Assignment Report

Real-time Operating System - 48450

Student Name: Dung Quoc Thai

Student ID: 11990405

Table of Contents

I. Introduction

The aim of this assignment is to get the student familiar with the concept of multi-threads and pipe-line. As this assignment is to be build using the programming language C, the core technical knowledge that shall be require are C programming and Google searching.

II. Theory of operation

The technical knowledge that are given for this assignment are basic understanding of C and how to apply this knowledge in OS. Practice programs for area such as multi-threads and pipe-line concept are provided in lab 3. This was used by the writer of this report (me) and is very much appreciated.

III. Operating condition

The assignment demanded the student to produce 2 programs.

Program 1 creates three threads for reading data from one file and writing the data to another file through the pip-line concept. The program must run the threads iteratively in sequential order. The total run time of the program shall be stores in shared memory to be read by program 2.

Program 2 reads information from shared memory used in program 1 and output this information to the user.

IV. Implementation

1. Method

The strategies used for this project is similar to many other programming projects.

The first and most important step is to clearly understand the project. This was completed by carefully reading through the assignment notes. The second step is to understand logically how the program would work. This was completed by looking at key points and listing down how the program would run logically, see figure 1.

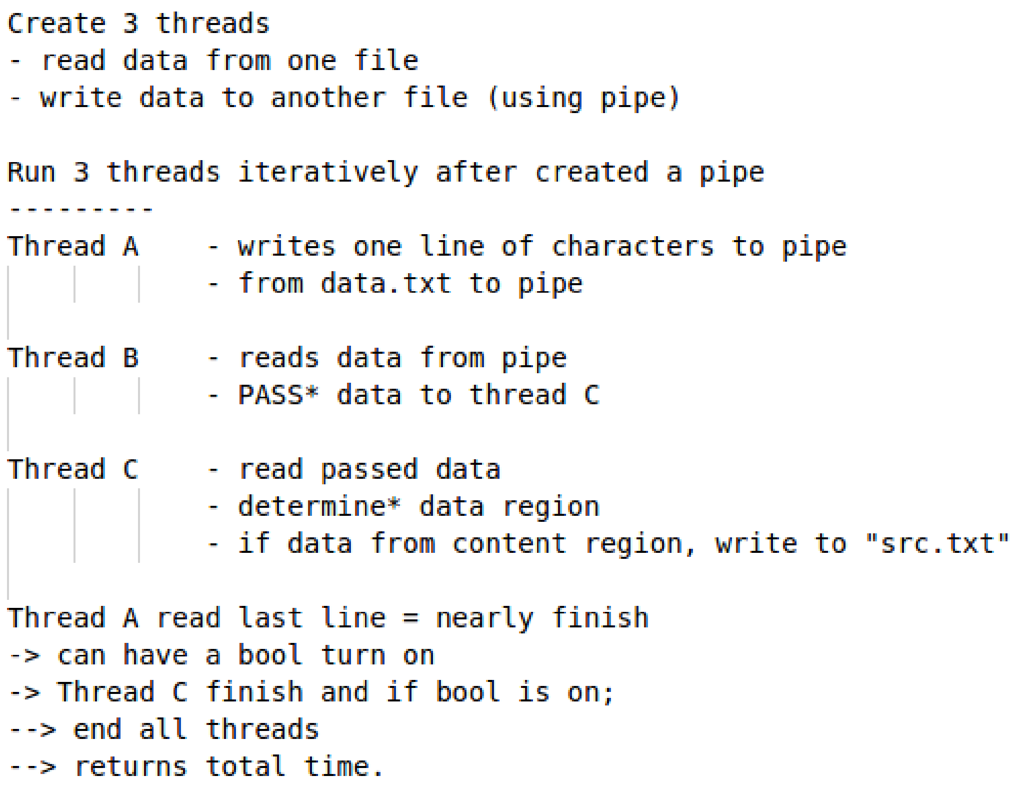
After the logic are listed, pseudo code was written, and the programming process was tackled in parts, following the listed logic (figure 1).

Strategy to solve problems during the coding period included; Google searching, consulting classmates, move away from the work space for a period of time to rethink about life, never give up.

After coding and debugging, the final step was to clean up all debugged output, nasty comments and recommended the project.

1. Flow chart

The flow chart on my personal take about this assignment is shown below.



```
Create 3 threads
- read data from one file
- write data to another file (using pipe)

Run 3 threads iteratively after created a pipe
---------
Thread A    - writes one line of characters to pipe
            - from data.txt to pipe

Thread B    - reads data from pipe
            - PASS* data to thread C

Thread C    - read passed data
            - determine* data region
            - if data from content region, write to "src.txt"

Thread A read last line = nearly finish
-> can have a bool turn on
-> Thread C finish and if bool is on;
--> end all threads
--> returns total time.
```

Figure 1: Logic flows

V. Experiments

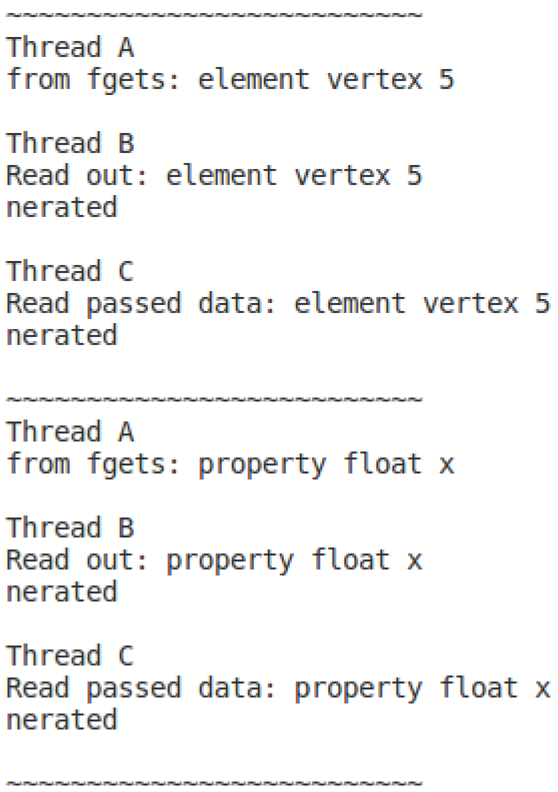
1. Hypothesis

The experiment question tasked the student (me) to hypothesis and test the outcome of the program when mutex is not applied.

The hypothesis would be that there will be an inconsistent in the characters that are written into pipe since the thread would be running simultaneously or concurrently instead of sequentially.

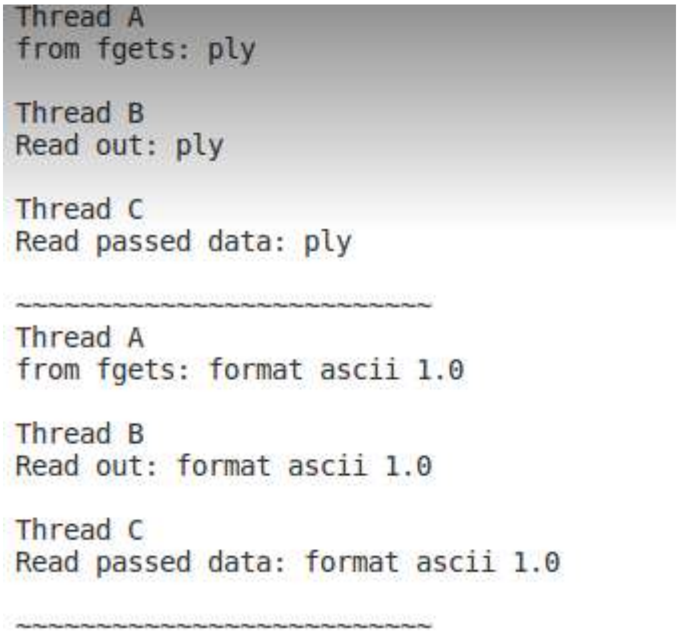
2. Results

The figures below provide output when mutex is not applied (figure 2) and when mutex is applied (figure 3).



```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Thread A
from fgets: element vertex 5

Thread B
Read out: element vertex 5
nerated

Thread C
Read passed data: element vertex 5
nerated

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Thread A
from fgets: property float x

Thread B
Read out: property float x
nerated

Thread C
Read passed data: property float x
nerated

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Figure 2: No mutex



```
Thread A
from fgets: ply

Thread B
Read out: ply

Thread C
Read passed data: ply

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Thread A
from fgets: format ascii 1.0

Thread B
Read out: format ascii 1.0

Thread C
Read passed data: format ascii 1.0

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Figure 3: Mutex applied

VI. Conclusion on result analysis

In conclusion, the result from outputs in figure 2 and figure 3 has confirm the hypothesis to be true. We can conclude that the use of mutex is important in this assignment for the outcome to be correct.