Gebze Technical University
CSE344 Systems Programming Course
-
HOMEWORK 2 REPORT

Muhammed OZKAN 151044084

April 11, 2021

Abstract

Process m (m for mother) will receive as a command-line argument the path to a regular ASCII file. The file will contain at least 8 rows, where each row will contain 16 comma separated real values. Every row is assumed to represent 8 coordinates in the form of : x0,y0,x1,y1,,x7,y7 where yi = f(xi) is an unknown function. There are 8 rows, so there are 8 unknown functions. Your task is to estimate these 8 functions through polynomial interpolation using the Lagrange form (check its wiki page) in two rounds. In the first round you will use the first 6 coordinates of each row, and in the second round you will the first 7 coordinates of each row, leaving the last coordinate for validation.

1 Overview

The homework I submitted is a complete assignment. It supports all the requirements and parameters requested in the homework document and opens the file with the given path. 8 children create processes. Children read the line given to them from the file and at the end of each step, they make the necessary calculations in 2 steps according to the signals of the mother and write them in the file. When the signal comes in the last step, it terminates itself by printing the coefficients of the polynomial on the screen.

Meanwhile, the mother waits for the children to finish their calculate with the relevant rows at each stage, takes the error averages of the sales according to the finished signal and writes them on the screen in both stages.

2 How did I solve this problem.

First of all, the program had to get the file path that came to it and open it with a file descriptor. After that, it was necessary to create 8 child processes by the mother and inform them of which lines to read in the order they were created. Child processes were created with the help of () and they were given which lines to read.

After this stage, we had 8 children and 1 mother process. 2 separate functions were designed for these two types of operations. For the Lagrange polynomial degree 5 and degree 6 estimation calculations of child processes, the code in the site specified in the section below has been used. In addition, for the Polynomial Fitting algorithm used to find the coefficients, the Gaussian elimination technique and codes on the site below are used.

File lock has been applied to prevent data loss and ensure synchronization when reading and writing files. SIGUSR1 and SIGUSR2 signals were used for synchronization between mother and children. With the sigsuspend() function, processes are provided to wait for signals where necessary.

When children finish their operations at the end of each episode, it sends a SIGUSR1 signal to their mothers. When the mother who received them has a total of 8, she calculates the average error she has to do and sends the SIGUSR2 signal to the children to switch to the next section. The children who take it move on to the other section. This process is repeated and the mother sends the SIGUSR2 signal for the last time for the last time, and the children who receive it end themselves by calculating the coefficients and printing them on the screen.

3 My design decisions.

The program we have prepared consists of 2 main functions and 10 auxiliary functions. The main functions are the children () function used by children. And it contains the necessary procedures for children to do, respectively. The second major main function is our mother function managing the children. This is the part that manages the necessary signals to the children according to the signals coming from them and contains the actions that it has to do.Our other auxiliary functions are the function that reads the relevant line from the file. Another related line is our function that adds data. We have 2 functions that manage file locks, lock and unlock them. We have 2 separate auxiliary functions that calculate Lagrange polynomials and coefficients. These are taken from the internet and the addresses are given below. Finally, we have 3 functions that we use to receive signals and a function that we use to convert the were read lines into numerical data and arrays. We have a modular structure using these functions. We reached the solution of our problem by using them sequentially.

4 What requirements did I meet and which did I fail?

All requirements are fully met. However, in some cases, depending on the structure and form of the system used (in the virtual machine), sometimes problems occur in capturing and sending signals. In such cases, it is necessary to run the program more than once.All processes created at the end of the program are terminated and resources are returned to the system.

5 Source Sites Used

https://www.codesansar.com/numerical-methods/lagrange-interpolation-method-using-c-programming.htm
https://www.bragitoff.com/2018/06/polynomial-fitting-c-program/

6 Tests

```
./processM example
Error of polynomial of degree 5: 23.3
Error of polynomial of degree 6: 9.7
Polynomial 0: -491.0,917.6,-547.0,153.0,-22.2,1.6,-0.0
Polynomial 1: -13.0,37.5,-26.5,9.9,-2.0,0.2,-0.0
Polynomial 2: 16.6,-8.4,-1.7,2.0,-0.4,0.0,-0.0
Polynomial 5: -51.7,89.3,-48.3,12.1,-1.5,0.1,-0.0
Polynomial 6: -1836.3,1791.2,-688.5,134.7,-14.2,0.8,-0.0
Polynomial 3: 799.0,-1125.8,615.2,-167.8,24.3,-1.8,0.1
Polynomial 7: 70.6,-113.3,69.4,-20.6,3.2,-0.3,0.0
```

```
./processM example
Error of polynomial of degree 5: 23.3
Error of polynomial of degree 6: 9.7
Polynomial 0: -491.0,917.6,-547.0,153.0,-22.2,1.6,-0.0
Polynomial 1: -13.0,37.5,-26.5,9.9,-2.0,0.2,-0.0
Polynomial 2: 16.6,-8.4,-1.7,2.0,-0.4,0.0,-0.0
Polynomial 4: -109.8,211.1,-134.7,40.1,-6.0,0.4,-0.0
Polynomial 3: 799.0,-1125.8,615.2,-167.8,24.3,-1.8,0.1
Polynomial 5: -51.7,89.3,-48.3,12.1,-1.5,0.1,-0.0
Polynomial 7: 70.6,-113.3,69.4,-20.6,3.2,-0.3,0.0
Polynomial 6: -1836.3,1791.2,-688.5,134.7,-14.2,0.8,-0.0
```