1. The vanilla FTP protocol has no built-in authentication. The only authentication performed is with USER/PASS commands. After that, both the client and the server assume that all requests/responses received are authentic. Otherwise, the server and client communicate a port and address to connect on for data connections. Any other requests not on those ports can be assumed to not be authentic. However, because this port information is transmitted over plaintext, it's easy for it to be intercepted. In summary, FTP is a very insecure protocol.

2. With a properly threaded server, FTP can scale one-to-many. At any given time, FTP can support as many data transfers as it has ports. After that, however, it would need to start queuing data transfers. But again, with a properly designed server, this wouldn't be an issue.

3. FTP servers respond with a 503 response code if commands are sent out of order. They respond with 2xx response codes when commands are performed successfully. Using these codes, the client can determine if it is talking in the right order.