1) Acquisition of frames from a Webcam in Linux and storage of the frames on disk. Variation in disk write shall not affect the frequency at which frames are acquired. Therefore the task carrying out frame write on disk shall be decoupled from the task supervising frames acquisition. Task implementation can be via Linux processes or Threads.

2) Acquisition of frames from a Webcam in Linux and storage of the frames on disk. Acquired frames shall be sent via TCP/IP to a separate server that shall store acquired frames on disk. In the protocol used, the filename shall be sent to the server when the connection has been established,

3) Producer-(multiple) consumers program with remote status monitoring. The producer and the consumers shall be implemented as threads and the message queue will be hosted in shared memory. Another thread, separate from the producer and the consumers, shall monitor the message queue length, the number of produced messages and the number of received messages for every consumer. A TCP/IP server will allow one or more clients to connect. When a client connects, a new thread is created, handling communication with that client and periodically sending the information collected by the monitor thread.

4) Producer-(single) consumer program with dynamic message rate adjustment. The consumer shall consume messages at a given rate, that is, with a given delay simulating the consumed message usage. An actor (task or process) separate from producer and consumer shall periodically check the message queue length and if the length is below a given threshold, it will increase the production rate. Otherwise (i.e. the message length is above the given threshold), it will decrease the production rate.

5) Simulation of dynamic periodic task execution. A pre-defined set of routines (with unique assigned name) shall be defined in advance with given processor usage, period and deadline. Every routine shall be composed of a program loop followed by a nanosleep() call. The exact amount of CPU time and consequently of the processor utilization can be done in advance using the ***time*** Linux command. The execution supervisor shall listen in TCP/IP for requests for task activation/deactivation. The received message shall specify the name of the task to be activated. A given task can be activated multiple times, starting every time a new thread running the selected routine. Before accepting a request for a new task, a response time analysis shall be carried out in order to assess the schedulability of the system.

Exercises 4 and 5 are more difficult than the others. This fact will be accounted for in the final evaluation.