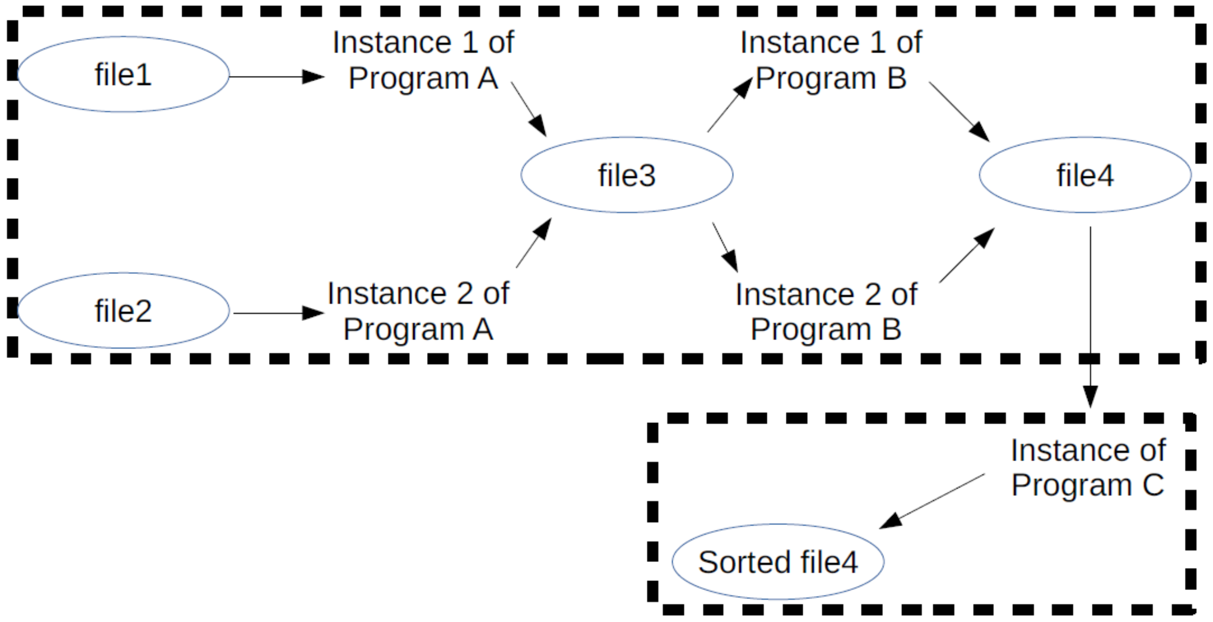
Gebze Technical University Computer Engineering

CSE 344 - 2019 Spring

HOMEWORK 1 REPORT

ELİF AKGÜN
1801042251

This assignment contains two parts and I did first part.



Part I

There are 2 programs(program A and program B), and 4 processes running in parallel because each process executed twice in arbitrary order.

Program A will receive 3 command line arguments:

```
./programA -i inputPathA -o outputPathA -t time
```

Both process of program A open and read different inputs(inputPathA1, inputPathA2) on read-only mode. There processes read 32 bytes from indicated file. Then for every 32 bytes that it reads, it'll convert them to 16 complex numbers. For example "bc" equals 9899 by their ASCII codes. So the complex number becomes "98+i99". After convertion, these processes writes these 16-complex-numbers to outputPathA file at the first available row. These prcesses read different files but write same file. If a row contains nothing or just '\n' char, it is empty. After writing a line, the program will call the sleep system call and go to sleep for time milliseconds.

Program B will also receive 3 commandline arguments:

```
./programB -i inputPathB -o outputPathB -t time
```

The input file of program B is the output file of program A. Processes of Program B read and write same files. When process B read a line then we expect that removing this line from it. It will open inputPathB, and then, it will read a random line from the file. If it is empty, it searchs non-empty line. When a non-empty line is found, it'll read those 16 complex numbers, delete that line by overwriting it with a '\n' and then calculate the discrete Fourier transform (using the FFT algorithm) of those 16 numbers, and write the calculation output to the file denoted by outputPathB as comma separated 16 complex numbers at the first available empty line. Then it'll sleep for time milliseconds. If there is no filled line by program A yet, then it'll sleep for time milliseconds, and try again.

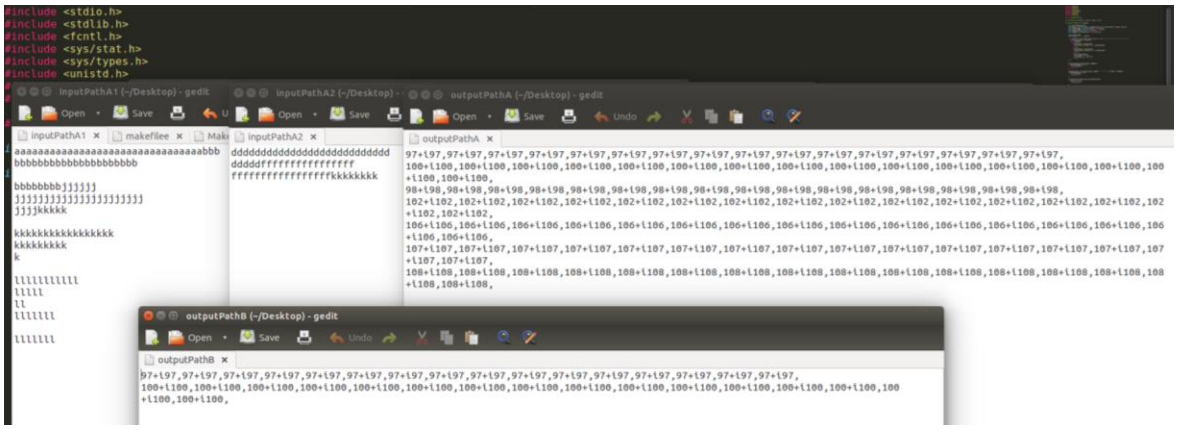
In process A, I created two processes by running them from two different terminals. I accessed the files by parsing command line arguments with getopt (). I can read my inputPahtA1 and inputPathA2 input files with two processes. By dividing the file I read into 32 bits, and I obtained 16 complex numbers. I ended the processes according to whether the number of bits remaining is less than 32. I created char *** arrays with column number 1 row number 16 and kept complex numbers in this array. I printed each one to the outputPathA file. I always printed at the end of the file, I could not check the empty line. Processes sleep for the specified time after writing each line. I used fcntl () and lock to prevent processes from running on the same file. When the processes are done, they close the related files and exit.

In process B, I created two processes B by running from two different terminals. I accessed the files by parsing command line arguments with getopt (). I can read the outputPathA file, the inputPathB file, with two different B processes. I could not delete the lines that I read. In both processes, I chose a random line from inputPathB and printed it to outputPathB by using rand() function. I checked the empty state of the selected random line. When it is empty, it looks at the next lines and returns to the beginning if the file reaches the end. I used fcntl () and lock to prevent processes from reading / writing from the file at the same time. If nothing has been written to the outputPathA file, they will sleep until they are written.

In totaly the program works with four different terminals.

A test for the program:

Input files and output files:



```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
```

inputPathA1 (~/Desktop) - gedit

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaabbb
bbbbbbbbbbbbbbbbbbbb

bbbbbbbbjjjjjj
jjjjjjjjjjjjjjjjjjjjjj
jjjjkkkkk

kkkkkkkkkkkkkkkkk
kkkkkkkkk
k

llllllllllll
lllll
ll
lllllll

lllllll
```

inputPathA2 (~/Desktop) - gedit

```
ddddddddddddddddddddddddddd
dddddfffffffffffffffff
ffffffffffffffffffkkkkkkkk
```

outputPathA (~/Desktop) - gedit

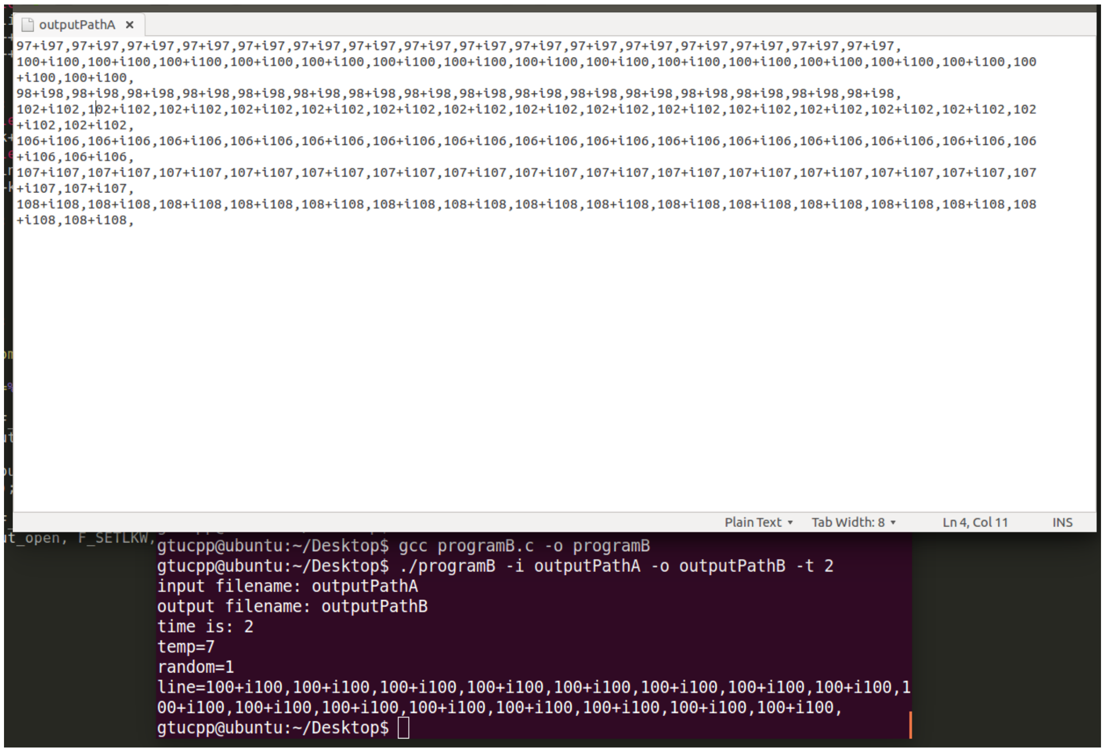
```
97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,
100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100
+i100,100+i100,
98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,
102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102
+i102,102+i102,
106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106
+i106,106+i106,
107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107
+i107,107+i107,
108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108
+i108,108+i108,
```

outputPathB (~/Desktop) - gedit

```
97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,
100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100
+i100,100+i100,
```

Selecting random line from inputPathB:

Note: I accepted the first line as the zeroth line.

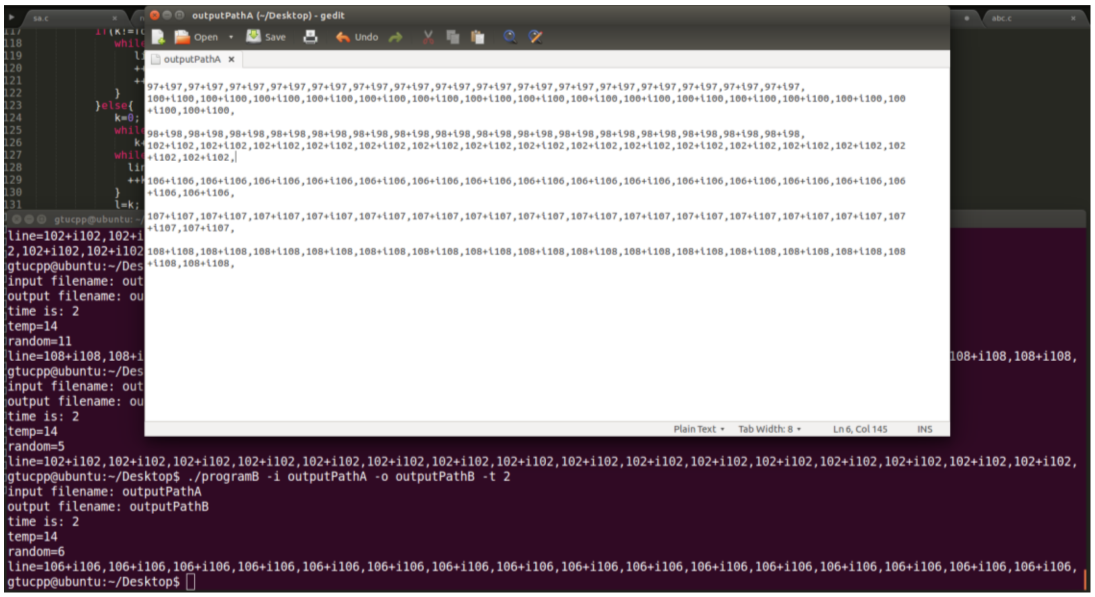


outputPathA

```
97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,
100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100
+i100,100+i100,
98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,
102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102
+i102,102+i102,
106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106
+i106,106+i106,
107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107
+i107,107+i107,
108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108
+i108,108+i108,
```

```
gtucpp@ubuntu:~/Desktop$ gcc programB.c -o programB
gtucpp@ubuntu:~/Desktop$ ./programB -i outputPathA -o outputPathB -t 2
input filename: outputPathA
output filename: outputPathB
time is: 2
temp=7
random=1
line=100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,1
00+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,
gtucpp@ubuntu:~/Desktop$
```

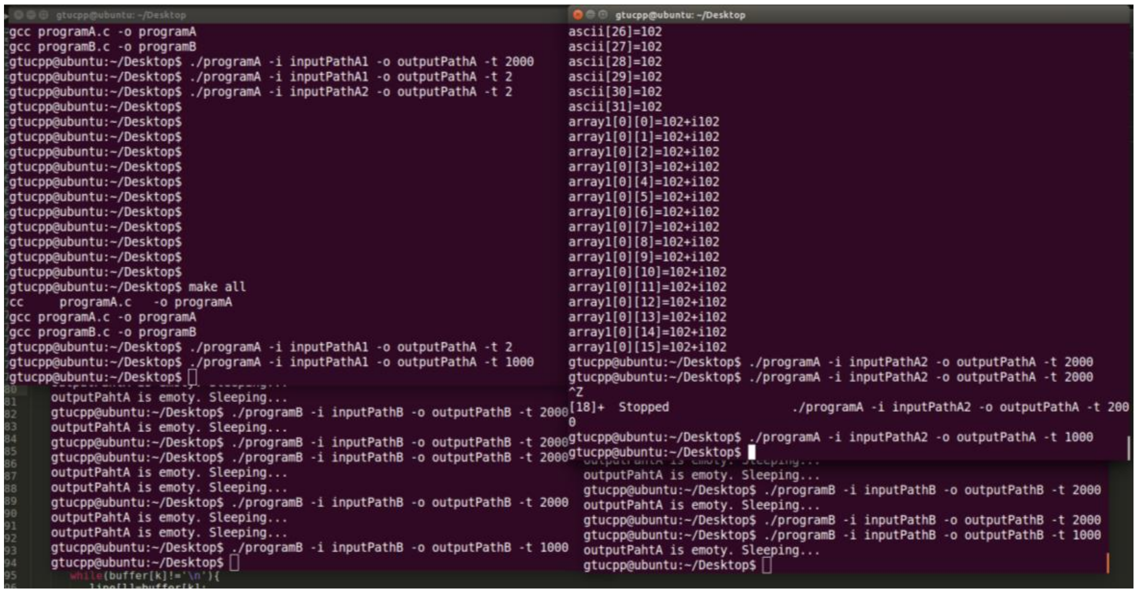
Selecting line from inputPathB randomly:



```
outputPathA (~/Desktop) - gedit

97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,97+i97,
100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100+i100,100
+i100,100+i100,

98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,98+i98,
102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102
+i102,102+i102,

106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106
+i106,106+i106,

107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107+i107,107
+i107,107+i107,

108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108+i108,108
+i108,108+i108,

Plain Text   Tab Width: 8   Ln 6, Col 145   INS
```

```
line=102+i102,102+i
2,102+i102,102+i102
gtucpp@ubuntu:~/Des
input filename: out
output filename: ou
time is: 2
temp=14
random=11
line=108+i108,108+i                                                                                    108+i108,108+i108,
gtucpp@ubuntu:~/Des
input filename: out
output filename: ou
time is: 2
temp=14
random=5
line=102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,102+i102,
gtucpp@ubuntu:~/Desktop$ ./programB -i outputPathA -o outputPathB -t 2
input filename: outputPathA
output filename: outputPathB
time is: 2
temp=14
random=6
line=106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,106+i106,
gtucpp@ubuntu:~/Desktop$
```

Program works with four different terminals:



```
gcc programA.c -o programA
gcc programB.c -o programB
gtucpp@ubuntu:~/Desktop$ ./programA -i inputPathA1 -o outputPathA -t 2000
gtucpp@ubuntu:~/Desktop$ ./programA -i inputPathA1 -o outputPathA -t 2
gtucpp@ubuntu:~/Desktop$ ./programA -i inputPathA2 -o outputPathA -t 2
gtucpp@ubuntu:~/Desktop$
gtucpp@ubuntu:~/Desktop$
gtucpp@ubuntu:~/Desktop$
gtucpp@ubuntu:~/Desktop$
gtucpp@ubuntu:~/Desktop$
gtucpp@ubuntu:~/Desktop$
gtucpp@ubuntu:~/Desktop$
gtucpp@ubuntu:~/Desktop$
gtucpp@ubuntu:~/Desktop$
gtucpp@ubuntu:~/Desktop$
gtucpp@ubuntu:~/Desktop$
gtucpp@ubuntu:~/Desktop$
gtucpp@ubuntu:~/Desktop$ make all
cc     programA.c   -o programA
gcc programA.c -o programA
gcc programB.c -o programB
gtucpp@ubuntu:~/Desktop$ ./programA -i inputPathA1 -o outputPathA -t 2
gtucpp@ubuntu:~/Desktop$ ./programA -i inputPathA1 -o outputPathA -t 1000
gtucpp@ubuntu:~/Desktop$
```

```
ascii[26]=102
ascii[27]=102
ascii[28]=102
ascii[29]=102
ascii[30]=102
ascii[31]=102
array1[0][0]=102+i102
array1[0][1]=102+i102
array1[0][2]=102+i102
array1[0][3]=102+i102
array1[0][4]=102+i102
array1[0][5]=102+i102
array1[0][6]=102+i102
array1[0][7]=102+i102
array1[0][8]=102+i102
array1[0][9]=102+i102
array1[0][10]=102+i102
array1[0][11]=102+i102
array1[0][12]=102+i102
array1[0][13]=102+i102
array1[0][14]=102+i102
array1[0][15]=102+i102
gtucpp@ubuntu:~/Desktop$ ./programA -i inputPathA2 -o outputPathA -t 2000
gtucpp@ubuntu:~/Desktop$ ./programA -i inputPathA2 -o outputPathA -t 2000
^Z
[18]+  Stopped                 ./programA -i inputPathA2 -o outputPathA -t 200
0
gtucpp@ubuntu:~/Desktop$ ./programA -i inputPathA2 -o outputPathA -t 1000
gtucpp@ubuntu:~/Desktop$
```

```
outputPahtA is emoty. Sleeping...
gtucpp@ubuntu:~/Desktop$ ./programB -i inputPathB -o outputPathB -t 2000
outputPahtA is emoty. Sleeping...
gtucpp@ubuntu:~/Desktop$ ./programB -i inputPathB -o outputPathB -t 2000
gtucpp@ubuntu:~/Desktop$ ./programB -i inputPathB -o outputPathB -t 2000
outputPahtA is emoty. Sleeping...
outputPahtA is emoty. Sleeping...
gtucpp@ubuntu:~/Desktop$ ./programB -i inputPathB -o outputPathB -t 2000
outputPahtA is emoty. Sleeping...
outputPahtA is emoty. Sleeping...
gtucpp@ubuntu:~/Desktop$ ./programB -i inputPathB -o outputPathB -t 1000
gtucpp@ubuntu:~/Desktop$
```

```
outputPahtA is emoty. Sleeping...
gtucpp@ubuntu:~/Desktop$ ./programB -i inputPathB -o outputPathB -t 2000
outputPahtA is emoty. Sleeping...
gtucpp@ubuntu:~/Desktop$ ./programB -i inputPathB -o outputPathB -t 2000
gtucpp@ubuntu:~/Desktop$ ./programB -i inputPathB -o outputPathB -t 1000
outputPahtA is emoty. Sleeping...
gtucpp@ubuntu:~/Desktop$
```