Asynchrony (computer programming)

Asynchrony, in computer programming, refers to the occurrence of events independent of the main program flow and ways to deal with such events. These may be "outside" events such as the arrival of signals, or actions instigated by a program that take place concurrently with program execution, without the program hanging to wait for results.[1] Asynchronous input/output is an example of the latter case of asynchrony, and lets programs issue commands to storage or network devices that service these requests while the processor continues executing the program. Doing so provides a degree of concurrency.[1]

A common way for dealing with asynchrony in a programming interface is to provide subroutines that return a future or promise that represents the ongoing operation, and a synchronizing operation that blocks until the future or promise is completed. Some programming languages, such as Cilk, have special syntax for expressing an asynchronous procedure call.[2]

Examples of asynchrony include the following:

- Asynchronous procedure call, a method to run a procedure concurrently, a lightweight alternative to threads.
- Ajax is a set of client-side web technologies used by the client to create asynchronous I/O web applications.
- **Asynchronous method dispatch** (AMD), a data communication method used when there is a need for the server side to handle a large number of long lasting client requests.[3] Using synchronous method dispatch (SMD), this scenario may turn the server into an unavailable busy state resulting in a connection failure response caused by a network connection request timeout. The servicing of a client request is immediately dispatched to an available thread from a pool of threads and the client is put in a blocking state. Upon the completion of the task, the server is notified by a callback. The server unblocks the client and transmits the response back to the client. In case of thread starvation, clients are blocked waiting for threads to become available.

See also

- Asynchronous system
- Asynchronous circuit

References

1. Davies, Alex (2012). *Async in C# 5.0* (https://books.google.com/books?id=xT45qhFrVnUC). O'Reilly. pp. 1–2. ISBN 9781449337124.
2. McCool, Michael; Reinders, James; Robison, Arch (2013). *Structured Parallel Programming: Patterns for Efficient Computation*. Elsevier. p. 30.
3. ICE usage of AMD (https://web.archive.org/web/20130721031512/http://zeroc.com/doc/Ice-3.3.0/manual/Overview.3.2.html).