



72.07 - Protocolos de Comunicación

Trabajo Práctico Especial

Grupo 7

Profesores:

Garberoglio, Marcelo Fabio; Kulesz, Sebastian; Stupenengo Faus, Hugo Javier; Oscar, Roberto y Valerio Ramos, Pedro Martín

Integrantes del grupo:

64396	Menshikoff, Katia	kmenshikoff@itba.edu.ar
63622	Santamarina Balbin, Manuel	msantamarinabalbin@itba.edu.ar
63587	Ader, Manuel	mader@itba.edu.ar
61067	Sapino, Matías	msapino@itba.edu.ar

Índice

Índice.....	1
1. Descripción detallada de los protocolos y aplicaciones desarrolladas.....	3
1.1. Componentes Principales.....	3
1.1.1. Servidor Principal.....	3
1.1.2. Máquina de Estados.....	3
1.1.3. Sistema de Comandos.....	3
1.1.4. Sistema de Almacenamiento.....	4
1.1.5. Sistema de Usuarios.....	4
1.1.6. Métricas y Monitoreo.....	5
2. Problemas encontrados durante el diseño y la implementación.....	5
2.1. Gestión de Memoria.....	5
2.2. Transformación de Mensajes.....	5
2.3. Interfaz de Administración.....	5
2.4. Sistema de Archivos.....	6
2.5. Concurrencia.....	6
3. Limitaciones de la aplicación.....	6
3.1. Métricas Volátiles.....	6
3.2. Almacenamiento Limitado.....	6
3.3. Seguridad Básica.....	6
3.4. Funcionalidad Esencial.....	7
4. Posibles Extensiones.....	7
4.1. Persistencia de Métricas.....	7
4.2. Seguridad Mejorada.....	7
4.3 Escalabilidad.....	7
5. Conclusiones.....	7
5.1. Arquitectura Eficiente.....	8
5.2. Modularidad.....	8
5.3. Protocolo POP3.....	8
5.4. Aprendizaje.....	8
6. Ejemplos de Prueba.....	8
6.1. Conexión y Autenticación.....	8
6.2. Consulta de Estado y Listado.....	9
6.3. Recuperación de Mensajes.....	9
6.4. Eliminación y Reset.....	10
6.5. Manejo de Errores.....	10
7. Guía de Instalación.....	11
7.1. Requisitos Previos.....	11
7.2. Pasos de Instalación.....	11
7.2.1. Obtener el Código Fuente.....	11
7.2.2. Compilar.....	11
7.2.3. Conectarse.....	11

8. Instrucciones de Configuración.....	11
9. Ejemplos de Configuración y Monitoreo.....	12
9.1 Configuración Básica.....	12
9.2. Configuración y monitoreo del Sistema.....	12
10. Documento de Diseño del Proyecto.....	14
10.1. Arquitectura General.....	14
10.2. Componentes Principales.....	14
10.3. Flujo de Datos.....	14
10.4. Decisiones de Diseño.....	14

1. Descripción detallada de los protocolos y aplicaciones desarrolladas

El proyecto implementa un servidor POP3 (Post Office Protocol 3) en C, diseñado para permitir a los clientes recuperar correos electrónicos de manera eficiente y segura. La implementación sigue una arquitectura basada en eventos y estados, lo que permite manejar múltiples conexiones concurrentes sin necesidad de crear hilos adicionales.

1.1. Componentes Principales

1.1.1. Servidor Principal

El núcleo del sistema (implementado en main.c) se encarga de:

- Inicializar el entorno y configurar los sockets
- Establecer un selector de eventos para manejar conexiones concurrentes
- Gestionar señales del sistema (SIGTERM, SIGINT)
- Configurar el directorio maildir para almacenamiento

1.1.2. Máquina de Estados

El servidor utiliza una máquina de estados (implementada en pop3.c) que maneja el ciclo de vida completo de cada conexión. Los estados principales son:

- WELCOME: Maneja la conexión inicial y muestra el mensaje de bienvenida
- READ: Espera y procesa los comandos entrantes
- WRITE: Envía respuestas al cliente
- FILE_WRITE: Maneja la escritura de archivos
- CLOSE: Gestiona el cierre ordenado de conexiones
- ERROR: Maneja situaciones de error

1.1.3. Sistema de Comandos

La implementación soporta los comandos estándar del protocolo POP3:

- USER/PASS: Autenticación
- STAT: Estado del buzón
- LIST: Listado de mensajes
- RETR: Recuperación de mensajes
- DELE: Marcado para eliminación
- QUIT: Cierre de sesión

- RSET: Reseteo de estado
- NOOP: No operación
- CAPA: Capacidades

Es importante aclarar que un usuario no autenticado solo puede utilizar los comandos USER, PASS, CAPA y QUIT. Mientras que un usuario que si está autenticado puede acceder, sacando USER y PASS (ya que ya está autenticado), a todo el resto de comandos

Además, el comando RETR permite la opción de aplicar transformaciones a los mails. Estos se ejecutan de la siguiente manera:

```
RETR <message_num>|<transformation_name>
```

en donde <transformation_name> es una de las transformaciones disponibles en el servidor. Las transformaciones disponibles por defecto son 'base64', 'cat' y 'uppercase'. Sin embargo, un usuario administrador puede agregar más transformaciones, así como habilitar y deshabilitar las ya existentes.

1.1.4. Sistema de Almacenamiento

El almacenamiento utiliza la estructura Maildir, que organiza los correos en directorios separados por usuario y estado.

```
-> maildir tree .
```

```
└─ user1
    ├── cur
    ├── new
    │   └─ mail1
    └─ tmp
```

Esta estructura permite:

- Operaciones atómicas en archivos
- Separación clara entre mensajes nuevos y procesados
- Manejo eficiente de concurrencia

1.1.5. Sistema de Usuarios

El manejo de usuarios implementa:

- Autenticación básica mediante usuario/contraseña

- Sistema de bloqueo para evitar accesos concurrentes al mismo buzón
- Gestión de estado de mensajes (eliminados/activos)
- Inicialización y mantenimiento de mailboxes

1.1.6. Métricas y Monitoreo

El servidor mantiene métricas en tiempo real sobre:

- Número de conexiones activas y totales
- Cantidad de bytes transferidos
- Registro de comandos ejecutados
- Estado general del sistema

Estas solo pueden ser consultada por un usuario administrador, con el comando 'metrics'

2. Problemas encontrados durante el diseño y la implementación

Durante el desarrollo del servidor POP3 se encontraron diversos desafíos técnicos que requirieron soluciones específicas:

2.1. Gestión de Memoria

El manejo de memoria presentó problemas, especialmente en el comando QUIT donde la liberación incorrecta de recursos y el acceso a punteros inválidos causaban cierres inesperados del servidor (core dumps). Esto requirió una revisión completa del ciclo de vida de las conexiones.

2.2. Transformación de Mensajes

La implementación del comando RETR enfrentó dificultades en el manejo de archivos temporales y el procesamiento de mensajes grandes. El sistema de transformación requería permisos específicos y un manejo cuidadoso de la memoria para evitar memory leaks.

2.3. Interfaz de Administración

El módulo de administración presentó retos en la sincronización con el servidor principal y la actualización de métricas en tiempo real.

2.4. Sistema de Archivos

El manejo del sistema Maildir mostró complicaciones en la gestión de archivos, la limpieza de archivos temporales y la gestión de directorios. requirieron una implementación robusta para mantener la integridad de los datos.

2.5. Concurrencia

El manejo de múltiples conexiones simultáneas presentó desafíos en términos de condiciones de carrera y transiciones de estado.

3. Limitaciones de la aplicación

3.1. Métricas Volátiles

El sistema de métricas opera exclusivamente en memoria, lo que significa que todos los datos de monitoreo y estadísticas se pierden al reiniciar el servidor. Esto impide realizar análisis históricos del rendimiento y dificulta la detección de patrones de uso a largo plazo.

3.2. Almacenamiento Limitado

Las estructuras de datos utilizan límites estáticos que restringen la capacidad del servidor. Con un máximo predefinido de usuarios y correos, el sistema puede resultar insuficiente para organizaciones en crecimiento o con alto volumen de mensajes.

```
#define MAX_USERS 1024
```

```
#define MAX_MAILS 1000
```

```
#define BUFFER_SIZE 1024
```

3.3. Seguridad Básica

La implementación de seguridad es bastante minimalista. Carece de soporte para conexiones cifradas y mecanismos avanzados de autenticación. Esto limita su uso en entornos donde la seguridad es un requisito crítico.

3.4. Funcionalidad Esencial

El servidor implementa solo las funcionalidades básicas del protocolo POP3, sin soporte para extensiones avanzadas como APOP, TOP, UTF8, SASL, etc.

4. Posibles Extensiones

4.1. Persistencia de Métricas

Se podría implementar un sistema de almacenamiento persistente para las métricas utilizando una base de datos SQLite o archivos de log estructurados. Esto permitiría:

- Análisis histórico del rendimiento
- Generación de reportes

4.2. Seguridad Mejorada

- Integración de SSL/TLS para conexiones cifradas
- Implementación de APOP para autenticación segura
- Sistema de rate limiting para prevenir ataques

4.3 Escalabilidad

El servidor podría extenderse para soportar:

- Modelo multi-proceso para aprovechar múltiples cores
 - Clustering para distribuir la carga
 - Cache distribuido para mejorar el rendimiento
 - Sistema de colas para operaciones pesadas
-

5. Conclusiones

El desarrollo de este servidor POP3 ha demostrado ser un ejercicio valioso en la implementación de protocolos de red y arquitecturas basadas en eventos. A través del proyecto, se han logrado varios objetivos importantes:

5.1. Arquitectura Eficiente

La elección de una arquitectura basada en estados y eventos ha probado ser efectiva para manejar múltiples conexiones concurrentes sin la complejidad adicional de la programación multi-hilos. El uso del selector de eventos permite una utilización eficiente de los recursos del sistema, aunque con las limitaciones mencionadas anteriormente.

5.2. Modularidad

El diseño modular del sistema, separando claramente las responsabilidades entre diferentes componentes (comandos, usuarios, almacenamiento, métricas), ha facilitado el desarrollo y permitirá futuras extensiones. La estructura del código permite agregar nuevas funcionalidades sin necesidad de modificar los componentes existentes.

5.3. Protocolo POP3

La implementación del protocolo POP3 ha seguido fielmente las especificaciones estándar, proporcionando una base sólida para la recuperación de correos. Aunque se implementaron solo los comandos básicos, la estructura permite fácilmente la adición de extensiones más avanzadas.

5.4. Aprendizaje

El proyecto ha servido como una excelente plataforma de aprendizaje

- Implementación de protocolos de red, específicamente POP3
- Manejo de concurrencia basada en eventos
- Gestión de recursos del sistema
- Diseño de sistemas modulares
- Manejo de errores y recuperación

6. Ejemplos de Prueba

A continuación se presentan ejemplos de interacción con el servidor POP3 que demuestran su funcionalidad básica.

6.1. Conexión y Autenticación

```
$ ncat localhost 1080
```

+OK POP3 server ready

USER user1

+OK user accepted

PASS password1

+OK maildrop locked and ready

6.2. Consulta de Estado y Listado

STAT

+OK 2 320

LIST

+OK 2 messages (320 octets)

1 120

2 200

.

6.3. Recuperación de Mensajes

RETR 1

+OK message follows (120 octets)

From: sender@example.com

To: user1@localhost

Subject: Test Message

This is a test message.

.

6.4. Eliminación y Reset

DELE 1

+OK message 1 deleted

RSET

+OK maildrop has been reset

LIST

+OK 2 messages (320 octets)

1 120

2 200

.

6.5. Manejo de Errores

USER invalid

-ERR invalid user

RETR 999

-ERR no such message

DELE 1

-ERR message already deleted

7. Guía de Instalación

7.1. Requisitos Previos

- Sistema operativo Linux/Unix
- Compilador GCC
- Make
- CMake (versión 3.0 o superior)
- Librerías de desarrollo estándar de C

7.2. Pasos de Instalación

7.2.1. Obtener el Código Fuente

```
git clone https://github.com/MatiSapino/POP3
cd POP3
```

7.2.2. Compilar

```
./build.sh
./pop3server
```

7.2.3. Conectarse

```
(en otra terminal)
ncat localhost 1080
```

8. Instrucciones de Configuración

Usage: ./pop3server [OPTION]...

```
-h                Imprime la ayuda y termina
-l <POP3 addr>    Dirección donde servirá el proxy POP3
-L <conf addr>    Dirección donde servirá el servicio de
management
```

-p <POP3 port> Puerto entrante conexiones POP3
-P <conf port> Puerto entrante conexiones configuración
-u <name>:<pass> Usuario y contraseña (hasta 10 usuarios)
-v Imprime información sobre la versión
-d Carpeta donde residen los Maildirs

9. Ejemplos de Configuración y Monitoreo

9.1 Configuración Básica

Iniciar servidor en puerto 1110 con usuarios predefinidos

```
./pop3server -p 1110 -u user1:pass1 -u user2:pass2
```

Iniciar con interfaz admin en puerto diferente

```
./pop3server -p 1110 -P 9090 -u admin:admin
```

Por otra parte, el usuario administrativo puede configurar cosas como las transformaciones posibles con

```
$ ncat localhost 9090
```

```
> transform add base64 'base64'  
Transformation added successfully.
```

9.2. Configuración y monitoreo del Sistema

Las métricas del servidor y la configuración, tanto del sistema como de las transformaciones disponibles para el RETR, se pueden acceder a través de la interfaz administrativa:

Available commands:

```
help - Show this help
```

```
metrics - Show server metrics
```

transform list - List available transformations
transform add <name> <cmd> - Add new transformation
transform test <name> <input> - Test transformation
transform enable <name> - Enable transformation
transform disable <name> - Disable transformation
config get - Show current configuration
config set timeout <sec[.nsec]> - Set select timeout
config set buffer_size <bytes> - Set I/O buffer size
config set max_connections <num> - Set max connections
config set verbose <true|false> - set verbose logging

Por ejemplo, al ejecutar 'metrics' :

```
$ ncat localhost 9090
> metrics
Current connections: 5
Total connections: 15
Bytes sent: 35
Bytes received: 40
Active users: 2
Total commands: 3
Failed commands: 2
Active users: 1
```

10. Documento de Diseño del Proyecto

10.1. Arquitectura General

El sistema está construido sobre una arquitectura basada en eventos, utilizando un selector no bloqueante como núcleo. Esta arquitectura permite manejar múltiples conexiones concurrentes de manera eficiente, distribuyendo las responsabilidades entre diferentes módulos especializados.

10.2. Componentes Principales

El servidor está dividido en cinco componentes fundamentales. El selector de eventos maneja todas las operaciones de I/O de manera no bloqueante. La máquina de estados controla el flujo de la aplicación a través de estados bien definidos. El sistema de usuarios gestiona la autenticación y los mailboxes. El sistema de comandos implementa el protocolo POP3, y el sistema de almacenamiento maneja la persistencia de datos usando la estructura Maildir.

10.3. Flujo de Datos

El procesamiento de datos sigue un flujo lineal y ordenado, comenzando con la conexión del cliente, pasando por la autenticación, y continuando con el procesamiento de comandos POP3. Todas las operaciones de I/O se realizan de manera no bloqueante para mantener la responsividad del sistema.

10.4. Decisiones de Diseño

Las decisiones de diseño se centraron en tres aspectos clave: la concurrencia basada en eventos para optimizar recursos, el uso de Maildir para garantizar operaciones atómicas y seguras, y una implementación básica del protocolo POP3 que nos da funcionalidades primitivas pero que permite futuras extensiones gracias a su desarrollo modular.