/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*

I declare that this lab is my own work in accordance with Seneca Academic Policy.  \*  No part of this assignment has been copied manually or electronically from any other source\*  (including web sites) or distributed to other students.\* \*

Name: Chetan Arora Student ID: 100976240 Date: 12 August 2025

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

a) Generally, what are syslog and rsyslog? Specifically, name three features of syslog/rsyslog and compare them to your embedded debug logging. Will there be any overlap of information?

Ans:- Classic Unix logging daemons that collect logs from the kernel and user processes, route them, and store/forward them (often via `/var/log/\*`).

Three features:

- Facility/priority routing: Rules route messages by facility and severity to files/remote sinks. Our embedded logger routes only by a single filter threshold and one remote server.

- Persistent disk files & rotation: Syslog uses rotating files and integrates with logrotate. Our embedded logger\* writes to one `server.log`; rotation/policy would need to be added.

- Network protocols: Syslog supports UDP/TCP/TLS and RELP in rsyslog. Our embedded logger uses UDP only for speed per spec.

Overlap: Yes—both capture severity, timestamp, message, and can forward to a central host. Ours carries file, function, and line for deep app context.

b) Name five features of syslog-ng.

Ans:- - Flexible filtering (match by program, host, regex).

- Multiple destinations (files, sockets, databases, SIEMs, cloud).

- Reliable transport options and flow-control.

- Structured logging (JSON) and message rewriting.

- Rich templating for message format and filenames.

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

Ans:- - More expressive config language and conditional logic.

- Better scalability and performance under load.

- Native support for structured/JSON logs and parsing.

- Advanced destinations (HDFS/NoSQL/HTTP).

- Built‑in throttling, buffering, and disk queues for reliability.

d) Consider a Log Server that has to manage embedded logs for a massive amount of processes on a massive amount of machines. Name three ways the server could manage the connections to each process.

Ans:- Connection Pooling – Reuse connections to reduce overhead and improve speed.

Asynchronous I/O (epoll/select) – Handle thousands of connections with minimal threads.

Message Broker Middleware – Use Kafka, RabbitMQ, or ZeroMQ between clients and the log server to manage load and queue logs.

e) Consider a Log Server that has to manage embedded logs for a massive amount of processes 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).

Ans:- Filtering by Severity/Keyword – Show only warnings, errors, or specific text patterns.

Log Aggregation and Search Tools – Use tools like ELK Stack (Elasticsearch, Logstash, Kibana) for indexing and querying.

Statistical Analysis – Generate metrics like error frequency, top error sources, or trend over time.

f) 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?

Ans:- Attach to Process:

gdb attach <PID>

This connects gdb to a running process without restarting it.

Get Thread Info:

Inside gdb:

info threads

thread <number>

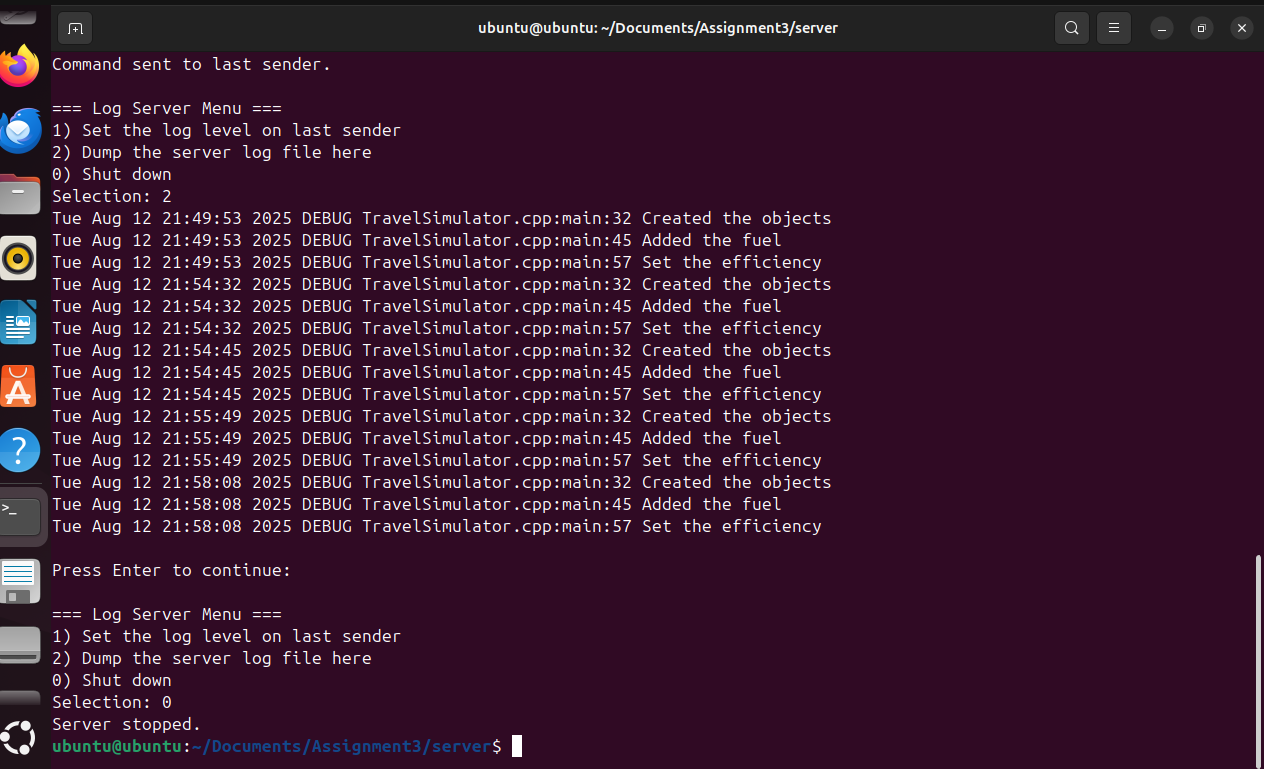
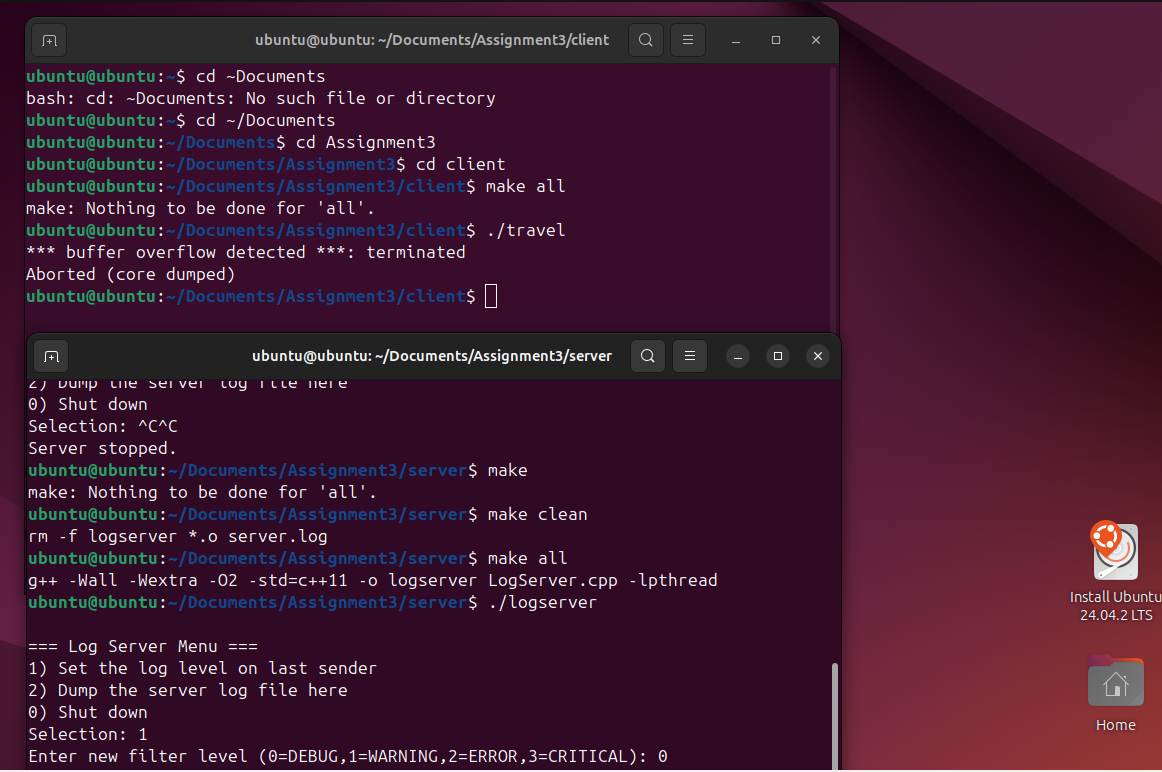
Shows all running threads and allows switching between them.

Usefulness:

Helps debug live issues without restarting the program.

Can inspect thread states, variables, and locks in real time.

Useful for diagnosing deadlocks, high CPU usage, or unexpected behavior in multithreaded programs.



Please answer the following two declarations:

o D1) On a scale from 1 to 5, How much did you use generative AI to complete this assignment?

▪

where:

▪

1 means you did not use generative AI at all

▪

2 means you used it very minimally

▪

3 means you used it moderately

▪

4 means you used it significantly

▪

5 means you relied on it almost entirely

▪

Your answer :2

o D2) On a scale from 1 to 5, How confident are you in your understanding of the generative AI support you utilized in this assignment, and in your ability to explain it if questioned?

▪

where:

▪

1 means "Not confident at all – I do not understand the generative AI support I used and cannot explain it."

▪

2 means "Slightly confident – I understand a little, but I have many uncertainties."

▪

3 means "Moderately confident – I understand the majority of the support, though some parts are unclear."

▪

4 means "Very confident – I understand most of the AI support well and can explain it with minor gaps."

▪

5 means "Extremely confident – I fully understand the generative AI support I used and can clearly explain or justify it if asked."

▪

Your answer :5