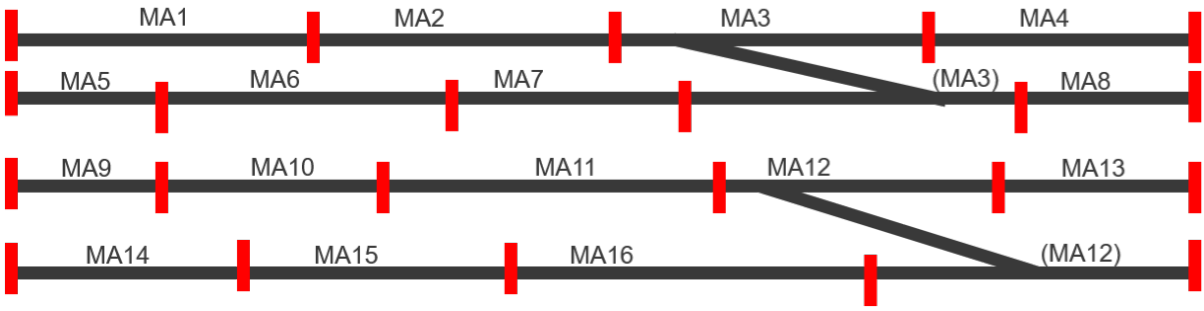
Operating Systems Project

Movement Authority ERTMS/ETCS LV 1 LV 2



Alberto Charabati

7031859

01.2023

ALBERTO.CHARABATI@STUD.UNIFI.IT

Project Specifics

Documentation

About the project

This project will be a mock of the Movement Authority ERTMS/ETCS LV 1 LV 2, it will aim to create a simulation of the way various trains cross track intersections and must be synchronised such that there will not be any collisions, allowing only one train to cross any railway segment at a time.

The project will be divided into 5 main elements:

-PADRE_TRENI

-TRENO

-REGISTRO

-RBC

-Signal

In addition, the program will be executable in two ways:

ETCS1: execution through shell.

ETCS2: execution through two different shells, the first directly by calling on ETCS2 and the latter by specifying ETCS2 RBC.

Project Elements

PADRE_TRENI: Process in charge of the creation of the 16 files representing the MAx segments. This files are set such that they have R/W access.
Each of the files will be marked by a 1 or a 0 indicating respectively whether they are occupied or not.
PROCESSI_TRENI: 5 processes that will be representing the different trains.
REGISTRO: Registry of the itineraries that each of the trains will follow.
RBC: Manages the AF_UNIX server socket. When called by executing the program in ETC2, the RBC gets the itineraries from REGISTRO and manages the MAx segments.
It creates a process for each request it receives.
Signal: Manages the SIGUSR1, SIGUSR2 signals that the processes use before terminating

Main

The input arguments are taken, and the program is executed accordingly (type of itinerary, type of execution, RBC).
If the program is run as ETCS1, it will be executed with PROCESSO_PADRE which will then create the 5 train processes (PROCESSI_TRENI) and PROCESSO_REGISTRO which will provide the itinerary to each of the trains.
Otherwise, if the program is run using the ETCS2, ETCS2 RBC settings, the program will be executed with RBC as a server socket, which will be the one to handle the itinerary from the registry for every train. The handling of the requests from the different process trains will be handled in parallel.

Makefile

A Makefile is provided for the assembling of the program. It sets the executable files ready and provides a "make clean" command to delete them accordingly when execution is over.

Execution Of The Program

- Run the Makefile with the “make” Command-line:
  - The bin and obj directories are created with the UNIX executable files.
  - “./run.sh” command to start the program:
  - Can take two parameters: -e, -m.
  - -e: Sets the ETC in which the program will be run; can be set to 1, 2. If no argument is specified will run on 1 on default.
  - -m: Sets the MAPPA in which the program will be run; can be set to 1, 2. If no argument is specified will run on 1 on default.
  - -h: Shows the arguments that can be parsed.

```
albertoch@Albertos-MacBook-Pro ProjOs % ./run.sh -h
Usage: run.sh [-e arg] [-m arg]
```

- When executing in ETC1 mode (./run.sh -m 1/2):

```
ETCS1 MAPPA1 RBC=0
REGISTRO process created
PADRE_TRENI process created
PADRE_TRENI Execution initialized.
PADRE_TRENI created process for TRENO 1
PADRE_TRENI created process for TRENO 2
PADRE_TRENI created process for TRENO 3
PADRE_TRENI created process for TRENO 4
PADRE_TRENI created process for TRENO 5
```

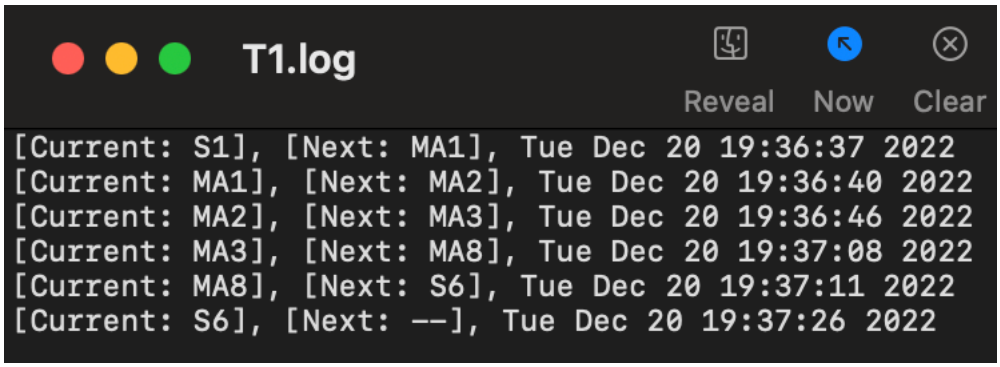
REGISTRO sends the itineraries directly to each TRENO process.

-When executing in ETC2 mode (./run.sh -e 2 -m 1/2):

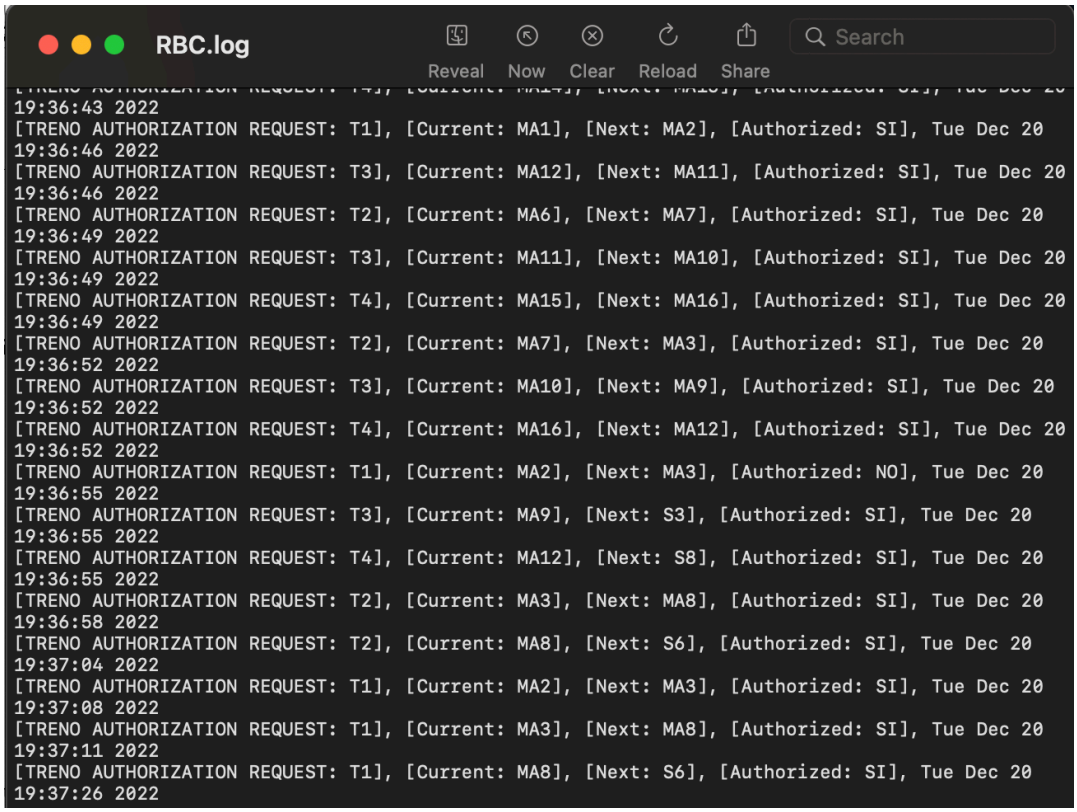
```
albertoch@Albertos-MBP ProjOs % ETCS2 MAPPA1 RBC=0
ETCS2 MAPPA1 RBC=1
```

LOGS

As the program is executed, a LOG is updated for each train (T1,..,T5). This log includes each step of the train until it reaches its destination, showing the current segment in each step, the next segment, and the date and time.
Ex:

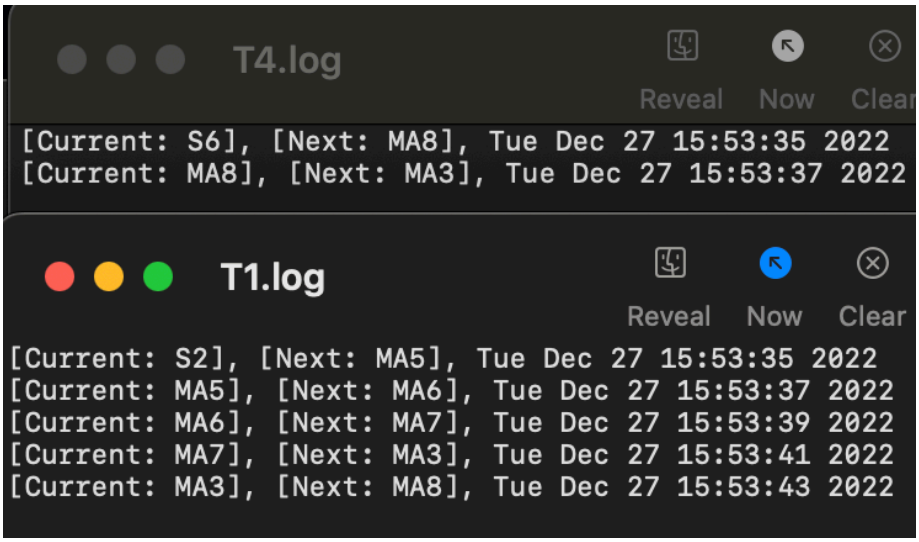


When the program is executed in ETC2, a RBC log is added showing the requests it gets from each of the trains, and wether these are accepted or not.
Ex:



Deadlocks

Aside the known errors which are handled or thrown inside the program, it may occur that a circular deadlock happens on the execution of the program. This might happen because given that the program simulates trains running on tracks, the possession-and-request condition must be allowed for the different processes (trains).
For example:
Train 1: currPos: MA3, nextPos: MA8.
Train 4: currPos: MA8, nextPos: MA3.
This example could theoretically occur on MAPPA 2, which would cause a deadlock.




```
[TRENO AUTHORIZATION REQUEST: T1], [Current: MA3], [Next: MA8], [Authorized: NO]
2022
[TRENO AUTHORIZATION REQUEST: T4], [Current: MA8], [Next: MA3], [Authorized: NO]
2022
```

Extra Elements

Elemento Facoltativo	Realizzato	Descrizione
Concurrent Read/Write management	SI	Segment Files are written in memory (isSegmentFree).
RBC - boe clash	Si	Checker that states match in current and next segments. (segmStatusChecker).
SIGUSR1 PADRE_TRENI, TRENO termination	Si	signalHandler
SIGUSR2 RBC termination	Si	signalHandler2

Example of how each TRENO process sends a SIGUSR1 signal when at the final station. When all TRENO processes have terminated, REGISTRO and PADRE_TRENI terminate.

```
TRENO 4 Execution terminated.
Sending SIGUSR1 to PADRE_TRENI, pid: 59538
TRENO 1 Execution terminated.
Sending SIGUSR1 to PADRE_TRENI, pid: 59538
TRENO process terminated.
TRENO 2 Current position: MA3, requesting permission to proceed to next position: MA8.
TRENO 3 Current position: MA9, requesting permission to proceed to next position: S3.
TRENO 3 Execution terminated.
Sending SIGUSR1 to PADRE_TRENI, pid: 59538
TRENO process terminated.
TRENO 2 Current position: MA8, requesting permission to proceed to next position: S6.
TRENO 2 Execution terminated.
Sending SIGUSR1 to PADRE_TRENI, pid: 59538
TRENO process terminated.
TRENO processes terminated.
REGISTRO, PADRE_TRENI: end of execution
```

When running in ETC 2 mode (-e 2), when all TRENO processes have terminated, a SIGUSR2 signal will be sent to terminate the RBC process, after all SIGUSR1 signals have been received by each TRENO process.

```
TRENO processes terminated.
Sending SIGUSR2 to RBC, pid: 59593
SIGUSR2 from Padre Treni to RBC, terminating RBC
Shared memory rbc_data removed.
RBC Server /tmp/rbc_server closed.
RBC Execution terminated.
REGISTRO, PADRE_TRENI: end of execution
```

System

Built, used and tested on latest to date (12.2022) Kali Linux 64 bit image running on a virtual machine, as well as on zsh + iTerm on an Intel Core i5 running on Ventura 13.1 (Darwin).