



Disseny i Implementació de la Fase 2

Sistema Gotham

AUTORS:

Pau Olea Reyes
Alfred Chávez Fernández

Assignatura: Sistemes Operatius
Universitat Ramon Llull – La Salle

November 17, 2024

1 *

Conte	nts*	1
2	Introducció	2
2.1	Esquema funcional de comunicació entre processos	2
2.2	Estructura de la implementació de la fase 2	3
3	Disseny del Sistema	4
3.1	Estructures de dades	4
3.2	Recursos del sistema utilitzats	4
4	Implementació	5
4.1	Mòduls principals	5
4.2	Flux d'execució	5
5	Opcions implementades	5
6	Conclusió	6

2 Introducció

La segona fase del projecte del Sistema Gotham està enfocada a la integració de múltiples processos i la comunicació efectiva entre ells, utilitzant recursos del sistema com **sockets**, **files (threads)**, **mutexos**, i fitxers de configuració.

2.1 Esquema funcional de comunicació entre processos

La Figura 1 mostra com es comuniquen els diferents processos del sistema:

- **Gotham.exe:** Actua com a servidor central que rep peticions i coordina la distribució de tasques entre els treballadors.
- **Enigma.exe:** Un tipus de treballador que es registra a Gotham i processa tasques específiques de text.
- **Harley.exe:** Un altre tipus de treballador que processa contingut multimèdia, com àudio o imatges.
- **Fleck.exe:** Un treballador encarregat de tasques d'indexació o processament addicional.
- La comunicació es fa mitjançant sockets TCP i segueix un protocol definit amb missatges estructurats per garantir la compatibilitat.

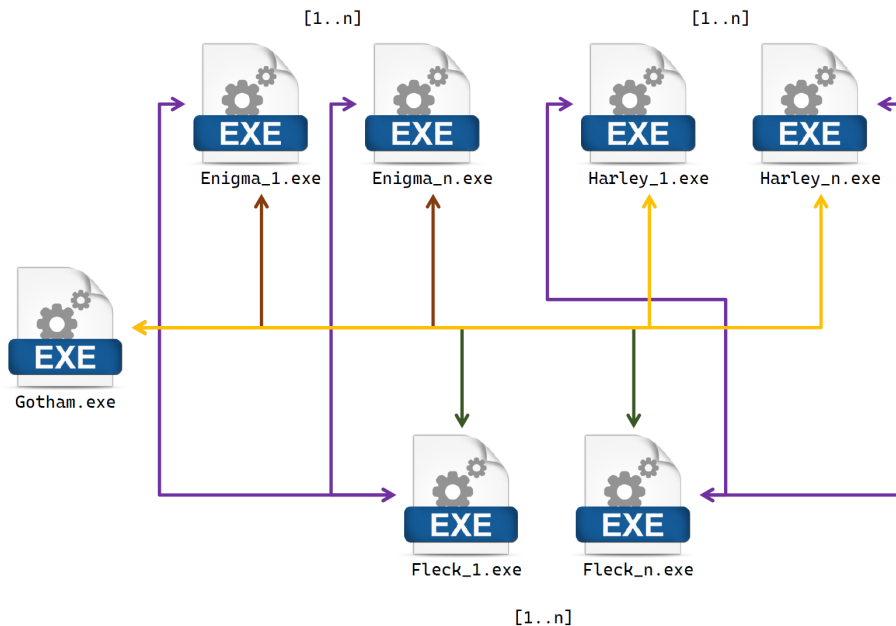


Figure 1: Esquema funcional de comunicació entre processos.

2.2 Estructura de la implementació de la fase 2

La Figura 2 presenta la jerarquia de fitxers i les seves dependències:

- Cada fitxer de configuració (amb extensió `.dat`) conté els paràmetres específics per al procés corresponent, com ara adreces IP, ports i tipus de treballadors.
- Els mòduls principals del sistema són `Gotham.c`, `Enigma.c`, `Harley.c` i `Fleck.c`.
- Els mòduls auxiliars (`Networking.c`, `FileReader.c`, `StringUtils.c`, `DataConversion.c`) proporcionen funcionalitats compartides, com la gestió de sockets, lectura de fitxers o manipulació de dades.
- El fitxer `Makefile` automatitza la compilació del sistema.

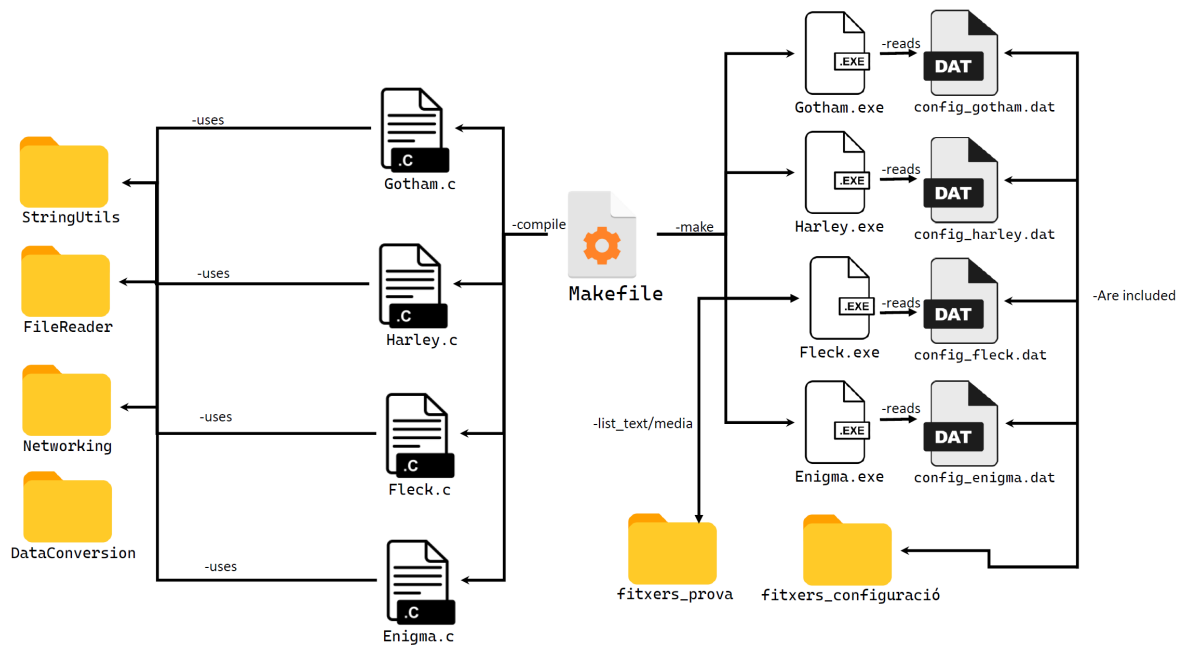


Figure 2: Estructura de la implementació de la fase 2.

3 Disseny del Sistema

3.1 Estructures de dades

Per a aquesta implementació, s'han definit estructures de dades clau per gestionar la informació dels *workers* i les comunicacions:

- **WorkerInfo:** Conté informació sobre la IP, port, tipus i descriptor de socket de cada *worker*.
- **WorkerManager:** Administra una llista dinàmica de *workers* amb accés sincronitzat mitjançant **mutexos**.

3.2 Recursos del sistema utilitzats

- **Sockets TCP:** Utilitzats per la comunicació entre processos (Gotham, Enigma, Harley, Fleck).
- **Threads:** Cada connexió amb un client és gestionada en un fil separat per millorar el rendiment.
- **Mutexos:** Garantitzen l'accés segur a les estructures compartides, com ara la llista de *workers*.
- **Fitxers de configuració:** Cada procés llegeix un fitxer de configuració específic que conté els paràmetres necessaris.

4 Implementació

4.1 Mòduls principals

El sistema s'ha dividit en diversos mòduls per garantir una millor modularitat i mantenibilitat:

- **Gotham.c:** Actua com a servidor central que coordina els *workers*.
- **Enigma.c, Harley.c i Fleck.c:** Clients que es connecten a Gotham per realitzar tasques específiques.
- **Networking.c:** Conté funcions genèriques per a la gestió de sockets i comunicacions.
- **FileReader.c:** Facilita la lectura de fitxers de configuració.
- **StringUtils.c i DataConversion.c:** Mòduls per al tractament de cadenes i conversió de dades.

4.2 Flux d'execució

1. **Inicialització:** Cada procés llegeix el seu fitxer de configuració i estableix connexions segons els paràmetres definits.
2. **Registre:** Els *workers* s'enregistren al servidor Gotham amb el seu tipus, IP i port.
3. **Processament:** Gotham delega tasques als *workers* segons el tipus de comanda rebuda.
4. **Finalització:** Els recursos es tanquen i la memòria dinàmica es gestiona adequadament.

5 Opcions implementades

- **Gestió de connexions simultànies:** Utilitzant *threads* per a cada connexió.
- **Suport per múltiples tipus de treballadors:** Enigma, Harley i Fleck poden processar diferents tasques.
- **Validació d'entrada:** Cada comanda és validada abans de processar-se per evitar errors.
- **Registre detallat:** Cada procés genera logs amb colors per facilitar el depurament.

6 Conclusió

La implementació de la fase 2 del projecte del Sistema Gotham ha estat un desafiament tècnic que ens ha permès aprofundir en conceptes fonamentals dels sistemes operatius, com la comunicació entre processos, l'ús de recursos del sistema i el disseny de programes escalables. Aquesta fase ens ha portat a dissenyar i implementar una arquitectura robusta i modular capaç de gestionar múltiples clients, tot mantenint una comunicació segura i eficient amb el servidor central, Gotham.

En primer lloc, hem aconseguit crear una estructura coherent basada en sockets TCP per establir comunicacions bidireccionals entre Gotham i els diversos tipus de treballadors: Enigma, Harley i Fleck. Aquesta elecció ha estat fonamental per garantir la fiabilitat de les transmissions, donant suport tant a la connexió com a l'intercanvi de dades. També hem implementat un sistema de gestió de connexions simultànies utilitzant fils (threads) en el servidor Gotham, que ens ha permès escalar el sistema per donar suport a múltiples clients de forma paral·lela. Aquesta estratègia ha estat clau per mantenir un bon rendiment en entorns amb altes càrregues de treball.

La integració de fitxers de configuració per a cada procés ha estat una decisió important en el nostre disseny. Aquest enfocament ens ha permès separar la lògica del codi de la configuració específica de cada component, facilitant així la portabilitat i la personalització del sistema. Els fitxers de configuració inclouen informació crucial com les adreces IP, els ports i els tipus de treballadors, i són gestionats mitjançant el mòdul 'FileReader'.

Durant aquesta fase, hem posat un èmfasi especial en garantir la sincronització i la integritat de les dades. Per aquest motiu, hem utilitzat mutexos per protegir les estructures compartides, com ara la llista dinàmica de treballadors, dins del servidor Gotham. Això ens ha permès evitar condicions de competició i assegurar que els recursos són gestionats de forma segura en un entorn concurrent.

Hem dissenyat els processos de manera que cadascun tingui una responsabilitat clara i ben definida. Gotham actua com el nucli central que coordina les accions dels treballadors i assigna tasques segons el tipus de comanda rebuda. Els treballadors, per la seva banda, compleixen rols específics (textual o multimèdia) i processen les tasques assignades de manera eficient. Aquesta distribució del treball ens ha permès desenvolupar un sistema escalable i fàcilment adaptable a futures extensions.

A més, hem implementat diverses funcionalitats opcionals que aporten un valor afegit al sistema. Per exemple, hem incorporat un sistema de validació d'entrades per assegurar que només es processen comandes vàlides. També hem creat un mecanisme de registre detallat amb codis de colors per a facilitar el depurament i el seguiment de les operacions del sistema.

Finalment, volem destacar que aquesta fase no només ha suposat un repte tècnic, sinó també una oportunitat per millorar les nostres habilitats de treball en equip i organització. La necessitat de coordinar diversos components, modularitzar el codi i gestionar recursos compartits ens ha obligat a treballar amb rigor i precisió, aspectes essencials en qualsevol projecte real de programari.

En resum, la implementació de la fase 2 ha estat un pas significatiu en el desenvolupament del Sistema Gotham, i estem orgullosos dels resultats obtinguts. El sistema resultant és robust, modular i escalable, i compleix els requisits plantejats en l'enunciat de la pràctica. A més, hem creat una base sòlida per afrontar futures ampliacions i millores, consolidant els coneixements adquirits i desenvolupant noves competències que seran invaluables en projectes futurs.