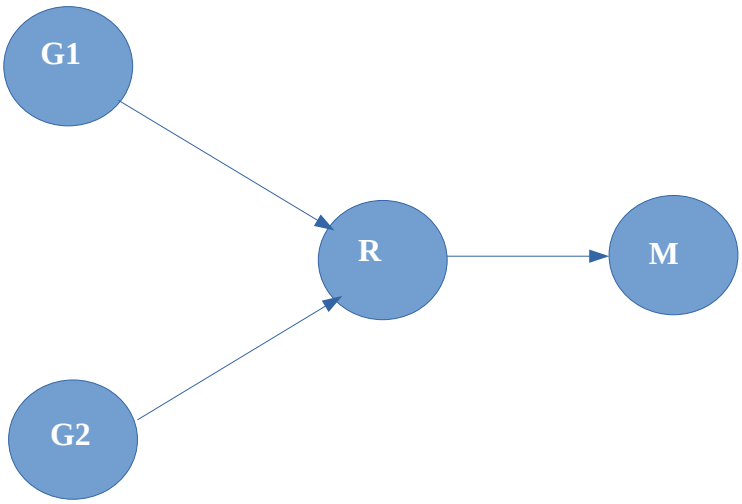
ARP – second assignment - 2nd semester



G1, G2 are processes that generate short messages composed by a time stamp, the process identifier, and a little datum identifying the message itself:

struct message {time_t timestamp; char g; int x);

timestamp is the actual time (by a suitable syscall)
g is a character identifying G1 or G2
x is the sequence number of the message by G1 or G2

Any message is generated by G_i inside a **strict loop** (the fastest possible); at any cycle a delay is applied before sending the message. The delay is computed as follows:

delay *[microseconds]* = offset + random(0,offset), where:

random(0,offset) is a random generator in the interval 0 – offset with uniform probability
offset is a parameter that must be changed in the experiment (see below)

This simplified model simulates the generation of quasi-periodic messages by a sensor at variable rate (depending on the value of offset).

R is a non deterministic receiver of the messages, with the aim of receiving as many messages as possible without blocking the generators G_i (as to say, allowing the G_i to execute at their maximum speed). R puts in a queue all received messages, in the same order they are received, adding each the actual time of reception. After the run is terminated (at least 1 million cycles of either G1 or G2), R transfers them to M; M computes the *bandwidth* (see below).

M analyzes data and outputs the result of the form:

no. of G1|G2 cycles: xx no. of messages received: xx offset delay in G1 | G2 cycles (usec): xx average delay (*latency*) between generation and reception of messages (usec): xx estimated bandwidth between G and R (bit/s): xx

Communication between G_i and R, and R and M are made through **pipes (*).**

The final aim is to find, by means of practical experiments, *the minimum offset value such as generators G_i are* ***not delayed*** *(or minimally delayed)* by R' execution.

With the so determined offset value the student must derive an estimate of the *throughput* of R, as to say, the maximum *bandwidth* (in bytes/s) of the communication between the G_i and R.

(*) write your program in two versions:
1. two unnamed pipes between G_i and R
2. one named pipe between G_i and R

and compare the results in the two cases.

Hints: keep the G loops codes as tiny as possible; the loop speed should be determined completely by the offset value; start the experiment with 10 offset values in the interval 10-50; start with a *timeout value* of the select ten times smaller than the minimum offset; then adjust slightly, one at a time, the offset and timeout values until the maximum bandwidth is achieved. Depending on your computer, values can vary much; possibly the starting 10 usec value will decrease or the 50 usec value will increase.