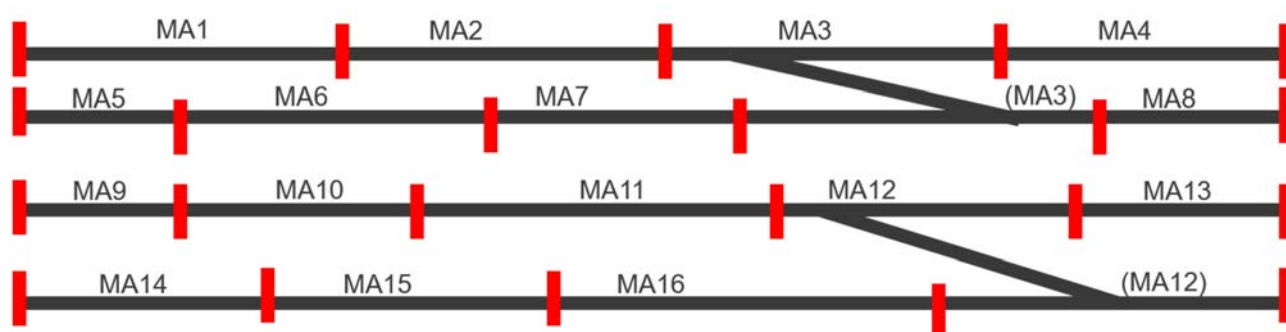


## Laboratorio di Sistemi Operativi

### **Progetto A.A. 2017/2018 – Uno schema di coordinamento vagamente ispirato alla Movement Authority per ERTMS/ETCS LV 1 e LV 2**

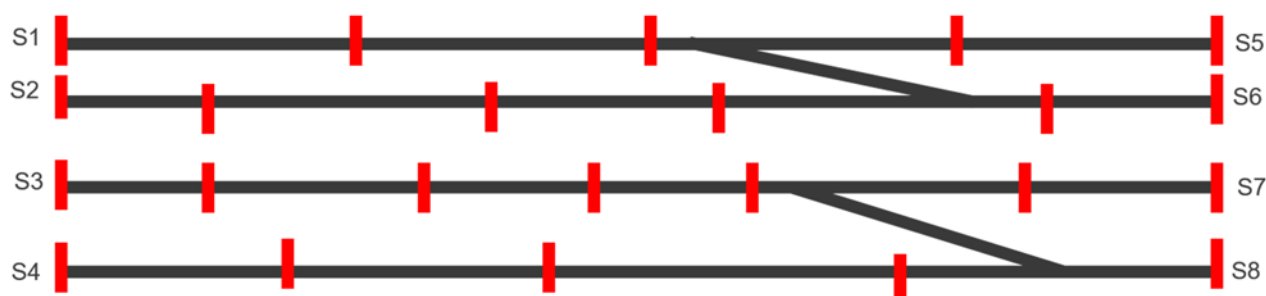
**Obiettivo complessivo.** Obiettivo del progetto è simulare il comportamento di 5 treni che, per realizzare la loro missione ferroviaria, attraversano vari segmenti di binario. La missione ferroviaria di ciascun treno è raggiungere una stazione definita. Il vincolo principale al movimento dei treni è che ciascun segmento di binario può essere occupato da un solo treno alla volta. Ciascun treno riceve di volta in volta, durante la sua missione ferroviaria, il permesso di accedere al segmento di binario successivo. Il progetto prevede la realizzazione dello schema soprastante in due modalità differenti, che corrispondono a due diversi modi per richiedere e ottenere il permesso di accedere ad un segmento di binario.

**Descrizione del sistema e del suo comportamento.** Nella Figura 1 sottostante, si riporta l'insieme di binari considerato. Si tratta di 16 segmenti, numerati da MA1 a MA16. Nel seguito del documento, con MA<sub>x</sub> si intende un qualunque segmento dei 16 presenti in Figura 1. Ciascun segmento è delimitato da 2 boe, con l'eccezione dei segmenti che contengono interconnessioni: questi ultimi sono delimitati da 4 boe, e sono i segmenti MA3 ed MA12. In Figura 1, ciascuna boa è rappresentata con una barra verticale.



*Figura 1 Binari e segmenti di binario MA<sub>x</sub> considerati.*

Si definiscono 8 stazioni, una per ciascun terminale di binario. Le stazioni sono numerate da S1 ad S8 come riportato nella Figura 2 sottostante.



*Figura 2 Stazioni numerate da S1 a S8.*

Si definiscono 5 treni, numerati da T1 a T5. Ciascun treno ha un itinerario associato, ovvero una stazione di partenza, un insieme di segmenti MA<sub>x</sub> da attraversare, ed una stazione di destinazione. La Tabella 1 sottostante associa un itinerario a ciascun treno.

Treno	Itinerario
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

*Tabella 1 Treni e percorsi.*

Si definiscono le seguenti regole del sistema ferroviario in oggetto:

- Ciascun treno avvia la propria missione ferroviaria da una stazione, ed esegue la propria missione ferroviaria fino al raggiungimento della stazione di destinazione, seguendo l'itinerario di Tabella 1.
- Tutti i treni avviano la propria missione ferroviaria nello stesso momento; in altre parole, partono contemporaneamente.
- La missione ferroviaria di un treno termina quando raggiunge la stazione di destinazione.
- Ciascun treno conosce il proprio itinerario, cioè la sequenza di segmenti  $MA_x$  che dovrà attraversare. I treni non possono deviare dal loro itinerario.
- Un segmento  $MA_x$  può essere attraversato da un solo treno alla volta; in altre parole, in un dato istante, ciascun segmento  $MA_x$  può essere occupato da al più un treno.
- Si impiegano 3 secondi per attraversare un qualsiasi segmento  $MA_x$ .
- Ciascun treno chiede, alla partenza dalla stazione oppure al termine dell'attraversamento di un qualsiasi segmento  $MA_x$ , il permesso di attraversare il segmento successivo o di entrare nella stazione di destinazione. Il permesso può essere richiesto in due modalità differenti, descritte nel seguito del progetto.
- Ovviamente, se due o più treni richiedono contemporaneamente accesso allo stesso segmento di binario  $MA_x$ , soltanto un treno potrà accedere (o meglio, *occupare*) il segmento; gli altri treni dovranno invece restare fermi, in attesa che il segmento di binario sia libero.
- Nelle stazioni di destinazione può risiedere un qualunque numero di treni; in altre parole, le stazioni non hanno il vincolo di accesso unico proprio del segmento di binario.

## Dettagli implementativi del programma richiesto.

Si richiede di sviluppare un programma che realizza il sistema descritto sopra, implementando i seguenti requisiti prescrittivi.

**Requisiti per l'avvio del programma.** Il programma è eseguibile in due modalità, che chiameremo ETCS1 ed ETCS2.

*Modalità ETCS1:* Il programma è avviato da una shell, specificando l'opzione ETCS1

Esempio: `shell# ./nomeprogramma ETCS1`

*Modalità ETCS2:* Il programma è avviato usando due shell diverse: in una shell, l'applicativo è lanciato specificando l'opzione ETCS2, mentre nell'altra shell è lanciato specificando l'opzione ETCS2 RBC.

Esempio:

`shell1# ./nomeprogramma ETCS2`

`shell2# ./nomeprogramma ETCS2 RBC`

Il ruolo delle opzioni ETCS1, ETCS2 ed ETCS2 RBC sarà descritto nel seguito (sezione *Modalità di richiesta di accesso*).

**Requisiti dell'esecuzione del programma.** I cinque treni sono rappresentati tramite 5 processi, che chiameremo nel seguito PROCESSI\_TRENI. I PROCESSI\_TRENI sono figli di un unico processo che chiameremo nel seguito PADRE\_TRENI.

PADRE\_TRENI è responsabile di creare 16 file di testo, che rappresentano i 16 segmenti MAX. Ad esempio, i file possono avere nome MA1, MA2, etc. Questi file sono inizializzati con il carattere "0" (zero) al loro interno (ovvero, PADRE\_TRENI scrive uno 0 all'interno di tali file). I file sono impostati per avere accesso in lettura e scrittura da parte di tutti, e privilegi posti a 777.

Ciascun PROCESSO\_TRENO, dopo la sua creazione, legge il proprio itinerario di missione (ovvero, l'insieme di segmenti MAX e stazioni che deve attraversare) da un file di testo, e lo mantiene in memoria in una struttura dati. Questi file di testo sono forniti nei file CSV chiamati T1, T2, T3, T4, T5, e sono disponibili sul sito del corso. Chiaramente, vi è un percorso differente per ciascun PROCESSO\_TRENO, ed i file T1, T2, T3, T4, T5 rispecchiano il contenuto di Tabella 1.

Tutti i PROCESSI\_TRENI avviano la propria missione contemporaneamente, a meno di inevitabili ritardi computazionali.

Iterativamente, ciascun PROCESSO\_TRENO:

- A. Legge il segmento MAX successivo o la stazione successiva, dalla propria struttura dati dove ha memorizzato l'itinerario di missione.
- B. Richiede l'autorizzazione di accedere a tale segmento MAX o alla stazione. Questa richiesta è effettuata in modo differente a seconda che il programma sia avviato con l'opzione ETCS1 o ETCS2 (si veda la sezione *Modalità di richiesta di accesso*).
  - o Se PROCESSO\_TRENO ha richiesto **ed ottiene** accesso ad una stazione, il PROCESSO\_TRENO entra nella stazione, imposta a 0 il contenuto del file MAX corrispondente al segmento che occupava precedentemente, quindi termina (ha completato la propria missione).
  - o Se PROCESSO\_TRENO ha richiesto **ed ottiene** accesso ad un segmento MAX, il PROCESSO\_TRENO:
    - imposta il contenuto del file MAX ad 1. Imposta a 0 il contenuto del file MAX corrispondente al segmento che occupava precedentemente.
- C. Dorme 3 secondi.
- D. *Ripete il ciclo ripartendo dal punto A*
- E. *(facoltativo: gestire la sincronizzazione tra i processi per assicurarsi che i PROCESSO\_TRENI avviino ciascuna iterazione in contemporanea, a meno di ovvi ritardi computazionali)*

*Chiarimento: in altre parole, quando il treno si sposta, imposta a 0 il contenuto del segmento che occupava precedentemente (scrive 0 nel file corrispondente): questo vuol dire che "libera" il segmento, e altri treni potranno accedervi. Invece, il treno imposta ad 1 il contenuto del segmento in cui si sposta, ovvero lo "occupa": nessun altro treno potrà accedervi. Se il permesso per spostarsi in un nuovo segmento è negato, il treno rimane fermo nel segmento attuale: riproverà a chiedere l'accesso al nuovo segmento nell'iterazione successiva. Sia che ottenga accesso, sia che non ottenga accesso, il treno dorme per 3 secondi.*

**Modalità di richiesta di accesso.** La richiesta di accesso ad un segmento MAX o una stazione è differente a seconda che il programma sia stato avviato con opzione ETCS1 o ETCS2.

**Avvio con ETCS1.** Se un PROCESSO\_TRENO richiede l'autorizzazione di accedere ad una stazione, l'accesso è sempre garantito. Se un processo treno richiede l'autorizzazione di accedere ad un segmento MAX, il PROCESSO\_TRENO verifica il contenuto del file MAX.

- Se è 0, il PROCESSO\_TRENO ha l'autorizzazione a spostarsi nel segmento MAX.
- Altrimenti, l'autorizzazione è negata.
- *(facoltativo: si consiglia di valutare l'opportunità di effettuare il lock dei file per letture/scritture concorrenti).*

*Avvio con ETCS2.* In questa versione, le autorizzazioni ad occupare i segmenti sono concesse da un server socket AF\_UNIX che chiameremo RBC. RBC gestisce le richieste di accesso ai segmenti ed alle stazioni. Il programma avviato con opzioni “ETCS2 RBC” genera questo server socket, mentre il programma avviato con la sola opzione “ETCS2” genera il PADRE\_TRENO e i PROCESSI\_TRENO.

Al suo avvio, RBC riceve dal PADRE tutti i percorsi dei treni. In altre parole, RBC riceve dal PADRE tutte le informazioni contenute nei file T1, T2, T3, T4, T5. Inoltre, RBC crea e mantiene una struttura dati per memorizzare lo stato dei segmenti MAX e delle stazioni, secondo le seguenti regole:

- Lo stato di un segmento MAX può essere libero (impostato a 0) o occupato (impostato ad 1). Ovviamente, in fase di inizializzazione tutti i segmenti sono liberi.
- Lo stato di una stazione è il numero di treni collocati in tale stazione.

Un PROCESSO\_TRENO deve richiedere ad RCB l’autorizzazione di accedere ad un segmento MAX o ad una stazione. Quando riceve tali richieste, RBC risponde al PROCESSO\_TRENO nel seguente modo:

- l’autorizzazione per la richiesta ad accedere ad una stazione è sempre concessa.
- Se invece PROCESSO\_TRENO richiede di accedere al segmento MAX, il server socket RBC controlla lo stato del segmento, e concede l’autorizzazione soltanto se lo stato è libero. Lo stato del segmento è quindi impostato ad *occupato*.

RBC si occupa anche di liberare lo stato dei segmenti: in altre parole, quando un PROCESSO\_TRENO lascia un segmento, comunica l’evento ad RBC che conseguentemente cambia lo stato del segmento in *libero*.

Nota: i PROCESSI\_TRENI continuano comunque a scrivere i valori nei file MAX, che infatti non sono utilizzati da RBC.

*(facoltativo: in caso di informazione discordante tra RBC e boe (ovvero, le informazioni dell’RBC non corrispondono con i requisiti delle boe), il TRENO rimane fermo).*

## Data logging

Ciascun PROCESSO\_TRENO riempie un file di log, per un totale di cinque file di log di nome T1.log, T2.log, T3.log, T4.log, T5.log. Nei file di log, ciascun PROCESSO\_TRENO registra informazioni sulla missione, cioè vi riporta continuamente il settore MAX in uso (oppure la stazione di partenza), il prossimo settore (oppure la stazione di destinazione), la data. A titolo esclusivamente di esempio, si mostra un possibile file di log per il PROCESSO\_TRENO corrispondente a T3 (quindi, T3.log):

```
[Attuale: S4], [Next: MA14], 27 Aprile 2018 16:14:10
[Attuale: MA14], [Next: MA15], 27 Aprile 2018 16:14:13
[Attuale: MA15], [Next: MA16], 27 Aprile 2018 16:14:16
[Attuale: MA16], [Next: MA12], 27 Aprile 2018 16:14:19
[Attuale: MA12], [Next: S8], 27 Aprile 2018 16:14:22
[Attuale: S8], [Next: --], 27 Aprile 2018 16:14:25
```

RBC invece scrive un file RBC.log, che contiene le autorizzazioni concesse e negate, i destinatari, e la data. A titolo esclusivamente di esempio, un possibile estratto del file di log RBC.log potrebbe essere:

```
[TRENO RICHIEDENTE AUTORIZZAZIONE: T4], [SEGMENTO ATTUALE: S6], [SEGMENTO RICHiesto: MA8],
[AUTORIZZATO: SI], [DATA: 27 Aprile 2018 16:15:22]
[TRENO RICHIEDENTE AUTORIZZAZIONE: T4], [SEGMENTO ATTUALE: MA8], [SEGMENTO RICHiesto: MA3],
[AUTORIZZATO: NO], [DATA: 27 Aprile 2018 16:15:25]
[TRENO RICHIEDENTE AUTORIZZAZIONE: T4], [SEGMENTO ATTUALE: MA8], [SEGMENTO RICHiesto: MA3],
[AUTORIZZATO: SI], [DATA: 27 Aprile 2018 16:15:28]
```

### **Regole per la presentazione del progetto.**

Il progetto assegnato è valido fino agli appelli di Febbraio 2019 compresi.

Il progetto deve essere svolto in gruppi di 2-3 persone. Gli studenti lavoratori possono svolgere il progetto individualmente.

E' necessario consegnare:

- Il **codice sviluppato** inclusi tutti i file necessari alla sua compilazione.
- Una **relazione** sul progetto. Ogni gruppo di lavoro dovrà produrre **una sola relazione**, i cui autori saranno quindi tutti i membri del gruppo stesso. La relazione dovrà essere in formato **pdf**.
- Un archivio **.zip** oppure **.tar.gz** contenente il codice e la relazione dovrà essere caricato sul sito del corso, seguendo l'apposito link che verrà reso disponibile alla pagina del corso.
- *Non si accettano consegne in formati differenti da quelli sopra indicati*

### **Relazione**

La relazione dovrà contenere:

- Informazioni sugli autori (per ciascun componente del gruppo: Nome, Cognome, Numero di matricola, indirizzo e-mail) e la data di consegna
- Descrizione dell'implementazione. La relazione dovrà essere esaustiva delle scelte progettuali e implementative. Alcuni possibili elementi specifici da discutere: descrizione dell'architettura software sviluppata e motivazione delle scelte progettuali, principali funzioni e strutture, principali algoritmi sviluppati.
- Evidenza del corretto funzionamento del programma rispetto alle specifiche fornite. Ciò può essere realizzato riportando e commentando opportunamente estratti del file di log, ovvero commentando tracce dell'esecuzione.
- **La relazione deve riportare tutte le istruzioni necessarie per compilare i file sorgenti e lanciare in esecuzione il programma.**