



B4 - Concurrent Programming

B-CCP-400

Bootstrap

Panoramix





During this Bootstrap, you will study examples of threads and mutex programming.

The context is trains running on parallel tracks, meaning each train has its own railway track. There is a bridge far way, but it is too fragile to hold the weight of several trains. Therefore, there can only be one train on the bridge at any given time.

Here are the symbols we'll be using;

- **n**: represents the number of the train
- **=** represents the railway
- **|** represents the bridge
- **#** represents the trains

The starting position is the following:

```
0123456789012345
-----
0: #===|----|=====
1: #===|----|=====
2: #===|----|=====
```

When a train is crossing the bridge, the other trains must wait...

```
0123456789012345
-----
0: =====| -#--|=====
1: =====|----|=====
2: =====|----|=====
```

...and then they get going again...

```
0123456789012345
-----
0: =====|----|==#==
1: =====| -#--|=====
2: =====|----|=====
```

...one at a time:

```
0123456789012345
-----
0: =====|----|====#
1: =====|----|==#==
2: =====| -#-|=====
```



STEP 1: ONE AT A TIME

Write a program that simulates the trains crossing the bridge, allowing only one train to cross at a time. The number of trains is defined by the **NB_TRAINS** macro.



Each train must be driven by a thread, and crossing the bridge must be controlled by a mutex.

Have an assistant check to see if your program functions properly.

STEP 2: SEVERAL... IS BETTER

Modify the previous program so that any given number of trains can simultaneously cross the bridge. This number will be defined by the **NB_MAX** macro.

```
0123456789012345
-----
0: ===== | ---- | ==#==
1: ===== | --#- | =====
2: ===== | -#-- | =====
3: ===== | ---- | =====
```

And that's all.

You should now be able to start your Rice-eating Philosophers project.