

# Homework 2

☰ Student Name	Muhammed Oğuz
☰ Student Number	1801042634

## Content

Content

Problem Understanding and Approach

File Structure

`makefile`

`processP.c` `processP.h`

`processR.c` `processR.h`

Error Enum Type

Coding Approach

Parse File

Write file and reparse

Valgrind Result

Missing Parts

## Problem Understanding and Approach

This homework has some critical points to consider.

First of all, I check class materials and text book for SIGINT and multi processors.

I wrote some test programs to recognize the basics. For every basic step, it should be avoided to make big mistake since it could be cause for computer crashes.

(Which I encountered while doing this homework)

## File Structure

`makefile`

I have the very same makefile from homework1.

I add some configurations to hanlde more easily.

```

You, 2 hours ago | 1 author (You)
1  CC = gcc
2  CFLAGS =-Wextra -Wall
3  HEADER = processP.h
4  SRC = main.c processP.c
5  HELPER = processR
6  HW = hw2
7  OUT = processP
8
9  muo: $(SRC) $(HEADER)
10     $(CC) $(HELPER).c $(CFLAGS) -o $(HELPER)
11     $(CC) $(SRC) $(CFLAGS) -o $(OUT)
12
13
14  debug: $(SRC) $(HEADER)
15     $(CC) $(SRC) $(CFLAGS) -g -o $(OUT)
16
17  memory:
18     valgrind ./${OUT}

```

**debug:** This option will compile program with `-g` flag. It will allows you to debug. For entering debug mode, you have to run your program with `gdb`.

**memory:** This option will call `valgrind`. But since, We have to provide arguments, It has no effect for this homework. For valgrind, you have to run command without makefile.

**processP.c** **processP.h**

Those files keeps parent process source code and declarations. I keep the functions as simple as since due to clean code.

Also header file contains well documented explanations about functions.

**processR.c** **processR.h**

Those files contains child process source code. This program's executable file is calling from `procesP` and execute coresponding.

It will append desired file to output and calculates the matrix.

## Error Enum Type

```
You, 27 minutes ago | 1 author (You)
12 typedef enum
13 {
14     INVALID_MALLOC = 1,
15     FILE_OPEN_ERROR,
16     FILE_READ_ERROR,
17     FILE_WRITE_ERROR,
18     FILE_LOCK_ERROR,
19     FILE_UNLOCK_ERROR,
20     FILE_CLOSE_ERROR,
21     FILE_SEEK_ERROR,
22     INVALID_ARGUMENTS,
23     INVALID_EXECVE,
24     INVALID_FORK,
25     INVALID_WAIT,
26
27 }
```

This enum type holds those provided enums and have function to well formatted way print to screen with using `write()` function.

Also has a global variable to set and use it.

## Coding Approach

### Parse File

I created an ASCII file and check if I am able to read it without any struggling. It will convert any ascii char to their corresponding ascii number values.

With those approach, I was able to parse every ascii char to respective number for keeping coordinates.

### Write file and reparse

After child processes calculate the matrixes, they will write to file and rereaded from parent and it will check for the closest matrixes.

## Valgrind Result

Valgrind has a option to track child processes memory leaks. I checked with it too and it says eveything seems fine.

## Missing Parts

Unfrotunaltly, program calculates matrix but not shows the difference value.

It sends env to process, process calculates but last difference does not shown