Gebze Technical University

Computer Engineering

CSE344 – System Programming

HOMEWORK #3

REPORT

NEVRA GÜRSES

161044071

Nevra Gürses
161044071

1 INTRODUCTION

1.1 Problem Definition

There are 5 processes ,one of them parent process and others are children processes.Parent process reads (2^n)x(2^n) character from two input files.Each character is converted to ASCII code integer equivalent.These (2^n)x(2^n) characters are elements of square matrix.So after reading two files,there are 2 square matrix A and B. Parent process P1 receives this two square matrices.After it creates 4 children process.Tasks of children processes are calculating one quarter of multiplication matrix of matrix A and matrix B.Each children process calculates one quarter of multiplication matrix,then gives result to parent.After all quarters are calculated,parent process combines them and calculates the singular values of multiplication matrix.

To achive this tasks,there must created bi-directional pipes between parent process and each of its children.And also parent process P1 must catch the SIGCHLD signal and perform a synchronous wait for each of its children.And in case of CTRL-C, all 5 processes must exit gracefully.

2 METHOD

2.1 Problem Solution

For solving this problem,firstly,I read (2^n)x(2^n) characters from each input files.After reading files, I convert each read character to ascii integer equivalent.After I create two matrices that are matrix A and matrix B.Then I divide matrices into arrays for quarters of matrix multiplication. I save up half and down half of matrix A into arrays.Then I also save left half and right half of matrix B into another arrays.I create bi-directional pipes between every child and parent.For example: A bidirectional pipe between P1 and P2, a bi-directional pipe between P1 and P3, and so on.Then I create 4 children process by fork system call.After, I write up half of matrix A and left half of matrix B in write-end of pipe of parent for children process P2 ; then, up half of matrix A and right half of matrix B in write-end of pipe of parent for children process P3; then , down half of matrix A and left half of matrix B in write-end of pipe of parent for children process P4 and down half of matrix A and right half of matrix B in write-end of pipe of parent for children process P5.In each children process,I calculate one quarter of matrix multiplication of matrix A and matrix B.Each children write own results in bi-directional pipes. While these are doing,parent process P1 catch the SIGCHLD signal and perform a synchronous wait for each of its children. P1 blocks its execution until all of its children have completed their calculations. This is a synchronization barrier and I apply this in my program.After all children have completed their calculations,parent process P1 read each outputs from bi-directional pipes and forms multiplication matrix C.And finally,parent process P1 calculates all singular values of C and prints them on screen.

Nevra Gürses
161044071

Functions that I use in my code for solving problem:

char* readForP1(char* inputFile,int n) : This function reads (2^n)x(2^n) character from given input files.

int * convertAsciiInt(char *buffer,int bytes): This function converts each read character to ASCII code integer equivalent.

ssize_t readFile(int fd, void *buffer, size_t byteCount): This function reads given byte character from given file according to file descriptor.

int convert2D(int array[],int row,int column,int startPoint):** This function converts 1D array to 2D array.

int* divide(int* total,int size,char pos): This function divides given matrix into 4 part.

int ** multiplication(int matrix1,int row1,int column1,int** matrix2,int column2):** This function makes matrix multiplication of given 2 matrix.

int createProductMatrix(int r,int first[][r],int second[][r],int third[][r],int fourth[][r]):**This function forms a matrix from given 4 quarter of it.

void sigChildHandler(int sigNo): SIGCHLD signal handler.

void sigIntHandler(int sigNo): SIGINT signal handler.

I create fork for children processes and bi-directional pipes in main function.

NOTE: Below functions that I use in my code for calculating singular value calculation stage.For this stage I use ,from "Numerical Recipes in C" (Cambridge Univ. Press) by W.H. Press, S.A. Teukolsky, W.T. Vetterling,and B.P. Flannery resource that is in link: http://cacs.usc.edu/education/phys516/src/TB/svdcmp.c

void svdcmp(double **a, int m, int n, double w[], double **v);

double pythag(double a, double b);

void free_dvector(double *v, int nl, int nh);

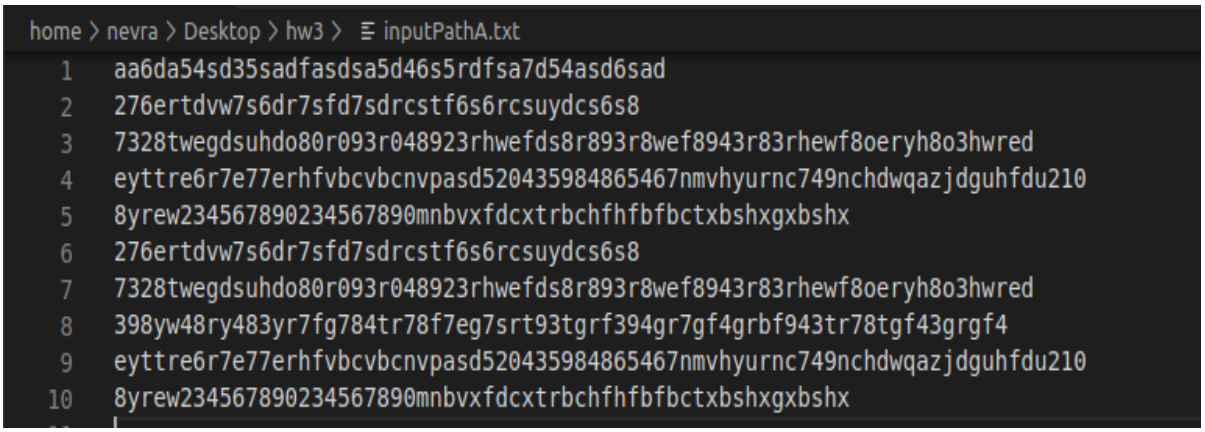
double *dvector(int nl, int nh);

double **dmatrix(int nrl, int nrh, int ncl, int nch);

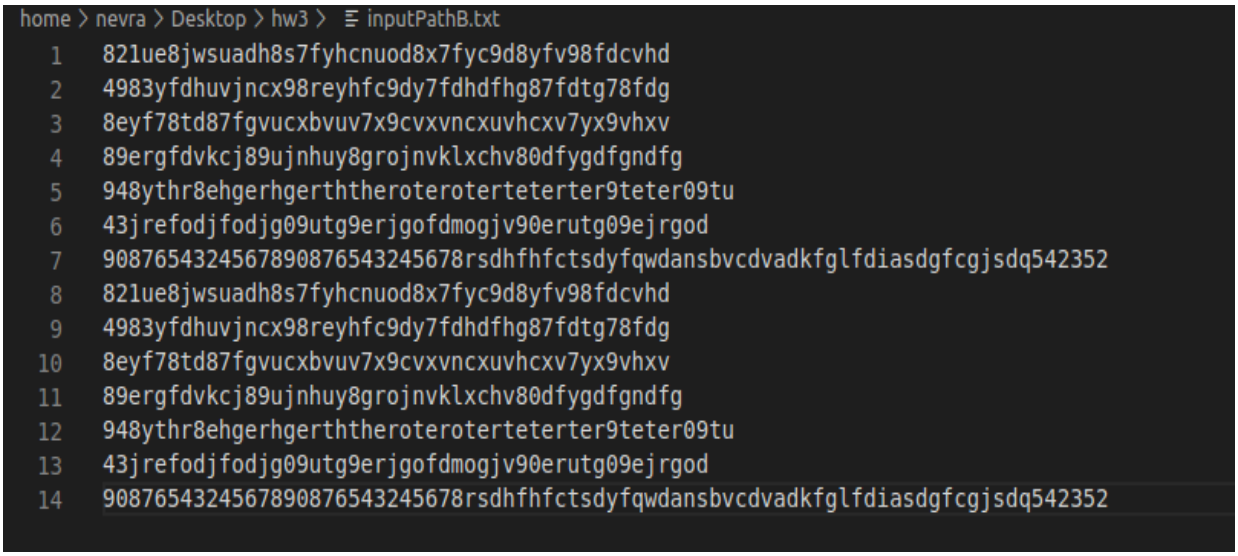
Nevra Gürses
161044071

3 RUNNING RESULTS:

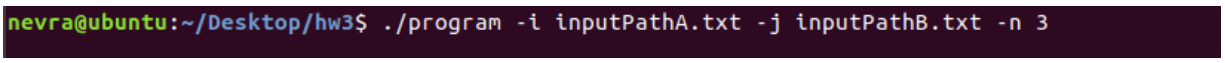
Input Files:



inputPathA.txt



inputPathB.txt



Commandline argument

Nevra Gürses
161044071

When n=3 in commandline arguments, Output is:

```
nevra@ubuntu:~/Desktop/hw3$ ./program -i inputPathA.txt -j inputPathB.txt -n 3

~~~~~~~~ MATRIX A ~~~~~~~~
97  97  54  100  97  53  52  115
100  51  53  115  97  100  102  97
115  100  115  97  53  100  52  54
115  53  114  100  102  115  97  55
100  53  52  97  115  100  54  115
97  100  10  50  55  54  101  114
116  100  118  119  55  115  54  100
114  55  115  102  100  55  115  100

~~~~~~~~ MATRIX B ~~~~~~~~
56  50  49  117  101  56  106  119
115  117  97  100  104  56  115  55
102  121  104  99  110  117  111  100
56  120  55  102  121  99  57  100
56  121  102  118  57  56  102  100
99  118  104  100  10  52  57  56
51  121  102  100  104  117  118  106
110  99  120  57  56  114  101  121

~~~~~IN SIGCHLD HANDLER~~~~~
Process ID of terminated child is:5891

~~~~~IN SIGCHLD HANDLER~~~~~
Process ID of terminated child is:5893

~~~~~IN SIGCHLD HANDLER~~~~~
Process ID of terminated child is:5892

~~~~~IN SIGCHLD HANDLER~~~~~
Process ID of terminated child is:5890

~~~~~~~~ PRODUCT MATRIX C ~~~~~~~~
53676  70401  59788  65096  55832  54464  63797  64373
54515  76662  64022  70952  57718  59666  66330  69354
56562  72856  60220  69266  58855  55506  64680  63331
57857  80839  66990  76612  61899  61957  70599  71577
54175  72767  62222  67567  52680  55323  63413  66454
46869  60294  53451  55927  47910  47853  57862  56067
64915  82717  69279  76482  64996  64461  72314  72768
58061  80695  68141  75127  66036  66332  73993  75661

~~~~~~~~ All the Singular Values of Product Matrix C ~~~~~~~~
517851.804 7046.323 6015.828 5068.427 2698.800 1171.040 4.265 563.197
nevra@ubuntu:~/Desktop/hw3$
```

Nevra Gürses
161044071

When n=2 in commandline arguments,Output is:

```
nevra@ubuntu:~/Desktop/hw3$ ./program -i inputPathA.txt -j inputPathB.txt -n 2


~~~~~~~~ MATRIX A ~~~~~~~~
97  97  54  100
97  53  52  115
100  51  53  115
97  100  102  97

~~~~~~~~ MATRIX B ~~~~~~~~
56  50  49  117
101  56  106  119
115  117  97  100
104  56  115  55

~~~~~IN SIGCHLD HANDLER~~~~~
Process ID of terminated child is:5868

~~~~~IN SIGCHLD HANDLER~~~~~
Process ID of terminated child is:5870

~~~~~IN SIGCHLD HANDLER~~~~~
Process ID of terminated child is:5869

~~~~~IN SIGCHLD HANDLER~~~~~
Process ID of terminated child is:5867

~~~~~~~~ PRODUCT MATRIX C ~~~~~~~~
31839  22200  31773  33792
28725  20342  28640  29181
28806  20497  28672  29394
37350  27816  36402  38784

~~~~~~~~ All the Singular Values of Product Matrix C ~~~~~~~~
120415.237 7.611 948.437 1420.839
nevra@ubuntu:~/Desktop/hw3$
```

When n=1 in commandline arguments,Output is:

```
nevra@ubuntu:~/Desktop/hw3$ ./program -i inputPathA.txt -j inputPathB.txt -n 1

~~~~~~~~ MATRIX A ~~~~~~~~
97  97
54  100

~~~~~~~~ MATRIX B ~~~~~~~~
56  50
49  117

~~~~~IN SIGCHLD HANDLER~~~~~
Process ID of terminated child is:5915

~~~~~IN SIGCHLD HANDLER~~~~~
Process ID of terminated child is:5914

~~~~~IN SIGCHLD HANDLER~~~~~
Process ID of terminated child is:5912

~~~~~IN SIGCHLD HANDLER~~~~~
Process ID of terminated child is:5913

~~~~~~~~ PRODUCT MATRIX C ~~~~~~~~
10185  16199
7924  14400

~~~~~~~~ All the Singular Values of Product Matrix C ~~~~~~~~
725.900 25214.374
nevra@ubuntu:~/Desktop/hw3$
```

Nevra Gürses
161044071

When n = 4 in commandline arguments , Output is :

```
nevra@ubuntu:~/Desktop/hw3$ ./program -i inputPathA.txt -j inputPathB.txt -n 4


~~~~~~~~ MATRIX A ~~~~~~~~
97  97  54  100  97  53  52  115  100  51  53  115  97  100  102  97
115  100  115  97  53  100  52  54  115  53  114  100  102  115  97  55
100  53  52  97  115  100  54  115  97  100  10  50  55  54  101  114
116  100  118  119  55  115  54  100  114  55  115  102  100  55  115  100
114  99  115  116  102  54  115  54  114  99  115  117  121  100  99  115
54  115  56  10  55  51  50  56  116  119  101  103  100  115  117  104
100  111  56  48  114  48  57  51  114  48  52  56  57  50  51  114
104  119  101  102  100  115  56  114  56  57  51  114  56  119  101  102
56  57  52  51  114  56  51  114  104  101  119  102  56  111  101  114
121  104  56  111  51  104  119  114  101  100  10  101  121  116  116  114
101  54  114  55  101  55  55  101  114  104  102  118  98  99  118  98
99  110  118  112  97  115  100  53  50  48  52  51  53  57  56  52
56  54  53  52  54  55  110  109  118  104  121  117  114  110  99  55
52  57  110  99  104  100  119  113  97  122  106  100  103  117  104  102
100  117  50  49  48  10  56  121  114  101  119  50  51  52  53  54
55  56  57  48  50  51  52  53  54  55  56  57  48  109  110  98

~~~~~~~~ MATRIX B ~~~~~~~~
56  50  49  117  101  56  106  119  115  117  97  100  104  56  115  55
102  121  104  99  110  117  111  100  56  120  55  102  121  99  57  100
56  121  102  118  57  56  102  100  99  118  104  100  10  52  57  56
51  121  102  100  104  117  118  106  110  99  120  57  56  114  101  121
104  102  99  57  100  121  55  102  100  104  100  102  104  103  56  55
102  100  116  103  55  56  102  100  103  10  56  101  121  102  55  56
116  100  56  55  102  103  118  117  99  120  98  118  117  118  55  120
57  99  118  120  118  110  99  120  117  118  104  99  120  118  55  121
120  57  118  104  120  118  10  56  57  101  114  103  102  100  118  107
99  106  56  57  117  106  110  104  117  121  56  103  114  111  106  110
118  107  108  120  99  104  118  56  48  100  102  121  103  100  102  103
110  100  102  103  10  57  52  56  121  116  104  114  56  101  104  103
101  114  104  103  101  114  116  104  116  104  101  114  111  116  101  114
111  116  101  114  116  101  116  101  114  116  101  114  57  116  101  116
101  114  48  57  116  117  10  52  51  106  114  101  102  111  100  106
102  111  100  106  103  48  57  117  116  103  57  101  114  106  103  111
```

```
~~~~~IN SIGCHLD HANDLER~~~~~
Process ID of terminated child is:5850

~~~~~IN SIGCHLD HANDLER~~~~~
Process ID of terminated child is:5851

~~~~~IN SIGCHLD HANDLER~~~~~
Process ID of terminated child is:5849

~~~~~IN SIGCHLD HANDLER~~~~~
Process ID of terminated child is:5848

~~~~~~~~ PRODUCT MATRIX C ~~~~~~~~
128577  140330  129676  133971  133018  131237  114864  129474  133927  147091  130760  141095  130192  141378  122054  136468
134822  145956  135219  143222  134580  133214  125972  131075  134543  148675  136083  148305  129973  142197  128073  137323
115442  126858  115837  118208  125167  118738  103417  123486  125557  129938  115914  127064  124548  129385  109634  121222
140722  155810  144698  152268  143591  140650  131395  141537  144233  157198  143815  156215  142781  152317  134735  147101
156343  168722  150904  157536  157263  154702  143390  153960  157847  176680  155093  170384  152208  165866  146911  161251
132553  135810  121904  125848  128945  125979  108614  118429  122136  141571  119550  140504  128126  136099  120125  132673
107591  111708  105410  107956  110585  105429  92126  107297  106017  119468  102298  116134  110499  112096  98148  106473
134335  151863  137833  143210  136708  133753  127595  140453  143718  151903  134798  149420  135260  147439  124066  139545
131992  138769  127735  129804  132416  129459  111247  124174  129269  144170  126622  142004  130308  140701  120776  134804
145146  158284  141176  148195  151502  145728  135513  151312  154982  163405  143340  159344  150694  160846  136958  155765
140397  150556  137163  143354  141302  138360  122842  136319  142882  159030  140728  155011  137299  149021  132950  143790
110661  126143  113111  116921  114043  112583  112920  119732  117860  124677  112271  123508  112677  120280  99527  112394
135400  140382  127786  131751  133432  133970  118914  125135  131191  147611  131581  146141  132220  144244  123532  141114
152720  166776  149335  151689  153413  151920  140362  151000  156067  167708  149905  166263  150006  165667  138298  158649
107243  113430  106052  112418  116927  112511  101164  106744  104811  126532  106160  119013  113461  115335  101702  114663
96743  105012  91656  96198  98694  93862  83919  93836  96056  106548  93426  104689  94240  103474  89787  100005

~~~~~~~~ All the Singular Values of Product Matrix C ~~~~~~~~
2123402.817 27158.056 20324.142 14075.148 12859.482 9826.703 7851.241 6433.002 4840.527 3964.243 2179.429 1881.678 1717.944 991.182 3.339 526.803
nevra@ubuntu:~/Desktop/hw3$
```

NOTE: My program is working true when commandline argument n <= 8