# Systems Engineering

| Level/Skill | Product Output | Communication/Writing | Data structures/Performance | Security | Networking | Systems Engineering | Tooling |
|---|---|---|---|---|---|---|---|
| 1 | Creates a design document based on well-defined scoped requirements and implements it. | Reports progress on a regular basis as required by the team's operational requirements. Actively solicits feedback. Participates on interview panels. | Demonstrates good understanding of basic data structures like hash tables, linked lists, and trees. Can reason about algorithm complexity. Applies relevant data structures in day to day activity.<br>Can implement a production quality software - it might be not the most efficient or secure, but correct. | Applies basic security principles to program design. For example, can set up HTTPS and password based auth. | Understands and reasons about networking concepts. Understands and can write production quality web servers. Understands common networking issues and troubleshooting techniques. | Understands the usage of POSIX and other APIs for Linux systems.<br>Understands synchronization primitives and their application, including reasoning about deadlocks and data races. Can write basic system-level code using the different types of memory and allocation. Understands inter process communication and can build systems leveraging it. Can implement data race and deadlock free code using basic production guidelines - using synchronization primitives and properly sharing state between components of the system. | Understands the usage of compilers, interpreters, build tools at the organization. |
| 2 | Can write high quality user and product focused documentation. | Provides constructive review on peers' code and design. Helps new team members during their first weeks. | | Applies industry best practices and security guidelines, like setting up strong TLS, can pick strong authentication and authorization mechanisms. | Has more granular understanding of the networking design, for example can reason about using GRPC vs HTTPS-JSON and their networking and scalability trade-offs. Can do the same for UDP vs TCP and lower level protocols. | Can reason about performance implications and risks of using synchronization primitives, understands granularity of locking, can debug and troubleshoot memory and synchronization issues. | Understands tools and compilers as an advanced user - for example, can refactor Makefiles to make them more efficient and parallel execution friendly or select the optimal set of compiler flags for Linux and macOS. |
| 3 | Collaborates with the team to scope requirements, based on good understanding of existing longer term product vision and estimates of the system design of a feature of a product. | Supports less experienced peers' technical skills, answering questions and being a resource. Documents and improves team practices. | Uses advanced data structures and algorithms, such as consensus, gossip protocols to implement key product features. | Understands security aspects of protocols on a deeper level, for example can explain differences between TLS 1.2 and 1.3 and security implications. Understands common attack vectors for server side or client side applications.<br>Can build secure systems that will pass quality security audit that will uncover few to no critical system design errors. | Understands scalability and resilience aspects of practical implementation of systems leveraging networking.<br>Can write fast, scalable servers and troubleshoot common networking issues such as connection lifecycle, pooling, backpressure and other aspects of the application design. | Not only can write data-race and deadlock free code, but implements safe and concurrent and/or parallel systems using minimum amount of shared state, granular locking - systems that are easy to read, extend and troubleshoot.<br>Writes high quality design documents with few to no critical system design errors. | |
| 4 | Leads the implementation of the isolated feature/improvement that measurably and significantly impacts business outcomes from gathering requirements to getting to the market stage. | Writes technical articles/blog posts, delivers tech and lightning talks representing the company's technical vision. | Leads implementation of products and standalone systems using advanced data structures and algorithms, such as upgrade/install distributed framework. | Can apply production quality novel cryptographic to build security systems. For example, can implement strong security support for the system with eBPF from scratch. | | Can implement production grade systems leveraging advanced low-level and novel components of the Linux design like BPF, control groups. | |
| 5 | Leads the implementation of a new product line or significant part of the product to deliver it to the market in collaboration with all other teams. | Writes advanced technical articles/blog posts, gaining significant industry traction or delivers technical talks on major conferences representing the company's vision. | Implements customer facing features and systems using advanced data structures and algorithms. | Writes technical articles on security aspects of the system, implements significant security product innovations in the area delivered to customers. | | Applies system level design to deliver new products or significant new components of an existing product to the market. | |
| 6 | Designs new data structures and algorithms solving relevant business problems and creating competitive advantage for the company. | Produces peer-reviewed research papers or patent applications. | | Researches and designs new security systems, cryptography and protocols. | | | |