

# “Computer Operations and Production Engineering”

CSE191, Spring 2018 (section 940918)

CSE 2154 (Wed @ 11am)

Taner Halicioglu <taner@eng.ucsd.edu> – <http://cseweb.ucsd.edu/~taner/>

Week 10 class is MANDATORY (no final during finals week). Also: Adjust class to ONE (1) unit, P/NP

**Overview:** We will explore topics that include System Administration/Operations and Network Engineering, and focus on aspects of troubleshooting and debugging. We'll also cover some "best practices" when it comes to architecting software and systems, as it relates to robustness, scaling, and ease of debugging. I'll be drawing from my experiences over the last 20+ years, working in operations-related roles.

**My background:** Graduated from UCSD in 1996 with a B.S. in Computer Science. While at UCSD, I worked at SDSC as a System Administrator. After graduation, I worked at a startup ("Globalcenter", 1996) doing Software and Network Engineering, and building system and network monitoring systems. I then moved to another startup ("Loudcloud", 1999) doing System Administration and Software Engineering (participated in one patent). After the "dot-com crash", I went to work at eBay (2002) as a Software Engineer in the Monitoring and Tools group. This was almost a "dev-ops" role, as we had to write software that helped control and monitor aspects of the eBay site (web server, app server, and database health). I then went to a no-name social networking startup called "TheFacebook" in late 2004, where I worked for over 5 years in various Operations-related roles, helping it evolve into what it is today. In 2009 I moved to south Orange County and worked for Blizzard Entertainment for a couple of years, in the Battle.net group as I lead the "Reliability" (dev-ops) group in their work doing triage debugging and acting as a liaison between Engineering and Operations.

Two sides to debugging and troubleshooting:

- 1) What can you do while coding to make troubleshooting easier?
- 2) What can you do to troubleshoot once there is a problem, and you DON'T have the source code?

**While coding, you'll want to think about:**

- **Race Conditions:** Threaded work, Multi-Process work (resource contention), Across-the-network work.
- **Edge Cases:** Sanitizing potentially bad data (any input), always ask yourself "Can this break? If so, how?" Even if you think "this will never happen", at least CATCH it and log it, etc.
- **Is this efficient (enough)?** (warnings about trying to be over-efficient)
- **How does this scale?** 1 user, 1000 users, 100k users, 10M users? What is the limiting factor? (Disk I/O, RAM, CPU, Network?) What happens when we approach the limit? Can this scale horizontally (this is the "easy" way -- adding machines)? **Counter:** Does this NEED to scale (efficiency vs. scalability)?
- **How will I debug this?** "console" output? logging? (too much? too little? tunable?) command-line switches ("debug" mode)? signals to adjust logging? config file (that can be reloaded live)?

**While debugging, these are things you'll need to consider:**

- **System health?**
  - **CPU:** Load avg, Processor utilization - individual and overall, utilities (top, vmstat)
  - **Memory:** Resident sizes, what else is running, near mem limit (hitting swap - is there swap?), utilities (ps, top, vmstat)
  - **Disk I/O:** IOPS (I/O Operations Per Second), what is the actual disk subsystem (local disk/SSD, fiber channel/SAN (Storage Area Network), Network File System(NFS)/NAS (Network Attached Storage), utilities (iostat)
  - **Network:** Interface utilization (bps and pps), packet loss (local, remote), protocol (UDP? TCP?), other (end users that may be slow or lossy or disappear -- or be malicious), utilities (netstat, ping, traceroute/mtr, tcpdump, iftop/iptraf)
- **Process health?** Pegging the CPU? Unexpected mem usage? Current state (R, S, D, etc)? utilities (ps, lsof, strace)
- **Logs?** Where are they? Do they show anything useful? Are they readable/can they be followed?
- **Does it depend on other resources?** Other local (same-system) resources? Remote (network) resources? Are THOSE healthy?

If there are any topics you think you'd like to hear more about, or things I didn't mention that you think I should, please do not hesitate to tell me or email me! :-)