Research project: **QuePy: Queueing with SymPy**
Keywords: Python, SW development, symbolic mathematics, Markov chains
Supervisor: Prof. Esa Hyytiä (esa@hi.is)

Background: SymPy package implements symbolic mathematics to Python. The basic capabilities include manipulation of mathematical expressions, simplifying them, computing derivatives and integrals, etc. The package also offers a basic support for random variables and standard distributions. This means that it is possible to compute means, variances, different moments, among other things, in symbolic form, thus avoiding pitfalls numerical computations may introduce.

The field of queueing theory studies *server systems* such as a pool of web-servers or a router in a network. The basic system consists of a queue (buffer) and a processor. A complete model defines also a job arrival process and service order (e.g., the first-come-first-served, FCFS). Several well-known results exist, e.g.,

1. The mean waiting time in the FCFS queue with arrival rate λ and (random) service time X is given by the Pollazcek-Khinchine formula,

   $$\mathbb{E}[W] = \frac{\lambda\,\mathbb{E}[X^2]}{2(1-\rho)},$$

   where ρ denotes the offered load, $\rho = \lambda\,\mathbb{E}[X]$. The above expression thus involves the first and two moments of X, and the arrival rate λ (which is a positive real number).
2. Little's result is a very useful result that applies to *any system*, and it says

   $$\mathbb{E}[N] = \lambda \cdot \mathbb{E}[T],$$

   where N denotes the number of jobs (customers, files, sick people, etc.) in the system, and T represents the sojourn time (time spent in the system).

Project description: The high level goal is to study and evaluate how different results and methods of queueing theory from the literature can be implemented in the context of Python and SymPy, eventually leading to a software library. It should be possible to carry out computations both in symbolic and numeric forms. The initial idea is to utilize the *Duck-typing*, but considering different design options is the first step of this project.

The resulting software package allows one to solve different performance evaluation and optimization tasks efficiently. This will be demonstrated by solving some example problems (that will be chosen jointly together with the supervisor). An example problem is how to determine the optimal routing in a network of queues with respect to different objectives (e.g., the mean delay, or some other QoE-specific objective).

Results: The results of this work will be three-folded:

1. The resulting package will be useful for researchers in the field by allowing fast prototyping and evaluation of new ideas.
2. Package will be useful supplementary teaching material (both in terms of programming and performance analysis)
3. SymPy package is not as mature as, e.g., Mathematica, and shortcomings and design issues are often discovered only when a library is actually used. Consequently, it is highly likely that this work leads to contributions to the SymPy package itself. (E.g., how to implement a random sum?)

Prerequisites::

- Skills in Python and ambition for elegant code
- Basic understanding of the probability theory
- REI503M *Performance Analysis of Computer Systems* (recommended)