8005 Asn2 - Design

Can two programmers ever a design fully share

Isaac Morneau; A00958405
Aing Ragunathan; A00765949

Considerations when designing

The following are key points that we focused on when planning how to handle the challenges of resource usage. While not all were acted on such as raw sockets they were attempted during the prototyping phase of the assignment. Limited TCP windows, manual threading, and increasing the file descriptor limit where the most notable actions that were accounted for.

Memory

- Smaller TCP window
- Logging all flushed to disk to be parsed afterward

CPU

- OpenMP or manual threading
    - Statically allocated scheduler
    - Core affinity set
- Upfront memory allocations
- Block allocations to avoid lots of mallocs

Networking

- Smaller packets
- Send keep alives to delay Userspace responses
- Raw sockets for prebuilt IP headers
- Maximum ephemeral ports

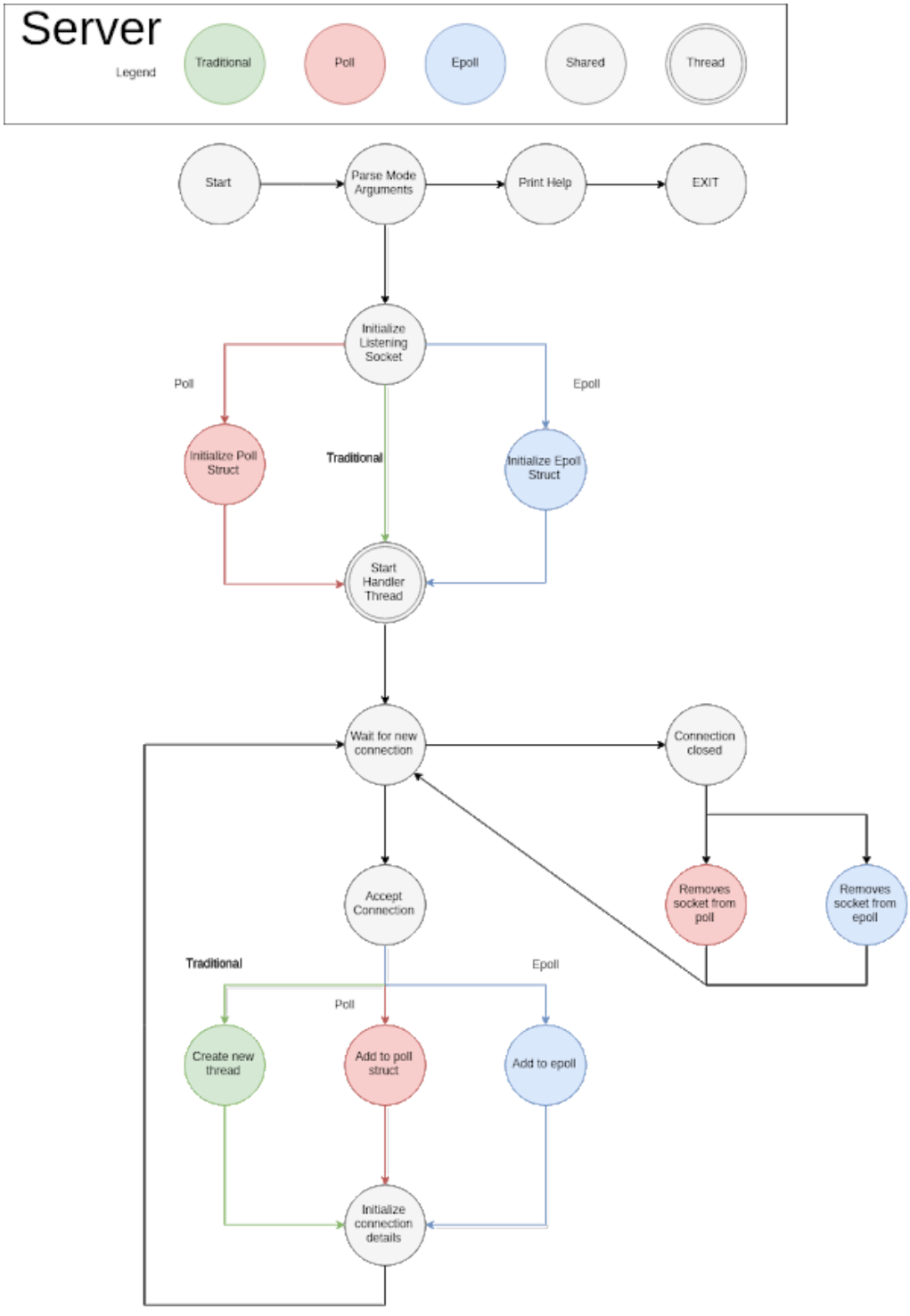
Linux kernel

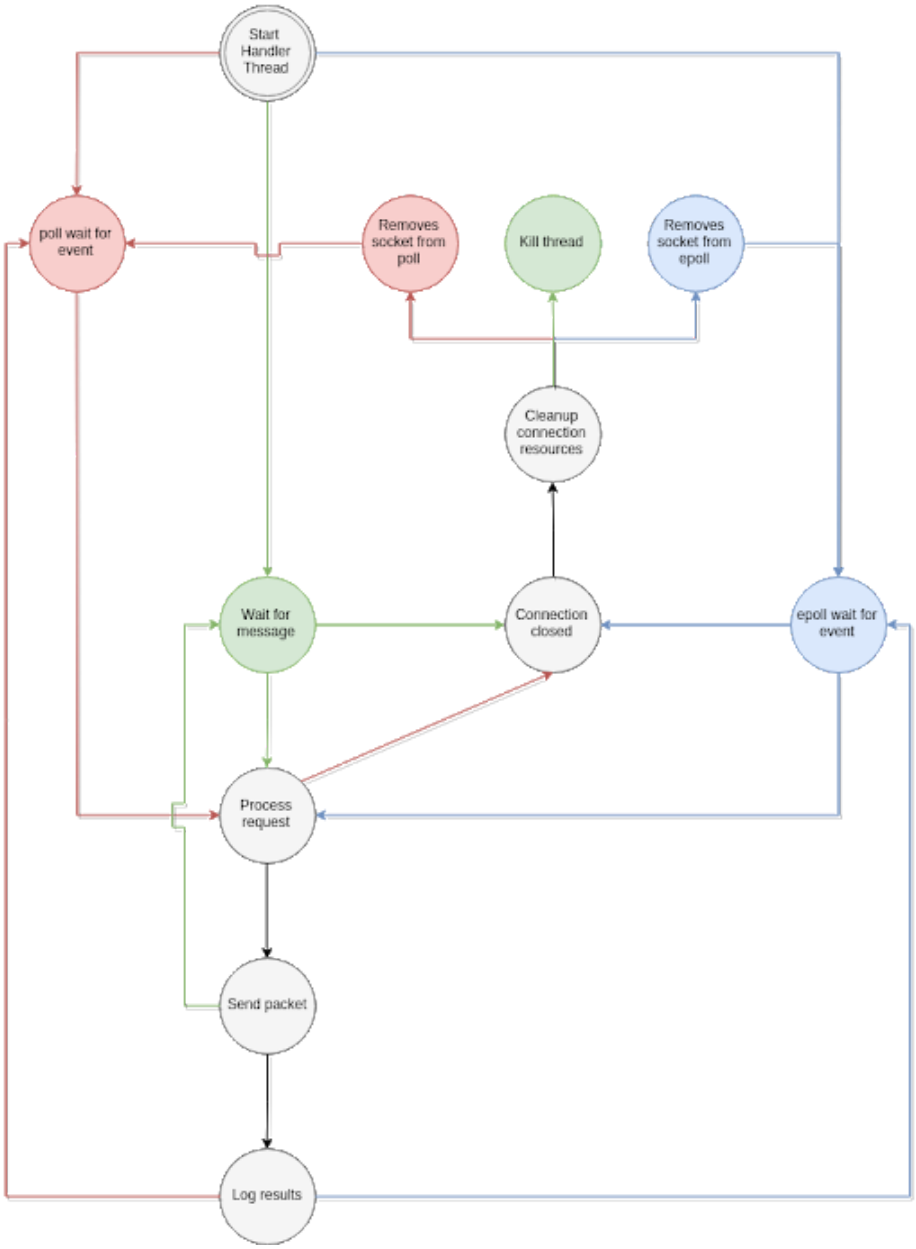
- Increase soft limit
- Increase hard limit
- Custom kernel to increase maximum hard limit

Logging

- Drop in format
- Minimal system usage
- Extract information postmortem from the log file

FSM


Server
Legend
Traditional
Poll
Epoll
Shared
Thread
Start
Parse Mode Arguments
Print Help
EXIT
Initialize Listening Socket
Poll
Epoll
Initialize Poll Struct
Traditional
Initialize Epoll Struct
Start Handler Thread
Wait for new connection
Connection closed
Accept Connection
Removes socket from poll
Removes socket from epoll
Traditional
Epoll
Poll
Create new thread
Add to poll struct
Add to epoll
Initialize connection details


Start Handler Thread
poll wait for event
Removes socket from poll
Kill thread
Removes socket from epoll
Cleanup connection resources
Wait for message
Connection closed
epoll wait for event
Process request
Send packet
Log results

Multithreaded Server Pseudocode

Start Server

Create a file descriptor for listening for clients
Set the file descriptor to listen for new connections
Goto Wait for Connection

Wait for Connection

Create a file descriptor for new connections
Wait for a new connection
Goto Establish Connection

Establish Connection

Set new connection to non blocking
Set the receiver window size
Create a Echo Thread
If maximum number of clients is not reached
    Create a Echo Thread
    Go to Wait for Connections
Otherwise
    Goto Exit

Echo Thread

While the thread is running
    Check for new data from client
    Send any new data back to client

Exit

Wait for the client to close the connections
Wait for the threads to return
Cleanup variables and quit the program

Poll Server

Start Server

Create epoll structures
Create a socket for listening
Set the file descriptor to listen for new connections
Set the first position of the Poll structure to the listening file descriptor
Set every other position in the Poll structure to unused
Set the number of connected clients to zero
Goto Create Pool of Poll_Hanlders
Goto Wait for Connection

Create Pool of Poll_Handle Threads

Get number of available CPUs
For every CPU, initialize a thread for handling established clients
On each thread
    Call Poll Handler

Wait for Connection

Create a file descriptor for new connections
While the program is running
    Wait for a new connection
    If a new connection is established
        Set new connection to non blocking
        Set receiver window size
        Add new connection to Poll structure
    Otherwise, print error

Poll_Handler

Set CPU affinity for this thread
While the thread is running
    Check for a poll events
    Echo all incoming packets

Epoll server

Start Server

Create a epoll structure
Create a file descriptor for listening for clients
Set the file descriptor to listen for new connections
Add new connections to epoll structures
Set every other position in the Poll structure to unused
Set the number of connected clients to zero
Goto Create Pool of Poll_Hanlders
Goto Wait for Connection

Create Pool of Epoll Handler Threads

Get number of available CPUs
For every CPU, initialize a thread for handling established clients
On each thread
    Call Epoll Handler

Wait for Connection

While the program is running
    Wait for a new connection
    If a new connection is established
        Set new connection to non blocking
        Set receiver window size
        Add new connection to epoll structure

Epoll Handler

Set CPU affinity for this thread
While the thread is running
    Wait for epoll events
    Echo all incoming packets for each read event
    Flush all buffers for each write event