CSE 344 SYSTEM PROGRAMMING

HW2 REPORT:

-SERDİL ANIL ÜNLÜ

-1801042672

How I solved this problem?

I create processR.c ve 1801042672.c file for parent and child process, in 1801042672.c, I have done the parent operations. In processR.c I have done the child operations. In parent file I used the SIGINT signal with in sigaction. Because SIGINT is one of the predefined signals that's associated with the terminal interrupt character (Ctrl+c). Also in parent file I forked the process_id first when creating the child. Then I increased the number of children where necessary. I use execv structure while I making fork() process. If the fork is successful I create an array of char args pointers. By the way, I used sprintf to communicate between the args char pointer and the buffer while creating the child. Then, I find the distance between the frobenius function I created in the forward parent and the distance between the two closest matrixes thanks to the covariance matrix created by the child. I used the wait() operation to avoid zombie children. I used sigfillset, sigemptyset and sigprocmask signals in the processR file, that is, the child file. I made these signals so that the children do not enter the process at the same time while they are processing, but queue up properly.Finally, since we can take the data from the input file and transfer it to the output file in the format we want, I transferred the data to the output file in binary format.

MY DESIGN DECISION?

I bring the data I get from the input file to the appropriate format and get the covariance matrices in the processR file. When transferring this data to the output file, I am sending it in binary form. I take the data I

received in the file number 1801042672 and take it as 3 by 3 floats. I found the shortest distance and pressed it to the terminal by applying the frobenius process in the matrices I wrote 3 by 3.The reason why I do the operations in 2 files is because I use the fork() + execv operation.

Which requirements I achieved and which I have failed?

I have used signals extremely well against interruptions in file operations. I used sigfillset, sigemptyset, sigprocmask signals for this. I did the covariance operations in the child file. I did the covariance operations in the child file. I do fork+exec and wait() operations in the parent file. I'm doing the Covariance and Frobenius operations, but there may be slight deviations. Also, there is no memory leak. Also, I have checked all the all the controls with -Wall, there is no warning.

The part that works wrong in my code is that when the same file is run over and over, it starts to give the closest 2 points as 0 because it gives the same coordinate values. Therefore, if you reset the output file or use a new output file when you enter a new value, you will get a healthy output.

OUTPUT EXAMPLE:

```
Created R_0 with:(65)(83)(70),(83)(65)(70),(83)(65)(70),(65)(83)(70),(65)(83)(70),(65)(83)(70),(65)(83)(70),(65)(83)(70),(65)(83)(70),(83)(65)(70),
Created R_1 with:(83)(65)(70),(83)(65)(71),(68)(71)(67),(83)(65)(67),(83)(65)(73),(70)(65)(83),(70)(85)(72),(83)(65)(85),(83)(65)(70),(119)(114)(97),
Created R_2 with:(119)(119)(97),(115)(97)(98),(102)(106)(98),(106)(97)(98),(102)(106)(97),(98)(106)(102),(98)(97)(115),(106)(102)(98),(97)(115)(106),(102)(106)(97),
Created R_3 with:(115)(106)(102),(115)(97)(102),(76)(65)(75),(70)(75)(65),(76)(83)(75),(76)(75)(83),(65)(75)(70),(83)(65)(70),(83)(65)(79),(70)(75)(65),
Created R_4 with:(83)(79)(75),(70)(79)(83),(65)(75)(70),(79)(75)(83),(65)(79)(70),(75)(79)(83),(65)(75)(70),(79)(83)(70),(75)(79)(65),(83)(75)(79),




 0.Covariance Matrix : {(68.040,-68.040,0.000),(-68.040,68.040,0.000),(0.000,0.000,0.000),}
 1.Covariance Matrix : {(185.650,145.150,84.950),(145.150,227.450,93.250),(84.950,93.250,85.250),}
 2.Covariance Matrix : {(48.450,6.550,-20.600),(6.550,50.090,-7.060),(-20.600,-7.060,30.440),}
 3.Covariance Matrix : {(285.690,171.310,202.560),(171.310,171.290,132.940),(202.560,132.940,165.840),}
 4.Covariance Matrix : {(47.290,3.480,15.180),(3.480,6.560,-3.040),(15.180,-3.040,40.760),}

The Closest 2 matricies are: 2.Matrix and 4.Matrix, and their distance is 16.000000.
```

-Since the format is up to us, I print the covariance values I get to the file in binary form to output file.

I created a function to use the SIGINT signal, set the flag value in it to 0 ,and performed the necessary operations according to whether the flag value changed or not.

```
{
    struct sigaction sa;
    sa.sa_handler=&handler;
    sigaction(SIGINT, &sa, NULL);
    int number_child = 0;
```

```
  if(flag == 1){

      close(fdWrite);
      remove(argv[0]);
      perror("Removed file process succesfully done !!!! \n");
  }
```

-In order to find the shortest distance using frobenius from the last operation, covariance, I call the function in which I wrote frobenius in this function and print in the terminal.

```
int readOutput(int *ans,int fdRead,int index){

    int saver;
    float arr[3][3];
    float buffer[9];


    if (fdRead < 0){
        perror("can't open\n");
        exit(1);
    }


    saver = read(fdRead,buffer,9*sizeof(float));
    if(saver == 0){
        *ans = -1;
        return saver;
    }

    for(int i = 0; i < 3; ++i){
        for(int j = 0; j < 3; ++j){
            arr[i][j]  = buffer[i*3 + j];
        }
    }
```

```
    *ans = frobeniusDistance(arr);
    return saver;
```

```
float frobeniusDistance(float matris[3][3]){

    int sum=0;
    float result;

    for(int i=0;i<3;i++){

        for(int j=0;j<3;j++){

            sum += pow(matris[i][j],2);
        }
    }

    result = sqrt(sum);

    return result;

}
```