Advanced Robot Programming - Assignment a.a. 19/20 - V 2.0

Author: Andrea Tiranti
EMAIL: andrea.tiranti97@gmail.com
Date: 20/09/21

Introduction

The scope of this assignment is to simulate a network of multi-process systems, each one running on a machine in the same LAN. The processes are connected through sockets, exchanging a data in the form of a token consisting of a *time stamp* and a *value*. In practice, due to Covid-19, all the multi process system is implemented in a single machine simulating the communication across multiple machines. In Figure 1 we can see the architecture of each machine ad the interface between them.

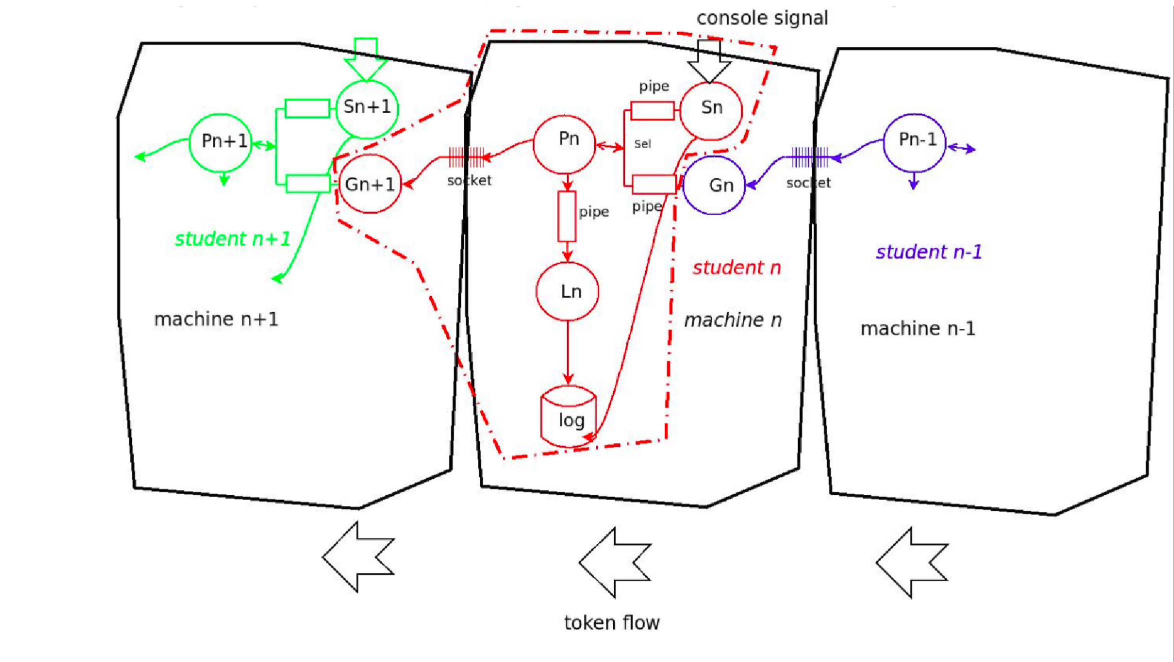

Figure 1: Overall Architecture

The idea is that the message exchanged between machines is a token with a timestamp and a *float* value. The value is computed as (T_{t-1} = old token, rf = reference frequency) if the value of $T_{t-1} < -1$:

$$signal_{increasing}(t) = T_{t-1} * cos(2\pi * rf * delay) + \sqrt{(1 - T_{t-1}^2/2) * sin(2\pi * rf * delay)} \tag{1}$$

and as below if the value of $T_{t-1} > 1$:

$$signal_{decreasing}(t) = T_{t-1} * cos(2\pi * rf * delay) - \sqrt{(1 - T_{t-1}^2/2) * sin(2\pi * rf * delay)} \tag{2}$$

For generate a sinusoidal wave depending on the *reference frequency* (rf) and on the **Time Delay** ($delay$) of the communication; moreover the wave oscillates always between $[-1; 1]$.

Finally the idea is to carry out some experiment testing how the wave change according to the time delay and test the system with different reference frequencies.

1 Overall project organization

For using the project simply clone the git repo at URL.

In the repository or in the compressed folder sent for the assignemtn we can find:

- The *ConfigurationFile.txt*.
- The */src* folder with the source files.
- The */executables* folder in which we have the executable.
- The */results* folder in which we can see the Log File and the signal file in which the wave is saved according to his frequency

NB: READ THE README.md file for more specification of the project and for all the instructions for compile, run and interact with the project

2 Implementation

The architecture already presented is implemented using:

- **fork()** : Is used for create processes P, G and L, all child of process S
- **pipe()** : Is used for create the pipes between P and L, P and G, P and S.
- **select()** : Is used for selecting in which pipe write or read (using **write()** and **read()**)
- **signal()** : Is used for handling the possible signals for interact with the execution of the program
- **socket()** : Is used for communicate with the next machine (localhost in V2.0)
- **exec()** : Is used for execute process G.

In the */src* folder inside the files *project.c* and *G.c* is possible to go deeply for the implementation.

3 Results

As we expected after the explanantion in the *Introduction* the signal must have a periodic behavior and we expect that the wave frequency change according to the reference frequency given in the *ConfigurationFile.txt*. Below in Figure we can check this out:

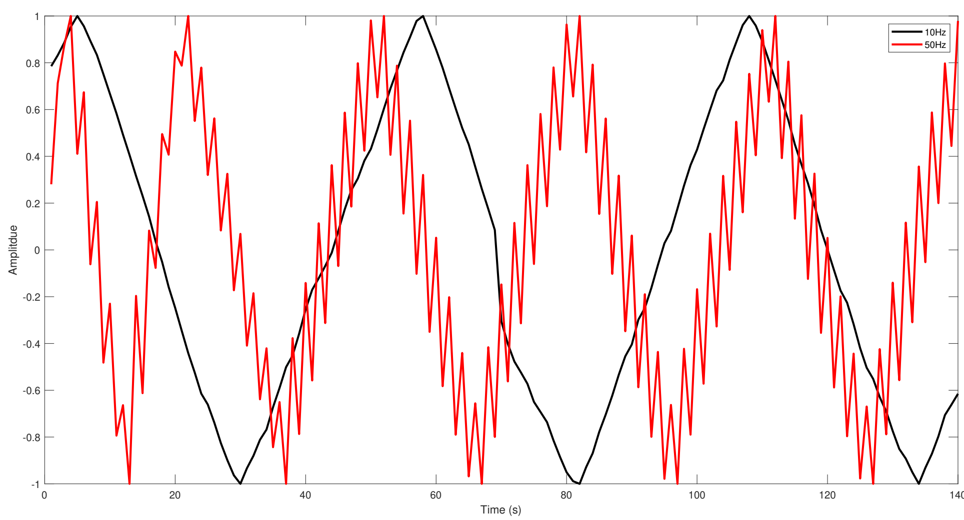

Figure 2: Overall Architecture

We can clearly see that effectively the signal with $rf = 50Hz$ oscillates more fast than the other signal with $rf = 10Hz$. The amplitudes of the signals are the same, as introduced before, in fact our signal can be brought back to the canonical sinusoidal signal:

$$x(t) = A * sin(\omega * t) \tag{3}$$

Where ($f = rf$ in our case):

$$\omega = 2 * \pi * f \tag{4}$$