

PRÁCTICA SO FASE 1

Ignacio Giral y Marti Farré
(ignacio.giral y marti.farre)

17/10/2024

Índice

| | |
|--------------------------------|---|
| Estructuración del diseño..... | 2 |
| Problemas observados..... | 4 |
| Estimación temporal..... | 5 |
| Conclusiones..... | 6 |
| Bibliografía..... | 7 |

Estructuración del diseño

Estructura general: El proyecto lo forman 9 elementos en total, entre los cuales por el momento (Fase 1) no hay comunicación entre ellos, los vamos a analizar uno a uno:

Directorio bin: Van a parar todos los archivos .o que genera el makefile y por tanto desde su interior es donde ejecutamos las llamadas para la ejecución de los diversos programas, por ejemplo para el programa fleck seria de la siguiente forma: `./fleck ../config/fleck.dat`

Directorio config: Encontramos todos los archivos de configuración necesarios, en concreto: `enigma.dat`, `harley.dat`, `gotham.dat` y `fleck.dat`

Directorio data: Encontramos todos los ficheros tipo "Media" y tipo "Text" con los que contamos.

Directorio enigma: Encontramos el código .c del proceso Enigma, encargado de distorsionar los ficheros tipo "Text". de momento solamente lee y muestra cómo se almacena correctamente su fichero de configuración.

Directorio harley: Encontramos el código .c del proceso Harley, encargado de distorsionar los ficheros tipo "Media", de momento solamente lee y muestra cómo se almacena correctamente su fichero de configuración.

Directorio gotham: Encontramos el código .c del proceso Gotham, encargado de conectar a los usuarios con los procesos enigma y harley, de momento solamente lee y muestra cómo se almacena correctamente su fichero de configuración.

Directorio fleck: Encontramos el código .c del proceso Fleck, los cuales serán los diferentes usuarios que se conectan al sistema, de momento lee y muestra cómo se almacena correctamente su fichero de configuración, reacciona a una interrupción como el CTRL+C y escucha comandas las cuales responde con Comanda OK o KO dependiendo de si está bien o mal escrita y en caso de Logout, List Media o List Text realiza la petición como es debido.

Directorio utils: Encontramos el código .c y .h de Utils, los cuales contienen ciertas funciones que son interesantes reutilizar y compartir desde los diferentes procesos del diseño.

Fichero Makefile: Fichero que ejecuta el ejecutable con la comanda "make".

Estructuras de datos:

En los códigos .c enigma, harley, gotham y fleck se han utilizado unas estructuras de datos structs para almacenar la información del fichero de configuración (IPs, puertos, directorios, tipo ("Media" o "Text"), username) en distintas variables, con esta forma:

```
typedef struct {  
    char *gotham_server_ip;  
    int gotham_server_port;  
    char *enigma_server_ip;  
    int enigma_server_port;  
    char *directory;  
    char *worker_type;  
} EnigmaConfig;
```

```
typedef struct {  
    char *gotham_server_ip;  
    int gotham_server_port;  
    char *harley_server_ip;  
    int harley_server_port;  
    char *directory;  
    char *worker_type;  
} HarleyConfig;
```

```
typedef struct {  
    char *fleck_server_ip;  
    int fleck_server_port;  
    char *external_server_ip;  
    int external_server_port;  
} GothamConfig;
```

```
typedef struct {  
    char *username;  
    char *directory;  
    char *server_ip;  
    int server_port;  
} FleckConfig;
```

Recursos del sistema utilizados:

En estas alturas de proyecto solamente se han utilizado signals como recurso del sistema destacado, se hace uso del signal para el control de la interrupción CTRL+C en el código del fleck.c. Con el uso del valgrind hemos observado cómo, aún en ese caso, no tenemos ninguna fuga de memoria.

Problemas observados

A estas alturas de práctica no hemos observado ningún problema notable. Hemos refrescado nuestros conocimientos del año pasado al ser repetidores, sólomente de la parte práctica, y ha salido de manera bastante fluida.

Estimación Temporal

Investigación: 0.5 hora.

Diseño: 0.5 hora.

Implementación: 6 horas.

Testing: 1 hora.

Documentación: 2 horas.

Conclusiones

Es aún pronto para sacar conclusiones muy grandes, pero la primera fase la hemos completado sin mucha dificultad, con buena comunicación y repartición del trabajo. Veremos como nos encontramos con la siguiente fase que seguro que es un reto más grande.

Bibliografía