Operating Systems ICE – AA2324

Train Journey Manager

Overall goal. The aim of the project is to simulate the behaviour of a set of trains that travel from one station to another through various track segments. The journey of each train is to reach a specific station. The main constraint on train movement is that each track segment can only be occupied by one train at a time. Each train asks permission to access the next segment of the track during the journey.

Description of the system and its behavior. Figure 1 shows railway tracks. There are 16 segments, numbered MA1 to MA16. In the rest of the document, MA*x* refers to any of the 16 segments in Figure 1. Each segment is bounded by 2 balises, with the exception of the segments that contain interconnections: the latter are bounded by 4 balises, and are the MA3 and MA12 segments. In Figure 1, each balise is represented with a vertical red bar.

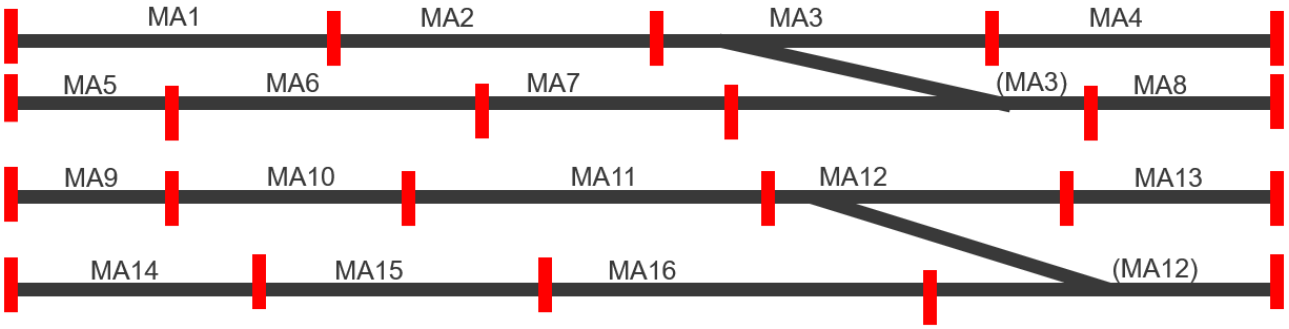

Figure 1 Tracks and track segments MAx considered.

8 stations are defined, at each end of each track, numbered S1 to S8 as shown in Figure 2 below.

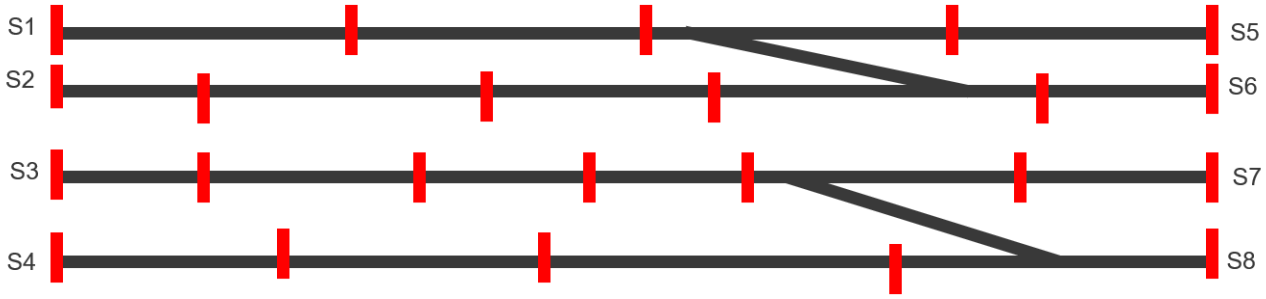

Figure 2 Stations numbered from S1 to S8.

Up to 5 trains, numbered from T1 to T5, want to use the tracks for their journeys. Each train has an associated journey profile / itinerary, i.e. a departure station, a set of MAx segments to traverse, and a destination station. There could be two possible situations, shown by MAP1 and MAP2 below.

Train	**Itinerary**
T1	S1, MA1, MA2, MA3, MA8, S6
T2	S2, MA5, MA6, MA7, MA3, MA8, S6
T3	S7, MA13, MA12, MA11, MA10, MA9, S3
T4	S4, MA14, MA15, MA16, MA12, S8
T5	--

Table 1: MAP1

Train	Itinerary
T1	S2, MA5, MA6, MA7, MA3, MA8, S6
T2	S3, MA9, MA10, MA11, MA12, S8
T3	S4, MA14, MA15, MA16, MA12, S8
T4	S6, MA8, MA3, MA2, MA1, S1
T5	S5, MA4, MA3, MA2, MA1, S1

Table 2 - MAP2

The following rules are defined for the railway system in question:

- Each train starts its own train mission from a station, and executes its train mission until it reaches the destination station. The itinerary, i.e. the sequence of segments MAx that each train will have to cross, is defined in MAP1 and MAP2.
- All trains start their journey at the same time; i.e., they leave at the same time from dept station.
- A train's journey ends when it reaches its destination station.
- MAP1 and MAP2 are stored in a file.
- Trains cannot deviate from the route specified in MAP1 and MAP2.
- A segment MAx can be crossed by only one train at a time; in other words, at any given time, each segment MAx can be occupied by at most one train.
- It takes 2 seconds to traverse any segment MAx.
- Each train shall, upon departure from the station or at the end of the crossing of any segment MAx, ask for permission to enter the next segment or to enter the destination station.
- Obviously, if two or more trains require access to the same segment MAx at the same time, only one train will be able to access (or rather, *occupy*) the segment; the other trains will have to wait til the the track segment gets freed.
- Any number of trains can reside in the destination stations (stations do not have the single access constraint that applies to the track segment).

Implementation Details - Assignment

You are required to develop a C program that implements the system described above, implementing the following requirements.

Requirements for starting the program. The route set to be used, i.e. MAP1 or MAP2, must be specified at the time of starting the program.

Example: shell# *./program_name MAP1*

Program Execution Requirements. The five trains are represented by 5 processes, which we will refer to in the following TRAIN_THR. The TRAIN_THR are the children of a single process that we will call CONTROLLER below.

The CONTROLLER is responsible for creating the shared memory containing the state of the 16 MAx segments, named MA1, MA2, etc. Each state is initialized with the character "0" (i.e., CONTROLLER writes a 0 inside memory upon creation). The pointer to the shared memory needs to be shared with all TRAIN_THR processes.

A JOURNEY process maintains the MAP1 and MAP2 routes. The JOURNEY process informs the various trains about the route to be followed whenever requested.

Each TRAIN_THR, after its creation, asks for its own itinerary to the JOURNEY process, and stores it locally.

All TRAIN_THR start their journey at the same time, aside from inevitable computational delays.

Iteratively, each TRAIN_THR:

A. reads the next MAx segment, or the next station to be reached from its itinerary.
B. asks for permission to access the next MAx segment or the next station, checking if the corresponding MAx area in the shared memory contains 0, otherwise the permission is denied (there are no constraints for access to the station).
    - If TRAIN_THR has requested **and gains** access to a station, the TRAIN_THR enters the station, sets the contents MAx area of the shared memory to 0, then ends (has completed its mission).
    - If TRAIN_THR has requested **and gains** access to an MAx segment, the TRAIN_THR sets the contents of the MAx shared area to 1. Sets the contents of the previous MAx shared area to 0.
    - When permission is denied, does nothing and goes to step C.
C. sleeps 2 seconds.
D. *repeats the cycle starting from point A til a station is reached*

When all trains get to their final destination, the CONTROLLER prints a message to screen and concludes the program.

In a nutshell: when the train moves, it sets the contents of the segment it previously occupied to 0 (it writes 0 in the corresponding shared memory) for freeing the segment, such that other trains will be able to access it. Instead, the train sets the content of the segment in which it moves to 1, i.e. it "occupies" it: no other train will be able to access it. If permission to move to a new segment is denied, the train remains stationary in the current segment: it will try again to request access to the new segment in the next iteration. Whether it gains access or not, the train waits 2 seconds.

Data logging

Each TRAIN_THR fills one log file, for a total of five log files named T1.log, T2.log, T3.log, T4.log, T5.log. In the log files, each TRAIN_THR records information about the mission as shown below:

[Current: S4], [Next: MA14], April 27, 2024 16:14:10
[Current: MA14], [Next: MA15], April 27, 2024 16:14:13
[Current: MA15], [Next: MA16], April 27, 2024 16:14:16
[Current: MA16], [Next: MA12], April 27, 2024 16:14:19
[Current: MA12], [Next: S8], April 27, 2024 16:14:22
[Current: S8], [Next: --], April 27, 2024 4:14:25 PM

Rules for the presentation of the project.

The assigned project is valid up to and including the February 2025 exam sessions, and can be delivered approximately one week before each Operating Systems exam session through a special Moodle activity. The project must be carried out individually.

You are expected to deliver a ZIP archive containing:

- The **developed code** included all the files necessary for its compilation.
- A **project report** in **pdf format of up to 6 pages (10pt Times, single spacing, including figures and tables).** This should describe what you did and **highlight the design choices** related to the overall architecture of the system, parallelism management, and the optional elements implemented.

Content of the report. The report should contain:

- Name, Surname, Registration number, E-mail address
- Delivery date
- Step-by-step instructions for compiling and running
- Target system: SW and HW features, e.g. the Linux distribution used
- Design and implementation: high-level presentation of the adopted solution, and main features or design choices, detailing how to avoid deadlocks and parallelism issues.
- Execution: present and comment on a typical execution of the program, providing evidence of its correct functioning for the most critical or interesting cases.