

Decodificador de Protocolo Industrial (PRT-7)

Materia: Estructura de Datos

Profesor: Dr. Said Polanco Martagón

Alumno: Emanuel Hernández Aguirre

Introducción

El presente documento describe el desarrollo e implementación del caso de estudio «Decodificador de Protocolo Industrial (PRT-7)». Este proyecto tiene como objetivo poner en práctica los conocimientos de Programación Orientada a Objetos (POO) y el manejo de estructuras de datos manuales, específicamente listas doblemente enlazadas y listas circulares, para simular el proceso de decodificación de un protocolo industrial de ensamblaje de mensajes ocultos.

El sistema PRT-7 simula la lectura de tramas enviadas por un dispositivo Arduino a través de un puerto serial. Cada trama puede ser de tipo LOAD o MAP, donde las primeras contienen datos de carga (caracteres) y las segundas instrucciones de rotación que alteran el mapeo del alfabeto. Mediante estas tramas, el programa reconstruye un mensaje cifrado, demostrando el uso de punteros, clases abstractas y herencia polimórfica.

Diseño y Desarrollo Técnico

El desarrollo del sistema se realizó en el lenguaje C++, implementando manualmente las estructuras de datos sin usar la STL. Se diseñó una clase base abstracta denominada TramaBase, de la cual derivan dos clases concretas: TramaLoad y TramaMap. Cada una implementa su propio comportamiento mediante el método virtual procesar().

Asimismo, se implementaron las clases RotorDeMapeo (lista circular doblemente enlazada) y ListaDeCarga (lista doblemente enlazada), encargadas de manejar el mapeo del alfabeto y almacenar los caracteres decodificados respectivamente. La interacción entre estas clases permite simular el movimiento de un disco cifrador y la decodificación dinámica de los caracteres recibidos.

El código fue compilado y ejecutado en Windows utilizando el compilador MinGW (g++ 6.3.0), logrando una correcta ejecución del algoritmo de decodificación, sin dependencias externas ni librerías de la STL.

Resultados y Evidencias

Durante la ejecución del programa, se ingresaron tramas de tipo L (LOAD) y M (MAP) siguiendo el formato del protocolo PRT-7. A continuación se muestra un ejemplo del flujo de ejecución en la consola:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
PS C:\Users\Emm11\OneDrive\Desktop\PRT7_Project> .\prt7.exe
Recibida L,H -> OK.
L,O
Recibida L,O -> OK.
L,L
Recibida L,L -> OK.
M,2
Recibida M,2 -> OK.
L,A
Recibida L,A -> OK.
L,Space
Recibida L,_ -> OK.
L,W
Recibida L,W -> OK.
M,-2
Recibida M,-2 -> OK.
L,O
Recibida L,O -> OK.
L,R
Recibida L,R -> OK.
L,L
Recibida L,L -> OK.
L,D
Recibida L,D -> OK.
^Z

---
MENSAJE ENSAMBLADO:
HOLC YORLD
---
```

○ PS C:\Users\Emm11\OneDrive\Desktop\PRT7_Project> []

El resultado mostrado confirma que el sistema procesa correctamente las tramas, realiza las rotaciones del rotor y ensambla el mensaje oculto de acuerdo con el protocolo de mapeo definido.

```
1 # Doxygen config for PRT-7
2 PROJECT_NAME          = "PRT-7 Decodificador"
3 OUTPUT_DIRECTORY      = docs
4 INPUT                 = src
5 FILE_PATTERNS        = *.h *.hpp *.c *.cpp
6 RECURSIVE              = YES
7 EXTRACT_ALL            = YES
8 GENERATE_HTML          = YES
9 GENERATE_LATEX         = NO
10 QUIET                  = YES
11 WARN_IF_UNDOCUMENTED = NO
```

Conclusión

El desarrollo del decodificador PRT-7 permitió reforzar los conceptos de programación orientada a objetos, uso de herencia y polimorfismo, así como la gestión manual de memoria mediante punteros. Además, se comprendió el funcionamiento de las listas doblemente enlazadas y listas circulares aplicadas a un problema práctico de decodificación.

La implementación del sistema sin el uso de librerías STL garantizó una comprensión más profunda del manejo de nodos, punteros y estructuras dinámicas, consolidando las bases para proyectos de mayor complejidad en el área de estructuras de datos.

Referencias

- Material del Caso de Estudio: Decodificador de Protocolo Industrial (PRT-7)
- Documentación oficial de C++ (cplusplus.com)
- Guía de estilo APA 7^a edición para trabajos académicos