

## Assignment 5 - Mitigating data loss

5/1/2025

40 Points Possible

Attempt 1



In Progress

NEXT UP: Submit Assignment



Add Comment

Unlimited Attempts Allowed

5/7/2025

## ▼ Details

The ideal solutions to overcome the Data loss in Assignment 4 are to implement:

1. One program with subprocess - "Scenario\_2\_One\_program\_with\_a\_sub\_process"
2. One process with 2 threads - "Scenario\_3\_One\_process\_with\_two\_threads"
3. Two processes - "Scenario\_4\_two\_processes"

The objective of this Assignment is to experiment with concurrency solutions for overcoming data losses.

Things needed for this assignment.

1. Virtual application file ([Assignment3.ova](https://unt.instructure.com/courses/117562/files/31279224?wrap=1) (<https://unt.instructure.com/courses/117562/files/31279224?wrap=1>). [↓](https://unt.instructure.com/courses/117562/files/31279224/download?download_frd=1) ([https://unt.instructure.com/courses/117562/files/31279224/download?download\\_frd=1](https://unt.instructure.com/courses/117562/files/31279224/download?download_frd=1)). There is no need to download this file again if you already have it from assignment 4.
2. Virtual machine manager (preferably Oracle VM VirtualBox. It is assumed that you have already downloaded and installed Oracle VM VirtualBox. If not, please download and install Oracle VM VirtualBox from <https://www.virtualbox.org/wiki/Downloads> (<https://www.virtualbox.org/wiki/Downloads>)).

**Task 1: Technique 1: Accept the data stream using the main program and process them using a sub-process.**

Navigate to the Scenario 2 code folder from the terminal using "`cd Desktop/'Scenario 2 - One program with a sub process'/Code`".

Then compile the following scripts:

- "`gcc -o calling_python_with_process calling_python_with_process.c -I/usr/include/python3.8 -lpython3.8`"
- "`gcc -o calling_python_with_process_wait calling_python_with_process_wait.c -I/usr/include/python3.8 -lpython3.8`" (Any warnings after compiling the scripts can be ignored)

Next, run these 2 scripts in different terminal windows:

- "`./calling_python_with_process`"
- "`./calling_python_with_process_wait`"

Once the program is started in both terminals, send the simulated data by running "`./a.out`" as per Assignment 3 (Please refer to **Step 4: Setting up the ECUSimulator** from Assignment 4). This will process the data and the processed time information will be stored in "`time.txt`".

Finally, modify the "`plot.py`" file in the "Home" directory to use the "`time.txt`" files that were created and run the script to generate the new plots. Then, compare the plots to the ones you obtained in assignment 4--do you see any improvements?

(Modification1: Update the path to your new time file. Modification2: Update the logic in **batch\_process\_time** function to extract the time information of **Time between Start and End** from your time file).

**Task 2: Technique 2: Use one thread to accept the data and another to process them.**

Navigate to the Scenario3 folder from the terminal using "`cd Desktop/'Scenario 3- One process with two threads'/Code`".

Now compile the script of Scenario3 in the terminal with the following command:

- "`gcc -o test_threads_single_process test_threads_single_process.c -lpthread -I/usr/include/python3.8 -lpython3.8`" (Any warnings after compiling the scripts can be ignored)

Now run the compiled script using:

- `"/test_threads_single_process"`

Once the program is started simulate the data `"/a.out."`

This will populate the time information of the data processed in `"time.txt"`

Finally, modify the `"plot.py"` in the "Home" folder to this file to generate the plot and run the script to generate the new plots. Then, compare the plots to the ones you obtained in task 1--do you see any improvements?

(Modification1: Update the path to your new time file. Modification2: Update the logic in `batch_process_time` function to extract the time information of **Time between Start and End** from your time file)

### Task 3: Technique 3: Use one process to accept the data and another to process them.

Navigate to the Sceanrio4 folder from the terminal using `"cd Desktop/'Scenario 4 - two processes'/Code"`

Once you are in this folder compile the following scripts

- `"gcc -o first_program first_program.c -lpthread -I/usr/include/python3.8 -lpython3.8"`
- `"gcc second_program.c -o second_program -I/usr/include/python3.8 -lpython3.8"`(Any warnings after compiling the scripts can be ignored)

Next, run the scripts in different terminal windows.

- `"/first_program"`
- `"/second_program"`

Then send the simulated data using `"/a.out."`

Once the data is processed the information on time taken for processing will be stored in `"time_program_1.txt"` and `"time_program_2.txt"`. (Note: `time_program_1.txt` will be generated once you terminate both the programs with `ctrl+c` in the terminal)

Now modify the `"plot.py"` in the "Home" folder to this file to generate the plot and run the script to generate one plot for each program. Then, compare the plots to the ones you obtained in task 2--do you see any improvements?

(Modification1: Update the path to your new time file. Modification2: Update the logic in `batch_process_time` function to extract the time information of **Time between Start and End** from your time file). Generate separate plots for first and second program.

### Task 4: Reflection

1. Which scenario was best and why?
2. What did you learn from this assignment?

#### Submission:

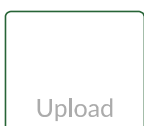
- Implement all 3 scenarios as per the description above and submit a report that addresses the following questions for **each scenario**.
  - Describe how the data is being processed. Add your plots to the report along with the screenshots of the code you modified in `plot.py`.
  - Compare the results with the data loss obtained from assignment 4 and describe your findings in one or two paragraphs.
- Reflection

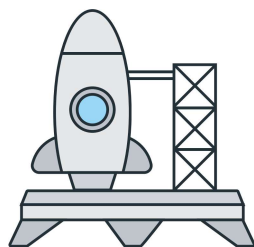
## Submission Details

Upload your **report in pdf format** and name the file using the structure:

`CSCE_5214_Assignment_5_Report_lastname_firstname.pdf` (for example, `CSCE_5214_Assignment_5_Report_doe_john.pdf`).

#### Choose a submission type





Choose a file to upload

File permitted: ZIP, RAR, PDF, DOCX

or

 Canvas Files

☐ I agree to the tool's [End-User License Agreement \(https://api.turnitin.com/api/lti/1p0/user/static\\_eula\)](https://api.turnitin.com/api/lti/1p0/user/static_eula)  
This assignment submission is my own, original work

\*



(<https://unt.instructure.com/courses/117562/modules/items/7909192>)



Assignment  
(<https://unt.instructure.com/courses/117562/modules/items/763>)