

# Esame Software Engineering (AA 2024/25)

09 Gennaio 2025 Lab. Colossus - Via salaria 113

*Enrico Tronci*

*Computer Science Department, Sapienza University of Rome*

*Via Salaria 113 - 00198 Roma - Italy*

tronci@di.uniroma1.it

<https://raise.uniroma1.it>

## Esercizio 5 (20 punti)

L'unità di tempo per il presente esercizio è il secondo e  $T = 1$  (secondi) è il time step per tutti i sistemi nel presente esercizio.

Il presente esercizio ha l'obiettivo di progettare un *dispatcher* per i *customers* dell'esercizio 3.

Il dispatcher ha  $N$  inputs ed  $N + 1$  outputs. Ad ogni input del dispatcher è connesso un customer. L'output  $i$  ( $i = 1, \dots, N$ ) del dispatcher ritorna il numero di messaggi ricevuti dall'input  $i$  (cioè dal customer  $i$ ). L'output  $N + 1$  del dispatcher ritorna tutti i messaggi ricevuti in input.

Il dispatcher garantisce la segunete proprietà.

### Order Preservation:

I messaggi dall'output  $N + 1$  hanno lo stesso ordine di quelli dai canali di input del dispatcher. Cioè, se il messggio  $m_1$  sul canale di input  $i$  ha timestamp  $t_1$  ed il messaggio  $m_2$  sul canale di input  $j$  ha timestamp  $t_2$  allora  $t_1 < t_2$  implica che  $m_1$  precede  $m_2$  nell'output del dispatcher. Si noti che se  $t_1 = t_2$  il dispatcher può scegliere qualsiasi ordine per i messaggi di output.

Per facilitare l'enforcement e la verifica del requisito di cui sopra è opportuno che ogni messaggio dai customers abbia il suo timestamp.

## 1 Formato dei parametri

Per semplicità si assume che tutti i customers hanno lo stesso valore per valore atteso a deviazione standard di  $T$ . Quindi il file `parameters.txt` consiste di tre righe:

```
N <numero di customers>
Avg <valore medio di  $\tau$ >
StdDev <deviazione standard di  $\tau$ >
```

Un esempio di file `parameters.txt` è:

```
N 100
Avg 3600
StdDev 500
```

## 2 Obiettivo

Si progetti il dispatcher di cui sopra ed un monitor con un solo output:

- MonitorOutput: 0 se la proprietà di *order preservation* è garantita dal dispatcher, 1 altrimenti.

La verifica viene fatta simulando il sistema per  $10^6$  time steps, cioè  $10^6$  secondi di tempo simulato (corrispondenti a circa 11 giorni di tempo simulato).

## 3 Formato di output

L'output dell'esercizio è memorizzato nel file `results.txt` la cui prima riga è formattata come indicato nelle istruzioni generali.

Le rimanenti righe del file `results.txt` hanno il formato:

```
<Canale di input 1 dispatcher> <numero messaggi ricevuti sul canale
1>
...
<Canale di input N del dispatcher> <numero messaggi ricevuti sul
canale N>
M2 <MonitorOutput>
```

Se  $N = 2$ , un esempio di file `results.txt` è:

```
2025-01-09-Mario-Rossi-1234567
1 100
2 150
M2 0
```