1.What are some advantages of peer-to-peer systems over client–server systems?

- No single point of failure. If any peer does fail, distributed resources make the system more resilient. In a client-server model, if the server crashes, all clients will lose the service that the server provides.
- Scalable. With additional peers there are additional resources (processing, storage, bandwidth), while P2P systems scale more effectively than centralized servers.
- More cost effective. Because every peer is effectively sharing their resources, P2P reduces the cost of a central server.
- Load distribution. The workload and traffic is distributed across all peers instead of a central server.
- Resource sharing. Each peer can share files, CPU cycles or other resources. This effectively acts as incentive for collaborative computing.
- Decentralization. By not having a centralized authority, P2P minimizes censorship and bottlenecks.

2. Describe some distributed applications that would be appropriate for a peer-to-peer System.

Examples of suitable distributed applications for a Peer-to-Peer systems are the ones below.

- File Sharing Systems: Such as BitTorrent, which breaks large files into pieces, allowing downloading of pieces from multiple peers at the same time.

- Content Distribution Networks (CDNs): Some decentralized CDNs that contain video, audio, and web content being distributed by peers.

- Blockchain & Cryptocurrencies: A system like Bitcoin or Ethereum uses P2P networks to replicate a ledger across peers and determine consensus.

- Collaborative Applications: Such as Google Docs-type applications for collaborative edition in a decentralized form, or for real-time multiplayer games.

- Voice over IP (VoIP) & Messaging Applications: Many applications and services rely on P2P networks for direct communication (e.g., Skype in its original form or a decentralized chat-based application).

- Grid/Volunteer Computing: Similar to SETI@home, Folding@home, or BOINC, many other projects are essentially a P2P participation for distributed scientific computation.

3. Identify several advantages and several disadvantages of open-source operating systems. Identify the types of people who would find each aspect to be an advantage or a disadvantage.

Pros: No-cost: Open-source OS like Linux can be downloaded and used at no cost (a plus to students, startups, or small-business owners).

Customized: Users can modify the source code to meet their particular needs (an advantage to developers and researchers).

Security/Transparency: The source code can be audited so vulnerabilities can be discovered and corrected; vulnerable code will be found and patched quickly (an advantage to those worried about security, such as to the U.S or an individual government).

Community Support: A strong global community documents, provides forums, and patches (an advanatge to learners and developers).

Innovation: Open-source allows for rapid experimentation and innovation (an advantage to tech companies or researchers).

Cons: No official company support: Unless purchased there's no company support (a disadvantage for enterprise needing help 24/7).

Software Compatibility Issues: Some proprietary software (like Adobe, MS Office) doesn't have full support (a disadvantage for a business that relies on that software).

Learning Curve: Generally requires more technical knowledge vs user-friendly proprietary OS (a disadvantage for someone who is just a casual user).

Hardware Compatibility: Some vendors provide only drivers for Windows/Mac (a disadvantage to end-users with this hardware).

Types of People & Viewpoints:

Students/Researchers: Pros - free and customizable, cons - may have difficulty with setup.

Enterprises: Pros - secure and scalable, cons - no officially supported and lack of compatibility.

Developers/Technologists: Pros - have access to the source code and you can innovate with, cons - the time it takes to set it up.

Casual Users: Pros- free, cons - usability issues and lack of usual apps.