictions, mispredictions, accuracy, confidence, etc.
- Free allocated memory:
 - Perceptron table.
 - Global history.
 - Path history.
- Close trace file and debug log (if enabled).





Thank you
everyone!

