**Exercise 1**

1. A Software Architecture defines a system in terms of computational components and interactions amongst those components. The benefit of a Software Architecture is that it enables us to create Architectural Styles. So, that we can define a family of systems in terms of a pattern of structural organization. We can apply these architectural styles to many systems.
2. Performance can be affected negatively because of overhead of passing through layers.  
   Layers introduce some development complexity because instead of calling a component directly, we need to pass through layers. This is not so suitable to implement in simple applications.
3. Producer and consumer like problems are suitable to solve with Specialist Parallelism. For example, this is suitable for computer graphics rendering in games. Graphics rendering is divided into 3 parts, Application -> Geometry -> Rasterization. We can divide these tasks for each processors, and every output is send to next step. This structure is useful to use with flow architecture.
4. Specialist Parallelism is basically pipelining. A chain of processing elements arranged in a way so that the output of each element is the input of the next.  
   Result Parallelism is where the each processor computes its own result. This is good for processing each element in a data structure, such as the pixels in an image or the frames in a movie.  
   Agenda Parallelism is where tasks are assigned for each processing units. Unlike in Specialist Parallelism, each processor is not specialized for a specific task. This is good for computing one result from a large number of inputs.
5. Flow architecture is a chain of processing elements linerly connected so that the output of each element is the input of the next.  
   Blackboard architectures are Result Parallelism, Agenda Parallelism and Specialist Parallelism.
6. Each blackboard style is suitable for multiple processors. Architecture selection is more depends on the problem type. However, Result Parallelism and Agenda Parallelism can benefit greatly from having more processors. Specialist Parallelism is more suitable for having fewer processors.
7. Unix pipes are bounded buffers that connect producer and consumer processes. They can be used by putting vertical bars between commands like:

netstat -pnlt | grep ':portno'

Netstat operation results are send to grep operation and a bounded buffer is used to connect these processes.

**Exercise 2**

1. I have some doubts about fsp modelling. I’m not sure if I can be able to model so complex processes. Should we expect complex fsp modelling questions in the exam? How much information do we need to know for fsp modelling?
2. Should we able to write the regular expression of the defined petri net? Is this important? Or is it enough for us to know how to define a petri net?
3. Can you explain the calculation of traces in fsp again?
4. I won’t attend the lecture on Dec 13th, because of a seminar meeting. Can you share your answers as pdf? Also, Can you share the final exam of the previous lecture?