Programming Models

- Boss/Workers
- Work Crew
- Pipeline
- Client/Server

Boss/Worker

- One *master* thread and one or more *workers*.
- The boss thread gathers or generates tasks that need to be done, then parcels those tasks out to the appropriate worker thread.
- This model is common in GUI and server programs

Boss/Worker

- A main thread waits for some event and then passes that event to the appropriate worker threads for processing.
- Once the event has been passed on, the boss thread goes back to waiting for another event.
- The boss thread does relatively little work.
- While tasks aren't necessarily performed faster than with any other method, it tends to have the best user-response times.

Work Crew

- Several threads are created that do essentially the same thing to different pieces of data.
- It closely mirrors classical parallel processing and vector processors, where a large array of processors do the exact same thing to many pieces of data.
- This model is particularly useful if the system running the program will distribute multiple threads across different processors.

Pipeline

- The pipeline model divides up a task into a series of steps, and passes the results of one step on to the thread processing the next.
- Each thread does one thing to each piece of data and passes the results to the next thread in line.

Pipeline

- This model tends to keep the individual tasks small and simple, as well as allowing some parts of the pipeline to block (on I/O or system calls, for example) while other parts keep going.
- It is also handy for a form of recursive programming where, rather than having a subroutine call itself, it instead creates another thread.

Client/Server

- A computational architecture that involves processes requesting service from other processes.
- The client is a process (program) that sends a message to a server process (program), requesting that the server perform a task (service).

Client/Server

- Client programs usually manage the user-interface portion of the application, validate data entered by the user, dispatch requests to server programs, and sometimes execute business logic.
- The client-based process is the frontend of the application that the user sees and interacts with.
- The client process contains solution-specific logic and provides the interface between the user and the rest of the application system.

Client/Server

- A server process (program) fulfills the client request by performing the task requested.
- Server programs generally receive requests from client programs, execute them and dispatch responses to client requests.
- The server process acts as a software engine that manages shared resources
- The server process performs the back-end tasks that are common to similar applications.