2019 ARP Assignment V1.0

The problem to be solved has the following specifications.

The goal is a network of multi-process systems, all *identical*, each one running on a machine in the same LAN, connected through sockets, exchanging a data token at a specified and measured *speed*. The number of machines is not given *a priori*. In figure 1 there are 4 machines as an example. They may be more, or less, or they may even reduce to *one single machine* for setup and debugging purposes.

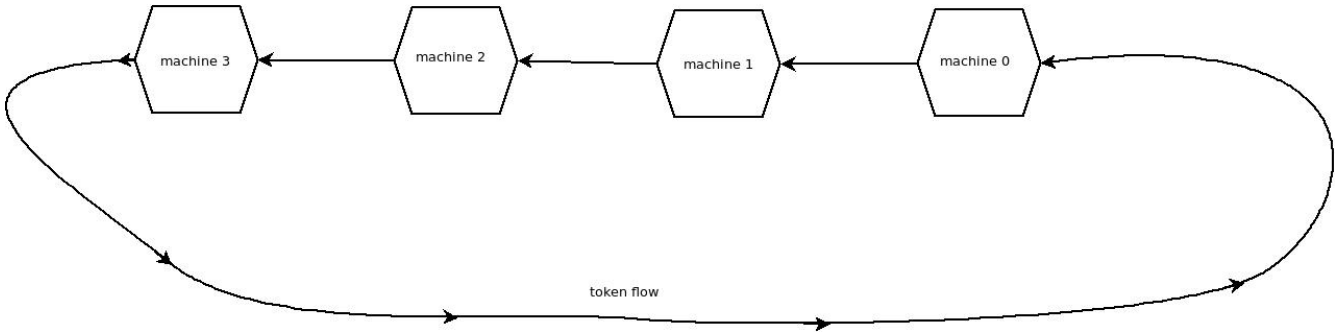

Figure 1

In figure 2 the structure of each machine is defined. Note that a black frame encompasses each machine's multi-process system, whereas a colored frame includes the system components that must developed by each student. Summing up, each student develops all system components of her/his machine, except for one, which is acquired by her/his partner "right side" student; in her/his turn, the student develops a component that is to be given to, and integrated by the"left side" student ("right" and "left" refer to the schema in figure 2).

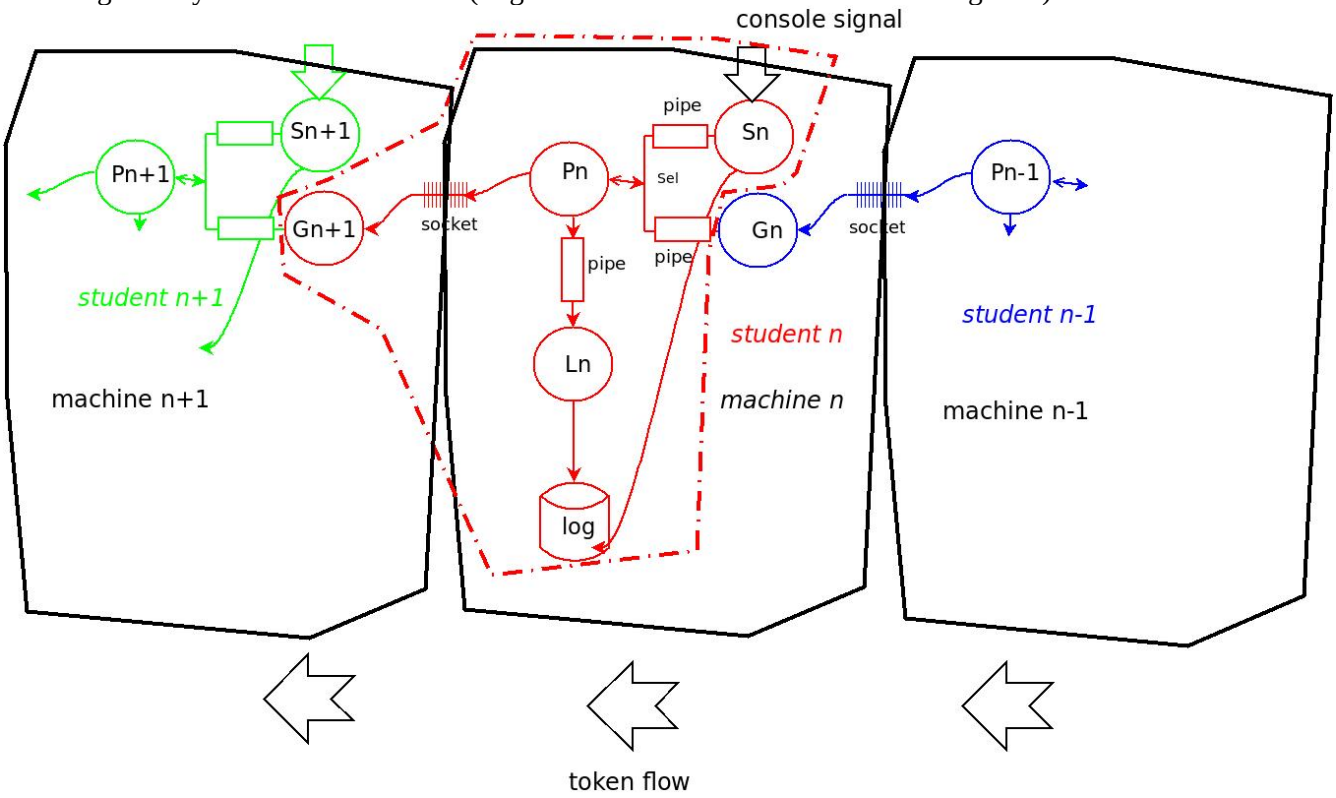

Figure 2

Glossary.

- Circles are Posix processes.
- Process P receives tokens, computes and sends an updated token (see below).
- Process L logs data end events in a log file.
- Process G receives tokens and dispatches them to P.
- Process S receives console messages as Posix signals.
- Processes S and G are connected to P through non-deterministic pipes (Posix *select*).
- A socket connects P to the partner's G process.

Detailed specifications.

A ***configuration file*** in each machine contains in text format the necessary parameters, and is edited by the user before running the multi-process:

- IP address of the machine, and of the adjacent machines
- reference frequency (RF)of the generated token wave
- other relevant data to be read before starting.

Token is a floating point number between -1. and 1.

P does the following computation:

new token = received token + DT x (1. - (received token)^2/2) x 2 pi x RF

where DT is the time delay between the reception and delivery time instants (that must be measured by the system). In this way the token series forms a ***sine wave*** of frequency RF.

Log file
holds a series of text lines in couples:
<timestamp> <from G | from S > <value>
<timestamp> <sent value> (a sample of the wave)

The log file contents is output anytime G receives a suitable request from the console

S receives signals for carrying out the following actions:

- start (receiving tokens, computing, sending tokens)
- stop ((receiving tokens, computing, sending tokens)
- dump log: outputs the contents of the log file, separating the wave from the actions.

Start phase
P, G, S, L start using the parameters stored in the configuration file.

The process G is given by the "right side" student in executable form, and is run using the *exec* syscall.

Main syscalls involved:

fork(), exec()
pipe()
select()

read() write()
socket()
signal()
and related ones.

Organization

Students should setup their own multi-process systems by implementing all components in a single machine.
The operating system is Ubuntu 18.04 LTS.
The final test will be carried out by triple or quad groups of randomly selected students.