Assignment 1

Question & answers

**Q1) What are syslog and rsyslog?**

● syslog: This is like your computer's diary. It writes down notes about everything that's

happening, from starting up a program to any errors that pop up. It's a way for the

computer to remember and track what it's doing.

● rsyslog: This is a smarter version of the computer's diary. It does everything syslog

does, but it can also send notes to other computers for backup or sort them better,

making it easier to find specific notes later.

Features of syslog/rsyslog:

Centralized Logging: Both tools are like having a central place where all notes from

different apps and parts of the computer are stored. This makes it easier to find and

review them later.

Network Logging: They can send these notes over a network to another computer. It's

like mailing your diary entries to a friend so they can read them too.

Flexible Configuration: You can set rules about what kind of notes to keep and which

ones to ignore, like only saving important diary entries and throwing out the rest.

Compared to Embedded Debug Logging:

● Embedded debug logging is like a special notebook for a specific project where you jot

down very detailed notes, especially when you run into problems. There might be some

common ground, like both noting down errors, but your special notebook has more

detailed notes about your specific project.

**Q2) Features of syslog-ng:**

Enhanced Filtering: Syslog-ng can ignore unimportant notes and only keep the ones you really care about. It's like having a filter that automatically throws out junk mail so you only see important letters.

Advanced Parsing and Rewriting: It can clean up and organize notes so they're easier to read and understand, sort of like having an assistant who tidies up your messy notes. Scalability and Performance: Syslog-ng can handle lots of notes without getting overwhelmed, making it great for when there's a lot going on and many notes to keep track of.

Secure Logging: It keeps your notes safe, especially when sending them over the internet, like sealing your letters in a tamper-proof envelope.

Flexible Destination Options: You can decide where to store your notes, like in a file on your computer, a database, or even in the cloud, giving you lots of choices on where to keep your information safe.

In essence, syslog-ng acts like a super-organized, secure, and efficient diary system for your computer, helping it keep track of what it's doing, stay tidy, and ensure important notes are always safe and easy to find.

**Q3) Name five ways syslog-ng is an improvement over syslog/rsyslog.**

1. Advanced Filtering Capabilities:

Comparing Syslog-ng to Syslog and Rsyslog, the latter provides a less versatile and less reliable filtering method. You may filter log messages using syslog-ng according to several parameters, including message priority, source IP, content, and more. This makes it possible to regulate the processing and forwarding of log messages more precisely.

2. Extensible Architecture:

Because of its modular and extensible architecture, Syslog-ng may easily have its capability expanded with the help of new modules and plugins. Compared to syslog and rsyslog, this offers more flexibility and customization possibilities, making it simpler to interact with other systems and add new capabilities as needed.

3. Reliable Message Delivery:

Features for fault tolerance and increased dependability are provided by Syslog-ng. To make sure that no log messages are lost in the event of a network outage or failure, it can buffer log messages locally and resend them. In distributed and high-availability systems, this is quite helpful.

4. Enhanced Security Features:

To guarantee the security and integrity of log messages, Syslog-ng comes with built-in authentication and encryption capabilities. It is more secure than conventional syslog and rsyslog since it supports encrypted transport protocols like TLS and can authenticate log sources using certificates.

5. Flexible log Destination Options:

Numerous log destinations are supported by Syslog-ng, including local files, distant servers, databases, and message queues. More adaptable than both syslog and rsyslog, this versatility enables you to quickly connect syslog-ng with a variety of logging and monitoring solutions.

**Q4) This assignment assumes one server and one application (the travel simulator plus the logger). How would the design of your server change if there were ten applications instead, each on a different machine?**

To manage several incoming log messages from various sources (machines), the server’s design would need to be changed if there were 10 apps, each running on a separate computer. Here are the changes:

1. **Dynamic IP and Port Binding:**

The server was initially intended to be restricted to a single IP address and port. The server must bind to INADDR\_ANY in order tot support several apps running on various computers. This will enable it to accept connections from any Ip address, not just one in particular.

Myaddr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

1. **Multiple Client Handling:**

Several client’s messages must be handled by the server. Separating and organizing logs from various sources would require a mapping or list of client addresses and ports.

1. **Thread Management:**

The server can process the incoming log messages for each client in a different thread for scalability and improved performance.

1. **Logging source identification:**

Every log message must have its source (i.e., the machine or application from which it originated) identified by the server. The log message may be made to reflect this by mentioning the client’s IP and port.

1. **Dynamic Logging level adjustment:**

Every client should be able to dynamically change of logging that the server allows. This necessitates monitoring the logging level for every client independently.

1. **Enhanced user interface:**

The server’s user interface needs to be improved so that logs from various sources may be viewed individually or filtered according to the source.

1. **Error Handling and Recovery:**

It is recommended to incorporate improved error handling to address possible problems that may arise from several concurrent connections.

These are the changes we made in the code;

The server’s IP address binding was modified to INADDR\_ANY, enabling connection from any IP address, not just one in particular. The server may now listen to incoming log messages from several client computers thanks to this update.

Myaddr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

Then in multiple client handling with ‘clientMap’; Created a map called clientMap to house data about every client. A ClientInfo struct with the client’s address and log level is the value, while the key is a string that represents the client’s IP address and port.

Struct ClientInfo{

Struct scokaddr\_in addr;

LOG\_LEVEL logLevel;

};

Map<string, ClientInfo> clientMap: // map to store client information

This line of code generates a unique clientkey using the client’s IP address and port when a log message is received, allowing the source of the message to be identified. It creates a new entry with the client’s address and default log level if the client is not already in clientmap.

String clientKey = string(inet\_ntoa(remaddr.sin\_addr)) + “.” + to string(ntohs(remaddr.sin\_port));

If (clientMap.find(clientkey) == clientMap.end()) {

ClientInfo newClient;

newClient.addr = remaddr;

newClient.logLevel = DEBUG;

clientMap[clientKey] = new Client;

}

The clientKey, which indicates the log message’s source, was added to the print statement. this aids in determining the client (application/machine) from whence the log message originated.

pthread\_mutex\_lock(&lock\_x);

Cout << clientKey << “ – “ << buf << endl;

pthread\_mutex\_unlock(&lock\_x);

improved error handling by introducing a 1-second snooze if no data is received through the use of an else condition. As a result, the server uses less CPU power and requires resources.

If (recvlen > 0) {

Buf[recvlen] = ‘\0’;

// existing code

} else {

Sleep (1);

}

**Q5) Consider a Log Server that has to manage embedded logs for a massive amount of applications on a massive amount of machines. With such a large amount of data in the logs, name three ways a user could extract useful information from them (be general).**

A user can extract information from the logs by filtering the logs by machine id or name. The other way is only looking at the logs that has a severity level above a certain level, for example only looking at logs that is generated by a crash. The user can also extract the number of critical logs being generated to see if the program is having a massive error everyone is facing and see if any important logs is being generated by many users.

**Q6) Explain how gdb could be used on a Linux machine to attach to a process and get thread information. Is this also useful in debugging?**

Gdb is used primarily for debugging because it allows you to inspect the state of threads, their call stacks, variables, and more, which can aid in diagnosing and fixing issues in multi-threaded programs. This is not only specified to multi threaded programs but also single threaded programs. It also helps determine where exactly the program crashed or stopped which helps debugging significantly.